

**Títol:** Interfície Gràfica per a software de control d'inspecció en temps real

**Volum:** 1

**Alumne:** Marc Pi Sardà

**Director/Ponent:** Ricard Vilalta Cañellas / Pere Botella López

**Departament:** LSI

**Data:** 07 / 01 / 2010



## DADES DEL PROJECTE

---

### DADES DEL PROJECTE

*Títol del Projecte:* Interfície Gràfica per a software de control d'inspecció en temps real

*Nom de l'estudiant:* Marc Pi Sardà

*Titulació:* Enginyeria superior en informàtica

*Crèdits:* 37.5

*Director/Ponent:* Ricard Vilalta Cañellas / Pere Botella López

*Departament:* Llenguatges i Sistemes Informàtics (LSI)

---

### MEMBRES DEL TRIBUNAL *(nom i signatura)*

*President:* Miquel Barceló García

*Vocal:* Narcís Nabona Francisco

*Secretari:* Pere Botella López

---

### QUALIFICACIÓ

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:*

---



## ÍNDEX DE CONTINGUTS

<b>1. Introducció</b>	<b>9</b>
1.1. Descripció	11
1.2. Descripció de l'empresa	12
1.3. Motivació personal	13
1.4. Problemàtica	14
1.5. Objectius i resultats esperats	15
1.6. Planificació inicial	16
1.6.1. Estudi temporal	16
1.6.2. Estudi econòmic	21
1.6.2.1. Costos de Hardware	21
1.6.2.2. Costos de Software	22
1.6.2.3. Costos de Recursos Humans	25
1.6.2.4. Cost Total	27
1.6.3. Estudi de riscos	28
1.7. Organització de la memòria	29
1.8. Glossari	31
<b>2. Contextualització</b>	<b>33</b>
2.1. Introducció	35
2.2. VisioInspect2	36
2.2.1. Explicació	36
2.2.2. Conceptes generals	38
2.2.3. Components	40
2.2.4. Estats	44
2.2.5. Seqüències típiques	46
2.3. Interfície gràfica	49
<b>3. Anàlisi de requisits</b>	<b>51</b>
3.1. Introducció	53
3.1.1. Abast del projecte	53
3.2. Requisits no funcionals	54

3.3. Requisits funcionals .....	55
<b>4. Especificació .....</b>	<b>57</b>
4.1. Introducció .....	59
4.2. Model de casos d'ús .....	60
4.2.1.Descripció dels actors del sistema .....	60
4.2.2.Descripció dels casos d'ús .....	61
4.2.2.1. Pantalles i widgets de l'aplicació .....	61
4.2.2.2. Menús de l'aplicació .....	74
4.3. Model conceptual de dades .....	91
4.3.1.Diagrama de classes global de la interfície gràfica .....	92
4.3.2.Diagrama de classes del Controlador .....	93
4.3.3.Diagrama de classes del Controlador de la interfície gràfica .....	93
4.3.4.Diagrama de classes de la Pantalla Principal .....	94
4.3.5.Diagrama de classes per la gestió dels Missatges .....	94
4.3.6.Diagrama de classes per la gestió dels Diàlegs .....	96
<b>5. Disseny .....</b>	<b>97</b>
5.1. Introducció .....	99
5.2. Disseny de l'arquitectura .....	100
5.2.1.Patrons utilitzats .....	101
5.3. Disseny de la Pantalla Principal .....	102
5.4. Disseny dels Menús i diàlegs .....	103
5.5. Disseny dels Widgets .....	105
5.6. Diagrames de seqüència .....	108
5.6.1.Inicialització de l'aplicació .....	108
5.6.2.Interacció Interfície gràfica – Controlador .....	110
5.6.3.Interacció Controlador – Interfície gràfica .....	111
5.6.4.Comunicació amb la base de dades i recàrrega de la configuració .....	112
5.6.5.Diàlegs dels menús de configuració de l'aplicació .....	114
<b>6. Implementació .....</b>	<b>119</b>
6.1. Introducció .....	121
6.2. Tecnologies utilitzades .....	122
6.3. Entorn de desenvolupament .....	123

6.3.1. Components del grup .....	123
6.3.2. Servidor del departament .....	124
6.4. Components implementats .....	126
6.4.1. Controlador de la interfície gràfica .....	126
6.4.2. Pantalla Principal .....	126
6.4.3. Menús i Diàlegs .....	127
6.4.4. GUI Widgets .....	130
6.5. Persistència de les dades .....	135
6.5.1. Configuració .....	135
6.5.2. SGBD .....	137
6.5.3. Estructures de dades .....	137
<b>7. Interfície gràfica desenvolupada .....</b>	<b>139</b>
7.1. Widgets de la Pantalla Principal .....	141
7.2. Diàlegs de configuració dels menús .....	145
7.3. Menús .....	150
7.4. Interfície gràfica global .....	151
<b>8. Revisió de la planificació .....</b>	<b>153</b>
8.1. Revisió de l'estudi temporal .....	155
8.2. Revisió de l'estudi econòmic .....	157
<b>9. Treball futur .....</b>	<b>159</b>
9.1. Immediat .....	161
9.2. A llarg termini .....	161
<b>10. Conclusions .....</b>	<b>163</b>
10.1. Punt de vista professional .....	165
10.2. Punt de vista del departament de I+D .....	165
10.3. Punt de vista personal .....	165
<b>11. Bibliografia .....</b>	<b>167</b>
11.1. Llibres .....	169
11.2. Referències Web .....	169





## 1. INTRODUCCIÓ

1.1	Descripció	11
1.2	Descripció de l'empresa	12
1.3	Motivació personal	13
1.4	Problemàtica	14
1.5	Objectius i resultats esperats	15
1.6	Planificació inicial	16
1.6.1	Estudi temporal	16
1.6.2	Estudi econòmic	21
1.6.2.1	Costos de Hardware	21
1.6.2.2	Costos de Software	22
1.6.2.3	Costos de Recursos Humans	25
1.6.2.4	Cost Total	27
1.6.3	Estudi dels riscos	28
1.7	Organització de la memòria	29
1.8	Glossari	31



## 1.1 Descripció

Aquest projecte consisteix en l'especificació, disseny i implementació d'una interfície gràfica per a una aplicació de control d'inspecció en temps real, basada en visió per computador. Aquesta forma part d'un projecte més gran que és el desenvolupament de la nova aplicació d'inspecció que s'ha realitzat a l'empresa **Estudis Electro-Mecànics S.L. (E2M)**.

Al llarg d'aquesta memòria es faran referències constants a elements del projecte general, ja que hi ha molts aspectes que hi mantenen relació. Igualment també s'ha dedicat un apartat a descriure el funcionament de l'aplicació completa, ja que així serà més fàcil d'interpretar les relacions entre els diferents components que el componen.

La finalitat de l'aplicació general és, a partir de les imatges rebudes d'una càmera, aplicar-hi unes anàlisis que permetran extreure uns resultats. Aquests resultats podran servir per a validar la imatge rebuda o per a rebutjar-la.

S'ha volgut crear un sistema que serveixi per a totes les màquines que es fabriquen a l'empresa, les quals poden tenir finalitats molt diverses. Per això s'ha dissenyat un sistema genèric que controla molts aspectes del tractament i processament d'imatges a través del computador.

Degut a la gran envergadura que comporta una aplicació d'aquest tipus, entre els components del departament hem decidit treballar en paral·lel i realitzar cada un una part de les que componen aquest software.

El projecte que es detalla en aquesta memòria consisteix en l'especificació, disseny i implementació de la interfície gràfica de l'aplicació descrita.



*Màquina d'inspecció d'etiquetes d'ampolles, construïda per l'empresa E2M*

## 1.2 Descripció de l'empresa



Imatges de l'empresa E2M

Estudis Electro-Mecànics S.L (**E2M**) és una empresa de caràcter familiar amb ganes i possibilitats d'expansió tant en l'àmbit nacional com internacional. Va néixer l'any 1993 i es troba situada en un polígon industrial de Sabadell, disposant d'una planta d'oficines i una altra amb el taller mecànic i zona de proves dels equips.

L'activitat principal de l'empresa engloba pràcticament tots els punts de la cadena de valor del sector. Es dedica al disseny, fabricació, comercialització, instal·lació i manteniment d'equips electrònics-informàtics per inspeccionar la qualitat de producció de productes fabricats, tant en sèrie com personalitzats a les necessitats del client.

La diversificació del producte és una de les claus de l'empresa, així com la seva flexibilitat a l'hora d'enfrontar-se a projectes no estàndards. Les línies de producte comercialitzables es centralitzen en el sector del *packaging*<sup>1</sup> i són les següents:

- **Inspectors mitjançant visió artificial:** de nivell, etiqueta, tap, tancament de llaunes i marcatge, per a la indústria de l'envasat, cosmètica, farmàcia, automòbil, etc.
- **Inspectors mitjançant sensors:** de nivell, etiqueta (tant dins com fora de l'etiquetadora) i tap
- **Equips de producció:** orientadors i separadors d'envasos i taps.
- **Expulsors i divisors:** expulsors i divisors d'alta velocitat per a elements inestables o fràgils.
- **Sistemes de visió:** Visió per a la lectura de dates de caducitat, números de lot, referències, codis de barres, etc.
- **Control d'impressió:** software pel control, fóra de la línia, de les etiquetes d'un producte.



Màquina de control d'etiquetes

En tot cas l'empresa és flexible a les necessitats del client, ja que tot el procés de disseny, creació, muntatge i proves de les màquines es fa a la mateixa empresa, permetent així una màxima flexibilitat a l'hora de crear el producte.

Per a controlar les màquines que l'empresa construeix, realitzar les diferents inspeccions, obtenir i mostrar els resultats d'aquestes inspeccions, es disposa d'un departament d' **I+D** encarregat de desenvolupar les aplicacions requerides per a cada projecte, introduint així la intel·ligència necessària al sistema.

És des d'aquest departament on ha sorgit la necessitat de desenvolupar el projecte que es tracta en aquesta memòria.

<sup>1</sup> *Packaging*: Embalatge de productes d'una línia de producció.

### 1.3 Motivació personal

Des del primer moment vaig tenir clar que volia fer el PFC en una empresa, ja que així, a part de poder realitzar aquests crèdits que em manquen per acabar la carrera, podia adquirir experiència professional.

Al treballar en una empresa podria aprendre moltes altres coses a part dels propis coneixements que em pogués proporcionar la realització del projecte, i que em serien útils de cara al futur.

Vaig estar buscant una empresa que m'oferís la possibilitat de la realització del PFC, fins que vaig trobar l'empresa E2M, on em van plantejar la possibilitat de realitzar el projecte que descriu en aquesta memòria.

Personalment suposava un repte per a mi un projecte d'aquest tipus, ja que a les empreses que havia estat fent pràctiques anteriorment no havia tractat mai amb les tecnologies amb les que treballen aquí, ni havia vist mai cap empresa especialitzada en el camp que treballa aquesta.

Des de E2M se'm va proposar realitzar la interfície gràfica de la nova aplicació que estaven creant, i després de mirar les possibilitats que tenia de presentar-ho com a PFC, no vaig dubtar en acceptar-ho.

## 1.4 Problemàtica

Actualment l'empresa ja disposa d'una aplicació instal·lada dins de cada màquina, i que es va modificant per tal de satisfer les diferents necessitats i problemes que cada client presenta. Aquesta aplicació està basada en una tecnologia que ha quedat obsoleta pels requeriments de l'empresa, i cada vegada es fa més difícil realitzar-li qualsevol canvi, així com adaptar els nous components que s'introdueixen a les màquines.

Aquesta aplicació estava basada en una tecnologia MFC (*Microsoft Foundation Class Library*), que utilitza la API de Windows per a la creació d'aplicacions. A part, la seva implementació feia que qualsevol modificació fos molt difícil o impossible de realitzar. Tot plegat feia que ens trobéssim amb una sèrie d'inconvenients:

- **Mono-plataforma:**
  - Només funcionava amb Windows XP: no es podien realitzar aplicacions per a sistemes de software lliure com Linux.
  - Es necessitava molta feina per a poder-lo adaptar al nou Windows 7.
- **Interfície gràfica deficitària:**
  - Look & Feel antiquat: La interfície gràfica que es podia crear amb aquesta tecnologia presentava un model antiquat que no permetia donar una imatge de modernitat a l'empresa.
  - No permetia diferents resolucions de pantalla (no s'adaptava correctament a pantalles amb diferents proporcions)
  - No es podia configurar la interfície mitjançant un arxiu de configuració: quan es volia canviar algun aspecte de la interfície s'havia de tocar directament el codi i tornar a compilar el software sencer.
  - Poc intuïtiva: S'havien de tenir coneixements avançats de l'aplicació per a realitzar-hi qualsevol acció, ja que no es disposava d'accessos directes, i les rutes d'accés a les diferents opcions no eren prou intuïtives.
  - No es podien visualitzar estadístiques en temps real: no es disposava de gràfiques per a visualitzar l'historial de resultats. S'havien d'observar els diferents arxius de log per a obtenir les conclusions de les anàlisis.

Tots aquests motius van provocar que es decidís desenvolupar una nova aplicació que fos més modificable, robusta i adaptable a futurs canvis. En definitiva es vol un software que s'adapti a les necessitats reals de l'empresa, amb previsió dels possibles canvis que pugui haver-hi.

La interfície gràfica de l'aplicació anterior també s'ha de modificar totalment, creant una nova visió de l'aplicació general, així com una nova estructuració dels menús i diàlegs.

Aquesta nova interfície s'ha de dissenyar tenint en compte que serà diferent per a cada client, és a dir, que ha d'estar molt ben estructurada i ha d'oferir tota mena de facilitats per a poder ser altament configurable.

## 1.5 Objectius i resultats esperats

L'objectiu d'aquest projecte consisteix en realitzar una interfície gràfica per a una aplicació de gran envergadura, facilitant la comunicació entre els diferents components de cada capa.

Aquesta interfície ha de ser usable i entenedora, i ha de gestionar tots els elements que ens ofereix l'aplicació global. A l'hora d'implementar-la s'han de tenir en compte les regles i patrons establerts en tot el projecte, així com contemplar la possibilitat de futures ampliacions.

Per això es vol que aquesta interfície sigui modular, permetent així, de manera senzilla, afegir noves funcionalitats o ampliar les que ja tingui.

Aquesta nova interfície gràfica ha d'estar feta basada en les funcionalitats que ofereix el sistema, i a les que ha de poder accedir cada tipus d'usuari. També s'ha tingut en compte els problemes que donava la interfície de l'aplicació anterior, millorant aspectes com:

- Suport per a multi plataforma: S'ha de poder instal·lar l'aplicació en Windows i Linux.
- Disposar d'uns menús simples i vistosos, amb accés a totes les opcions que ens ofereix l'aplicació
- Realitzar un arxiu de configuració que permeti controlar l'estructuració de la interfície gràfica, oferint així una fàcil modificació d'aquesta, per a adaptar-la a les necessitats de cada client.
- Permetre una bona visualització independentment de la resolució de la pantalla que s'estigui utilitzant. Aquestes poden anar des de 5 a 21 polzades.
- Oferir la visualització a temps real de les estadístiques dels resultats, mitjançant gràfiques.

## 1.6 Planificació inicial

En aquest apartat es representa la planificació inicial del projecte, en el que hi ha un estudi del temps estimat que es trigarà a fer cada etapa, així com el cost econòmic que inicialment es creia que costaria.

Els costos tant temporals com econòmics representen només una guia representativa, ja que estan subjectes a canvis depenent de diferents factors externs que poden sorgir durant el projecte. Aquests possibles factors també els contemplem alhora de realitzar aquesta planificació, amb la intenció de tenir controlats el major nombre de variables per tal de realitzar una planificació més acurada.

### 1.6.1 Estudi temporal

A continuació es detalla la planificació temporal que es va creure convenient inicialment per a realitzar aquest projecte. Per a aquesta finalitat es van definir diferents tasques, cada una amb un temps aproximat d'execució.

Cada tasca es troba englobada dins d'alguna de les etapes del projecte.

A continuació es mostren detalladament aquestes tasques, segons l'etapa a la qual pertanyen:

#### Etapa d'inicialització:

<b>Tasca</b>	Estudi i definició del projecte
<b>Descripció</b>	Realització d'un estudi per a conèixer els objectius i l'abast del projecte.
<b>Durada</b>	5 dies
<b>Recursos</b>	Gerent / Comercial / Director del projecte / Client

<b>Tasca</b>	Anàlisi de requeriments
<b>Descripció</b>	Es plantegen les necessitats i funcionalitats que haurà de tenir la interfície gràfica.
<b>Durada</b>	5 dies
<b>Recursos</b>	Gerent / Comercial / Director del projecte / Client



**Etapa d'especificació i disseny:**

<b>Tasca</b>	Especificació
<b>Descripció</b>	Es defineixen les diferents funcionalitats que el sistema oferirà.
<b>Durada</b>	10 dies
<b>Recursos</b>	Director del projecte / Analista programador / Programador

<b>Tasca</b>	Disseny
<b>Descripció</b>	Es donaran les directrius per a la posterior implementació.
<b>Durada</b>	20 dies
<b>Recursos</b>	Analista programador / Programador

**Etapa de construcció del sistema:**

<b>Tasca</b>	Implementació
<b>Descripció</b>	Es generarà el codi de l'aplicació.
<b>Durada</b>	100 dies
<b>Recursos</b>	Analista programador / Programador

<b>Tasca</b>	Proves
<b>Descripció</b>	Es realitzaran tests per a comprovar el bon funcionament del sistema
<b>Durada</b>	100 dies
<b>Recursos</b>	Analista programador / Programador

**Etapa de posada en funcionament:**

<b>Tasca</b>	Posada en funcionament
<b>Descripció</b>	El sistema serà introduït en un ambient de producció
<b>Durada</b>	5 dies
<b>Recursos</b>	Analista programador / comercial / client / director projecte

**Etapa de documentació:**

<b>Tasca</b>	Redacció de la memòria
<b>Descripció</b>	Elaboració de la memòria final del projecte.
<b>Durada</b>	30 dies
<b>Recursos</b>	Programador

<b>Tasca</b>	Documents de final d'etapa
<b>Descripció</b>	Documents que es generen al finalitzar alguna de les etapes del projecte.
<b>Durada</b>	0 dies. Es consideraran fites del projecte.
<b>Recursos</b>	Programador

Algunes d'aquestes tasques s'executaran de manera paral·lela, com podrien ser l'etapa d'implementació i la de proves, ja que es creu convenient anar provant el que s'està implementant periòdicament, per tal de comprovar que tot compleix amb el que s'ha dissenyat.

A més, al final de la implementació, es comença a realitzar la posada en funcionament, per així poder fer proves d'ús intensiu per part dels usuaris.

Seguidament es mostra la planificació inicial mitjançant la taula de tasques i el diagrama de GANTT temporal que ens ajuda a situar gràficament el transcurs de les diferents etapes del projecte. S'han extret directament les captures de pantalla de l'esquema realitzat mitjançant el programa "*Microsoft Office Project 2007*".

S'han tingut en compte les vacances de Setmana Santa (una setmana) i de l'Agost (tres setmanes), així com les diferents festivitats durant les quals l'empresa està tancada.

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	[-] PFC	175 días	lun 02/03/09	vie 04/12/09		
2	[-] Inicialització	10 días	lun 02/03/09	vie 13/03/09		
3	Estudi i definició del projecte	5 días	lun 02/03/09	vie 06/03/09		Gerent;Comercial;Director projecte;Client
4	Anàlisi de requeriments	5 días	lun 09/03/09	vie 13/03/09	3	Gerent;Comercial;Director projecte;Client
5	[-] Especificació i disseny	30 días	lun 16/03/09	mar 05/05/09		
6	Especificació	10 días	lun 16/03/09	vie 27/03/09	4	Director projecte;Analista programador;Programador
7	Disseny	20 días	lun 30/03/09	mar 05/05/09	6	Analista programador;Programador
8	[-] Construcció del sistema	105 días	mié 06/05/09	vie 23/10/09		
9	Implementació	100 días	mié 06/05/09	vie 16/10/09	7	Analista programador;Programador
10	Proves	100 días	mié 13/05/09	vie 23/10/09		Analista programador;Programador
11	[-] Posada en funcionament	5 días	lun 26/10/09	vie 30/10/09		
12	Posada en funcionament	5 días	lun 26/10/09	vie 30/10/09	10	Analista programador;Comercial;Client;Director projecte
13	[-] Documentació	165 días	vie 13/03/09	vie 04/12/09		
14	Redacció de la memòria	30 días	lun 26/10/09	vie 04/12/09	10	Programador
15	[-] Documents finals d'etapa	140 días	vie 13/03/09	vie 30/10/09		
16	Definició inicial del projecte	0 días	vie 13/03/09	vie 13/03/09	4	Programador
17	Document de l'especificació	0 días	vie 27/03/09	vie 27/03/09	6	Programador
18	Document del disseny	0 días	mar 05/05/09	mar 05/05/09	7	Programador
19	Document de revisió de les proves	0 días	vie 23/10/09	vie 23/10/09	10	Programador
20	Document de revisió de la posada en funcionament	0 días	vie 30/10/09	vie 30/10/09	12	Programador

Taula de tasques extreta directament des d'una captura de pantalla del Microsoft Project

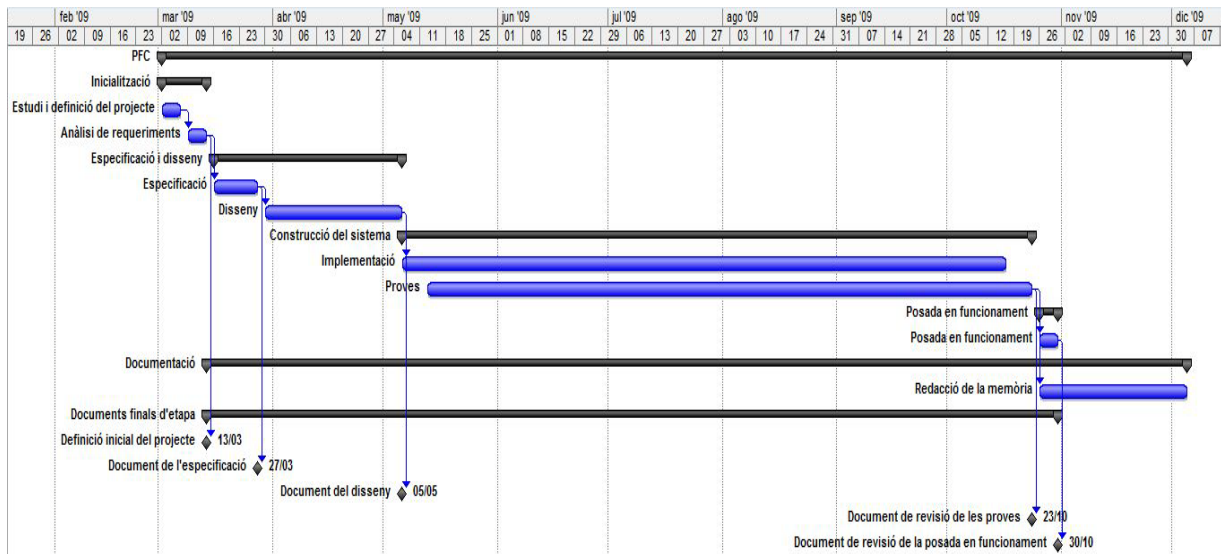


Diagrama de Gantt extret directament des d'una captura de pantalla del Microsoft Project

El número d'hores empleades en cada jornada és de 4, ja que només es treballa a mitja jornada. Per tant, de manera aproximada es podria definir una taula resum de la planificació de les hores de cada tasca, i global del projecte, com la següent (tenint en compte les tasques que es poden realitzar paral·lelament):

<b>Tasca</b>	<b>Duració (dies)</b>	<b>Hores</b>
Estudi i definició del projecte	5	20
Anàlisi de requeriments	5	20
Especificació	10	40
Disseny	20	80
Implementació	100 (+5 de proves)	400
Proves	100	400
Posada en funcionament	5	20
Redacció de la memòria	30	120
<b>TOTAL</b>	<b>175</b>	<b>700</b>

*Taula 1: Resum dels temps de cada tasca  
(tenint en comptes les tasques que es realitzen paral·lelament)*




### 1.6.2 Estudi econòmic

L'objectiu d'aquest apartat és oferir una previsió econòmica del desenvolupament del projecte, avaluant els costos que la realització d'aquest projecte té.

S'ha intentat incloure tot el que podria estar relacionat amb el projecte, començant amb l'amortització de les llicències d'ús del hardware i software usat, i acabant amb el cost del personal que ha intervingut en el desenvolupament d'aquest projecte.

#### 1.6.2.1 Costos de Hardware

Per al desenvolupament del projecte s'ha fet ús d'un ordinador personal, un servidor i un ordinador de proves més complert, amb les mateixes característiques que els que hi ha instal·lats a les màquines finals.

Equip	Característiques	Preu aproximat
	<ul style="list-style-type: none"> <li>• Intel Core 2 Duo, 2 GHz.</li> <li>• 2 GB RAM DDR2.</li> <li>• Disc dur 120 Gb .</li> <li>• Pantalla TFT 17 polsades.</li> </ul>	500 €
	<ul style="list-style-type: none"> <li>• Intel Core 2 Quad, 2.6 GHz.</li> <li>• 4 GB RAM DDR3.</li> <li>• Disc dur 500 Gb .</li> <li>• Pantalla TFT 17 polsades.</li> </ul>	1.000 €
	<ul style="list-style-type: none"> <li>• Intel Core 2 Duo, 2.2 Ghz.</li> <li>• 2 GB RAM DDR2.</li> <li>• Disc dur 200 Gb .</li> <li>• Pantalla TFT 17 polsades.</li> </ul>	600 €

Taula 2: Resum dels equips utilitzats en el projecte

Com que el temps d'amortització d'aquests equips és d'uns 3 anys, s'ha de ponderar el cost amb el temps que es creu que es necessitarà pel desenvolupament d'aquest projecte, aproximadament uns 9 mesos pel PC, uns 8 mesos pel servidor, i uns 2 mesos pel computador de proves.

Preu aproximat	% d'amortització	Cost
500 €	9 mesos / 36 mesos = 25 %	<b>125 €</b>
1.000 €	2 mesos / 36 mesos = 5 %	<b>50 €</b>
600 €	8 mesos / 36 mesos = 22 %	<b>133 €</b>
<b>TOTAL</b>		<b>308 €</b>




*Taula 3: Cost dels equips per al projecte*


### 1.6.2.2 Costos de Software

En aquest apartat es contempla el valor de les diferents aplicacions que s'han usat en el desenvolupament del projecte.

Es tenen en compte els diferents programes usats en les màquines descrites a l'apartat anterior, i només els que s'han utilitzat per a la realització d'aquest projecte.





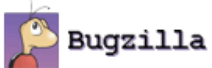


#### Eines d'ofimàtica i bàsiques del PC de treball:

Aplicació	Descripció	Cost
	<b>Microsoft Windows Xp.</b> Preu de la llicència inclòs en el preu del PC	-
	<b>The Gimp.</b> Llicència GPL	-
	<b>Microsoft Office Professional 2007.</b> Llicència anual	500 €

 Google Chrome	<p><b>Google Chrome.</b></p> <p>Codi obert</p>	<p>-</p>
<p><b>TOTAL</b></p>		<p><b>500 €</b></p>






Taula 4: Software d'ofimàtica utilitzat en el projecte

Eines del servidor:

Aplicació	Descripció	Cost
	<p><b>Ubuntu Server Edition.</b></p> <p>Llicència GPL</p>	<p>-</p>
	<p><b>MySQL.</b> Llicència d'empresa.</p> <p>Llicència anual</p>	<p>420 €</p>
	<p><b>MediaWiki.</b></p> <p>Codi obert</p>	<p>-</p>
	<p><b>Tortoise SVN.</b></p> <p>Llicència GPL</p>	<p>-</p>
	<p><b>Bugzilla.</b></p> <p>Llicència GPL</p>	<p>-</p>
	<p><b>Coppermine photo gallery.</b></p> <p>Codi obert</p>	<p>-</p>
	<p><b>Doxygen.</b></p> <p>Llicència GPL</p>	<p>-</p>
<p><b>TOTAL</b></p>		<p><b>420 €</b></p>

Taula 5: Software del servidor utilitzat en el projecte

**Eines de programació:**

Aplicació	Descripció	Cost
	<b>Llibreries Qt.</b> Llicència LGPL	-
	<b>Qt Creator.</b> Llicència LGPL	-
	<b>MinGW.</b> Llicència GPL	-
	<b>OpenCV.</b> Llicència BSD	-
	<b>StarUML.</b> Codi obert	-
<b>TOTAL</b>		<b>0 €</b>

*Taula 6: Software de programació utilitzat en el projecte*

Igual que en l'apartat anterior (en el que s'analitzen els costos de hardware) els costos de software també estan subjectes a un percentatge d'amortització corresponent al temps que s'han fet servir per aquest projecte. Aquests percentatges són identificatius, ja que és pràcticament impossible calcular aquests costos exactament.

En la següent taula es mostren els preus totals del software del projecte:

Tipus de cost	% d'amortització	Cost
Cost de les eines d'ofimàtica i bàsiques del PC de treball	15% del preu original (aproximadament)	<b>75 €</b>
Cost de les eines del servidor	Amortització del 50% (mig any)	<b>210 €</b>
Cost de les eines de programació	-	<b>0 €</b>
<b>TOTAL</b>		<b>285 €</b>

*Taula 7: Costos totals del software utilitzat en el projecte*



### 1.6.2.3 Costos Recursos Humans

Per a realitzar un estudi de possibles costos de recursos humans més verídic, s'han establert uns perfils, el preu per hora dels quals va variant segons el seu rol.

Aquests perfils són els corresponents a les persones que han intervingut en el transcurs del projecte:

<b>Perfil</b>	Gerent de l'empresa
<b>Descripció</b>	És el responsable dels contactes a alt nivell entre l'empresa i el client, així com de representar a l'empresa en els diferents actes públics.
<b>Cost</b>	0 € l'hora. No suposa cap cost afegit al còmput total del projecte que s'està tractant, ja que és qui s'encarrega de gestionar els sous dels treballadors de l'empresa.
<b>Unitats</b>	1

<b>Perfil</b>	Comercial
<b>Descripció</b>	Es el responsable de transmetre els possibles requeriments que el client demana per a la interfície gràfica de l'empresa, i del contacte directe amb aquest per a qualsevol tracte.
<b>Cost</b>	20€ l'hora. Aquest sou pot variar depenent dels èxits que comporti el projecte, ja que part del seu sou va per comissions.
<b>Unitats</b>	2

<b>Perfil</b>	Client
<b>Descripció</b>	És el que transmet les necessitats i contracta els serveis de l'empresa.
<b>Cost</b>	0€ l'hora. No suposa cap cost afegit al còmput total del projecte que s'està tractant.
<b>Unitats</b>	1

<b>Perfil</b>	Director del projecte
<b>Descripció</b>	És l'encarregat de controlar que el conjunt del projecte s'està duent amb consonància amb els terminis que s'havien marcat. També és el responsable de parlar amb els comercials i establir noves fites o canvis en la planificació del projecte.
<b>Cost</b>	35 € l'hora
<b>Unitats</b>	1

<b>Perfil</b>	Analista programador
<b>Descripció</b>	És l'encarregat de l'especificació, disseny i implementació del projecte, així com de provar que tot funcioni correctament.
<b>Cost</b>	25 € l'hora
<b>Unitats</b>	2

<b>Perfil</b>	Programador
<b>Descripció</b>	És l'encarregat de la implementació del projecte, encara que també se li atribueixen tasques d'especificació i disseny del software, així com de documentació del projecte.
<b>Cost</b>	10 € l'hora
<b>Unitats</b>	1

Si tenim en compte les hores que cada persona ha dedicat al projecte, podem resumir els costos deduïts dels recursos humans en la següent taula:

<b>Perfil</b>	<b>Nº d'hores</b>	<b>Cost</b>
Gerent de l'empresa	20	0 €
Comercial	50	1.000 €
Client	30	0 €
Director del projecte	90	3.150 €
Analista programador	240	6.000 €
Programador	660	6.600 €
<b>TOTAL</b>	<b>1.090</b>	<b>16.750 €</b>

*Taula 8: Costos de Recursos Humans*

---

#### 1.6.2.4 Cost Total

---

El cost total que es desprèn del desenvolupament del projecte és el següent:

<b>Tipus de cost</b>	<b>Cost</b>
Hardware	308 €
Software	285 €
Recursos Humans	16.750 €
<b>TOTAL</b>	<b>17.343 €</b>

*Taula 9: Cost total del projecte*

Com a consideració final, cal indicar que el cost real del hardware segurament serà inferior a l'indicat, ja que la inversió tant dels PC's com del servidor ha estat anterior a l'inici del projecte, amb el que la seva amortització ja estava assignada a d'altres projectes, però per a fer una representació més fidel del cost de l'entorn de treball utilitzat, s'ha cregut necessari incloure'ls en l'estudi.

A part també s'ha de tenir en compte que moltes de les hores assignades a recursos humans, que no són del recurs "Programador" o "Analista programador", han estat calculades fent servir una aproximació el més acurada possible, ja que durant aquestes hores també es dedicava a analitzar i desenvolupar el software global de l'aplicació, així com d'altres projectes. Igualment s'ha intentat extreure les hores que correspondrien al treball d'aquest projecte.

---

### 1.6.3 Estudi dels riscos

---

En aquest apartat es mostra un llistat amb els diferents factors que, quan es va realitzar la planificació inicial, es va creure que podrien fer variar aquesta planificació.

A l'hora de realitzar les estimacions inicials ja es van tenir en compte alguns d'aquests aspectes, però degut a la seva incertesa, no es podia definir quin seria el seu grau d'afectació dins de cada tasca del projecte. Així que es mostren com a possibles riscos:

- Riscos interns:
  - o Primera experiència en desenvolupament d'aplicacions utilitzant la dupla **Qt + software lliure**:
    - El procés d'aprenentatge es pot complicar depenent de la dificultat que presenti alguna de les opcions d'implementació del sistema.
- Riscos empresarials:
  - o Durant l'execució del projecte no es disposa del 100% del temps per a implementar-lo, sinó que també s'han d'atendre altres tasques del departament, com podrien ser:
    - Ajuda en la configuració d'altres equips.
    - Instal·lació i proves d'algunes màquines.
  - o Recursos limitats:
    - Es podria donar el cas de no tenir alguns dels recursos necessaris per a la implementació o proves del software en el termini establert, degut a problemes amb els costos o temps d'adquisició d'aquests recursos.
- Incidències personals:
  - o Possibles imprevistos personals d'alguna de les persones que intervé en el projecte. Aquests imprevistos podrien ser malalties, vacances, no disponibilitat, etc.
- Riscos externs:
  - o Influència del client final:
    - El client pot modificar algun dels requeriments durant l'execució del projecte que suposaria alguna variació en la planificació inicial.
  - o Multi plataforma:
    - Dificultats afegides a l'hora de portar el software a d'altres plataformes, com podria ser Windows 7.

## 1.7 Organització de la memòria

Aquesta memòria està dividida en diferents capítols que es descriuen breument a continuació:

- **Contextualització:**
  - Descripció del sistema global. Explicació de les seves funcions, així com petites pinzellades dels principals components i característiques que el formen.
- **Anàlisi de requisits:**
  - S'explica de forma detallada quines són les necessitats i funcionalitats que ha de cobrir el projecte
- **Especificació:**
  - En aquest apartat es presentarà el model conceptual de dades que s'ha fet servir per a realitzar el projecte, així com el model de casos d'ús.
- **Disseny:**
  - Aquí s'exposen les decisions de disseny que s'han pres en cada bloc del projecte per a poder complir amb l'especificació plantejada. També es pot veure el flux d'execució d'algunes de les funcionalitats descrites mitjançant els diagrames de seqüència.
- **Implementació:**
  - En aquest apartat es mostra quines han estat les estratègies adoptades (tecnologies utilitzades, gestió de la persistència de les dades...) per a implementar cada component que compon aquesta interfície gràfica.
- **Interfície gràfica desenvolupada:**
  - En aquest apartat podem fer-nos una idea de la visualització i funcionament dels diferents blocs de la interfície gràfica, mitjançant les diferents imatges de les captures de pantalla de cada part.
- **Treball Futur:**
  - Hi ha certs aspectes que s'han de millorar o acabar de concretar. Tot i que no estan requerits per a aquesta versió de la interfície, en aquest apartat s'exposen per a tenir-los en compte per a versions futures.
- **Revisió de la planificació:**
  - En aquest apartat es farà una valoració (revisió) dels temps i costos presentats al principi del projecte, fixant-nos sobretot ens les variacions que s'han produït.
- **Conclusions:**

- Es mostren separatament les conclusions del projecte des de diferents punts de vista: des del punt de vista professional, des del departament de visió i des del personal.
- **Bibliografia:**
  - Per tancar la memòria, en aquest apartat es cita tota la bibliografia que s'ha consultat per a elaborar el projecte.

## 1.8 Glossari

A continuació es mostra una llista de paraules (amb la seva corresponent definició) que apareixeran sovint al llarg d'aquesta memòria, i que se suposarà que es coneixen, permetent així una narració més fluida dels diferents aspectes del projecte:

- **VisioInspect2:**
  - o Nom de l'aplicació global que s'està desenvolupant al departament d' I+D de l'empresa E2M, i que servirà per a realitzar les inspeccions i controlar les futures màquines que es construeixin.
  
- **Widget:**
  - o Objectes d'interacció d'entrada o sortida amb l'usuari (que poden incorporar elements com botons, text, diàlegs o gràfiques) que permeten agrupar funcionalitats del programa.
  
- **Layout:**
  - o És la taula on es distribueixen els diferents widgets. Conté diverses files i columnes, on s'hi poden inserir widgets, permetent així una distribució correcta d'aquests.
  
- **Format:**
  - o És un conjunt de diferents aspectes de la configuració global del sistema. Un format en concret pot tenir una sèrie de processos, tractaments, informació de càmeres...
  - o Es fan formats per tal de poder carregar i guardar més ràpidament algunes opcions de configuració pels productes de la línia. Com que en una línia ens podem trobar més d'un tipus de producte (o la mateixa màquina pot estar en diverses línies), podem definir un conjunt de configuració per a cada tipus (format) de producte.
  
- **Tractament:**
  - o És l'element que realitza les anàlisis, encarregant-se de realitzar totes les transformacions necessàries a les imatges entrants i retornar els resultats i imatges esperats.
  - o Està implementat en un plugin i en podem trobar de diferents tipus: anàlisis d'etiqueta, anàlisi de nivell, anàlisi d'hamburguesa...
  
- **Procés:**
  - o És l'execució seqüencial d'un conjunt de tractaments. Cada procés és llançat en el seu propi thread. Per llençar un procés aquest ha de tenir com a mínim un dispositiu d'entrada ( una càmera per exemple) i un tractament assignats.
  
- **Plugin**
  - o És un complement de l'aplicació, que es relaciona amb aquesta per tal d'aportar-li nova informació i funcionalitats. Utilitzem els plugins definits per a QT, els quals es gestionen a través de llibreries dinàmiques (*dll* en sistemes windows i *so* en sistemes linux).

- En la nostra aplicació trobem plugins per a implementar tractaments i mecanismes d'adquisició del senyal, com les càmeres i l'entrada de senyals digitals.
- Aquestes funcionalitats s'implementen en un plugin per tal de ser independents del nucli de l'aplicació, permetent així la seva modificació sense haver de recompilar o reinstal·lar el software base.
- **Thread:**
  - És un fil d'execució del programa. Cada thread s'executa utilitzant recursos de la CPU independentment, permetent així que si hi ha diversos threads corrent paral·lelament, no es sobrecarregui només una part de la CPU, sinó que es pugui repartir més.
  - A més, el timing d'execució d'un thread no depèn de cap altre thread. Així que es poden estar executant paral·lelament diversos components (cada un en un thread diferent) dins de l'aplicació.
  - El Sistema Operatiu és el responsable de la gestió dels threads, realitzant l'assignació d'aquests a cada processador del sistema.
- **Grup de processos:**
  - Cada procés forma part d'un grup. Aquests grups serveixen per definir conjunts de processos que realitzen alguna feina similar o que es necessiten els uns als altres per poder funcionar.
  - Fins que no s'acaba un grup de processos no s'actualitza la pantalla principal amb les noves dades.
- **Resultats abstractes:**
  - Són resultats calculats fora dels processos, i representen la suma dels resultats del sistema (per exemple el resultat total és la suma dels resultats parcials de cada tractament realitzat).
- **Resultats de procés:**
  - Són un tipus de resultats que retornen alguns tractaments i que només s'utilitzen per a ser servits com a entrada d'altres tractaments.
  - Normalment són imatges que se'ls ha aplicat alguna transformació i que s'envien a d'altres tractaments perquè les analitzin.
- **Subscriure un resultat:**
  - Si un widget vol mostrar algun resultat o imatge resultat s'ha de subscriure a aquest, ja que els resultats només podran ser vistos per aquells widgets que hi estiguin subscrits. Un cop es genera un resultat, s'observa la llista de widgets subscrits al resultat i aquest és distribuït.
- **Arxiu de Log:**
  - Arxiu de text on hi ha registrat els diferents events, errors, canvis d'estat i "warnings" del sistema. Hi ha 2 tipus d'arxius de Log: *Log debug* (mostra tots els alertes del sistema, per a poder debugar més fàcilment en cas d'error) i *Standard Log* (mostra les alertes estàndards del sistema).



## 2. CONTEXTUALITZACIÓ

2.1	Introducció	35
2.2	VisiInspect2	36
2.2.1	Explicació	36
2.2.2	Conceptes generals	38
2.2.3	Components	40
2.2.4	Estats	44
2.2.5	Seqüències típiques	46
2.3	Interfície gràfica	49



## 2.1 Introducció

En aquest apartat de la memòria es mostraran les principals característiques i funcions que té el nou sistema que s'ha desenvolupat. No està centrat només en la part de la interfície gràfica (que és a la qual es refereix aquest projecte), sinó que es mostra una visió global del software abans d'aprofundir del tot en l'explicació, l'especificació i el disseny de la interfície gràfica.

Amb els diagrames i esquemes que es mostren en aquest apartat es podrà tenir una idea de la comunicació i jerarquia que hi ha entre els diferents components del sistema.

També es podran veure alguns esquemes referents a les funcionalitats més importants o més comunes del sistema. A l'hora d'escollir les funcionalitats a mostrar en aquest punt s'han tingut en compte aquelles que es podrien considerar bàsiques per al funcionament de l'aplicació, i que seran sempre igual independentment del client o de la màquina en la que estigui instal·lat el sistema.

Al final es detalla un punt dedicat a explicar quina és la funció de la interfície gràfica dins d'aquest sistema, així com les relacions i comunicacions que té amb la resta de components. Aquest punt servirà també com a introducció dels següents apartats de la memòria, on es detallen més profundament les característiques d'aquesta interfície.

Aquesta nova aplicació que s'ha creat s'anomena "**VisioInspect2**", fent referència a la funcionalitat principal pel qual ha estat dissenyat, que és la inspecció i anàlisi d'imatges a través de visió per computador.

## 2.2 VisioInspect 2

### 2.2.1 Explicació

El VisioInspect2 és una aplicació creada amb la principal finalitat de controlar el funcionament de les màquines que es construeixen a l'empresa E2M. Aquestes màquines capturen imatges dels productes que s'estan elaborant en algun punt de la línia de producció de l'empresa client, i les envien a l'aplicació. És aquí on es tracten les imatges rebudes, s'analitzen i s'extreuen uns resultats. A partir d'aquests resultats la màquina ha de ser capaç de realitzar l'acció pertinent, com podria ser, per exemple, el fet d'expulsar un producte defectuós de la línia.

Com s'ha vist a grans trets, la principal funció de l'aplicació és la de realitzar unes anàlisis a unes imatges entrants per tal de poder extreure'n uns resultats. A part d'això també ha de ser capaç de configurar alguns dels aspectes del sistema.

Tot seguit es mostra un llistat de les principal funcions que té l'aplicació VisioInspect2, i a les que es podrà accedir a través de la interfície gràfica. En aquest llistat només es mostren les funcionalitats principals que ha de tenir l'aplicació per funcionar correctament. A part d'aquestes en poden aparèixer de noves depenent de la màquina que s'hagi de controlar o del client amb el que s'està tractant.

- **Configuració del sistema i de les màquines:**
  - o L'aplicació ofereix a l'usuari l'opció de poder escollir i configurar el mode d'inspecció de les imatges, és a dir el mode en que la màquina realitza les fotografies. Aquest pot ser mitjançant un temporitzador (capturant una imatge cada cert temps) o realitzant una fotografia cada vegada que detecta que hi ha un element (mitjançant fotosensors).
- **Configuració de la captura d'imatges del programa:**
  - o L'aplicació permet seleccionar i configurar quina font d'entrada d'imatges escollim per al programa. Aquesta font pot ser alguna carpeta del disc dur que contingui imatges (bàsicament per a poder realitzar les proves i configuracions del sistema), o alguna de les càmeres que té la màquina.
- **Configuració de les anàlisis**
  - o Una anàlisi consisteix en l'execució d'una sèrie de processos sobre una imatge. Cada procés conté diferents tractaments que s'apliquen sobre la imatge.
    - Del resultat de tots aquests processos dependrà el resultat total de l'anàlisi.
    - Cada tractament està implementat en una llibreria (plugin) a part, permetent així una certa independència de les funcions bàsiques del VisioInspect2 i les específiques de cada tractament.
    - En aquests plugins és on s'aplica més en profunditat les tècniques de visió de computador per a tractar i inspeccionar les imatges.
    - El programa permet carregar i configurar els plugins, així com els processos i els tractaments a través de diferents menús de configuració.

A part també es permeten tenir diferents configuracions de les anàlisis englobades en diferents formats, permetent així carregar una configuració ràpidament per a cada producte que s'estigui tractant al sistema.

- **Control de la inspecció**
  - L'aplicació permet engegar i parar la inspecció en qualsevol moment.
  
- **Visualització de resultats**
  - L'aplicació ofereix diferents opcions de visualització de resultats, ja sigui mitjançant resultats numèrics, gràfiques, histogrames...

A part també es pot visualitzar, escollir i guardar la imatge a mostrar a la pantalla després d'aplicar-hi els diferents tractaments. Es pot escollir si es vol veure la imatge original o la resultant d'algun d'aquests tractaments.
  
- **Gestió d'usuaris**
  - L'aplicació ofereix una base de dades d'usuaris amb privilegis per accedir a l'aplicació. Des de la interfície gràfica es poden gestionar aquests usuaris.
  
- **Arxiu de configuració**
  - Tota la configuració del programa (configuració de les anàlisis, dels resultats, de la base de dades, de la interfície gràfica...) es troba en un arxiu XML que es llegeix cada vegada que s'engega l'aplicació.

El programa pot carregar un arxiu de configuració diferent o guardar la configuració actual. Igualment, quan es tanca l'aplicació, es guarda novament tota la informació a l'arxiu.

El fet de tenir-ho englobat en un sol arxiu fa que sigui molt més senzill i directe qualsevol canvi a la configuració.

Paral·lelament es disposa d'un arxiu XSD per a comprovar, cada vegada que iniciem l'aplicació, que l'arxiu de configuració està ben escrit i definit.



*Màquina de proves de l'empresa E2m,  
on hi va connectat un PC amb l'aplicació **VisioInspect2***

---

## 2.2.2 Conceptes generals

---

L'aplicació VisioInspect2 està implementada utilitzant el llenguatge C++ mitjançant les llibreries de Qt, amb l'ajut d'altres llibreries i utilitats complementàries per a realitzar la part gràfica i el tractat d'imatges. A l'apartat dedicat a la implementació estan més detalladament explicades totes les tecnologies i mètodes utilitzats [Apartat 6].

El software està dividit en diferents components, per tal de poder estructurar totes les funcions que ha de cobrir.

Tots aquests components estan estructurats i dissenyats per tenir un acoblament baix [WWW01], i el més flexible i adaptable possible, en el qual els canvis es puguin fer de la manera més simple i les ampliacions siguin factibles.

### Arquitectura de l'aplicació

L'Orientació a objectes està present en tot el disseny. A més s'ha decidit utilitzar el patró arquitectònic en **tres capes** per tal de definir l'arquitectura de l'aplicació:

- **Capa de presentació:** Capa mitjançant la qual els usuaris interaccionen amb el sistema fent ús de les funcionalitats que aquest ofereix.
- **Capa de domini:** Encarregada d'implementar el domini de la solució i proporcionar les funcionalitats que ofereix el sistema. És aquí on, a part d'habilitar les funcionalitats ofertes per la capa de presentació, estan implementats els plugins que fan les anàlisis.
- **Capa de gestió de dades:** Encarregada de mantenir la persistència de les dades. En aquesta capa hi podem trobar una base de dades i d'altres estructures que mantenen en memòria certes dades i informació necessàries pel funcionament de l'aplicació.

### Controladors i missatges

Per a comunicar components de capes diferents es fan servir els controladors, encarregats d'enllaçar les funcions dels mòduls d'una capa amb les seves correlatives a l'altra capa.

Cada controlador té una classe "*Manager*", que és un *Solito*<sup>1</sup> que l'inicialitza al principi de l'aplicació. Això s'ha fet així per a poder tenir un control sobre l'ordre d'inicialització de cada controlador i de cada mòdul. Hi ha una classe encarregada d'anar cridant els "*Managers*" de tots els controladors en l'ordre que toca al principi de l'execució del programa. Un cop iniciats tots els controladors, ja es pot començar a usar la resta de components del sistema.

Per a comunicar els controladors entre sí i entre els diferents components, s'ha definit una política de missatges. Aquesta consisteix en que si un mòdul o component ha de cridar alguna funció a través del controlador, li envia un missatge informant-li de l'acció que ha de realitzar. El controlador rep el missatge i el posa en una cua de missatges, els quals s'aniran atenent seqüencialment quan el controlador no estigui ocupat. Amb això evitem un possible

---

<sup>1</sup> Patró de disseny que consisteix en fer que una classe tingui una sola instància, proporcionant-li també un punt d'accés global a ella [SOF01]

coll d'ampolla als controladors, ja que els missatges sempre s'atendran un per un en l'ordre en que han arribat.

L'aplicació té un controlador principal a la capa de domini que s'encarrega d'atendre la majoria dels missatges que corren per l'aplicació. Aquest serà l'encarregat de distribuir la feina entre els diferents components.

A la capa de domini també trobem altres controladors que gestionen diferents mòduls d'aquesta capa, i d'altres que s'encarreguen de la comunicació amb la capa de dades.

A la capa de dades no hi trobem cap controlador, ja que s'accedeix directament als mòduls d'aquesta capa des dels controladors de la capa de domini o de presentació.

A la capa de presentació trobem un sol controlador que gestiona els elements de la finestra principal. També serveix per passar missatges al controlador principal, i cridar altres funcions d'altres controladors de la capa de domini o de dades.

Per a evitar una possible saturació del sistema, s'han repartit l'execució dels components més crítics en threads diferents de computació:

- El Controlador general i el controlador de la interfície gràfica estan en un thread separat de la resta, per a poder tractar els diferents missatges tranquil·lament. Aquests poden acceptar fins a 1000 missatges<sup>2</sup> a la cua de processament.
- La finestra principal està també en un thread separat, per a poder gestionar tots els Widgets i diàlegs sense dependre de l'execució d'altres accions de l'aplicació. Així es permet una reacció immediata a qualsevol acció que realitzi l'usuari sobre aquesta pantalla.
- Cada procés (conjunt de tractaments) s'executa en un thread diferent per tal de no interferir en l'execució global del programa.

En els següents punts es pot observar més detalladament i gràficament l'estructuració, els estats i el funcionament del sistema.

---

<sup>2</sup> Aquest número s'ha definit després de realitzar unes proves no gaire exhaustives. S'hauria de realitzar un estudi més a fons per a determinar-lo exactament

---

### 2.2.3 Components

---

En aquest apartat s'expliquen els components (o mòduls) que formen part del sistema, així com la interrelació que hi ha entre ells.

Cada component conté diferents classes que gestionen una mateixa funcionalitat.

Ja s'ha parlat dels diferents tipus de controladors que hi ha, però aquí es pot observar on estan situats conceptualment a dins del sistema, i veure a quin o quins components controlen.

A continuació es mostra una descripció de cada un dels mòduls, seguit de l'esquema global de components, que ens ajudarà a situar-los gràficament.

#### Capa de presentació

- GuiController: És l'encarregat de controlar els components de la interfície gràfica, i de comunicar-los amb els controladors de la capa de domini, enviant els missatges o realitzant les peticions necessàries.
- MainWindow: És l'encarregat de gestionar tots els elements de la pantalla principal, entre els que hi podem trobar les diferents pantalles que s'hi mostren, així com els diferents menús i diàlegs que hi poden aparèixer. També és l'encarregat de comunicar-se amb el controlador de la capa de presentació, enviant-li els missatges que li arriben dels diàlegs i dels widgets o pantalles.
- Diàlegs: Gestiona els diferents diàlegs de l'aplicació. Aquí hi podem trobar els diàlegs de la pròpia pantalla principal, així com els que apareixen de les opcions dels menús.
- GUIWidgets: Conté els diferents Widgets que es poden mostrar a la pantalla principal. Aquí hi podem trobar els widgets per a mostrar resultats, per a controlar el sistema, o per a obrir diàlegs de configuració, entre d'altres.

#### Capa de domini

- Controller: És el controlador general de l'aplicació. S'encarrega de tractar tots els missatges que es produeixen durant l'execució, consultant constantment a la cua de missatges (MessageQueue). Al tractar aquests missatges, delega la feina als components corresponents, siguin de la capa de domini o de presentació.
- Process Manager: Crea i destrueix els processos que hi ha configurats. També és l'encarregat de gestionar els diferents tractaments (implementats en els plugins), així com les entrades (càmeres) i les sortides (resultats i imatges resultants), de tots els processos.
- Acquire: Gestiona els diferents mètodes d'adquisició d'imatges, així com les càmeres i els dispositius d'entrada / sortida analògiques i digitals.

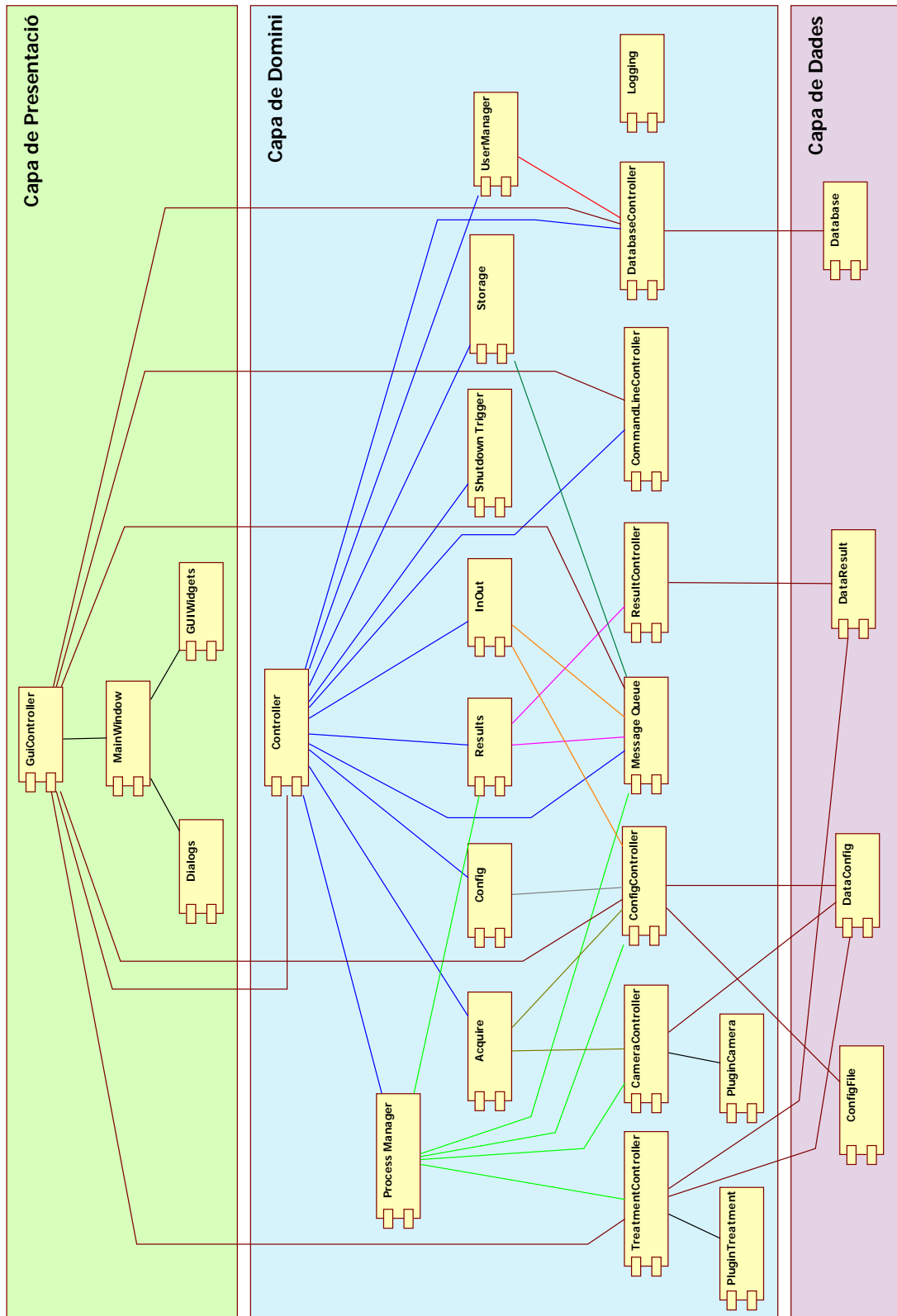


- Config: Gestiona la configuració global del sistema. És l'encarregat de llegir i escriure a l'arxiu de configuració (a través d'un controlador), així com de mantenir la configuració global de tot el sistema.
- ConfigController: És l'encarregat de gestionar la configuració del sistema. Llegeix i escriu a l'arxiu de configuració, i hi realitza les modificacions pertinents. També gestiona les estructures de dades que guarden informació de la configuració.
- Results: Gestiona les sortides dels processos. Entre aquestes sortides hi podem trobar els resultats de cada tractament i les imatges resultants de l'aplicació de cada un.
- ResultController: Gestiona els resultats de les anàlisis, i comunica aquests resultats amb les estructures de dades necessàries per a emmagatzemar-los en memòria.
- InOut: Gestiona tots els elements externs a l'aplicació, és a dir tots els dispositius de la màquina que són necessaris de controlar i d'obtenir-ne dades.
- Shutdown Trigger: Gestiona el tancament de l'aplicació. S'encarrega que tots els components finalitzin correctament.
- Storage: Guarda imatges resultants a disc.
- Message Queue: Conté una cua amb tots els missatges que han enviat els diferents components del programa i que estan pendents de ser processats pel controlador principal o pel controlador de la interfície gràfica.
- Logging: Gestiona els missatges que han d'anar als arxius de Log de l'aplicació. A part d'escriure els missatges, també s'encarrega de mostrar-los a les pantalles indicades per aquesta funció.  
  
\*En el diagrama general de components no s'hi poden observar les connexions dels diferents components amb aquest, ja que tots hi tenen accés i hi poden accedir, per tant s'han omès les línies que representen la connexió amb aquest mòdul per tal de simplificar el diagrama i fer-lo més llegible.
- TreatmentController: Controla i gestiona els diferents plugins de tractament.
- PluginTreatment: Representa un plugin de tractament de les anàlisis.
- CameraController: Controla i gestiona els diferents plugins de càmera.
- PluginCamera: Representa un plugin de càmera del sistema.
- UserManager: Controla totes les gestions referides als usuaris del sistema, i s'encarrega de la comunicació amb el controlador de la base de dades per a obtenir i modificar aspectes dels usuaris.
- CommandLineController: És l'encarregat de llegir la línia de comandes quan s'executa l'aplicació. És l'encarregat de llegir i interpretar els arguments que se li poden passar a l'executable de l'aplicació quan s'inicia.

- DatabaseController: És l'encarregat de la comunicació amb la base de dades. És el que es connecta amb el gestor de base de dades i hi envia les sentències SQL.

### Capa de dades

- DataConfig: Gestiona l'escriptura i lectura de l'arxiu de configuració de l'aplicació.
- DataResult: Gestiona l'emmagatzematge en memòria dels diferents resultats i imatges resultats de cada tractament. No es guarda tot l'historial d'imatges i de resultats a memòria, ja que sinó s'acabaria molt aviat, sinó que només es manté l'última que s'ha rebut de cada tractament.
- Database: Representa la base de dades de l'aplicació, controlada per un sistema gestor de base de dades MySQL.
- ConfigFile: Representa l'arxiu XML de configuració del sistema.



Esquema de components de l'aplicació VisioInspect2

---

## 2.2.4 Estats

---

Fins ara s'ha explicat l'estructura de l'aplicació així com les interrelacions entre els diferents controladors i components que la componen.

Un altre aspecte a destacar són els diferents estats en els que podem trobar el programa. Segons les funcions que està realitzant l'aplicació en cada moment podem considerar que aquest es troba en un estat o en un altre.

A part, també cal mencionar que hi ha un component encarregat de comprovar les transmissions entre estats, evitant així algun possible error de codi o de funcionament incorrecte de l'aplicació.

A continuació es mostra una descripció de cada un dels estats, seguit de l'esquema que ens ajudarà a saber la interrelació entre ells, permetent-nos saber així quina navegació podem realitzar a través de cada un.

### **Init**

És l'estat en que es troba el programa mentre s'estan inicialitzant tots els seus components.

### **Closing**

És l'estat en que es troba el programa mentre s'estan inhabilitant tots els seus components, i tancant totes les seves finestres i diàlegs.

### **Idle**

És l'estat en que es troba el programa quan no està fent res, és a dir quan està a l'espera que l'usuari hi realitzi alguna acció.

### **Analysis**

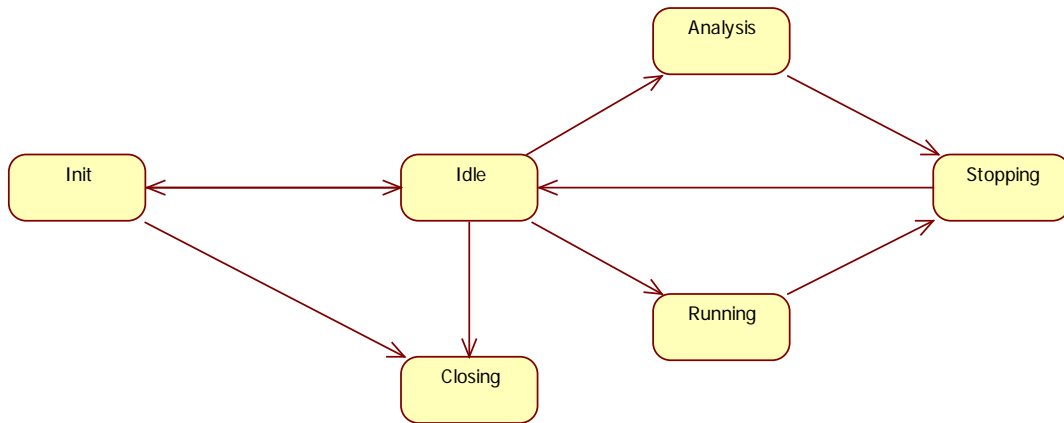
És l'estat en que es troba el programa quan està executant una anàlisi.

### **Running**

És l'estat en que es troba el programa quan s'ha iniciat la inspecció, és a dir quan estan corrent totes les anàlisis.

### **Stopping**

És l'estat en que es troba el programa quan s'ha premut el botó d'aturar la inspecció, és a dir quan el programa està finalitzant totes les anàlisis.



*Esquema dels possibles estats en que pot estar l'aplicació **VisioInspect2**.*

### 2.2.5 Seqüències típiques

Després d'haver descrit l'estructuració i els estats del programa, ja ens podem fer una idea de quines seran les funcionalitats que aquest ens ha d'oferir. En aquest apartat es mostren les principals accions que es realitzen en el programa, de manera esquemàtica, per a tenir una idea de com es comuniquen els diferents controladors i components descrits anteriorment.

#### Inicialització del sistema

En aquest esquema podem veure quin és el camí que es segueix quan s'inicia el programa, per tal d'iniciar tots els controladors i components. S'ha posat d'una forma genèrica en aquest esquema, però el funcionament és exactament el mateix per a tots els controladors i components.

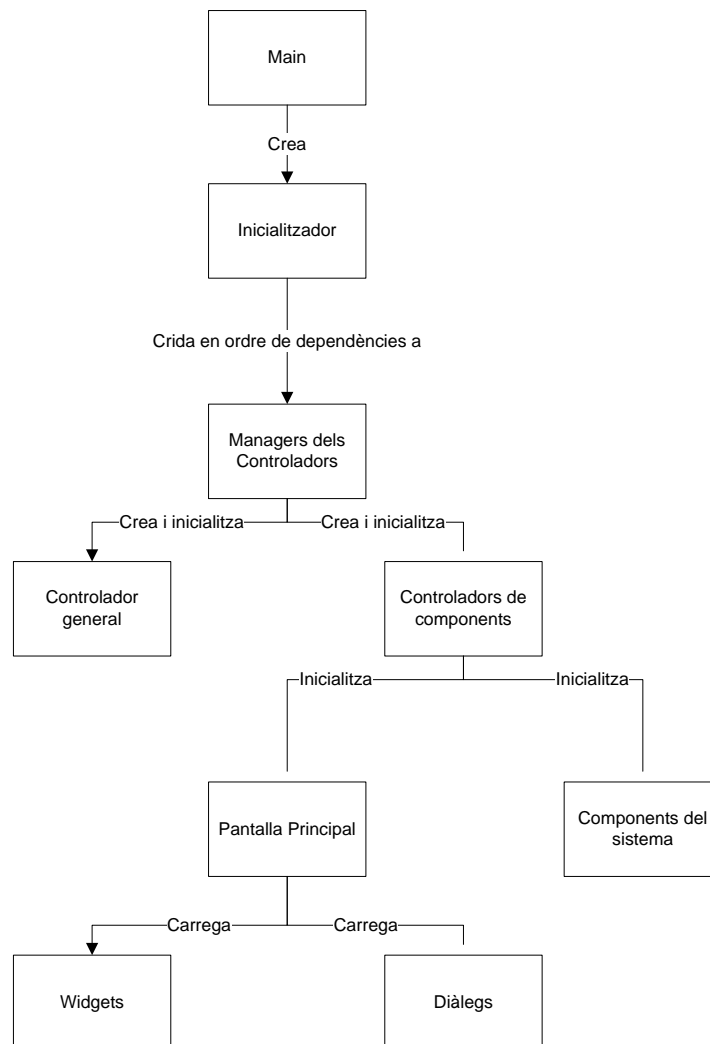


Diagrama explicatiu de la inicialització del sistema

#### Iniciar la inspecció

En aquest esquema podem veure el camí que es segueix quan l'usuari decideix iniciar la inspecció, prement el botó corresponent. Aquí podem veure com s'activen les anàlisis implementades en els diferents tractaments (plugins).

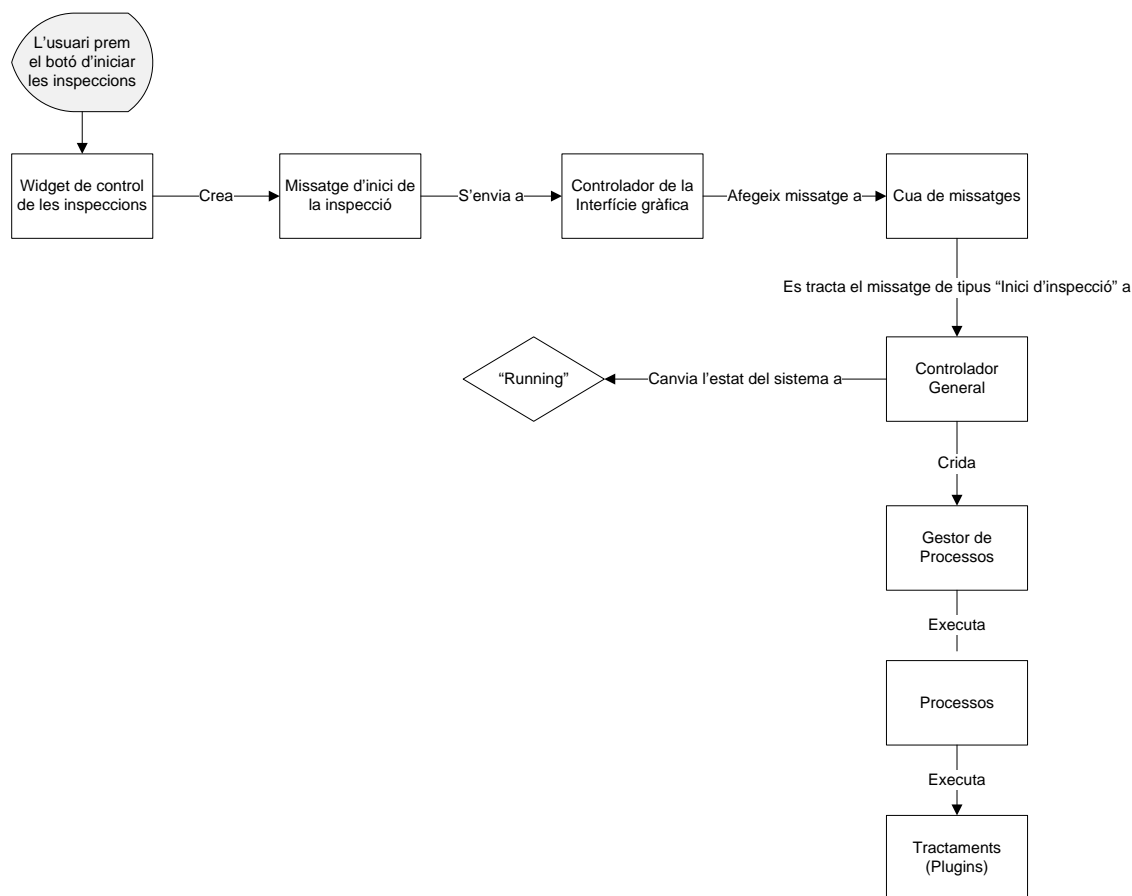


Diagrama explicatiu del camí que es segueix quan s'inicia una inspecció

### Mostrar els resultats de les anàlisis

En aquest esquema podem veure el camí que es segueix per a mostrar els resultats dels tractaments per pantalla. Es mostra des que es genera el resultat al tractament (és a dir, a partir de la finalització del diagrama anterior), fins que és demanat pel procés i s'envia a la pantalla principal per mostrar-se. Tot aquest camí d'enviament dels resultats està representat en aquest esquema.

Paral·lelament també podem observar que primerament s'ha de subscriure cada widget al resultat que ha de mostrar.

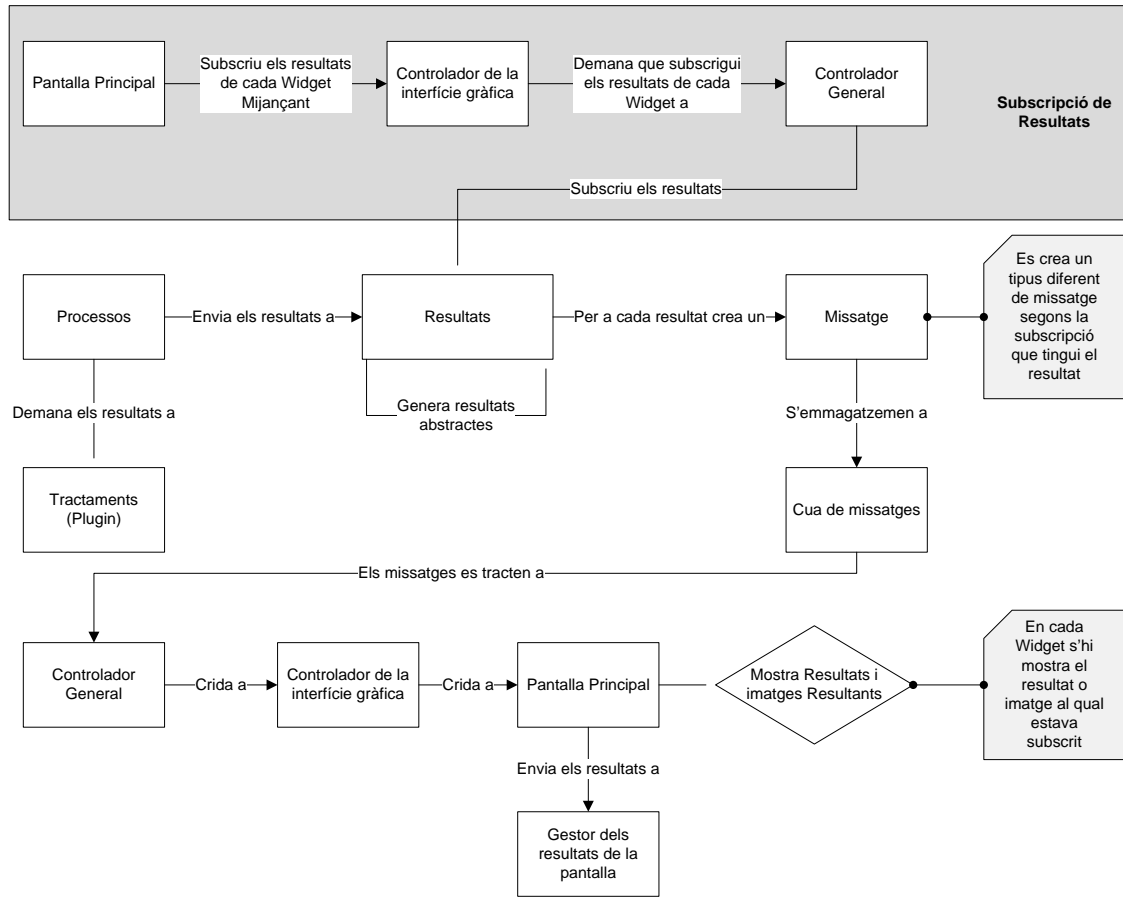


Diagrama explicatiu del camí que es segueix per a poder mostrar els resultats i les imatges resultants per pantalla després d'haver realitzat una inspecció



## 2.3 Interfície gràfica

Un cop explicada l'estructuració, les funcionalitats bàsiques i els estats de l'aplicació ja ens podem centrar en profunditat en la interfície gràfica.

Aquest projecte d'interfície gràfica implica l'especificació, el disseny i la implementació de la capa de presentació de l'aplicació descrita anteriorment.

Alguns dels aspectes bàsics que s'han de tenir en compte a l'hora de realitzar aquesta interfície gràfica són:

- La comunicació entre els widgets i diàlegs de la pantalla principal i la capa de domini i la de dades es realitzarà a través del controlador de la interfície gràfica (GuiController). Aquest serà l'encarregat de comunicar-se amb els altres controladors del sistema.
- La comunicació des dels widgets i diàlegs cap el controlador serà mitjançant missatges que aquest emmagatzemarà a una cua i els gestionarà quan estigui disponible.

Altrament des del controlador es poden cridar directament les funcions de la pantalla principal que no impliquin la creació de cap element nou. Si la funcionalitat a realitzar implica la creació d'algun element nou s'haurà de fer mitjançant l'enviament de "*signals*" [Apartat 6.2] des del controlador fins a la pantalla principal, ja que es troben en threads diferents. Això es fa per tal de poder mantenir el sincronisme de cada un dels components, permetent així que la pantalla principal els atengui quan toqui.

- Cada plugin tindrà la seva pròpia interfície gràfica de configuració, i podrà ser mostrada des del diàleg de la interfície gràfica global que li toqui. En aquesta memòria però, no es tindrà en compte el disseny d'aquestes pantalles de configuració dels plugins, ja que cada un pot ser diferent i contenir elements totalment independents de la resta del sistema. Dependran exclusivament de les funcions que requereix cada client.
- L'estructuració de la interfície gràfica dependrà de la configuració que es llegeixi de l'arxiu XML destinat a aquest fi. Si es vol realitzar algun canvi en el funcionament d'algun dels widgets i disposició d'aquests a la pantalla principal, s'haurà de fer editant directament aquest arxiu.



### 3. ANÀLISI DE REQUISITS

3.1	Introducció	53
3.1.1	Abast del projecte	53
3.2	Requisits no funcionals	54
3.3	Requisits funcionals	55



## 3.1 Introducció

La realització de l'anàlisi de requisits actualment és indispensable per qualsevol projecte, ja que permet definir els objectius i funcionalitats que haurà d'oferir el sistema mitjançant idees simples, sense necessitat d'entrar en detalls tecnològics.

Per dur a terme aquesta fase s'ha de recopilar informació de què és el que s'espera de la interfície o què és el que els usuaris necessiten. Per reunir tota aquesta informació s'han realitzat diverses reunions entre els diferents representants dels departaments de l'empresa, així com un estudi dels aspectes positius i negatius de l'actual sistema de visió, analitzant les seves mancances.

En aquest capítol es mostra una primera definició dels objectius que s'han de complir en el projecte, i més precisament dels requisits que ha de tenir, explicat d'una manera senzilla i no ambigua.

### 3.1.2 Abast del projecte

El sistema ha de representar una interfície gràfica que permeti a l'usuari realitzar totes les funcions que hi ha implementades a la capa de domini, d'una manera còmoda, senzilla, ràpida i vistosa.

Els beneficis que s'obtindran amb la inclusió d'aquesta interfície gràfica al sistema seran:

- Permetre als usuaris manipular els components del sistema d'una manera més senzilla i ràpida.
- Millorar la imatge de l'empresa realitzant una interfície vistosa i professional.
- Millorar la qualitat d'atenció al client.
- Millorar la interfície anterior, creant-ne una de més funcional, responent a les exigències i requisits tant del client com dels tècnics de l'empresa E2M que hauran de configurar els sistemes.
- Facilitar els possibles canvis que hi pugui haver a la interfície, permetent que es puguin realitzar interfícies amb distribucions diferents per a cada client, de manera senzilla i relativament ràpida.

## 3.2 Requisits no funcionals

### Eficiència

- El sistema ha de resultar suficientment eficient per tal de permetre un ús adequat.
  - o Per això, s'han de tenir en compte les funcionalitats de la capa de domini que requereixin més temps, i gestionar-les des de la interfície sense que apareguin errors ni que l'usuari es pugui pensar que alguna cosa no funciona.
- La interfície exercirà com a primera capa d'errors, inhabilitant a l'usuari les funcionalitats que no es poden realitzar en algun moment determinat, ja que es podria provocar algun error.

### Usabilitat

- El sistema ha de ser intuïtiu i fàcilment usable. La complexitat de les funcions ofertes ha d'anar amb consonància amb els coneixements del perfil d'usuari que estigui usant l'aplicació.
- Les accions han d'estar presentades en una interfície senzilla, amb icones clares i representatives.
- La interfície ha d'estar pensada per a possibles futures instal·lacions de l'aplicació en sistemes amb pantalles tàctils.
- S'ha de tractar de respectar al màxim el procés mental dels usuaris, realitzant camins fàcils i intuïtius per accedir a les funcionalitats.
- La imatge global de la interfície ha de ser visualment atractiva, ben estructurada i ben repartida, però mantenint una aparença professional, que no sembli un joc.

### Tecnologies

- La tecnologia utilitzada per desenvolupar l'eina serà C++ combinat amb la llibreria de Qt.
  - o El motiu d'aquesta decisió és degut a la voluntat de mantenir una coherència amb el llenguatge utilitzat per implementar els mòduls de les capes de domini i de dades.
- El software ha de poder ser multi plataforma, s'ha de poder executar sobre Windows XP, Windows 7 i Ubuntu 9.10 (Linux), com a mínim.

### Seguretat i privilegis

- Els usuaris hauran d'accedir a l'aplicació mitjançant un nom d'usuari i una contrasenya, la qual els oferirà uns privilegis sobre les funcionalitats de l'aplicació.
- Els requisits de seguretat vénen determinats per la normativa CFR21 Part 11[PHAR02].
  - o Aquesta normativa defineix aspectes com que s'han de registrar tots els intents d'accés a l'aplicació, s'ha de registrar cada acció realitzada per l'usuari, etc.

### 3.3 Requisits funcionals

#### Execució de l'aplicació

- Els usuaris s'han de poder registrar a l'inici de l'aplicació, per a poder ser identificats amb el seu rol.
- Els usuaris han de poder tenir control sobre l'execució de les inspeccions. Han de poder iniciar i aturar les anàlisis en qualsevol moment.
- S'han de poder recollir i mostrar els missatges del Log del sistema.
- Els usuaris han de poder veure les imatges resultants de les anàlisis, la imatge original, i el temps de cada anàlisi.
  - o També s'ha de poder escollir quina imatge resultant es vol veure.
- Els usuaris han de poder visualitzar els resultats, mostrant aquells quan han estat satisfactoris, erronis i el total.
- A part dels resultats numèrics, aquests també s'han de poder visualitzar gràficament a través de gràfics identificatius. S'han de poder configurar manualment, i han de ser:
  - o Gràfic de barres per a mostrar tots els resultats de les anàlisis.
  - o Histograma per a mostrar el resultat d'algun tractament que tregui algun valor numèric que es pugui visualitzar.
  - o Una imatge o requadre de color identificatiu per a mostrar el resultat de l'última anàlisi.
  - o Mostrar un sol resultat.
- Aquests resultants, juntament amb la informació dels gràfics i les imatges, s'han de poder reiniciar i posar a zero en qualsevol moment.
- S'ha de poder canviar de format, així com eliminar o afegir-ne algun en qualsevol moment que el programa no estigui realitzant la inspecció.
- Hi ha d'haver accessos directes als menús de configuració dels tractaments de les anàlisis. Aquests accessos directes s'han de poder configurar manualment per tal d'indicar quins es volen mostrar.
- S'ha de mostrar en tot moment el Logo de l'empresa E2M (fabricant del sistema) i de l'empresa client, permetent accedir a la web de cada una senzillament des de l'aplicació.

### Configuracions del sistema

- S'ha de poder carregar una nova configuració des d'un arxiu XML, així com guardar la configuració actual.
- S'ha de poder tancar l'aplicació alternativament al botó de tancar de la finestra.
- Els usuaris que el seu rol els ho permeti, han de poder configurar:
  - o L'adquisició de les imatges.
  - o Les anàlisis, permetent la configuració dels processos i tractaments de cada un.
  - o Les rutes dels plugins.
  - o El mode d'inspecció de les imatges.
- S'ha de poder mostrejar el resultat d'un sol tractament de forma gràfica i numèrica.
- Els usuaris han de poder guardar les imatges resultants o les errònies a disc, si ho desitgen.
- Els usuaris que tinguin privilegis per a fer-ho, han de poder gestionar el grup d'usuaris de l'aplicació, permetent afegir-ne de nous, eliminar-ne algun que ja existeixi, o canviar d'usuari mentre l'aplicació està engegada.
- Els usuaris han de poder veure la informació referent a la versió de l'aplicació (revisió del control de versions del software), i la informació de l'empresa.



## 4. ESPECIFICACIÓ

4.1	Introducció	59
4.2	Model de casos d'ús	60
4.2.1	Descripció dels actors del sistema	60
4.2.2	Descripció dels casos d'ús	61
4.2.2.1	Pantalles i widgets de l'aplicació	61
4.2.2.2	Menús de l'aplicació	74
4.3	Model conceptual de dades	91
4.3.1	Diagrama de classes global de la interfície gràfica	92
4.3.2	Diagrama de classes del Controlador	93
4.3.3	Diagrama de classes del Controlador de la interfície gràfica	93
4.3.4	Diagrama de classes de la Pantalla Principal	94
4.3.5	Diagrama de classes per la gestió dels Missatges	94
4.3.6	Diagrama de classes per la gestió dels Diàlegs	96



## 4.1 Introducció

A continuació es procedirà a l'especificació del projecte. Aquest pas consisteix en definir el què fa la interfície i com es comporta, així com descriure tots els elements que la componen.

Aquesta operació és molt important a l'hora de la generació de software, ja que ens definirà el mapa o l'estructura a seguir alhora de construir l'aplicació. Per a fer-ho ens basarem en els models que ens proporciona el llenguatge UML (*Unified Model Language*) [ES07].

El llenguatge UML es basa en models per a poder representar sistemes, no només la seva visió estàtica (Diagrames de classes), sinó que també permet definir la seva visió dinàmica (Diagrames de seqüència, diagrames d'estat, etc.). Al mateix temps, al ser un llenguatge de modelatge comú, es crea una documentació i uns esquemes que qualsevol desenvolupador amb coneixements de UML és capaç d'entendre, independentment del llenguatge de programació utilitzat en la implementació.

Per a poder realitzar tot això, UML defineix uns models que estan representats mitjançant diagrames per a una major comprensió. Entre els diagrames més importants s'hauria de destacar els de les classes, els de seqüència, els de casos d'ús i els d'estat.

En aquest apartat es mostrarà primerament el model de casos d'ús del sistema, que ens servirà per començar a representar les diferents funcionalitats que ha de tenir la interfície, i a estructurar-les segons una tipologia marcada.

Aquest model de casos d'ús, i l'estructuració marcada, ens servirà per realitzar el model conceptual de dades, amb els diagrames de classes necessaris per al disseny i implementació de la interfície gràfica i les seves funcionalitats.





## 4.2 Model de casos d'ús

El model de casos d'ús descriu els actors que intervindran, i la interacció que aquests tenen amb el sistema. Per fer-ho es descriu cada actor així com tots els casos d'ús que ens podem trobar.

Hi pot haver algun cas d'ús que, per la seva simplicitat o semblança amb algun altre, s'hagi descartat o es realitzi només una breu explicació descriptiva.

### 4.2.1 Descripció dels actors del sistema

Els actors que intervindran en els diversos casos d'ús seran els futurs empleats de l'empresa E2M i de les empreses clients que hagin de treballar i interactuar amb la interfície gràfica. En la següent taula es mostra una explicació de cada un d'ells.

Actor	Descripció
 <b>Tècnic E2M</b>	<p>És el tècnic de l'empresa E2M, encarregat de la configuració global del sistema.</p> <p>Tindrà permisos per accedir a qualsevol lloc del programa.</p>
 <b>Tècnic empresa client</b>	<p>És el tècnic responsable de l'empresa client, encarregat de configurar alguns aspectes del sistema, ja sigui de la interfície gràfica com del funcionament global del sistema</p>
 <b>Control qualitat empresa client</b>	<p>És el responsable del control de qualitat de l'empresa client, encarregat de mirar les dades i comprovar les inspeccions.</p> <p>No tindrà permisos per configurar cap aspecte del funcionament del programa</p>
 <b>Operari empresa client</b>	<p>És el treballador de la línia de l'empresa client, encarregat de controlar que la producció es realitza correctament. Només haurà d'engegar i apagar la màquina, a part de visualitzar els resultats.</p> <p>No tindrà permisos per configurar cap aspecte del funcionament del programa.</p>

#### 4.2.2 Descripció dels casos d'ús

En aquest apartat es mostrarà, individualment per a cada funcionalitat de la interfície gràfica, la descripció dels casos d'ús que es fan servir, implicant-hi els actors descrits a l'apartat anterior.

Per a fer-ho s'han dividit les funcionalitats en grups, segons la tipologia de cada funcionalitat:

- Pantalles i widgets de l'aplicació
- Menús de l'aplicació

##### 4.2.2.1 Pantalles i widgets de l'aplicació

En aquest apartat es mostren els casos d'ús referents a les diferents pantalles i widgets que podem trobar a l'aplicació. Bàsicament es descriuen la pantalla d'inici de l'aplicació i els diferents apartats en que pot estar composta la pantalla principal del programa.

##### Pantalla inicial de l'aplicació

Serà la primera pantalla que es mostrarà a l'iniciar l'aplicació, per a poder identificar l'usuari que hi vol entrar.

##### Login d'usuaris:

<b>Cas d'ús:</b>	Login d'usuaris	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client	
<b>Propòsit:</b>	Autenticació de cada client abans d'entrar a l'aplicació	
<b>Resum:</b>	Els actors s'hauran d'autenticar davant una pantalla prèvia a l'arrencada de l'aplicació. Cada un tindrà uns privilegis diferents, i se li mostrarà una visualització diferent de la interfície	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'usuari inicia l'aplicació	2. El sistema mostra un diàleg per permetre a l'usuari introduir les seves credencials (usuari i

<p>3. L'usuari introdueix el seu nom d'usuari i la contrasenya i prem el botó d'acceptar.</p>	<p>contrasenya).</p> <p>4. El sistema rep la petició i consulta la base de dades per comprovar les credencials. Habilita la visualització de la pantalla principal corresponent als privilegis de l'usuari introduït.</p>
<b>Curs alternatiu</b>	
<p>Línia 4: L'usuari introduït no està registrat a la base de dades, o la contrasenya no és la correcta: El sistema mostra un missatge d'error i torna a habilitar el diàleg inicial de registre d'usuari perquè ho torni a provar.</p>	

#### Widget de control

És el widget que permet controlar el funcionament del sistema: l'arrencada i aturada de les inspeccions.

#### Botó de "Run":

<b>Cas d'ús:</b>	Arrencar les inspeccions
<b>Actors:</b>	Tècnic E2M, Tècnic client, Operari client
<b>Propòsit:</b>	Iniciar les inspeccions
<b>Resum:</b>	Els actors podran iniciar les inspeccions, permetent així el funcionament de les anàlisis.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
<p>1. L'usuari activa el botó de "Run" del widget de control</p>	<p>2. El sistema posa en funcionament el sistema</p>
<b>Curs alternatiu</b>	

Línia 2: El sistema no pot arrencar degut a algun error en la configuració o mal funcionament d'algun dels seus components: Es mostra un error en el widget habilitat a aquest fi, i es cancel·la la petició d'inici de les anàlisis.

**Botó de "Stop":**

<b>Cas d'ús:</b>	Parar les inspeccions	
<b>Actors:</b>	Tècnic E2M, Tècnic client, Operari client	
<b>Propòsit:</b>	Aturar les inspeccions	
<b>Resum:</b>	Els actors podran aturar el sistema, parant així les captures i les anàlisis.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
1. L'usuari activa el botó de "Stop" del widget de control		2. El sistema para tots els components del sistema
<b>Curs alternatiu</b>		
Línia 2: El sistema retorna un error a l'aturar el sistema. Podria ser que les últimes anàlisis no s'hagin realitzat correctament: Es mostrarà un missatge per pantalla.		

**Botó de "Run Once":**

<b>Cas d'ús:</b>	Realitzar una sola anàlisi a la imatge actual.	
<b>Actors:</b>	Tècnic E2M, Tècnic client, Operari client.	
<b>Propòsit:</b>	Comprovar els resultats d'una anàlisi.	
<b>Resum:</b>	Els actors podran realitzar només una inspecció a la imatge actual, provocant així l'anàlisi només per a una imatge. El sistema s'aturarà automàticament quan hagi realitzat la inspecció.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
1. L'usuari activa el botó de "Run Once" del widget de control		

2. El sistema realitza una inspecció sobre la imatge actual i atura l'aplicació.
<b>Curs alternatiu</b>
Línia 2: El sistema no pot realitzar l'anàlisi degut a algun error en la configuració o mal funcionament d'algun dels seus components: Es mostra un error en el widget habilitat a aquest fi, i es cancel·la la petició d'inici de les anàlisis.

### Widget de log's

És el widget que mostrarà el Log del sistema. Concretament es mostraran els missatges de tipus "Event" i "Error" que es puguin donar en qualsevol component, així com en la mateixa aplicació.

#### Mostrar Log del sistema

<b>Cas d'ús:</b>	Mostrar el Log del sistema
<b>Actors:</b>	Tècnic E2M, Control qualitat client , Tècnic client, Operari client
<b>Propòsit:</b>	Mostrar les entrades de l'arxiu de Log de l'aplicació
<b>Resum:</b>	Els actors podran consultar l'estat de l'aplicació i del sistema en general mitjançant aquest widget, ja que s'hi mostraran les diferents línies de l'arxiu de Log del sistema que siguin de tipus events o errors, en el que s'hi reflecteixen les incidències de tots els seus components.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
1. L'usuari habilita la visualització del widget de Log del sistema	2. Es mostra una llista amb les últimes incidències registrades a l'arxiu de Log del sistema, diferenciant entre les que han estat errònies i les que no. Es mostren les línies de Log des de l'últim canvi d'usuari o de format, o des de l'arrencada de l'aplicació.
<b>Curs alternatiu</b>	



Widget de visualització de les imatges

És el widget on es mostraran les diferents imatges que obtenim de les càmeres, així com les imatges resultants de les anàlisis.

**Mostrar imatge actual:**

<b>Cas d'ús:</b>	Mostrar l'última imatge.
<b>Actors:</b>	Tècnic E2M, Control qualitat client , Tècnic client, Operari client.
<b>Propòsit:</b>	Mostrar l'última imatge que s'acaba de processar i analitzar.
<b>Resum:</b>	Els actors podran veure l'última imatge que ha rebut el sistema, i que s'acaba de processar i analitzar. Aquesta imatge serà l'original o la resultant d'alguna anàlisi, segons estigui seleccionat al selector de visualització d'imatges.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
<ol style="list-style-type: none"> <li>L'usuari habilita la visualització del widget de visualització de l'última imatge.</li> <li>L'usuari pot escollir si veure la imatge original o la resultant de l'última anàlisi.</li> </ol>	<ol style="list-style-type: none"> <li>El sistema mostra un widget amb l'última imatge que ha rebut el sistema.</li> <li>El sistema mostra al widget la imatge que ha seleccionat l'usuari.</li> </ol>
<b>Curs alternatiu</b>	
<p>Línia 2: El sistema no ha rebut cap imatge: es mostra una imatge per defecte que indica que no s'ha rebut cap imatge.</p> <p>Línia 4: L'anàlisi no ha generat cap imatge resultant: es mostra una imatge per defecte que indica que no està disponible la imatge sol·licitada.</p>	

### Widget de resultats

És el widget on es mostraran els resultats de l'execució de les diferents anàlisis realitzades des de l'arrencada del sistema.

#### Mostrar l'últim resultat:

<b>Cas d'ús:</b>	Mostrar l'últim resultat.	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client.	
<b>Propòsit:</b>	Mostrar els resultats de les inspeccions.	
<b>Resum:</b>	Els actors podran veure el resultat de l'última anàlisi de forma gràfica. Si el resultat ha estat correcte es mostrarà una imatge identificativa, sinó se'n mostrarà una altra que indicarà el contrari.	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'usuari habilita la visualització del widget de visualització de l'últim resultat.	2. El sistema mostra un widget on es pot observar una imatge identificativa del resultat de l'última anàlisi.
<b>Curs alternatiu</b>		
Línia 2: Encara no s'ha realitzat cap anàlisi: es mostra una imatge que identifica que encara no s'ha analitzat cap imatge.		

#### Mostrar resultats:

<b>Cas d'ús:</b>	Mostrar els resultats	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client	
<b>Propòsit:</b>	Mostrar els resultats dels anàlisis	
<b>Resum:</b>	Els actors podran veure els resultats globals de les anàlisis des que s'ha iniciat l'aplicació. En aquests resultats podrem observar la quantitat d'inspeccions realitzades, entre les que podrem veure el número de correctes i d'incorrectes.	
<b>Curs típic d'esdeveniments</b>		

Accions dels actors	Resposta del sistema
1. L'usuari habilita la visualització del widget de visualització resultats.	2. El sistema mostra un widget on es pot observar uns comptadors amb la quantitat d'inspeccions realitzades, diferenciant entre les que han estat correctes i les incorrectes.
Curs alternatiu	

**Reiniciar resultats:**

<b>Cas d'ús:</b>	Reiniciar els resultats
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client
<b>Propòsit:</b>	Reiniciar el comptatge de resultats
<b>Resum:</b>	Els actors podran reiniciar o netejar els comptadors de resultats per a establir un nou punt de control de resultats.
Curs típic d'esdeveniments	
Accions dels actors	Resposta del sistema
1. L'usuari habilita la visualització del widget de visualització resultats.	2. El sistema habilita un botó per a poder reiniciar els comptadors.
3. L'usuari prem el botó de netejar (o reiniciar) els resultats.	4. El sistema posa a 0 els comptadors de resultats totals, resultats correctes i resultats incorrectes. Neteja també tots els widgets que mostrin resultats de les inspeccions, ja sigui de manera numèrica o gràfica.
Curs alternatiu	

## Widget dels formats

És el widget on es gestionaran els diferents formats d'anàlisis de les imatges que tractarem. Cada format conté diversos processos i tractaments que configuren les anàlisis que apliquem a la imatge rebuda.

### Seleccionar format:

<b>Cas d'ús:</b>	Seleccionar un format	
<b>Actors:</b>	Tècnic E2M, Tècnic client	
<b>Propòsit:</b>	Seleccionar un dels formats existents.	
<b>Resum:</b>	Els actors podran seleccionar un format de la llista de formats creats amb anterioritat. Al seleccionar un format, aquest serà el que s'utilitzarà per a realitzar les inspeccions.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
1. L'usuari habilita la visualització del widget de visualització de la llista de formats		2. El sistema mostra una llista amb tots els formats creats anteriorment
3. L'usuari clica sobre el format desitjat		4. El sistema recarrega la configuració de les anàlisis amb els paràmetres del nou format
<b>Curs alternatiu</b>		
Línia 2: No hi ha cap format creat: es mostrarà una llista buida i l'usuari no podrà seleccionar cap format.		
L'ínia 4: L'aplicació està corrent en el moment d'intentar canviar de format: es mostrarà un error a l'arxiu de Log, ja que no es pot canviar de format mentre s'està corrent alguna inspecció. L'operació de canvi o selecció de nou format quedarà cancel·lada.		

### Afegir format:

<b>Cas d'ús:</b>	Afegir un nou format
<b>Actors:</b>	Tècnic E2M, Tècnic client

<b>Propòsit:</b>	Afegir un nou format a la configuració del sistema.	
<b>Resum:</b>	Els actors podran afegir nous formats per a carregar noves configuracions.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
1. L'usuari clica sobre el botó d'afegir un nou format.		2. El sistema mostra un nou diàleg on l'usuari hi podrà introduir el nom del nou format.
3. L'usuari introdueix el nom del nou format.		4. El sistema registra el nou format i l'introdueix a la configuració global de l'aplicació.
<b>Curs alternatiu</b>		
Línia 4: L'aplicació està iniciada quan es vol afegir un nou format: El botó d'afegir un nou format estarà deshabilitat mentre l'aplicació estigui iniciada.		
Línia 4: L'usuari ha introduït un nom en blanc o un nom que ja existeix: el sistema mostrarà un error a l'arxiu de Log i cancel·larà la creació del nou format.		

**Eliminar format:**

<b>Cas d'ús:</b>	Eliminar un format	
<b>Actors:</b>	Tècnic E2M, Tècnic client	
<b>Propòsit:</b>	Eliminar un dels formats existents.	
<b>Resum:</b>	Els actors podran eliminar un format de la llista de formats creats amb anterioritat	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
2. L'usuari selecciona un format		1. El sistema mostra la llista de formats creats
3. L'usuari clica al botó d'eliminar format		

4. El sistema elimina el format seleccionat
<b>Curs alternatiu</b>
Línia 1: No hi ha cap format creat: es mostrarà una llista buida i el botó d'eliminar estarà deshabilitat

### Widget de les gràfiques

És el Widget on es mostraran les diferents gràfiques que tenim per a l'aplicació, permetent així una visualització ràpida de l'evolució dels resultats i diferents aspectes relacionats amb aquests.

#### Gràfic resultats:

<b>Cas d'ús:</b>	Gràfic de l'evolució dels resultats.
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client.
<b>Propòsit:</b>	Visualitzar un gràfic amb l'evolució dels resultats.
<b>Resum:</b>	Els actors podran visualitzar una gràfica de barres que mostrarà l'evolució dels diferents resultats obtinguts de les anàlisis.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
1. L'usuari habilita la visualització del widget de visualització del gràfic de resultats, indicant-hi quins resultats vol veure.	2. El sistema mostra un widget on es pot observar un gràfic de barres amb l'evolució dels resultats de cada tractament configurat.
<b>Curs alternatiu</b>	
Línia 2: No hi ha cap resultat configurat per a mostrar: el sistema no mostrarà el widget de les gràfiques.	

**Histograma:**

<b>Cas d'ús:</b>	Histograma de resultats.	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client.	
<b>Propòsit:</b>	Visualitzar un resultat en forma d'histograma.	
<b>Resum:</b>	Els actors podran visualitzar l'histograma de l'evolució d'un resultat d'alguna de les anàlisis. Aquest resultat ha de ser de tipus numèric perquè tingui sentit mostrar-lo.	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'usuari habilita la visualització del widget Histograma, indicant-hi quin resultat vol veure.	2. El sistema mostra un widget on es pot observar un histograma amb l'evolució del resultat de l'anàlisi configurada.
<b>Curs alternatiu</b>		
Línia 2: No hi ha cap resultat configurat per a mostrar: el sistema no mostrarà el widget de les gràfiques.		

[Widget barra d'accessos directes](#)

Aquest widget serveix per a poder tenir accessos directes a diferents funcionalitats de l'aplicació, estarà compost per diferents botons que ens duren ràpidament a la funció desitjada. Serviran, tant per configurar l'aplicació, com per poder veure diferents coses per pantalla.

**Barra d'accessos directes**

<b>Cas d'ús:</b>	Barra d'accessos directes.	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Visualitzar una barra de botons amb accessos directes.	
<b>Resum:</b>	Els actors podran visualitzar una barra amb diferents botons que seran accessos directes a diferents menús de configuració. Normalment seran els menús més utilitzats o més habituals.	

Curs típic d'esdeveniments	
Accions dels actors	Resposta del sistema
<p>1. L'usuari habilita la visualització del widget d'accessos directes. Des de l'arxiu de configuració podrà configurar quins accessos s'han de mostrar.</p> <p>3. L'usuari clica a sobre d'un accés directe</p>	<p>2. El sistema mostra un widget on es pot observar una sèrie de botons amb accessos directes a diferents menús de l'aplicació, segons s'hagi configurat des de l'arxiu XML de configuració.</p> <p>4. El sistema mostra el menú de configuració del tractament corresponent a l'accés directe que s'ha clicat.</p>
Curs alternatiu	
<p>Línia 2: No hi ha cap botó configurat a l'arxiu XML: El sistema no mostra el widget.</p> <p>Línia 4: El menú que s'ha de mostrar correspon al menú de configuració d'un plugin que no ha estat carregat: El sistema mostra un error a l'arxiu de LOG</p>	

### Widget de visualització del tipus d'imatge

Aquest widget serveix per escollir el tipus d'imatge que volem que es mostri per pantalla, que normalment serà la imatge original, o la imatge resultat d'alguna de les anàlisis que tinguem activades.

### Selector d'imatge a visualitzar

<b>Cas d'ús:</b>	Selector d'imatge a visualitzar
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client
<b>Propòsit:</b>	Visualitzar selector per a poder triar quina imatge es vol visualitzar al widget "WImage"



<b>Resum:</b>	Els actors podran escollir quina imatge es vol visualitzar al widget "WImage" de la pantalla principal. Es podrà triar si es vol la imatge original, o alguna de les imatges resultants després d'aplicar les anàlisis.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>	<b>Resposta del sistema</b>	
<ol style="list-style-type: none"> <li>1. L'usuari habilita la visualització del widget Selector d'imatges, així com les opcions que vol que s'hi mostrin</li>   <li>3. L'usuari escull una de les opcions del selector</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema mostra un widget on es pot observar un selector amb tots els tipus que s'han llegit de l'arxiu de configuració.</li>   <li>4. El sistema mostra la imatge desitjada al widget "WImage" de la pantalla principal.</li> </ol>	
<b>Curs alternatiu</b>		
<p>Línia 2: No hi ha cap resultat configurat per a mostrar: el sistema no mostrarà el widget de les gràfiques.</p> <p>Línia 4: La imatge que s'ha de mostrar correspon a la sortida d'un plugin que no ha estat carregat: El sistema mostra un error a l'arxiu de LOG.</p>		

#### 4.2.2.2 Menús de l'aplicació

En aquest apartat es mostren els casos d'ús referents als diferents menús que hi ha a la interfície gràfica.

Es mostren els esquemes dels menús als quals accedim a través de la barra d'eines, a través de la barra d'accessos directes o clicant amb el botó dret en algun apartat de la pantalla que permeti obrir-hi un nou menú

#### Menú "Fitxer" de l'aplicació

És el menú que ens permetrà realitzar les funcionalitats globals de l'aplicació: guardar configuració, carregar configuració, tancar l'aplicació...

#### Obrir un fitxer de configuració:

<b>Cas d'ús:</b>	Obrir arxiu de configuració	
<b>Actors:</b>	Tècnic E2M, Tècnic client	
<b>Propòsit:</b>	Obrir un arxiu de configuració prèviament generat	
<b>Resum:</b>	Els actors podran carregar un nou arxiu de configuració, en el que hi haurà definides noves estructuracions de la pantalla principal, així com noves configuracions d'anàlisis, formats, càmeres...	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>L'usuari selecciona l'opció "Obrir arxiu" dins del menú "Arxiu".</li> <li>L'usuari escull un arxiu XML</li> </ol>	<ol style="list-style-type: none"> <li>El sistema mostra un diàleg que permet a l'usuari escollir un arxiu de tipus "XML".</li> <li>El sistema torna a carregar la configuració de l'aplicació amb el nou arxiu.</li> </ol>
<b>Curs alternatiu</b>		
Línia 3: L'usuari no selecciona cap arxiu i tanca el diàleg: El sistema ignorarà l'ordre de tornar a carregar una nova configuració i seguirà com estava.		

**Guardar un fitxer de configuració:**

<b>Cas d'ús:</b>	Guardar un fitxer de configuració	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client	
<b>Propòsit:</b>	Guardar el fitxer de configuració	
<b>Resum:</b>	Guardar en el fitxer de configuració la configuració actual del programa. En aquesta configuració s'hi inclou tot el que fa referència a les anàlisis, interfície gràfica, informació de la base de dades...	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
1. L'usuari selecciona l'opció "Guardar arxiu de configuració" dins del menú "Arxiu".		2. El sistema guarda a l'arxiu XML la configuració global del sistema.
<b>Curs alternatiu</b>		
Línia 2: Error al guardar la configuració: Es mostra un error al Log i es cancel·la l'operació.		

**Tancar l'aplicació:**

<b>Cas d'ús:</b>	Tancar l'aplicació	
<b>Actors:</b>	Tècnic E2M, Tècnic client, Operari client	
<b>Propòsit:</b>	Tancar l'aplicació	
<b>Resum:</b>	Els actors podran tancar l'aplicació, finalitzant totes les inspeccions que estan corrent.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
1. L'usuari selecciona l'opció "Tancar" dins del menú "Arxiu".		2. El programa finalitza totes les anàlisis que estiguin corrent, guarda la configuració i tanca la finestra principal.
<b>Curs alternatiu</b>		

Línia 2: S'ha produït algun error mentre es tanca l'aplicació: es mostra l'error a l'arxiu de Log.

### Menú "configuració" de l'aplicació

És el menú que ens permetrà configurar els diferents aspectes del sistema, no només de l'aplicació. Aspectes com la gestió de processos i tractaments, configuració de les càmeres, configuració dels plugins... són aspectes que es poden gestionar des dels submenús que formen part d'aquest apartat.

#### Configuració de l'adquisició de les imatges:

<b>Cas d'ús:</b>	Configuració de l'adquisició de les imatges.	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Configurar els mètodes d'adquisició de les imatges.	
<b>Resum:</b>	Els usuaris podran configurar els diferents mètodes que ofereix el sistema per a obtenir les imatges. Els usuaris podran veure els diferents modes d'adquisició i configurar-ne els seus paràmetres.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>	<b>Resposta del sistema</b>	
<ol style="list-style-type: none"> <li>L'usuari selecciona l'opció "Configuració d'adquisició" dins del menú "Configuració".</li> <li>L'usuari escull el mètode de captura que vol configurar i modifica els paràmetres desitjats.</li> <li>L'usuari clica el botó d'acceptar per tal de guardar els canvis.</li> </ol>	<ol style="list-style-type: none"> <li>El sistema mostra el diàleg corresponent a la configuració d'adquisició, en el que podem veure les diferents càmeres i mètodes de captura d'imatges</li> <li>El sistema aplica els canvis a la configuració global de l'aplicació i ho escriu a l'arxiu XML.</li> </ol>	
<b>Curs alternatiu</b>		
Línia 4: L'usuari clica al botó de cancel·lar: El sistema tanca el diàleg sense guardar els canvis i manté la configuració global del sistema tal i com estava anteriorment.		

### Configuració de les anàlisis:

En aquests casos d'ús es mostra com es gestionen els processos que s'executen en cada inspecció. Cada procés pot contenir un o més tractaments. Al mateix temps, cada procés i tractament poden tenir una sèrie de paràmetres.

Tant la gestió de processos com la de tractaments es redueix a un cas d'ús per mostrar-los, afegir-ne un de nou, un per renombar-lo i un per esborrar-lo. Per a realitzar els següents casos d'ús s'han extret aquells que són redundants o molt similars a d'altres, resultant així una llista dels més representatius, i que expliquen millor les funcionalitats més complexes que ofereix aquest menú.

<b>Cas d'ús:</b>	Mostrar la llista de processos	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Mostrar una llista amb tots els processos que tenim configurats	
<b>Resum:</b>	Els actors podran veure una llista amb tots els processos que s'hagin configurat des de l'arxiu de configuració o a través d'aquest mateix menú en anteriors ocasions.	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	1. L'usuari selecciona l'opció "Configuració de les anàlisis" dins del menú "Configuració".	2. El sistema mostra una llista amb tots els processos que té carregats en la seva configuració global.
<b>Curs alternatiu</b>		

<b>Cas d'ús:</b>	Afegir un procés
<b>Actors:</b>	Tècnic E2M, Tècnic client.
<b>Propòsit:</b>	Afegir un nou procés
<b>Resum:</b>	Els actors podran afegir un nou procés per a realitzar les anàlisis. Aquest nou procés haurà de contenir un o més tractaments, que seran els que realitzaran les anàlisis.

<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
<p>1. L'usuari selecciona l'opció d'afegir un nou procés al conjunt de processos.</p> <p>3. L'usuari introdueix el nom del procés.</p> <p>5. L'usuari introdueix el nom del grup</p> <p>7. L'usuari introdueix el nom i el tipus de plugin que utilitzarà el nou tractament</p>	<p>2. El sistema mostra un nou diàleg preguntant quin serà el nom del nou procés.</p> <p>4. El sistema enregistra el nom i mostra un nou diàleg preguntant a quin grup ha de pertànyer el nou procés.</p> <p>6. El sistema enregistra el grup i mostra un nou diàleg preguntant el nom del primer tractament del procés (sempre es mostrarà ja que no es pot crear un procés sense cap tractament).</p> <p>8. El sistema guarda el nou procés a la configuració local del diàleg</p>
<b>Curs alternatiu</b>	
<p>Línia 4: El nom del nou procés ja existeix en algun altre procés: El sistema mostra un missatge d'alerta i fa que l'usuari torni a provar de posar un nou nom.</p>	

<b>Cas d'ús:</b>	Eliminar un procés.
<b>Actors:</b>	Tècnic E2M, Tècnic client.
<b>Propòsit:</b>	Eliminar un procés existent.
<b>Resum:</b>	Els actors podran eliminar un procés de la llista de processos, eliminant així tots els seus tractaments.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>

<p>1. L'usuari selecciona l'opció d'eliminar un nou procés de la llista de processos.</p> <p>2. El sistema elimina el procés i tots els seus tractaments de la configuració local del diàleg.</p>
<b>Curs alternatiu</b>
Línia 2: El procés és l'únic que queda a la llista: el sistema retorna un error, ja que com a mínim hi ha d'haver un procés perquè l'aplicació pugui realitzar anàlisis.

<b>Cas d'ús:</b>	Renomenar un procés.
<b>Actors:</b>	Tècnic E2M, Tècnic client.
<b>Propòsit:</b>	Renomenar un procés existent.
<b>Resum:</b>	Els actors podran canviar el nom d'un dels processos de la llista.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
<p>1. L'usuari selecciona l'opció de renomenar un nou procés de la llista de processos.</p>	<p>2. El sistema canvia el nom del procés de la configuració local del diàleg.</p>
<b>Curs alternatiu</b>	
Línia 2: El nou nom del procés ja existeix com a nom d'un altre procés de la llista: el sistema mostra un missatge d'error i cancel·la l'operació, oferint la possibilitat a l'usuari de tornar a posar un altre nom.	

<b>Cas d'ús:</b>	Mostrar els tractaments d'un procés.
<b>Actors:</b>	Tècnic E2M, Tècnic client.
<b>Propòsit:</b>	Mostrar tots els tractaments del procés seleccionat
<b>Resum:</b>	Mostrar una vista amb tots els tractaments que conté el procés seleccionat. Es quedarà seleccionat i obert el primer tractament.

<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
<p>1. L'usuari selecciona un dels processos de la llista.</p>	<p>2. El sistema mostra una nova llista amb tots els tractaments del procés seleccionat.</p> <p>3. El sistema selecciona automàticament el primer tractament del procés i mostra la seva informació.</p>
<b>Curs alternatiu</b>	

<b>Cas d'ús:</b>	Mostrar la informació d'un tractament.
<b>Actors:</b>	Tècnic E2M, Tècnic client.
<b>Propòsit:</b>	Mostrar la informació corresponent a un tractament dins d'un procés.
<b>Resum:</b>	Mostrar una vista amb tota la informació corresponent a un tractament, on hi ha els seus paràmetres i la interfície gràfica de configuració del plugin que representa.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
<p>1. L'usuari selecciona un tractament de la llista de tractaments que formen part d'un procés.</p>	<p>2. El sistema mostra la informació del tractament seleccionat, en el que es pot observar els paràmetres del tractament i la interfície de configuració del plugin que representa.</p>
<b>Curs alternatiu</b>	



<b>Cas d'ús:</b>	Guardar la nova configuració.	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Guarda les modificacions fetes a la configuració global.	
<b>Resum:</b>	Quan es prem el botó d'aplicar o acceptar del diàleg. S'han de guardar a la configuració global del sistema tots els canvis que s'han fet des que s'ha obert el diàleg, o des de l'última vegada que s'ha guardat.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>	<b>Resposta del sistema</b>	
1. L'usuari prem el botó d'"acceptar" o "aplicar" del diàleg.	2. El sistema aplica els canvis que s'han fet des de que s'ha obert el diàleg a la configuració global del sistema.	
<b>Curs alternatiu</b>		

**Configuració de les rutes dels plugins:**

<b>Cas d'ús:</b>	Canviar el path d'un plugin	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Canviar la ruta on es va a buscar la llibreria del plugin.	
<b>Resum:</b>	Els actors podran modificar la ruta on es troba la llibreria del plugin que es vol carregar. Fins que no es carrega aquesta llibreria no es pot fer servir cap de les funcions que el plugin ofereix, mostrant un error cada vegada que es vulgui utilitzar. Inicialment els valors que apareixen en aquest diàleg són els que s'han carregat des de l'arxiu de configuració.	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'usuari selecciona l'opció "Configuració dels path's dels plugins" dins del menú "Configuració".</li> <li>3. L'usuari modifica el path d'un plugin</li> <li>4. L'usuari clica el botó d'acceptar per guardar els canvis.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema mostra el diàleg corresponent a la configuració dels path's, en el que podem veure els diferents plugins i una línia amb el path de la llibreria a on està carregat.</li> <li>5. El sistema aplica els canvis a la configuració global de l'aplicació i ho escriu a l'arxiu XML.</li> </ol>
<b>Curs alternatiu</b>		

**Configuració del mode d'inspecció:**

<b>Cas d'ús:</b>	Modificar el mode d'inspecció	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Modificar el mode d'inspecció de les imatges	
<b>Resum:</b>	Els actors podran canviar el mode en que es capturen les imatges. Per exemple, es podrà escollir si es vol inspeccionar una imatge cada vegada que arriba un "triguer", cada cert temps utilitzant un temporitzador...	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>		<b>Resposta del sistema</b>
<ol style="list-style-type: none"> <li>1. L'usuari selecciona l'opció "Mode d'inspecció" dins del menú "Configuració".</li> <li>3. L'usuari selecciona el mode d'inspecció que desitja i hi introdueix un temps d'inspecció (si el mode ho permet).</li> <li>4. L'usuari clica sobre el botó d'acceptar el diàleg</li> </ol>		<ol style="list-style-type: none"> <li>2. El sistema mostra el diàleg de configuració dels diferents modes d'inspecció que hi ha configurats</li> <li>5. El sistema enregistra els canvis i els guarda a la configuració global del programa</li> </ol>
<b>Curs alternatiu</b>		

## Menú “resultats” de l’aplicació

És el menú que ens permet gestionar els resultats de les diferents anàlisis. Des d’aquí podem veure els resultats, veure una gràfica d’algun resultat en concret, guardar imatges resultants...

### Mostrar resultats:

<b>Cas d’ús:</b>	Mostrar els resultats de les anàlisis
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client
<b>Propòsit:</b>	Mostrar el resultat de les anàlisis
<b>Resum:</b>	Els actors podran veure els resultats de les anàlisis que s’hagin configurat a l’arxiu de configuració. Es podran veure els valors resultants de les anàlisis (siguin enters, cadenes de caràcters, booleans...).
Curs típic d’esdeveniments	
Accions dels actors	Resposta del sistema
1. L’usuari selecciona l’opció “Mostrar resultats” dins del menú “resultats”.	2. El sistema mostra un diàleg amb una llista on s’hi poden observar els diferents resultats que s’han anat obtenint de les anàlisis que s’han fet fins al moment. Es mostren els resultats de les anàlisis que s’han configurat a l’arxiu de configuració.
Curs alternatiu	

**Mostrejar un sol resultat:**

<b>Cas d'ús:</b>	Mostrar un sol resultat.	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client.	
<b>Propòsit:</b>	Mostrar l'evolució dels resultats d'una sola anàlisi.	
<b>Resum:</b>	Els usuaris podran veure de forma gràfica, l'evolució dels resultats de l'anàlisi que escullin.	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'usuari selecciona l'opció "Mostrejar un resultat" dins del menú "resultats".</li> <li>3. L'usuari selecciona el resultat que vol veure.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema mostra un diàleg preguntant a l'usuari quin és el resultat que vol mostrejar.</li> <li>4. El sistema mostra un diàleg amb un gràfic que va canviant segon els resultats que va obtenint de l'anàlisi escollida.</li> </ol>
<b>Curs alternatiu</b>		

**Guardar les imatges resultants:**

<b>Cas d'ús:</b>	Guardar les imatges resultants.	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client.	
<b>Propòsit:</b>	Guardar les imatges resultants després de realitzar-hi d'una anàlisi.	
<b>Resum:</b>	Els usuaris podran definir una carpeta del sistema on es guardaran les imatges resultants de l'anàlisi que es vulgui. Les imatges es guardaran a la carpeta escollida mantenint les característiques després d'haver-li realitzat una anàlisi.	
<b>Curs típic d'esdeveniments</b>		
<b>Accions dels actors</b>	<b>Resposta del sistema</b>	
<ol style="list-style-type: none"> <li>1. L'usuari selecciona l'opció "Guardar les imatges resultants" dins del menú "resultats".</li> <li>3. L'usuari introdueix la ruta del directori i especifica les imatges que hi vol guardar.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema mostra un diàleg on permet a l'usuari definir en quin directori es guardaran les imatges, quines imatges resultants agafem o si volem guardar les dolentes.</li> <li>4. El sistema modifica la configuració global del sistema perquè després de realitzar-se l'anàlisi escollida es guardin les imatges resultants a la carpeta que s'ha triat.</li> </ol>	
<b>Curs alternatiu</b>		

### Menú “Usuaris” de l’aplicació

És el menú que ens permet gestionar els usuaris que tenen permisos per accedir al sistema.

#### Canviar d’usuari:

<b>Cas d’ús:</b>	Canviar d’usuari	
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client	
<b>Propòsit:</b>	Canviar l’usuari que està usant l’aplicació	
<b>Resum:</b>	Els actors podran canviar el tipus d’usuari que està fent servir l’aplicació. Cada tipus d’usuari té un rol, que li permet poder visualitzar i gestionar més o menys opcions de l’aplicació i del sistema.	
<b>Curs típic d’esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>L’usuari selecciona l’opció “Canviar d’usuari” dins del menú “Usuaris”.</li> <li>L’usuari selecciona el nou usuari al qual vol canviar i hi introdueix la contrasenya</li> </ol>	<ol style="list-style-type: none"> <li>El sistema mostra un diàleg on li permetrà escollir el nou usuari i introduir la seva contrasenya.</li> <li>El sistema enregistra el canvi d’usuari, i torna a recarregar tots la configuració per mostrar l’aplicació tal i com els permisos del nou usuari li permeten</li> </ol>
<b>Curs alternatiu</b>		

**Afegir usuari:**

<b>Cas d'ús:</b>	Afegir un usuari	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Afegir un nou usuari amb accés a l'aplicació	
<b>Resum:</b>	Els actors podran afegir nous usuaris amb permisos per accedir a l'aplicació. Per a cada nou usuari l'actor haurà de definir quin és el rol que li vol donar.	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>
	<ol style="list-style-type: none"> <li>1. L'usuari selecciona l'opció "Afegir usuari" dins del menú "Usuaris".</li> <li>3. L'usuari introdueix el nom, contrasenya i rol del nou usuari.</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema mostra una sèrie de diàlegs on se li permetrà introduir el nom, la contrasenya i el rol del nou usuari. Només li permetrà crear usuaris amb un rol igual o superior al seu.</li> <li>4. El sistema guarda les dades del nou usuari a la base de dades</li> </ol>
<b>Curs alternatiu</b>		

**Eliminar usuari:**

<b>Cas d'ús:</b>	Eliminar un usuari	
<b>Actors:</b>	Tècnic E2M, Tècnic client.	
<b>Propòsit:</b>	Eliminar un usuari dels que tenen accés a l'aplicació	
<b>Resum:</b>	Els actors podran eliminar usuaris dels que tenen permisos per accedir a l'aplicació.	
<b>Curs típic d'esdeveniments</b>		
	<b>Accions dels actors</b>	<b>Resposta del sistema</b>



<p>1. L'usuari selecciona l'opció "Eliminar usuari" dins del menú "Usuaris".</p> <p>3. L'usuari selecciona l'usuari que vol eliminar de la base de dades</p>	<p>2. El sistema mostra un diàleg on se li permetrà escollir un usuari. Només li permetrà eliminar usuaris amb un rol igual o superior al seu.</p> <p>4. El sistema elimina l'usuari de la base de dades</p>
<b>Curs alternatiu</b>	

### Menú "Ajuda" de l'aplicació

És el menú que ens permetrà veure diferents opcions d'ajuda i d'informació de l'aplicació, com per exemple: el log complet d'events, informació sobre la versió del software...

#### Mostrar el Log d'events:

<b>Cas d'ús:</b>	Mostrar el log d'events
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client
<b>Propòsit:</b>	Mostrar el log d'events del sistema i els temps de les anàlisis
<b>Resum:</b>	Els actors podran veure en un mateix diàleg tot l'historial del Log des que s'ha iniciat l'aplicació, i els temps de les anàlisis (el temps dels processos i tractaments que s'han executat en l'última inspecció). Es mostren a la mateixa pantalla, ja que internament aquests temps es propaguen pel codi mitjançant el Log.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
<p>1. L'usuari selecciona l'opció "Mostrar el log d'events" dins el menú "Ajuda".</p>	<p>2. El sistema mostra un diàleg on es pot observar el log global del sistema</p>

	(totes les entrades a l'arxiu de Log de tipus error o canvi d'estat, des de que s'ha iniciat l'aplicació). També es mostren els temps d'execució dels últims processos i tractaments.
<b>Curs alternatiu</b>	

**Informació sobre el software:**

<b>Cas d'ús:</b>	Informació de l'aplicació
<b>Actors:</b>	Tècnic E2M, Control qualitat client, Tècnic client, Operari client
<b>Propòsit:</b>	Informació sobre la versió de l'aplicació
<b>Resum:</b>	Els usuaris podran veure la informació referent a la versió del software que està instal·lada, així com d'altres aspectes relacions amb l'origen d'aquest.
<b>Curs típic d'esdeveniments</b>	
<b>Accions dels actors</b>	<b>Resposta del sistema</b>
1. L'usuari selecciona l'opció "Informació sobre el software" dins del menú "Ajuda".	2. El sistema mostra un diàleg on es poden observar diversos aspectes de la versió del software.
<b>Curs alternatiu</b>	

### 4.3 Model conceptual de dades

En aquest apartat es mostraran els diagrames de classes que s'han utilitzat per a implementar les diferents funcionalitats de la interfície gràfica i que formen part del model conceptual de dades.

Aquests diagrames són el pilar de tota l'especificació, ja que mostren de manera estàtica totes les classes que compondran el sistema, així com les seves interrelacions (incloent herència, agregació, etc.) i les relacions amb les diferents classes de la capa de domini i de la capa de dades de l'aplicació global. Així es mostra d'un simple cop d'ull tot el que el sistema pot contenir.

En els següents diagrames de classes es representen totes les classes que es fan servir per a implementar la interfície gràfica, així com aquelles classes de la capa de domini i de dades que són necessàries per a la implementació de les funcionalitats que es requereixen en aquesta interfície.

S'han organitzat les diferents classes necessàries per a la interfície gràfica segons el seu tipus, per tal de poder realitzar els diagrames d'una forma estructurada.

En els següents punts es mostren cadascun d'aquests tipus i el diagrama corresponent. Primerament però es mostra un diagrama que engloba les principals classes que intervindran en aquesta interfície.

Aquesta classificació s'ha fet per tal de poder mostrar totes les classes del projecte d'una manera vistosa i llegible. Per això la major part dels diagrames contenen la definició de les classes sense mostrar els seus atributs ni operacions. El fet que des de l'empresa no es requerís que en els diagrames es mostrés aquesta informació, ha fet que tampoc es mostri en aquesta memòria, permetent així una millor visió de cada un.

4.3.1 Diagrama de classes global de la interfície gràfica

En aquest apartat es mostra un diagrama global de les classes generals que intervenen en les funcionalitats de la interfície gràfica.

Les classes que es mostren en aquest diagrama són les que gestionen els elements (diàlegs, widgets, missatges...) necessaris en el disseny i implementació del projecte.

Es mostra també la classe "CMessage" ja que és necessària per a comunicar les diferents accions entre les classes del diagrama.

En aquest diagrama es mostren els atributs i les operacions de cada classe, ja que només està format per 4 classes, i la seva inclusió no impedeix una bona comprensió.

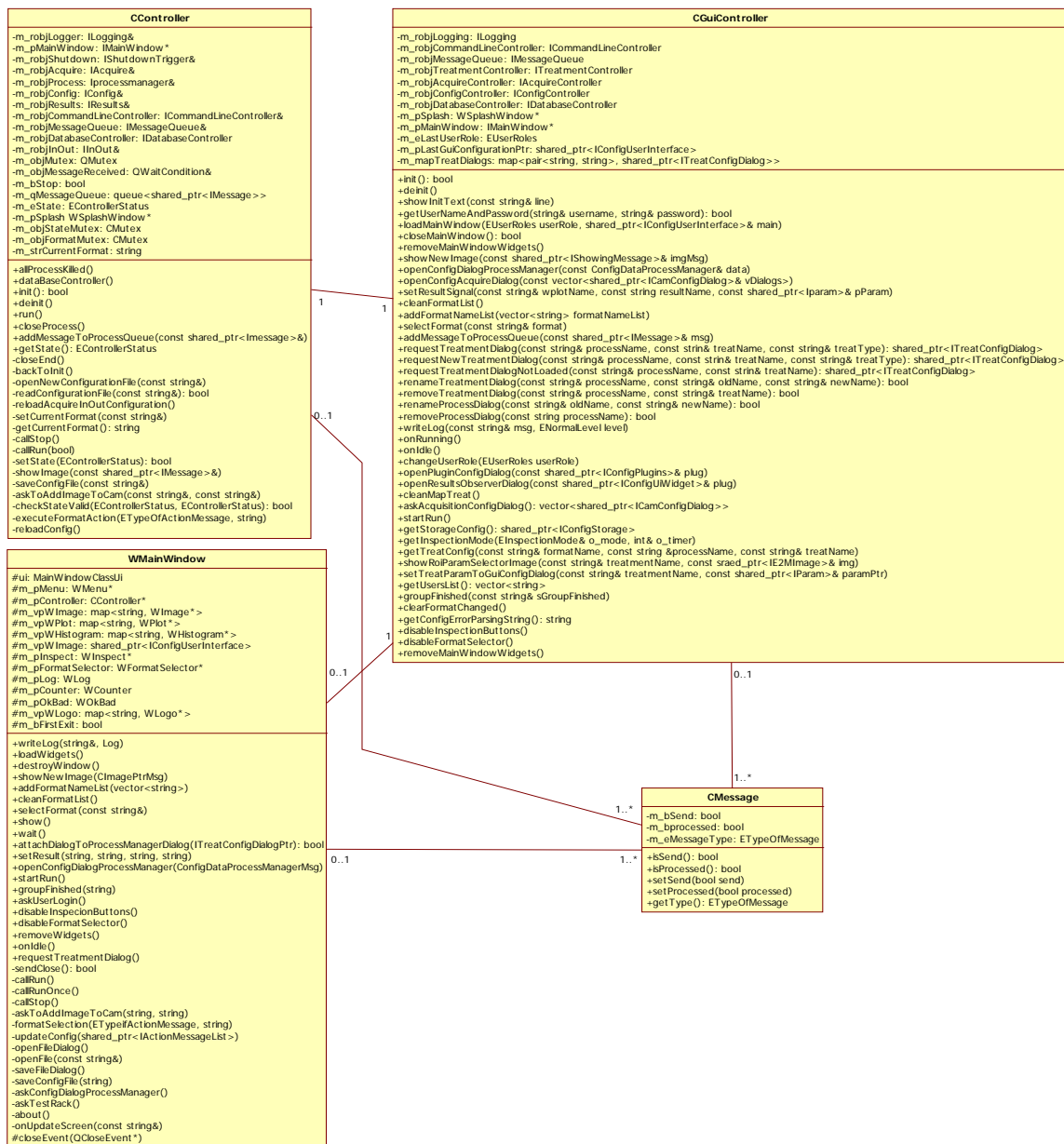


Diagrama de classes global

### 4.3.2 Diagrama de classes del Controlador

En aquest apartat es mostra en forma de diagrama UML, les classes de la interfície gràfica que es relacionen amb el controlador global de l'aplicació (*CController*), situat a la capa de domini.

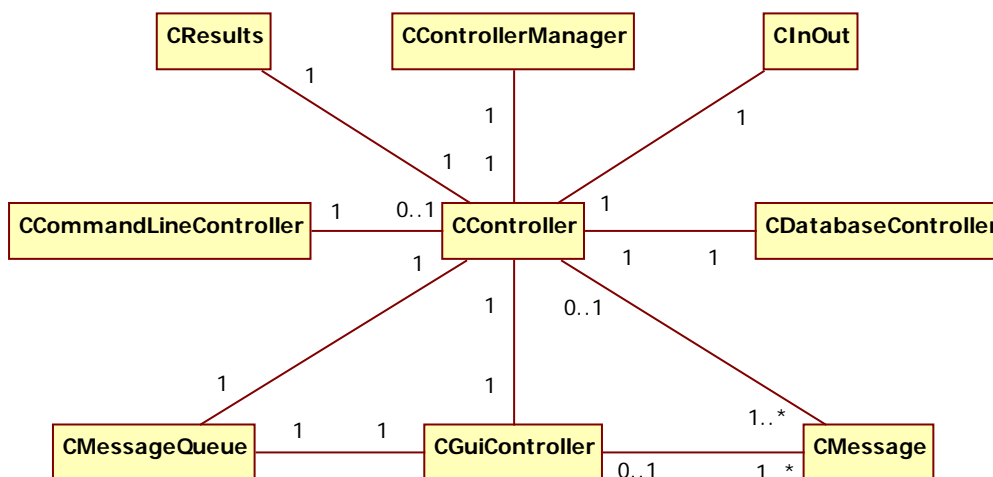


Diagrama de classes del controlador

### 4.3.3 Diagrama de classes del Controlador de la Interfície gràfica

En aquest apartat es mostra en forma de diagrama UML, les classes que intervenen amb el controlador de la interfície gràfica (*CGuiController*), situat a la capa de presentació.

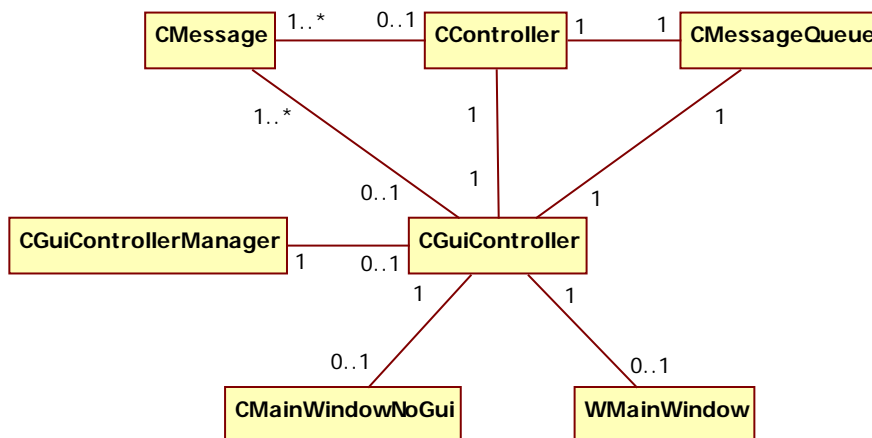


Diagrama de classes del controlador de la interfície gràfica

#### 4.3.4 Diagrama de classes de la Pantalla Principal

En aquest apartat es mostra en forma de diagrama UML, les classes que gestiona la pantalla principal de l'aplicació (*WMainWindow*) situada a la capa de presentació. La major part de les classes que gestiona aquesta classe són Widgets, representats per una *W* al davant del nom de cada classe que l'implementa. L'explicació de cada ún està a l'apartat de disseny d'aquesta memòria [Apartat 5.5].

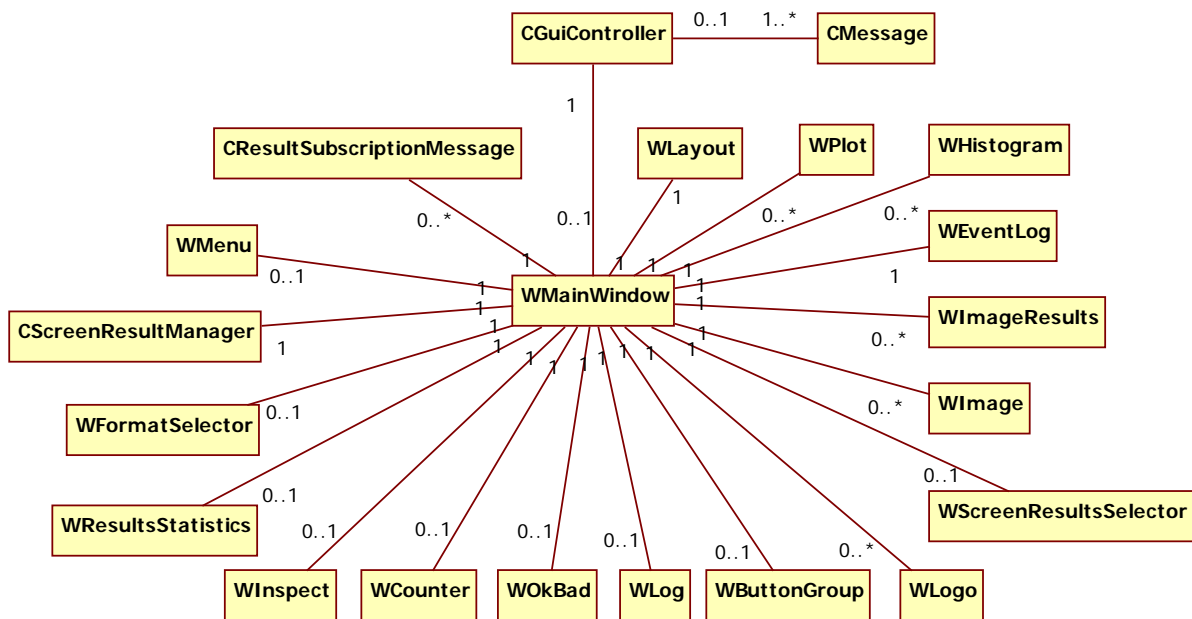


Diagrama de classes de la pantalla principal

#### 4.3.5 Diagrama de classes per la gestió dels Missatges

En aquest apartat es mostra en forma de diagrama UML, les classes que serveixen per a la gestió dels diferents tipus de missatges.

Aquests missatges hereten tots de la classe "*CMessage*", i serveixen per representar les diferents accions que tracta el controlador.

Per englobar aquests missatges s'utilitzen unes classes que gestionen el seu emmagatzematge (*CMessageQueue* i *CMessageQueueManager*).

Totes aquestes classes formen part d'un mòdul de la capa de domini (*MessageQueue*), però s'utilitzen també per gestionar les accions a través del controlador de la interfície gràfica i de la pantalla principal.

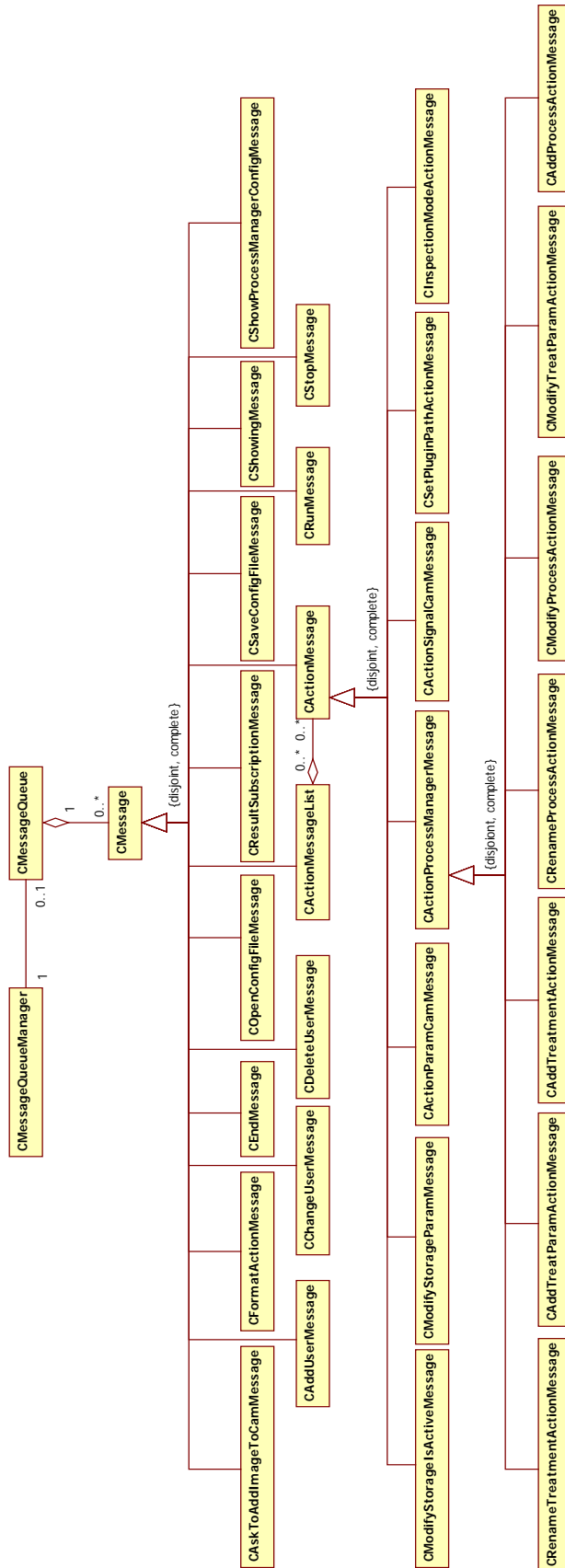


Diagrama de classes dels missatges

### 4.3.6 Diagrama de classes per la gestió dels Diàlegs

En aquest apartat es mostra en forma de diagrama UML, les classes que representen els diferents diàlegs de l'aplicació. Tots ells es relacionen amb la pantalla principal, i contenen una lletra *W* al davant ja que s'implementen com uns Widgets.

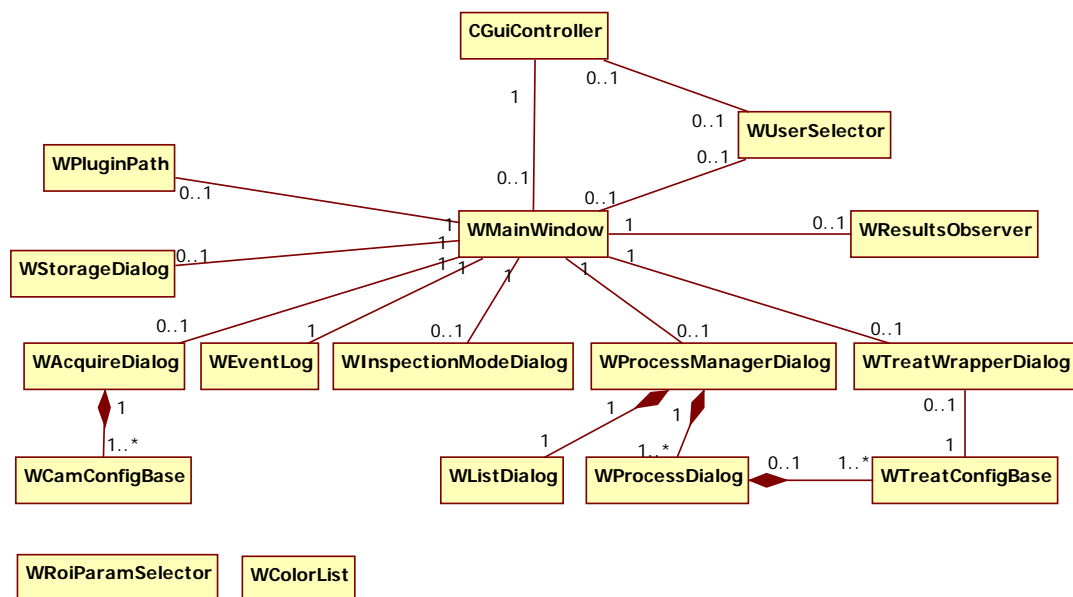


Diagrama de classes dels diàlegs

En aquest diagrama es pot observar que hi ha 2 classes que no estan unides a cap altre. Això és degut perquè aquestes classes serveixen com a classes auxiliars per a utilitzar en la creació dels diferents diàlegs de configuració dels plugins, així com en futures funcionalitats de la interfície gràfica.



## 5. DISSENY

5.1	Introducció	99
5.2	Disseny de l'arquitectura	100
5.2.1	Patrons utilitzats	101
5.3	Disseny de la pantalla principal	102
5.4	Disseny dels Menús i Diàlegs	103
5.5	Disseny dels Widgets	105
5.6	Diagrames de seqüència	108
5.6.1	Inicialització de l'aplicació	108
5.6.2	Interacció Interfície gràfica – Controlador	110
5.6.3	Interacció Controlador – Interfície gràfica	111
5.6.4	Comunicació amb la base de dades i recàrrega la configuració	112
5.6.5	Diàlegs dels menús de configuració de l'aplicació	114



## 5.1 Introducció

Si bé en el capítol anterior es definia què és el que havia de fer el nostre sistema, això no és suficient per a dur a terme la implementació d'aquest, ja que s'orienta a la descripció del problema, per tant falta decidir com resoldre'l.

L'etapa de disseny consisteix en prendre les decisions que ens portin a realitzar un sistema capaç de dur a terme les funcions definides prèviament. Aquesta etapa és crucial per a dur a terme un projecte amb èxit, ja que donarà pas a la implementació, on es començaran a veure els resultats de tot el treball i l'esforç realitzat.

Per guiar les construcció de la solució idònia en cada cas s'hauran d'adoptar certs criteris de disseny i establir les tecnologies i estructures generals del sistema seguint aquests criteris.

En aquest capítol es detallaran i justificaran les decisions preses a l'hora de dur a terme el disseny del sistema.

Primerament es mostra el disseny de l'arquitectura que s'ha usat en aquest projecte, així com els diferents patrons utilitzats en el disseny del software, i que ens han servit per a agilitzar i millorar algunes accions.

Tot seguit es mostra una descripció de cada un dels components de la interfície gràfica, com estan estructurats i dissenyats.

Finalment, fent ús dels diagrames de seqüència, s'han representat gràficament les funcionalitats més importants o complexes de cada apartat del projecte.

## 5.2 Disseny de l'arquitectura

El primer pas que s'ha de dur a terme és escollir l'arquitectura que serà la més adequada pel sistema.

La determinació de l'arquitectura no és un procés senzill, ja que consisteix en la presa de decisions respecte a:

- L'organització del sistema software.
- La selecció dels elements estructurals i les seves interfícies.
- El comportament d'aquests elements estructurals.
- La possible composició dels elements estructurals en subsistemes més grans.
- L'estil i polítiques que guien aquesta organització.

Com s'ha vist a l'apartat on es detallaven les propietats de l'aplicació *VisioInspect2* [Apartat 2.2], el sistema utilitza l'**orientació a objectes** per encapsular les operacions i les dades, tractant-los com a objectes, facilitant la reutilització i la facilitat de manteniment. A part, s'ha mostrat també com, globalment, s'utilitza una **arquitectura en tres capes** per a organitzar tots els components.

Per a dissenyar la interfície gràfica s'ha seguit la política que segueix l'arquitectura en tres capes del sistema global, utilitzant els beneficis de l'orientació a objectes per a crear tots els components necessaris. Per tant s'ha intentat seguir el mateix mètode de comunicació entre components definit a la capa de domini.

A més a més s'ha decidit utilitzar el **patró controlador** per comunicar la interfície gràfica (capa de presentació) amb la capa de domini. En concret s'ha decidit utilitzar el patró **façana** que proporciona les funcionalitats requerides [DES03].

Amb l'ajut d'aquest patró s'ha definit un sol controlador per a la capa de presentació, el qual té tantes operacions com esdeveniments ha de capturar el sistema. Aquest controlador també pot tenir algun atribut que permeti compartir informació entre les diferents classes.

---

### 5.2.1 Patrons utilitzats

---

Per a realitzar el disseny i posterior implementació de l'aplicació, s'ha comptat amb la utilització d'una sèrie de patrons de disseny, que ens han ajudat a facilitar algunes decisions i alguns apartats d'aquesta feina.

Els patrons que s'han fet servir per a dissenyar i implementar aquest projecte són [DES03]:

- Singletó:
  - o Consisteix en què una classe només tingui una instància, i proporciona un punt d'accés global a ella.
  - o Cada component del sistema és un singletó. Només existeix una instància de cada un.
- Observador:
  - o És una estructura de classes (subjectes) que notifiquen els seus canvis a unes altres classes (observadores). Quan un observador rep una notificació, actualitza el seu estat, executant la funció que correspongui.
  - o El controlador general actua com a observador de la cua de missatges de l'aplicació. Una nova actualització d'aquesta cua despertarà el controlador.
- Chain of responsibility:
  - o Permet establir la línia que han de seguir els missatges o les crides per a que els objectes realitzin la feina indicada o sol·licitada.
  - o En diferents diàlegs com per exemple el diàleg de configuració de les anàlisis, es delega la responsabilitat entre les diferents classes que hi intervenen, passant els missatges entre elles fins arribar a la que realment ha d'executar l'acció requerida.
- Bridge:
  - o Desacobla una abstracció de la seva implementació.
  - o S'utilitza aquest patró per a realitzar les diferents interfícies de les classes del domini (necessàries per a la gestió d'alguns diàlegs), i algunes classes de la capa de presentació que no representen widgets, ni controladors, ni diàlegs.

S'utilitzen classes que són interfícies per a desacoblar la definició de les operacions de la seva implementació.

### 5.3 Disseny de la pantalla principal

La pantalla principal és un component essencial del sistema desenvolupat. És l'esquelet on es suporten tots els components gràfics.

Serà l'encarregada de gestionar la interacció de l'usuari amb el sistema, la qual cosa fa que a l'hora de dissenyar-la s'hagin tingut en compte els requeriments i funcionalitats del sistema especificats en les primeres etapes del projecte.

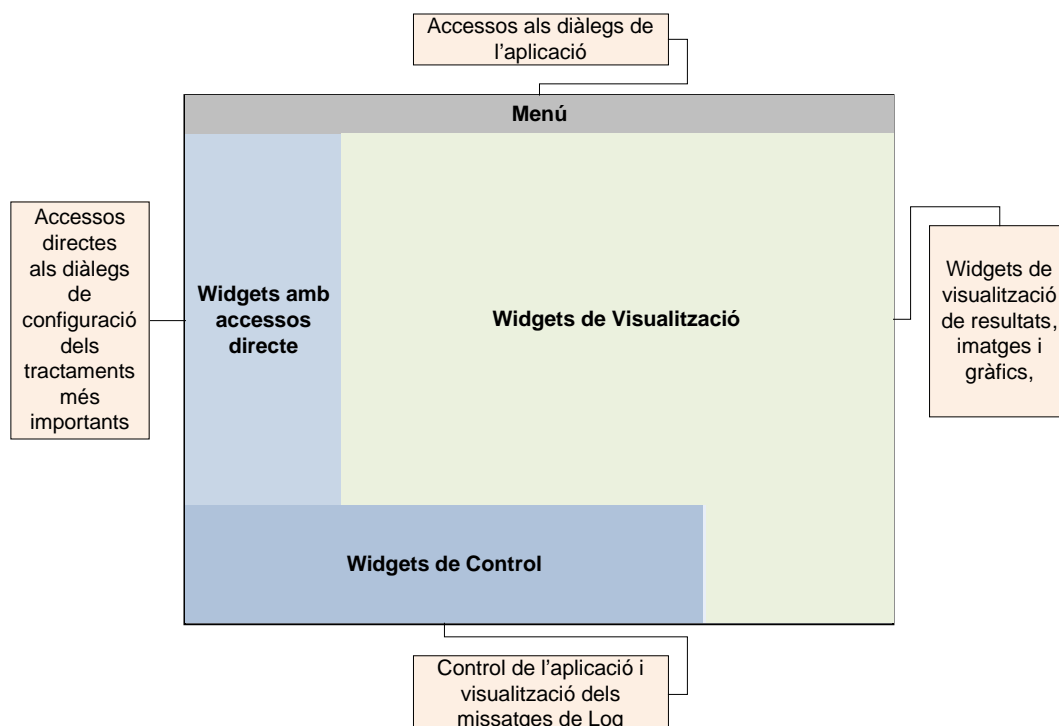
Es suposa que l'aplicació serà instal·lada en una màquina que disposarà de pantalla, teclat i ratolí, encara que aquest últim pot ser dispensable ja que la interfície gràfica està pensada per a poder ser gestionada a través de pantalles tàctils.

Pel que fa les interaccions amb l'usuari, les accions que se li demanen són les mínimes necessàries per tal de proporcionar-li el que desitja. S'ha donat especial importància a fer que la manera en què l'usuari consulti les dades, controlï l'aplicació i configuri els paràmetres, sigui el més simple, ràpida i còmoda possible, de manera que l'usuari no trobi dificultat en la utilització de l'aplicació.

Per tal de poder satisfer tots aquests elements, la pantalla principal ha d'estar dissenyada pensant en les necessitats reals dels usuaris.

Aquesta pantalla carrega els diferents widgets i menús segons els paràmetres llegits de l'arxiu de configuració, i els distribueix al lloc on els toca. A part també serveix com a passarel·la per a passar els missatges i accions entre el controlador i els widgets.

A continuació es mostra un esquema de com es pot estructurar la pantalla principal per a introduir-hi els widgets i els diàlegs del menú.



*Esquema de l'estructuració dels tipus de widgets dins de la pantalla principal*

## 5.4 Disseny dels Menús i Diàlegs

A través de les opcions dels menús de la pantalla principal podem accedir als diferents diàlegs de l'aplicació. Aquests diàlegs, com els widgets, poden realitzar feines diferents segons la seva finalitat.

A continuació es mostra un llistat amb els diferents tipus de diàlegs que podem trobar:

### **Visualització d'informació sobre els resultats de les anàlisis.**

Reben la informació del domini a través del controlador de la interfície gràfica, i passant per la pantalla principal.

### **Informació del sistema i opcions bàsiques**

Realitzen accions comuns com ara guardar o obrir un arxiu de configuració, tancar l'aplicació...

No necessiten informació de les altres capes del domini. Només notifiquen la seva acció al controlador de la interfície gràfica i aquest ja sabrà què fer-ne.

### **Gestió d'usuaris**

Necessita informació de la base de dades per a mostrar els usuaris, així que l'aconsegueix gràcies al controlador de la interfície gràfica, que es comunica amb el controlador de la base de dades.

### **Configuració del sistema**

Podem trobar aquest tipus de diàlegs de configuració dins del menú "Configuració" i dels accessos directes de la pantalla principal, així com en el diàleg de guardar les imatges resultants de les anàlisis.

Aquests diàlegs són els que mostren i modifiquen algun aspecte de la configuració del programa. El que els fa més complicats és el fet de poder canviar la configuració mentre el programa està obert, així com la possibilitat de canviar algun paràmetre i després poder desfer els canvis, prement el botó de cancel·lar del diàleg.

Per resoldre aquest problema s'ha creat una política que segueixen tots aquests tipus de diàlegs, i que s'aplica en algunes de les accions que aquests realitzen:

- Al crear el diàleg:
  - o Es crea una còpia de la configuració del programa. Aquesta còpia serà la que es visualitzarà en els camps del diàleg, i és la que es modificarà quan es canviï algun paràmetre d'aquest.

- Es crea també una cua de missatges on s'aniran emmagatzemant totes les accions que es facin en el diàleg.
- Al modificar algun camp:
  - Es modifica el camp o paràmetre corresponent a la còpia de la configuració local del diàleg.
  - Es crea un missatge identificatiu de l'acció que s'ha realitzat i s'envia a la cua de missatges.
- Al prémer el botó de *Cancel·lar*:
  - S'eliminen totes les instàncies obertes pel diàleg, així com la còpia local de la configuració i la cua de missatges. Després es tanca el diàleg.
- Al prémer el botó d'*Aplicar* o *Acceptar*:
  - S'envia la cua de missatges a la pantalla principal que serà l'encarregada de passar-los al controlador de la interfície gràfica. Aquest els posarà a la cua de missatges pendents de ser processats pel controlador principal.

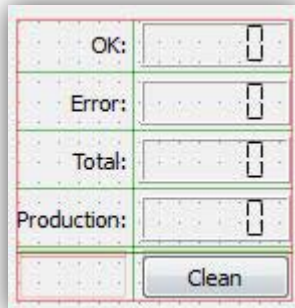
Amb això podrem actualitzar la configuració global del sistema amb els canvis que hem estat fent des del diàleg.

- La configuració global del sistema acabarà sent igual a la local.



## 5.5 Disseny dels widgets

La major part de la superfície de la pantalla principal la formen els **Widgets**, que poden contenir diferents elements gràfics com ara botons, línies de text, selectors, imatges gràfics...



*Exemple de Widget. Comptador de resultats*

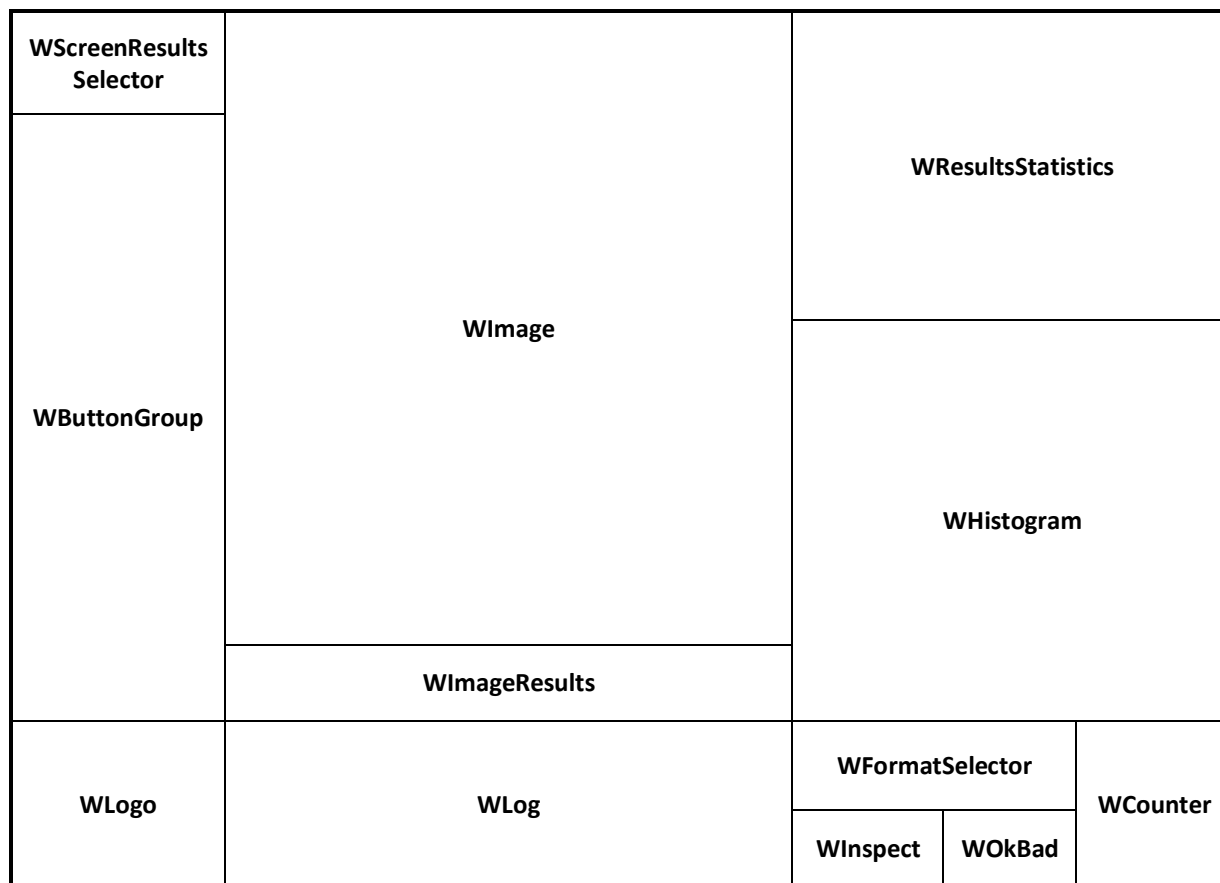
Per a cada funcionalitat que es requereix a la pantalla principal s'ha creat un Widget, i s'han intentat agrupar espacialment dins de la pantalla segons el seu tipus.

Aquests widgets seran els encarregats de:

- Mostrar informació del sistema:
  - o Rebran informació de la capa de domini que s'haurà de mostrar, com ara:
    - Mostrar el Log del sistema
    - Mostrar diferents aspectes dels resultats de les anàlisis com ara comptadors, gràfics, imatges resultants...
- Gestionar l'execució i configuració del sistema:
  - o Modificaran algun aspecte de la configuració del sistema, que s'haurà de comunicar als controladors del domini mitjançant la pantalla principal i el controlador de la interfície gràfica. Les accions que han de permetre aquest tipus de widgets són:
    - Oferir control sobre l'arrencada i l'aturada de les anàlisis
    - Oferir accessos directes als diàlegs de configuració del sistema
    - Oferir la possibilitat de canviar de format.

La seva localització dins de la pantalla principal ve donada a través del fitxer de configuració.

En la pàgina següent es mostra una possible estructuració dels widgets dins la pantalla principal, tenint en compte les explicacions mostrades al punt en el que es parla del disseny de la pantalla principal [Apartat 5.3]:



A l'apartat dedicat a les captures de pantalla de l'aplicació es pot visualitzar gràficament els diferents widgets que apareixen en aquest esquema [Apartat 7.1 i 7.4].

A continuació es mostra un llistat amb l'explicació de cada un dels widgets:

- **WScreenResultsSelector:**
  - Selector per a poder escollir quina imatge es vol visualitzar al widget WImage.
- **WButtonGroup:**
  - Widget que conté diferents botons amb accessos directes a algun dels principals diàlegs de configuració dels tractaments.
- **WLogo:**
  - Conté una imatge amb el Logo de l'empresa E2M (amb un link a la web oficial). Pot contenir també altres logos d'empreses clients.
- **WImage:**
  - Widget on es mostra la imatge seleccionada al selector del widget "WScreenResultsSelector".
- **WImageResults:**

- Mostra de forma gràfica els resultats de les últimes anàlisis.
- **WLog:**
  - Mostra el Log de l'aplicació, ensenyant els events, canvis d'estat i els errors que es produeixin.
- **WResultsStatistics:**
  - Mostra un gràfic de barres amb diferents resultats de les anàlisis.
- **WHistogram:**
  - Mostreja, mitjançant un gràfic de tipus "Histograma", un resultat de les anàlisis.
- **WFormatSelector:**
  - Mostra un selector per a poder canviar de format, així com un botó per eliminar un format existent i un altre per afegir-ne un de nou.
- **WInspect:**
  - Mostra els controls (botons) per poder iniciar i parar les inspeccions.
- **WOkBad:**
  - Mostra de manera gràfica, el resultat global de la última inspecció.
- **WCounter:**
  - Mostra diferents comptadors numèrics per a ensenyar la quantitat de resultats bons i dolents, producció actual, i inspeccions realitzades.

## 5.6 Diagrames de seqüència

En aquest apartat es mostren els diagrames de seqüència de les operacions que s'han sol·licitat per part dels promotors del projecte. S'ha descartat fer els diagrames de totes les operacions perquè hi ha moltes accions que són repetitives, simples o que segueixen un flux molt similar a alguna altra. Així que es mostra un conjunt de les que són més representatives, complexes i importants.

Per cada diagrama o conjunt de diagrames es mostra una llista amb les operacions que podria representar, ja que el funcionament és similar.

### 5.6.1 Inicialització de l'aplicació

Aquest diagrama presenta el flux que segueix l'aplicació durant la **inicialització**. No és aplicable a cap altra acció ja que es duu a terme únicament cada vegada que iniciem el programa.

S'hi mostra la interacció pertanyent a la part de la interfície gràfica i del controlador general, ometent els passos de les capes de domini i dades.

Només es mostra la creació del Widget comptador de resultats (WCounter) com a exemple d'inicialització d'un widget.

La pantalla principal (MainWindow) crea tots els altres widgets de la mateixa forma que aquest.

Aquest widget, s'ha de subscriure als resultats que vol mostrar. Això ho ha de fer per tal de poder obtenir els valors que necessita mostrar. Perquè un widget pugui veure els valors d'alguns resultats, així com algunes imatges resultants, s'hi ha de subscriure.

Els resultats dels tractaments només estan disponibles per a aquells widgets que hi estiguin subscriïts. Així es pot controlar més tota la informació que es manipula per l'aplicació.

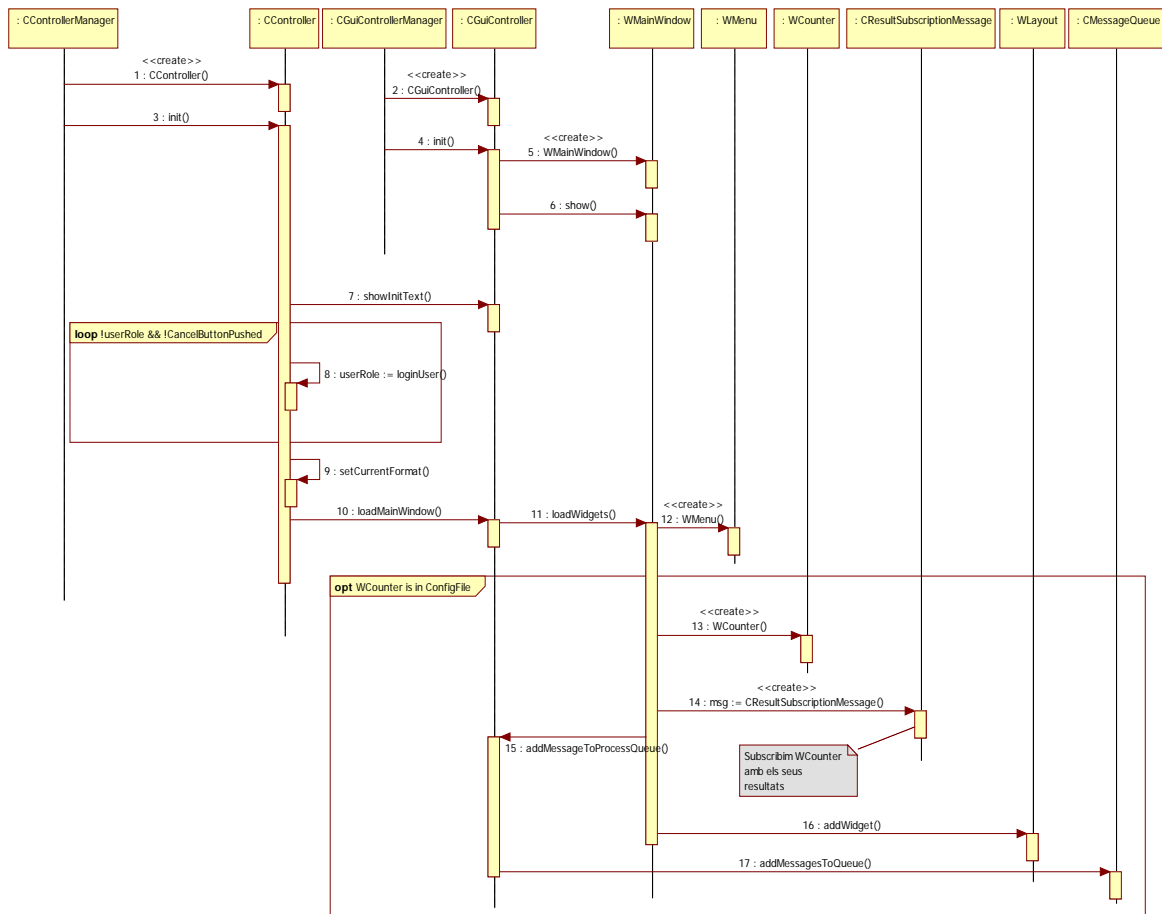


Diagrama d'inicialització de l'aplicació

### 5.6.2 Interacció Interfície gràfica - Controlador

En aquest apartat es mostra un diagrama que podria ser vàlid per a qualsevol acció en que l'usuari clica algun botó d'algun widget de la interfície gràfica. Amb aquest exemple es pot tenir una idea del camí que es segueix des que succeeix algun event a la interfície gràfica i es tracta a la capa de domini a través del controlador.

Aquest diagrama acaba el seu flux al controlador, que serà l'encarregat de transferir i executar les accions a la capa de domini.

Es mostra com a exemple el diagrama pertanyent a la funcionalitat que permet **iniciar la inspecció** (quan es prem el botó d'iniciar), és a dir la representació gràfica del cas d'ús "Arrencar les inspeccions" [Apartat 4.2.2.1].

Aquest diagrama podria ser extrapolable a qualsevol dels següents casos d'ús:

- Parar el funcionament del sistema.
  - o S'executa quan es rep un missatge del tipus "STOP\_MESSAGE"
- Realitzar una sola inspecció a la imatge actual.
  - o S'executa quan es rep un missatge del tipus "RUN\_MESSAGE"
- Tancar l'aplicació.
  - o S'executa quan es rep un missatge del tipus "END\_MESSAGE"
- Obrir arxiu de configuració.
  - o S'executa quan es rep un missatge del tipus "OPEN\_CONFIG\_FILE\_MESSAGE"
- Guardar un fitxer de configuració.
  - o S'executa quan es rep un missatge del tipus "SAVE\_CONFIG\_FILE\_MESSAGE"

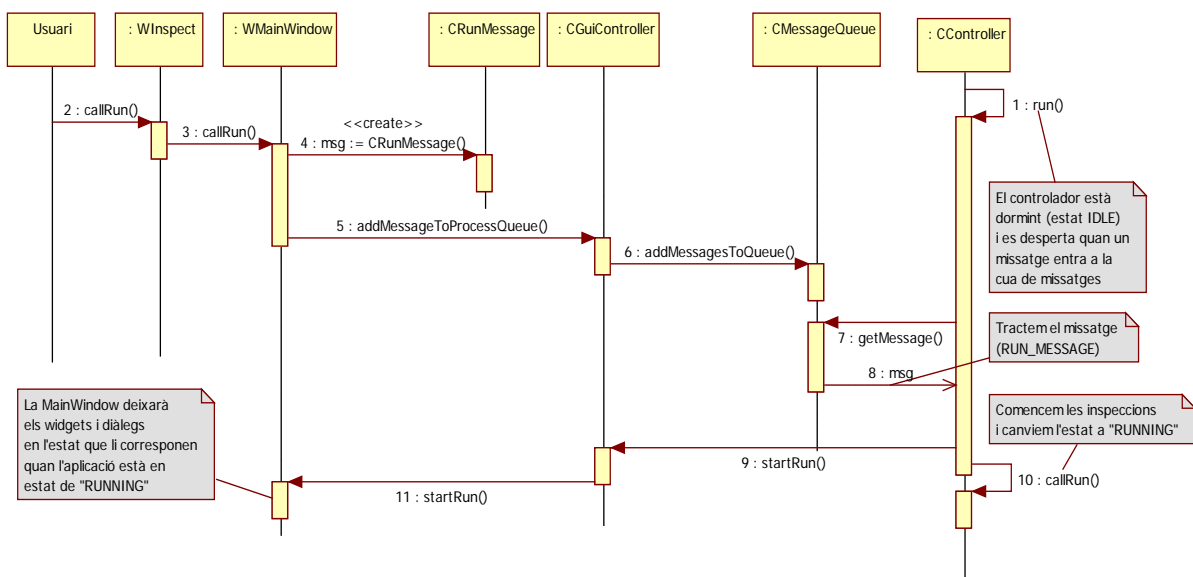


Diagrama d'inici de les inspeccions

### 5.6.3 Interacció Controlador - Interfície gràfica

En aquest apartat es mostra un diagrama que podria ser vàlid per a qualsevol acció en que el controlador rep un missatge que el fa mostrar o modificar algun aspecte per la interfície gràfica. Amb aquest exemple es pot tenir una idea del camí que es segueix des que es rep un missatge al controlador fins que es tracta a la interfície gràfica, a través del controlador de la interfície gràfica i de la pantalla principal.

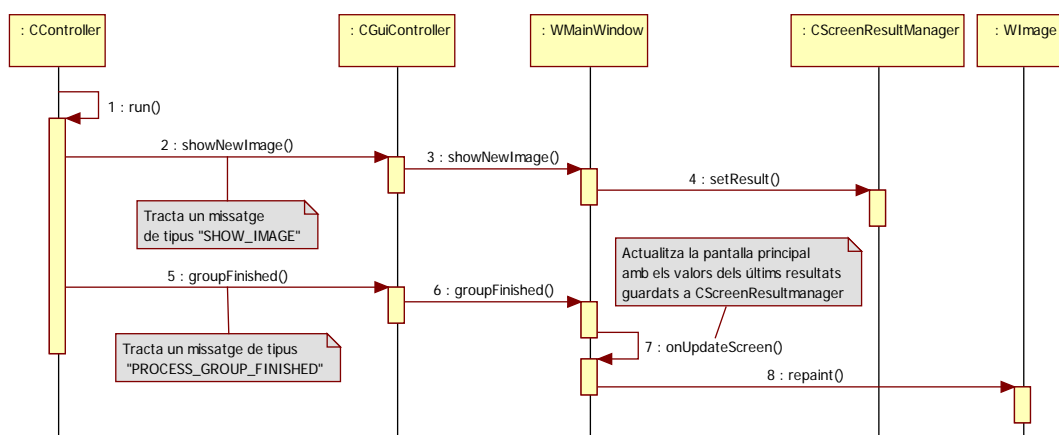
Es mostra com a exemple el diagrama pertanyent a la funcionalitat que permet **mostrar la imatge resultant de l'última anàlisi**, és a dir la representació gràfica del cas d'ús "Mostrar l'última imatge" [Apartat 4.2.2.1].

Per a mostrar aquesta última imatge, com es pot veure al diagrama següent, són necessaris dos passos:

1. Quan es rep un resultat o imatge resultant, es guarda en una estructura de dades utilitzant la classe "CScreenResultManager".
2. Quan arriba un missatge confirmant que el grup de processos ja està finalitzat, s'actualitzen els components de la pantalla principal bolcant els resultats i les imatges resultants guardades a aquesta classe.

Aquest diagrama podria ser extrapolable a qualsevol dels següents casos d'ús:

- Mostrar el Log del sistema
- Mostrar l'últim resultat
- Mostrar els resultats
- Gràfic de l'evolució dels resultats
- Histograma de resultats
- Mostrar els resultats de les anàlisis
- Mostrar un sol resultat
- Mostrar el log d'events



*Diagrama que representa el camí per a mostrar una imatge per pantalla*

\* La comunicació entre el CGuiController i la WMainWindow és asíncrona, ja que es troben en threads diferents

### 5.6.4 Comunicació amb la base de dades i recàrrega de la configuració

Algunes funcionalitats requereixen tornar a carregar la configuració del sistema, així com les opcions de la interfície gràfica. Ens trobem en un cas així quan canviem d'usuari o de format amb el qual estem treballant.

Aquests diagrames mostren el flux que s'obté en el moment en que l'usuari realitza l'acció de **canviar d'usuari**. Aquest cas realitza les dues accions que es volen representar en aquest apartat: la comunicació amb la base de dades d'usuaris i l'acció de recarregar la configuració.

El primer diagrama (el que representa la comunicació amb la base de dades) podria ser extrapolable als següents casos d'ús:

- Afegir un usuari
- Eliminar un usuari

El segon diagrama (el que representa l'acció de recarregar la configuració i la interfície gràfica) podria ser extrapolable als següents casos d'ús:

- Seleccionar un format
- Afegir un nou format
- Eliminar un format

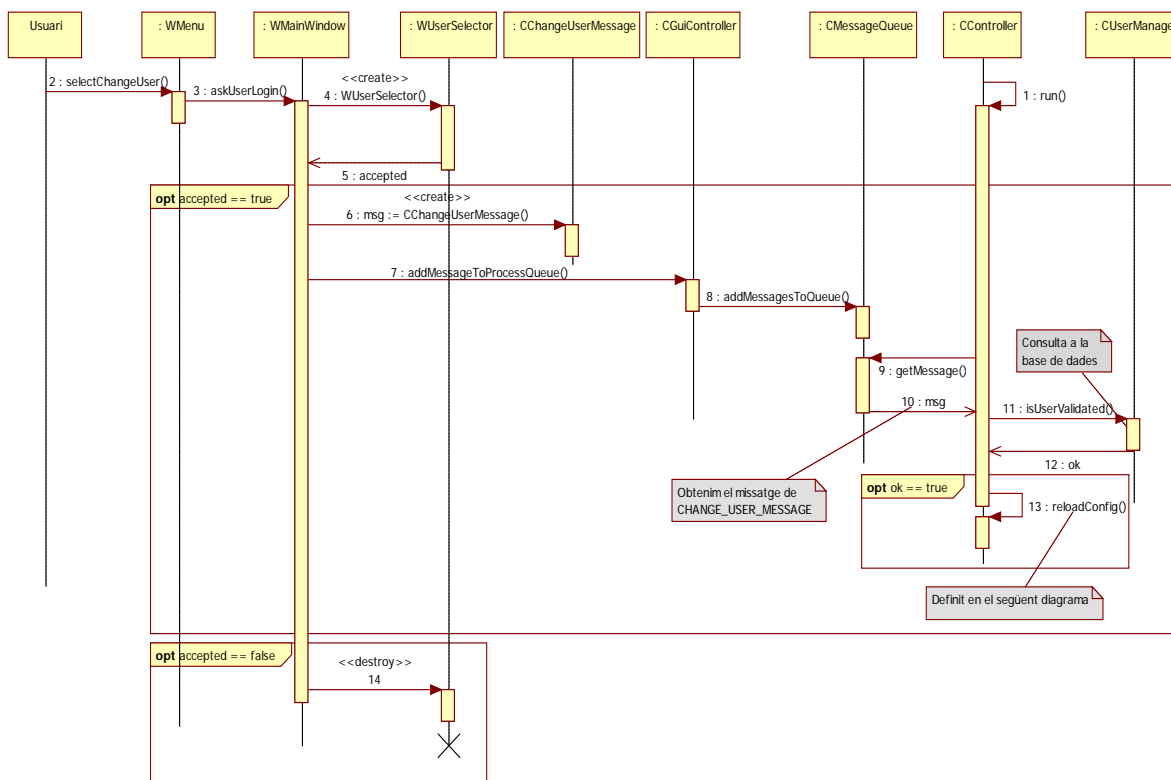


Diagrama que representa el camí per a poder canviar d'usuari



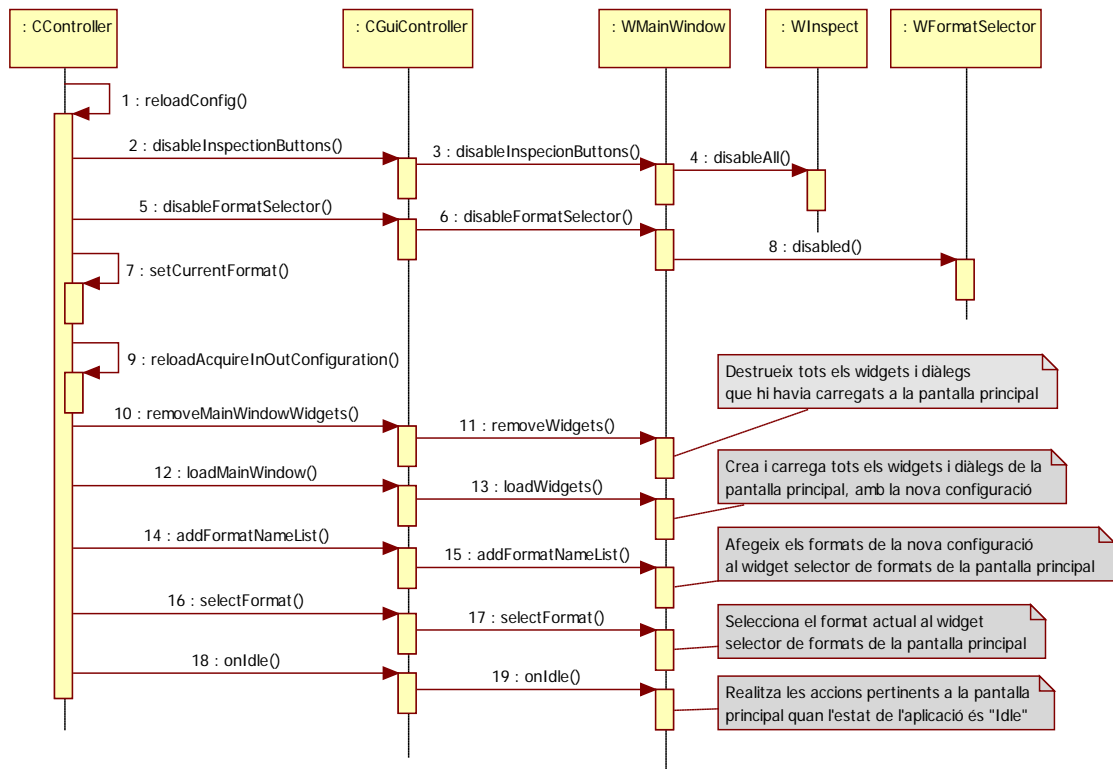


Diagrama que representa el camí per a poder recarregar la configuració

---

### 5.6.5 Diàlegs dels menús de configuració de l'aplicació

---

En aquests diagrames es pot observar el flux de l'aplicació que realitzen els diàlegs de configuració.

Aquests diàlegs contenen una còpia local de la configuració del programa, amb la que treballen mentre estan oberts. Cada acció que es realitza s'aplica sobre aquesta còpia local i es guarda en forma de missatge en una cua de missatges, la qual serà bolcada cap al controlador en acceptar el diàleg (prémer el botó d'aplicar o el d'acceptar).

Cada diàleg realitza accions diferents a la seva còpia local, sense modificar la configuració global de l'aplicació. Aquestes accions dependran de les opcions que tingui cada interfície gràfica.

Per tal de poder interpretar els diagrames correctament, es procedeix a la definició de les classes més importants que hi apareixeran, indicant quina és la seva funció en aquests diagrames:

- **WProcessManagerDialog**
  - o Widget que conté i gestiona els processos i els tractaments, així com la configuració local del diàleg. És la classe que engloba tots els components del diàleg.
  
- **WListDialog**
  - o Widget que conté la llista de processos.
  
- **WProcessDialog**
  - o Widget que representa un procés. Conté i gestiona els tractaments del procés, així com els seus paràmetres.
  
- **CConfigProcessManager**
  - o Classe que representa i gestiona la còpia local de la configuració. Conté tota la informació dels processos i tractaments del diàleg. Rep els missatges i els processa, delegant les accions a la classe que li pertoca.
  
- **CTreatmentController**
  - o Gestiona els plugins dels tractaments. En aquest cas ens serveix per a carregar els diàlegs de configuració corresponents al plugin del tractament que volem mostrar.

En els següents diagrames es mostren les accions més complexes que es realitzen al menú de configuració de les anàlisis, i que es poden extrapolar als altres diàlegs de configuració. Es mostren les accions d'aquest menú perquè és el més complex i el que realitza més accions. Cada acció està separada en un diagrama diferent.

**Obrir el diàleg de configuració de les anàlisis:**

Es mostra la seqüència a partir del moment que es rep un missatge del tipus "SHOW\_PROCESS\_MANAGER\_CONFIG\_MESSAGE" al controlador general.

És el procés que es segueix quan s'obre el diàleg de configuració de les anàlisis, i es carreguen tots els diàlegs i widgets de configuració dels processos i tractaments.

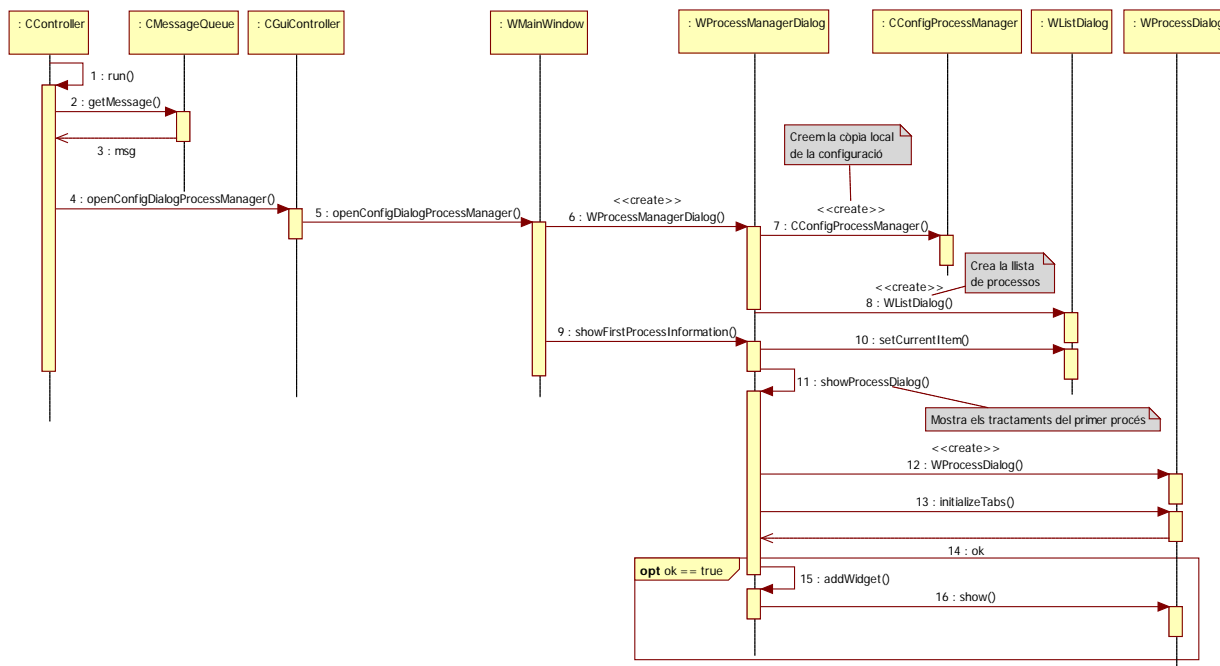


Diagrama que representa el camí per a poder mostrar el diàleg de configuració de les anàlisis

**Carregar tots els tractaments del procés**

Es mostra la seqüència que inicialitza les pestanyes (tabs) que contenen els widgets de configuració de cada tractament. A cada una se li carrega el widget de configuració del plugin que li correspon al tractament.

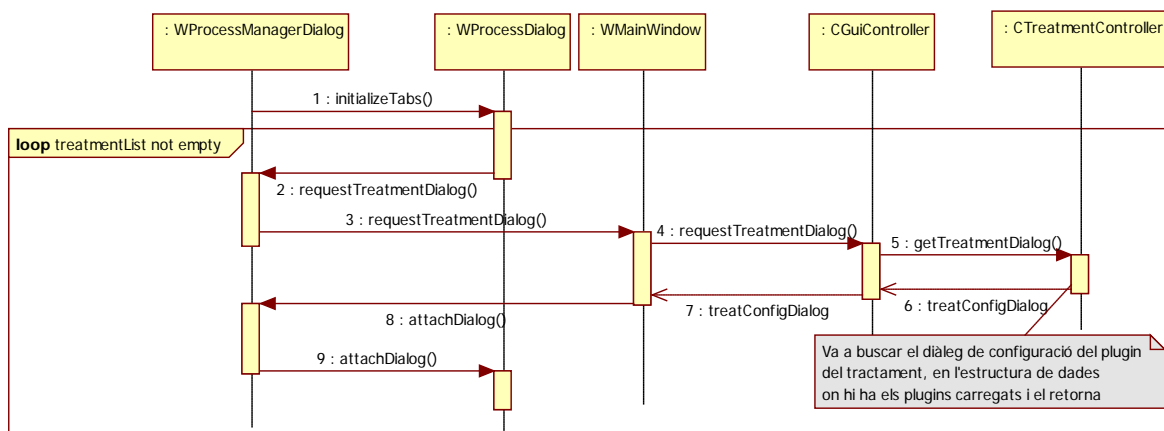


Diagrama que representa el camí per poder carregar tots els tractaments d'un procés

### Inserir un nou procés

Es mostra la seqüència d'execució a partir que l'usuari vol inserir un nou procés a partir del Widget que conté la llista de processos. Aquest diagrama pot englobar l'acció de renombrar i eliminar un procés, ja que es segueixen els mateixos passos. Al final es mostra com es modifica la configuració local mitjançant el tractament d'un missatge de tipus "ADD\_PROCESS\_ACTION\_MESSAGE".

Quan s'insereix un nou procés s'ha d'inserir també un nou tractament, ja que no hi pot haver un procés sense cap tractament, per això també està reflectit en aquest diagrama.

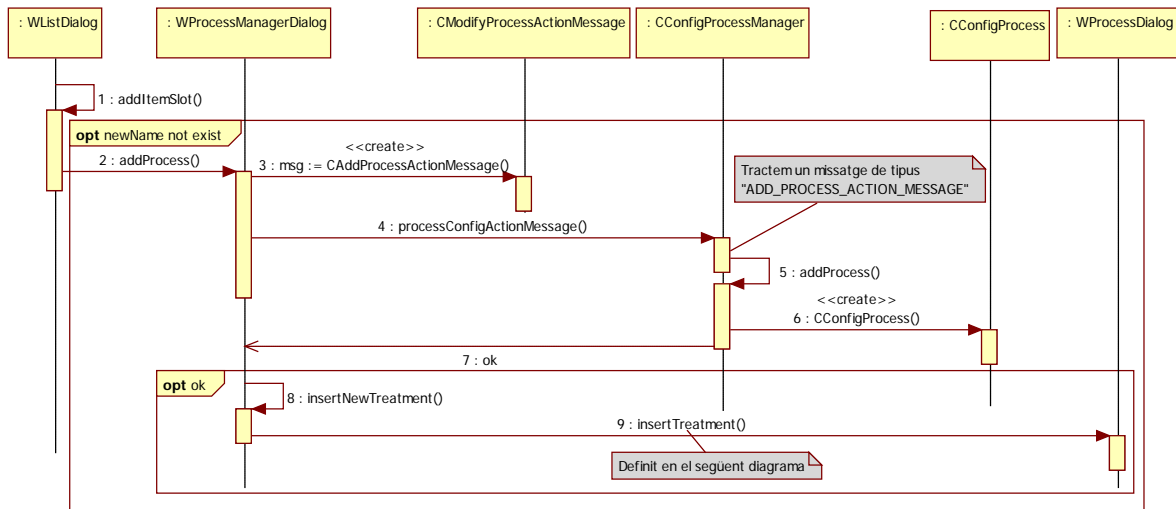


Diagrama que representa el camí per poder inserir un nou procés

### Inserir un nou tractament

Aquest flux d'execució té molta relació amb l'anterior, ja que com s'ha dit abans, sempre que es crea un procés s'ha de crear com a mínim un tractament. Pot englobar l'acció de renombrar i eliminar un tractament, ja que es segueixen els mateixos passos, i s'hi mostra la seqüència a partir del moment que es crida a la funció d'afegir un tractament nou al diàleg de configuració de processos i tractaments, el diàleg general (WProcessManagerDialog).

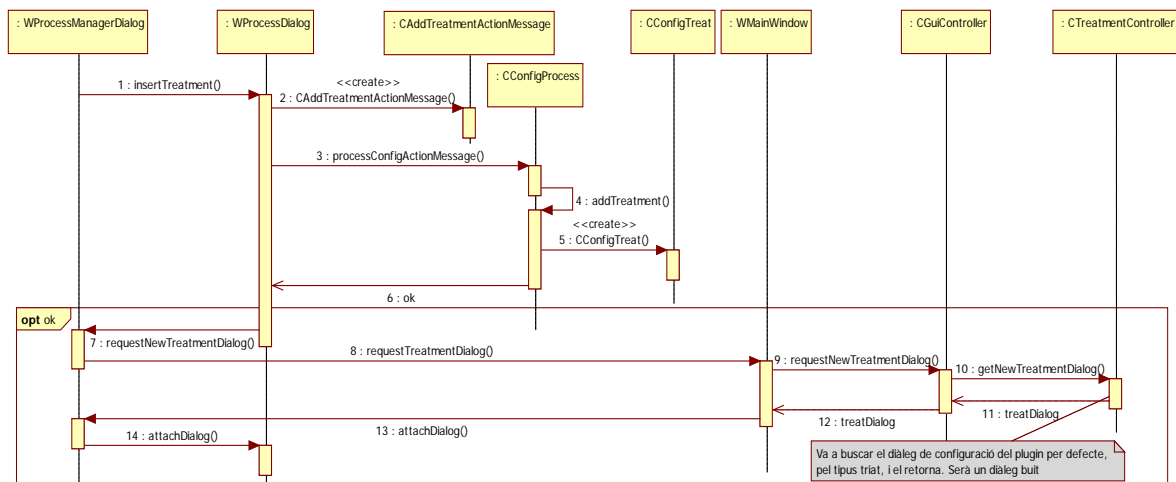


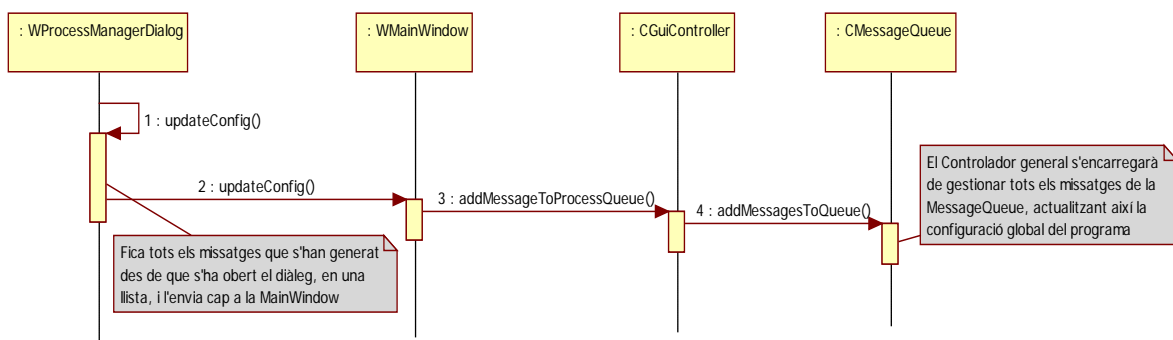
Diagrama que representa el camí per a poder inserir un nou tractament

### Guardar els canvis

Aquest flux d'interacció s'aplica quan s'activa el botó d' "Aplicar" o d' "Acceptar" (en aquest últim cas es tanca el diàleg després de realitzar els canvis descrits aquí).

El que es vol representar en aquest diagrama és com s'envien tots els missatges que s'han generat amb les modificacions als processos i tractaments, cap a la cua de missatges, de la que llegeix constantment el controlador general. Amb això permetem actualitzar la configuració global de l'aplicació.

Després d'aplicar aquests canvis, la configuració global de l'aplicació serà la mateixa que la local del diàleg.



*Diagrama que representa el camí per poder guardar els canvis fets en el diàleg*



## 6. IMPLEMENTACIÓ

6.1	Introducció	121
6.2	Tecnologies utilitzades	122
6.3	Entorn de desenvolupament	123
6.3.1	Components del grup	123
6.3.2	Servidor del departament	124
6.4	Components implementats	126
6.4.1	Controlador de la interfície gràfica	126
6.4.2	Pantalla Principal	126
6.4.3	Menús i Diàlegs	127
6.4.4	GUI Widgets	130
6.5	Persistència de les dades	135
6.5.1	Configuració	135
6.5.2	SGBD	137
6.5.3	Estructures de dades	137





## 6.1 Introducció

En aquest apartat es tractaran els principals detalls de la implementació del disseny obtingut en l'apartat anterior.

El pas que es realitza a la implementació és agafar el disseny del sistema, i mitjançant la codificació a través d'un llenguatge de programació, obtenir un conjunt que permeti ser executat, aconseguint acabar tot el desenvolupament del sistema.

En aquest capítol doncs, ens centrarem en l'explicació de les polítiques d'implementació que s'han fet servir per a realitzar tots els components de la interfície gràfica.

També es detalla un apartat dedicat a l'explicació de la persistència de les dades que s'ha utilitzat en l'aplicació "VisioInspect 2", i que cal saber-la i entendre per tal de realitzar la interfície gràfica, ja que aquesta hi accedeix en diferents ocasions.

Es poden observar alguns detalls del codi utilitzat per a implementar aquests components amb alguns exemples. No s'ha volgut introduir gran quantitat de codi en aquest apartat per a poder-lo fer més llegible i amè.

Igualment s'explica com s'ha realitzat la creació de codi de cadascún dels components descrits als apartats anteriors d'especificació i disseny, així com la manera en que es comuniquen aquests components, siguin o no de la mateixa capa.

## 6.2 Tecnologies utilitzades

Com ja s'ha explicat en apartats anteriors, el llenguatge de programació que s'ha utilitzat per a implementar la interfície gràfica és **C++** fent ús en tot moment de les funcionalitats que ens ofereix la llibreria **Qt**.

Entre d'altres funcionalitats, la llibreria Qt ens permet l'ús de **Signals** i **Slots** per a gestionar algunes crides a funcions. Aquest mecanisme permet implementar el patró Observador.

La idea és que des d'una classe es poden enviar *Signals* que poden contenir informació, i que poden ser tractats per altres classes fent servir unes funcions especials anomenades *Slots*.

Aquest mecanisme és molt útil per a gestionar i comunicar les interfícies gràfiques amb la capa de domini, ja que permet una **comunicació asíncrona** entre el component que envia el *Signal* i el que el tracta amb l'*Slot*. Per aquest mateix motiu també és de molta utilitat aquest mecanisme per a tractar components que estan executant-se en threads diferents.

La part gràfica tant dels diàlegs com dels Widgets està implementada usant els components gràfics que ens ofereix el Qt. Així mateix, moltes d'aquestes utilitats han estat creades mitjançant l'editor gràfic del Qt (integrat dins l'eina de desenvolupament *Qt Creator*).

El motiu de la utilització d'aquestes tecnologies, rau principalment en el fet que la resta de components de l'aplicació estan implementades amb aquest llenguatge, i que com que aquest ja ens ofereix les eines que ens permeten implementar totes les funcionalitats definides pels requeriments del projecte, s'ha decidit optar per aquest camí.

A part d'aquestes llibreries també se n'han usat d'addicionals per a implementar algunes funcionalitats de la interfície gràfica:

- **OpenCV**
  - o És una llibreria multi-plataforma que facilita la manipulació i el tractament d'imatges mitjançant visió artificial.
  - o S'ha fet servir per a implementar la classe que representa les imatges que es mostren al widget "*WImage*". S'ha creat una classe anomenada "*CE2MImage*" que és una encapsulació de la "*IplImage*", classe que representa una imatge de la llibreria OpenCV.
  
- **Llibreries QWT**
  - o Llibreria que conté components (Qt Widgets) i classes útils per a representar diferents tipus de gràfiques.
  - o S'ha fet servir per a implementar alguns dels widgets que mostren gràfiques de resultats, com per exemple el *WPlot* (mostreja una senyal o resultat) i *WHistogram* (histograma que mostreja el valor d'un resultat).

## 6.3 Entorn de desenvolupament

En aquest apartat es descriurà l'entorn de treball on s'ha desenvolupat el projecte, tant el grup de persones que l'han fet possible com les eines que s'han utilitzat durant la seva execució.

### 6.3.1 Components del grup

Hi ha hagut tres persones que, dins del departament d'**I+D** de l'empresa, ens hem encarregat de la implementació d'aquest projecte. Tot seguit mostro quines han estat les feines de cadascú en el projecte que es tracta aquí, així com dins el projecte global del VisioInspect2 [Apartat 1.6.2.3]:

- 2 Analistes programadors:
  - De la interfície gràfica, han ajudat a implementar algunes funcions, així com a desenvolupar les diferents interfícies de configuració dels plugins.
  - Són els que s'encarreguen de la implementació de la capa de domini i dades de l'aplicació VisioInspect2, així com dels plugins de tractaments i de càmeres. S'encarreguen també de l'administració del servidor i de donar suport a d'altres treballadors de l'empresa.
- 1 Programador:
  - S'encarrega de la implementació de la interfície gràfica, així com de tota la documentació del projecte (documentació de la planificació, realització de diagrames, redacció de la memòria, etc.) .
  - Ajuda en la implementació d'algunes funcions de la capa de domini i dades de l'aplicació VisioInspect2, com ara la creació i manipulació de la base de dades, investigació sobre algun plugin, com el del tractament de codi de barres, etc.

---

### 6.3.2 Servidor del departament

---

A part de l'ordinador personal que cadascú té per a poder desenvolupar la seva part del projecte, es disposa d'un servidor al departament que ens ajuda a l'hora de comunicar-nos, i de mantenir les còpies de seguretat.

Aquest disposa del hardware necessari [Apartat 1.6.2.1] per a gestionar els serveis que conté, així com una distribució de Linux especial per a servidors (**Ubuntu Server Edition**).

Els serveis que s'utilitzen d'aquest servidor per a desenvolupar el projecte són:

- **Subversion** (Control de versions de codi - SVN):
  - S'utilitza una aplicació pel control de versions del software que es genera. Al realitzar el software de manera distribuïda entre els diferents membres del departament, aquesta eina és molt útil per a mantenir la coherència en el codi i poder-lo tenir actualitzat en tot moment amb les últimes versions. A més a més s'utilitza per a la gestió de la configuració del programa.
  
- **MediaWiki**:
  - S'utilitza una web interna de l'empresa, emmagatzemada en aquest servidor, en la que hi ha instal·lada una aplicació que permet gestionar informació de diferents aspectes de l'empresa, i específicament relacionada amb aquest projecte.  
Molta de la documentació, informació, dubtes i preguntes referides al projecte que es tracta en aquesta memòria, estan recollits en aquesta web, mitjançant l'estructuració que ens ofereix l'aplicació MediaWiki (aplicació amb la que es desenvolupa la coneguda Wikipedia).
  
- **Compilació automàtica**:
  - Es disposa d'un script al servidor que realitza compilacions automàtiques del codi que hi ha al SVN cada hora, disposant així sempre d'un codi compilat amb la última versió que hi ha al servidor. Això ens permet agilitzar algunes proves i poder comprovar alguns errors que es puguin donar amb el codi que hi ha al SVN.
  
- **DoxyGen**:
  - És una aplicació que ens permet generar documentació del codi que es crea directament quan s'està implementant.  
Seguint unes regles d'implementació, es pot generar en qualsevol moment una documentació de totes les classes i funcions del codi mitjançant aquesta aplicació.  
També s'obté aquesta documentació actualitzada de forma horària a través del script de compilació.

- **BugZilla:**
  - És una aplicació que ens permet portar un control dels bugs (errors) que es van trobant quan es genera el codi. Aquí es poden publicar aquests errors, i queden enregistrats fins que el responsable del bug no els soluciona. Emet un recordatori al seu responsable.
  
- **CopperMine Photo Gallery:**
  - És un servei que ens ofereix la manipulació i gestió d'una galeria d'imatges emmagatzemada al servidor. Ens permet pujar noves fotos, eliminar-ne, estructurar-les en diferents categories, etc.  
Ens serveix per a guardar les diferents imatges de proves per a l'aplicació, així com diferents imatges resultants d'algun tractament.

*\* Es pot veure una relació de les llicències i costos de cada una d'aquestes aplicacions a l'apartat de planificació de costos referent al software [Apartat 1.6.2.2].*

## 6.4 Components implementats

### 6.4.1 Controlador de la Interfície gràfica

El controlador de la interfície gràfica és l'encarregat de la comunicació entre la capa de domini i la capa de presentació (la interfície gràfica).

Com s'ha vist en els apartats anteriors, és l'encarregat de crear la pantalla principal de la interfície, així com d'inicialitzar-la. Així mateix també és l'encarregat d'informar de qualsevol canvi que provingui de la capa de domini a aquesta pantalla principal.

A part d'actualitzar la pantalla principal amb els events provinents de la capa de domini, també és l'encarregada de propagar els missatges que arriben des de la pantalla principal, i que han de ser executats pel controlador de la capa de domini.

### 6.4.2 Pantalla Principal

La pantalla principal (MainWindow) ha de definir una base per encabir tots els widgets i menús que realitzen les funcions requerides, així com accessos directes als diàlegs que també s'han creat.

Per a desenvolupar aquest component s'ha fet una reimplementació de la classe que ens ofereix la llibreria Qt per a aquests tipus de pantalles: *QMainWindow*. Aquesta classe ens ofereix diverses utilitats que podem utilitzar, com és la possibilitat d'afegir un menú automàticament a la part superior de la finestra, i proporcionar-nos les funcions necessàries per tancar i iniciar la pantalla principal.

A part d'aquestes funcions, s'ha introduït un *Layout* que exerceix com a base on hi aniran situats els diferents widgets.

Tota la informació dels diàlegs i widgets que s'han de carregar a la pantalla principal, així com la posició que aquests últims han de tenir, està definida a l'arxiu XML de configuració de l'aplicació, i del qual es parla a l'últim punt d'aquest apartat.

Aquest arxiu es llegeix des de la capa de domini i carrega la pantalla principal amb la informació que conté.

El component de la pantalla principal és l'encarregat de gestionar tots aquests widgets i diàlegs, duent a terme accions com per exemple:

- Carregar els widgets al lloc que els pertoca amb la informació que els correspon.
- Eliminar els widgets quan l'aplicació s'està tancant o quan s'està recarregant la configuració.

- Actualitzar la informació i les imatges dels widgets quan es requereix.
- Enviar els missatges que li arriben des dels diàlegs i widgets cap al controlador de la interfície gràfica. Actua de propagador dels missatges quan aquests necessiten accedir a la capa de domini a través del controlador.
- Implementar alguna de les accions dels diàlegs o dels widgets que requereixen d'informació de la capa de domini o de dades. Aquest pot aconseguir aquesta informació demanant-la directament al controlador de la interfície gràfica.

### 6.4.3 Menús i Diàlegs

Com s'ha explicat a l'apartat anterior, la pantalla principal conté una barra de menú a la part superior. Per a implementar aquest menú també s'ha fet ús d'una classe que ens ofereix Qt: la *QMenuBar*.

Aquesta classe ens ofereix la possibilitat de definir diferents Menús que s'afegiran a aquesta barra. Per a realitzar aquests menús s'ha usat la classe de Qt *QMenu* que ens facilita l'opció de poder afegir accions a aquests menús.

Cada acció obre un diàleg dels que hi ha implementats al sistema. Depenent de la tipologia del diàleg i de les seves funcionalitats (explicades a l'apartat de disseny), s'ha decidit optar per una política d'implementació o una altra.

Es poden observar les captures de pantalla d'aquests diàlegs a l'apartat de les imatges de la interfície gràfica d'aquesta memòria, dedicades als diàlegs [Apartat 7.2].

Tot seguit mostrem els diferents models de diàlegs i els detalls de la seva implementació:

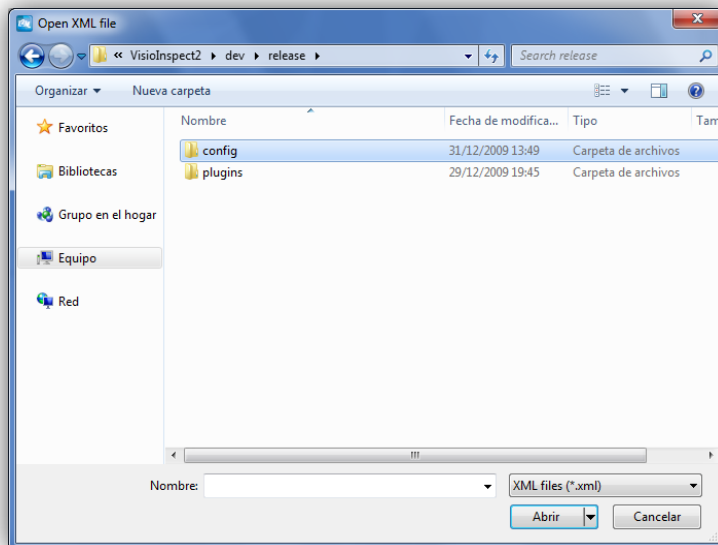
#### Diàlegs d'informació i d'opcions bàsiques

Aquests diàlegs són els que executen accions bàsiques i que poden ser implementats simplement amb l'ajuda de diàlegs que ens ofereix el mateix Qt.

Els diàlegs per poder obrir i guardar la configuració, per afegir o eliminar un usuari o per mostrar la informació del sistema, són exemples d'aquest punt.

Les classes de Qt que s'han usat per a implementar aquestes funcions són:

- *QFileDialog*: per obrir i guardar l'arxiu de configuració.
- *QInputDialog*: per a afegir i eliminar usuaris.
- *QMessageBox*: per a mostrar el diàleg amb la informació del sistema.



*Diàleg per obrir un arxiu de configuració XML (exemple del QFileDialog).*

### Diàlegs de visualització

Aquests diàlegs són els que no requereixen d'accions complexes per a poder realitzar la funcionalitats que tenen assignades. Simplement són diàlegs que no estan implementats a la llibreria de Qt, i que han estat creats només per aquesta aplicació.

Aquests diàlegs no modifiquen la configuració global de l'aplicació, ja que només serveixen per mostrar coses.

Les funcionalitats que usen diàlegs d'aquests tipus són:

- Visualització de resultats: contenen una pantalla connectada als resultats de les anàlisis i els mostren de forma gràfica i numèrica
- Visualització del plot d'un sol resultat: després d'escollir quin resultat individual es vol mostrar, apareix un gràfic amb l'evolució que aquest resultat està tenint en les anàlisis.
- Visualització de les entrades de l'arxiu de Log, i dels temps de les anàlisis.

### Diàleg de selecció d'usuari

Només hi ha un diàleg d'aquest tipus (*WUserSelector*), i és el que s'utilitza per a escollir un usuari de la base de dades. Es fa servir a l'inici de l'aplicació (quan s'ha d'escollir amb quin usuari es vol entrar), i quan es vol canviar d'usuari a través del menú.

La peculiaritat d'aquest diàleg és que necessita informació de la base de dades, però tal i com s'ha mostrat a l'apartat de disseny [Apartat 5.6.4], es recolza amb el controlador de la interfície gràfica per a obtenir-la.



### **Diàleg de configuració**

Aquests diàlegs són diferents als altres, ja que són més complexes. La diferència recau en el fet de que necessiten una còpia de la configuració global del sistema, així com una cua de missatges de les accions que estan realitzant [Apartat 5.6.5].

Per a poder resoldre aquest tema s'ha creat una classe per a cada diàleg, encarregada de gestionar aquests missatges i còpies de la configuració. Aquestes estan situades en classes diferents segons el diàleg, però seran les encarregades de:

- Gestionar els missatges:
  - o S'han creat diferents tipus de missatges identificatius [Apartat 4.3.5] de totes les accions que es poden realitzar dins d'aquests diàlegs. La seva estructura és l'adequada perquè després puguin ser interpretats pel controlador.
  
- Gestionar la còpia local:
  - o S'ha creat un nou paràmetre que conté una còpia de la part de la configuració que s'està modificant. Com que tots els elements de la configuració (processos, tractaments, informació de les càmeres...) estan guardades en memòria mitjançant la utilització de classes, s'ha de fer una còpia de la informació d'aquestes classes.

#### 6.4.4 GUI Widgets

Cada Widget és una reimplementació de la classe *QWidget* de Qt, aprofitant les propietats bàsiques que aquest ofereix i afegint-hi les específiques que cadascun requereix.

Així i tot s'ha intentat seguir una estructura d'arxius bàsica per cada widget, en la que podem trobar que per a cada widget s'ha creat una carpeta amb el seu nom. Dins d'aquesta carpeta hi ha l'arxiu del codi font (amb extensió .cpp), l'arxiu de les capçaleres (amb extensió .h) i l'arxiu amb la representació gràfica d'aquest (amb extensió .ui).

La representació gràfica de cada diàleg (tant del menú com dels accessos directes) també s'ha realitzat mitjançant Widgets, seguint l'estructuració d'arxius descrita anteriorment. Es poden observar les captures de pantalla d'aquests widgets a l'apartat de les imatges de la interfície gràfica d'aquesta memòria, dedicades als widgets de la pantalla principal [Apartat 7.1].

Seguidament es mostra la implementació d'un widget simple (*WInspect*) que conté 3 botons per controlar les inspeccions del sistema. Es poden observar els 3 arxius descrits anteriorment, i què és el que conté cadascun:

**WInspect.cpp** (conté el codi font):

```
#include "WInspect.h"

WInspect::WInspect(EUserRoles::EUserRoles userRole,
                  QWidget *parent)
    : QWidget(parent),
      m_eUserRole(userRole)
{
    ui.setupUi(this);
    ui.pushButtonStop->setEnabled(false);
    connect(ui.pushButtonRun, SIGNAL(clicked()), this,
            SLOT(callRun()));
    connect(ui.pushButtonStop, SIGNAL(clicked()), this,
            SLOT(callStop()));
    connect(ui.pushButtonRunOnce, SIGNAL(clicked()) , this,
            SIGNAL(onRunOnce()));
    connect(ui.pushButtonRun, SIGNAL(clicked()), this,
            SLOT(disableAll()));
    connect(ui.pushButtonStop, SIGNAL(clicked()), this,
            SLOT(disableAll()));
    connect(ui.pushButtonRunOnce, SIGNAL(clicked()) , this,
            SLOT(disableAll()));
}

WInspect::~WInspect()
{}

void WInspect::disableAll(){
    ui.pushButtonRun->setEnabled(false);
    ui.pushButtonStop->setEnabled(false);
    ui.pushButtonRunOnce->setEnabled(false);
}
```

```

}

void WInspect::callRun(){
    emit onRun();
}

void WInspect::showRunButton(){
    ui.pushButtonRun->setEnabled(true);
    ui.pushButtonStop->setEnabled(false);
    ui.pushButtonRunOnce->setEnabled(true);
}

void WInspect::showStopButton(){
    ui.pushButtonRun->setEnabled(false);
    ui.pushButtonStop->setEnabled(true);
    ui.pushButtonRunOnce->setEnabled(false);
}

void WInspect::callStop(){
    emit onStop();
}

```

**WInspect.h** (conté les capçaleres de les funcions i les variables globals de la classe):

```

#ifndef WINSPECT_H
#define WINSPECT_H

#include <QtGui/QWidget>
#include "ui_WInspect.h"
#include <src/Controller/EUserRoles.h>

class WInspect : public QWidget
{
    Q_OBJECT
public:
    WInspect(EUserRoles::EUserRoles userRole = EUserRoles::NONE,
            QWidget *parent = 0);
    ~WInspect();
signals:
    void onRun();
    void onRunOnce();
    void onStop();
public slots:
    void showRunButton();
    void showStopButton();
    void disableAll();
private slots:
    void callRun();
    void callStop();
private:
    Ui::WInspectClass ui;
    EUserRoles::EUserRoles m_eUserRole;
};
#endif // WINSPECT_H

```

**WInspect.ui** (Conté el disseny de la interfície gràfica del widget).

\*Es mostra una captura de pantalla de pantalla del Widget, i el codi XML necessari per a crear-lo (generat automàticament a través del QtCreator).



```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>WInspectClass</class>
  <widget class="QWidget" name="WInspectClass">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>182</width>
        <height>70</height>
      </rect>
    </property>
    <property name="sizePolicy">
      <sizepolicy hstretch="Expanding" vsizetype="Expanding">
        <horstretch>0</horstretch>
        <verstretch>0</verstretch>
      </sizepolicy>
    </property>
    <property name="windowTitle">
      <string>WInspect</string>
    </property>
    <layout class="QGridLayout" name="gridLayout_3">
      <item row="0" column="0">
        <layout class="QGridLayout" name="gridLayout_2">
          <item row="0" column="0">
            <widget class="QPushButton" name="pushButtonRun">
              <property name="sizePolicy">
                <sizepolicy hstretch="Maximum" vsizetype="Fixed">
                  <horstretch>0</horstretch>
                  <verstretch>0</verstretch>
                </sizepolicy>
              </property>
              <property name="minimumSize">
                <size>
                  <width>50</width>
                  <height>50</height>
                </size>
              </property>
              <property name="maximumSize">
                <size>
                  <width>50</width>
                  <height>50</height>
                </size>
              </property>
              <property name="toolTip">
                <string>Start inspection</string>
              </property>
            </widget>
          </item>
        </layout>
      </item>
    </layout>
  </widget>
</ui>
```

```

</property>
<property name="text">
  <string/>
</property>
<property name="icon">
  <iconset resource="../../resources/resourcesE2M.qrc">
    <normaloff>:/icons/control_play_big.png</normaloff>
    :/icons/control_play_big.png
  </iconset>
</property>
<property name="iconSize">
  <size>
    <width>32</width>
    <height>32</height>
  </size>
</property>
</widget>
</item>
<item row="0" column="1">
  <widget class="QPushButton" name="pushButtonStop">
    <property name="sizePolicy">
      <sizepolicy hstretch="Maximum" vsizetype="Fixed">
        <horstretch>0</horstretch>
        <verstretch>0</verstretch>
      </sizepolicy>
    </property>
    <property name="minimumSize">
      <size>
        <width>50</width>
        <height>50</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>50</width>
        <height>50</height>
      </size>
    </property>
    <property name="toolTip">
      <string>Stop inspection</string>
    </property>
    <property name="text">
      <string/>
    </property>
    <property name="icon">
      <iconset resource="../../resources/resourcesE2M.qrc">
        <normaloff>:/icons/control_stop_big.png</normaloff>
        :/icons/control_stop_big.png
      </iconset>
    </property>
    <property name="iconSize">
      <size>
        <width>32</width>
        <height>32</height>
      </size>
    </property>
  </widget>
</item>
<item row="0" column="2">
  <widget class="QPushButton" name="pushButtonRunOnce">

```

```
<property name="sizePolicy">
  <sizepolicy hstretch="Maximum" vsizetype="Fixed">
    <horstretch>0</horstretch>
    <verstretch>0</verstretch>
  </sizepolicy>
</property>
<property name="minimumSize">
  <size>
    <width>50</width>
    <height>50</height>
  </size>
</property>
<property name="maximumSize">
  <size>
    <width>50</width>
    <height>50</height>
  </size>
</property>
<property name="toolTip">
  <string>Single test</string>
</property>
<property name="text">
  <string/>
</property>
<property name="icon">
  <iconset resource="../../resources/resourcesE2M.qrc">
    <normaloff>:/icons/control_end_big.png</normaloff>
    :/icons/control_end_big.png
  </iconset>
</property>
<property name="iconSize">
  <size>
    <width>32</width>
    <height>32</height>
  </size>
</property>
</widget>
</item>
</layout>
</item>
</layout>
</widget>
<layoutdefault spacing="6" margin="11"/>
<resources>
  <include location="../../resources/resourcesE2M.qrc"/>
</resources>
<connections/>
</ui>
```

## 6.5 Persistència de les dades

Aquest apartat, encara que parli dels elements de la capa de dades de l'aplicació, s'ha introduït en aquest punt ja que per implementar algunes funcionalitats de la interfície gràfica, s'ha d'accedir directament a la capa de dades. Així es pot tenir una idea bàsica dels elements que formen part d'aquesta capa i quins d'ells són necessaris per a la interfície gràfica.

### 6.5.1 Configuració

La configuració global del sistema es guarda en un arxiu XML que es llegeix en obrir l'aplicació, i que conté totes les definicions necessàries per mantenir l'estat del programa quan es tanca. D'aquesta manera no es perd la configuració d'una execució anterior de l'aplicació, ja que cada vegada que el programa es tanca, es torna a guardar aquest arxiu amb la nova configuració.

Quan s'inicia l'aplicació, tota la informació del fitxer de configuració es bolca a unes estructures de dades que seran amb les que es treballarà durant tota l'execució. Alguns dels elements que guarden informació de l'arxiu XML en estructures de dades són els tractaments, els formats, les càmeres, els paràmetres de la base de dades, etc.

És interessant conèixer els elements que formen part d'aquest fitxer, ja que l'estructuració i el contingut de la interfície gràfica en depenen.

A continuació es mostra de manera esquemàtica el contingut d'aquest fitxer de configuració, explicant què hi podem trobar a cada punt:

```
< Configuració del VisioInspect2 >
  < Versió del Software >
    Informació sobre la versió del software.
  < Configuració dels modes d'adquisició >
    Configuració dels plugins de càmeres.
  < Configuració dels formats >
    Configuració dels formats, processos, tractaments i paràmetres.
  < Configuració de la interfície gràfica >
    Configuració dels widgets de la pantalla principal.
  < Configuració de la Base de dades >
    Configuració dels paràmetres de connexió a la base de dades
  < Configuració dels elements d'entrada / sortida >
    Configuració dels diferents dispositius d'entrada / sortida del sistema
  < Configuració dels plugins >
    Rutes del diferents plugins de tractaments
  < Configuració de les opcions >
    Especificació d'algunes opcions d'inspecció de l'aplicació
  < Configuració de l'emmagatzematge de les imatges >
    Configuració de l'opció d'emmagatzematge d'imatges resultants de les inspeccions
</Configuració del VisioInspect2>
```

El següent fragment de codi serveix com a exemple de la definició d'un Widget de la interfície gràfica a l'arxiu XML. Concretament es mostra el Widget que conté el gràfic de barres dels resultats:

```
<UiMainWindowWidget spanRow="3" row="0" spanCol="4"
  type="WResultsStatistics" col="2" name="ResultsHist">

  <UiWidgetParam value="IMG_ERROR_CLOSING_AREA" type="Result"
    name="Closing Area"/>
  <UiWidgetParam value="SHAPE_INTEGRITY" type="Result"
    name="Shape"/>
  <UiWidgetParam value="BURGER_INTEGRITY" type="Result"
    name="Burger Full"/>
  <UiWidgetParam value="BURGER_ORIENTATION" type="Result"
    name="Burger Angle"/>
  <UiWidgetParam value="STRANGE_OBJECT_IMAGE" type="Result"
    name="Strange Object"/>
  <UiWidgetParam value="PAPER_ORIENTATION" type="Result"
    name="Paper Angle"/>
  <UiWidgetParam value="TOTAL" type="Result" name="TOTAL"/>

</UiMainWindowWidget>
```

*Codi XML que defineix el Widget "WResultsStatistics"*

En aquest codi hi podem observar com l'element amb nom "ResultsHist" està representat per un widget de tipus "WResultsStatistics", que és un gràfica de barres. Es pot observar la captura de pantalla d'aquest widget a l'apartat de les imatges de la interfície gràfica d'aquesta memòria, dedicat als widgets de la pantalla principal [Apartat 7.1].

Aquest està situat a la fila 0 (*row="0"*), columna 2 (*col="2"*) i recull els valors dels resultats:

- IMG\_ERROR\_CLOSING\_AREA → representat a la barra amb títol "Closing Area".
- SHAPE\_INTEGRITY → representat a la barra amb títol "Shape".
- BURGER\_INTEGRITY → representat a la barra amb títol "Burger Full".
- BURGER\_ORIENTATION → representat a la barra amb títol "Burger Angle".
- STRANGE\_OBJECT\_IMAGE → representat a la barra amb títol "Strange Object".
- PAPER\_ORIENTATION → representat a la barra amb títol "Paper Angle".
- TOTAL → representat a la barra amb títol "TOTAL".



---

### 6.5.2 SGBD

---

L'aplicació necessita una base de dades per tal de poder tenir una taula amb la relació dels usuaris amb permisos per accedir-hi. En aquesta base de dades hi trobem únicament una taula on hi ha definit, per a cada tipus d'usuari, el seu identificador, el seu nom, la seva contrasenya i el rol que té al sistema.

El disseny de la interfície gràfica dependrà totalment del rol de l'usuari que accedeixi al programa, ja que depenent del tipus d'usuari, s'habilitaran o s'inhabilitaran algunes opcions.

També és necessari accedir a la base de dades des de la interfície gràfica per obtenir la llista d'usuaris, quan es vol mostrar el selector d'usuaris (a l'iniciar l'aplicació o per canviar d'usuari).

No s'ha volgut guardar res més a la base de dades perquè el programa no ho requereix, però també perquè si s'hi volguéssim guardar més coses com ara resultats, informació de les anàlisis, etc. es perdria molt de temps en accedir-hi, i això ralentitzaria l'execució.

A l'hora d'escollir el sistema gestor de la base de dades, no se'n requeria un de gaire complet ni amb moltes funcions, però sí que fos fiable i robust. Per tots aquests motius es va decidir utilitzar el MySQL, que ofereix un servidor i un client SQL que s'adequa a les nostres necessitats.

També s'han utilitzat les classes *QSqlDatabase*, *QSqlQuery* i *QSqlError* que ens ofereix el Qt per tal de poder establir la connexió i realitzar les consultes i modificacions des del codi C++ cap a la base de dades.

---

### 6.5.3 Estructures de dades

---

Com s'ha explicat a l'apartat anterior, a la base de dades no es guarda cap informació dels resultats dels tractaments de les anàlisis, però en canvi s'han d'emmagatzemar a algun lloc per poder-hi accedir des dels diferents components del sistema.

S'han creat estructures de dades per poder guardar informació de l'última imatge resultant de cada tractament i els últims resultats dels tractaments.

A part d'això també es guarden imatges auxiliars necessàries per realitzar les anàlisis.

La informació que es guarda a les estructures de dades (excepte la referent a la configuració), és només la que correspon a l'última anàlisi, ja que si es guardés més volum d'informació, al llarg del temps s'acabaria exhaurint la memòria del computador on està instal·lat el sistema.



## 7. INTERFÍCIE GRÀFICA DESENVOLUPADA

7.1 Widgets de la Pantalla Principal	141
7.2 Diàlegs de configuració dels menús	145
7.3 Menús	150
7.4 Interfície gràfica global	151

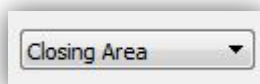
Interfície Gràfica per a software de control  
d'inspecció en temps real

**Interfície gràfica desenvolupada**

## 7.1 Widgets de la Pantalla principal

En aquest apartat es mostren captures de pantalla dels diferents widgets que es poden mostrar a la pantalla principal, i que es troben explicats a l'apartat 5.5 d'aquesta memòria. Es mostra cada Widget individualment (amb el seu nom al peu), per tal de poder veure clarament els seus components:

**Selector d'imatges resultants a visualitzar des del widget WImage:**



*WScreenResultsSelector*

**Buttons amb accessos directes a alguns diàlegs de configuració dels tractaments:**



*WButtonGroup*

**Logo de l'empresa, amb el link a la pàgina web:**

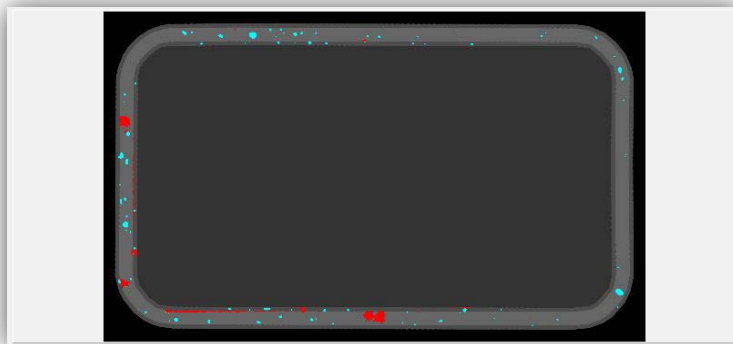


*WLogo*

**Widget per a visualitzar les imatges:**



*WImage: Visualitzant la imatge original*



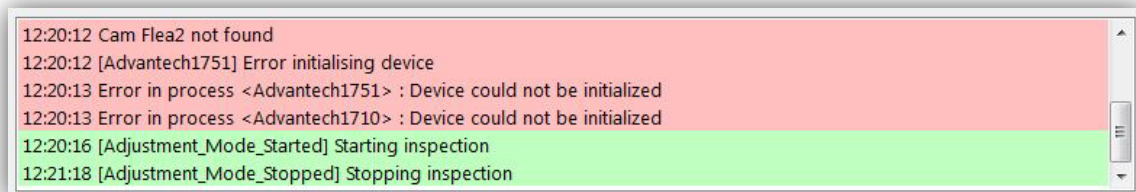
*WImage: Visualitzant la imatge resultant del tractament que analitza la zona de tancament de la barqueta*

**Mostrar els resultats dels últims tractaments:**



*WImageResults: Visualitzant en vermell els tractaments que han retornat error i en verd els correctes.*

**Mostrar els events, canvis d'estat i errors del sistema:**



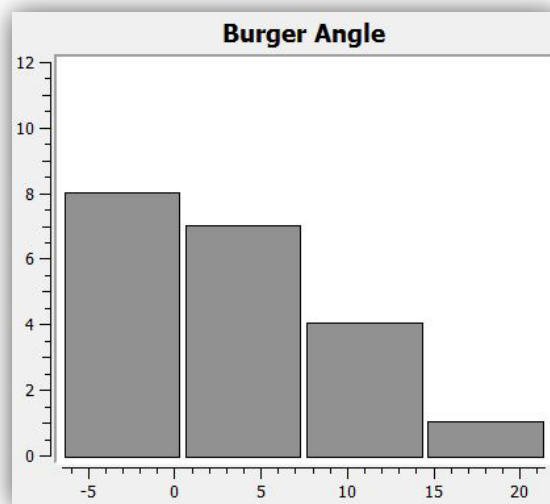
*WLog: Visualitzant en vermell els errors i en verd els canvis d'estat i els events.*

**Gràfica de resultats:**



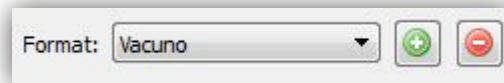
*WResultsStatistics: Visualitzan en verd els resultats correctes i en vermell els incorrectes, per a cada tractament.*

**Visualització gràfica de la progressió d'un sol resultat mitjançant un histograma:**



*WHistogram: Visualitzant el resultat "Burger Angle"*

**Selector de format:**



*WFormatSelector: Visualitzant el format actual de "Vacuno"*

**Control de les inspeccions:**



*WInspect: Visualitzant l'estat en que les inspeccions estan aturades.*

**WOkBad:**



*WOkBad: Visualitzant la imatge que representa que la última inspecció ha sigut errònia, i la que ha sigut correcte*

**Comptador de resultats:**



*WCounter*



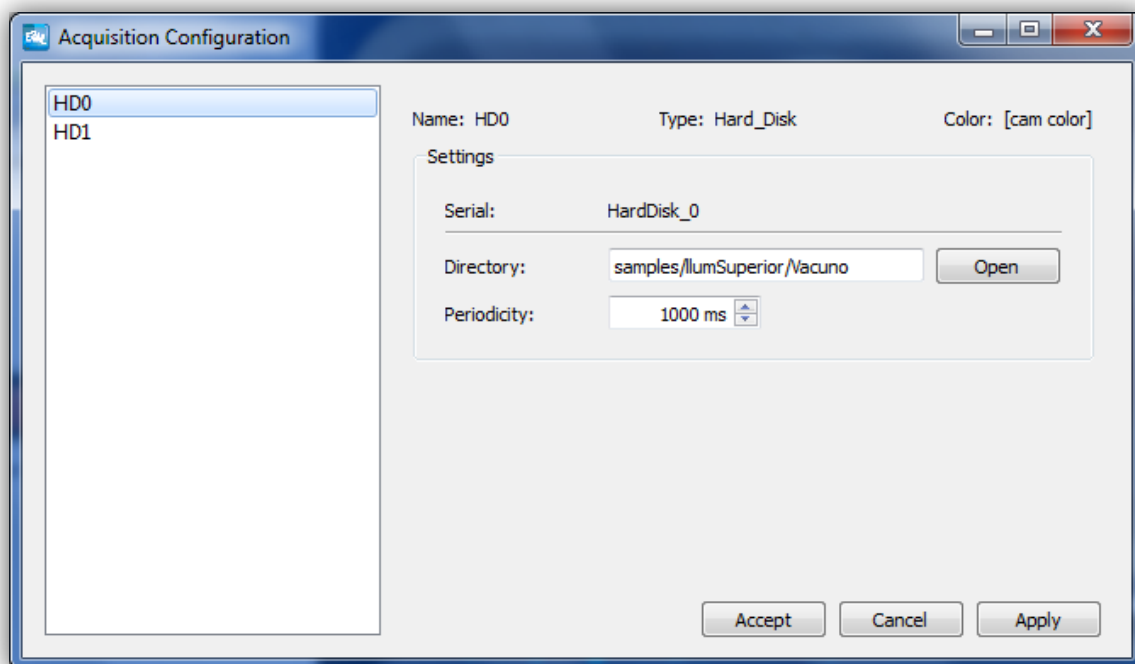
## 7.2 Diàlegs de configuració dels menús

En aquest apartat es mostren captures dels diferents diàlegs als que podem accedir mitjançant el menú de la pantalla principal, i que es troben explicats a l'apartat 5.4 d'aquesta memòria.

Hi ha alguna opció d'aquests menús que no necessita un diàleg de configuració complex, sinó que utilitza diàlegs senzills (com per exemple el d'afegir un usuari) o estàndards de Windows (com per exemple el d'obrir un arxiu de configuració). Aquests tipus de diàlegs no es mostren en aquestes captures.

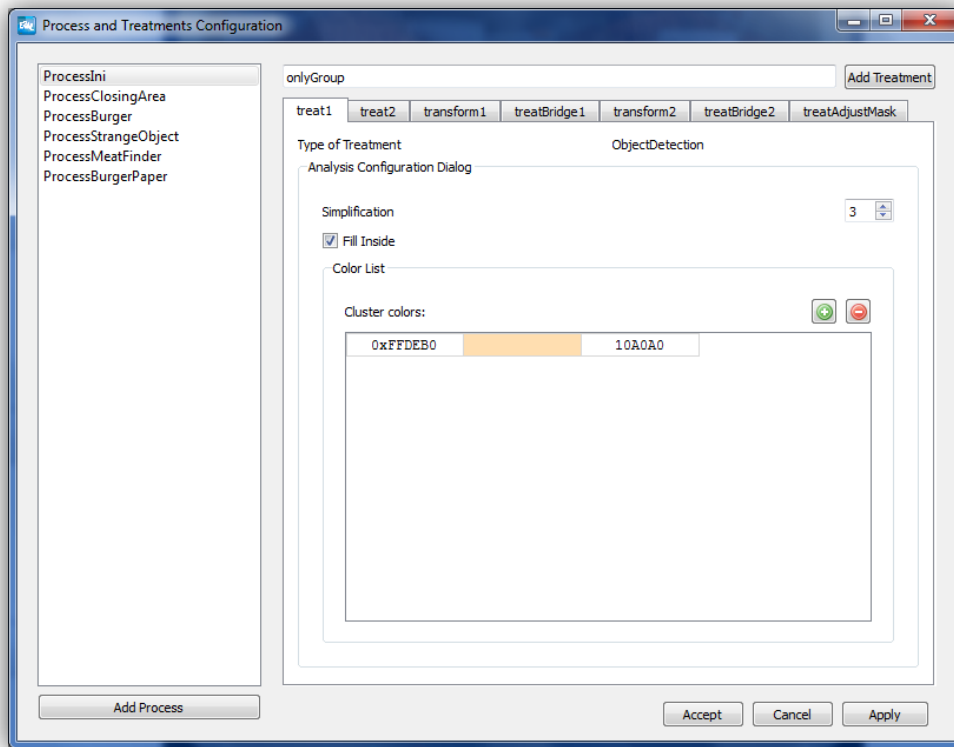
Pel que fa als altres casos, es mostra cada diàleg individualment, per tal de poder veure clarament els seus components (amb el seu nom al peu):

### Configuració del mode d'adquisició:



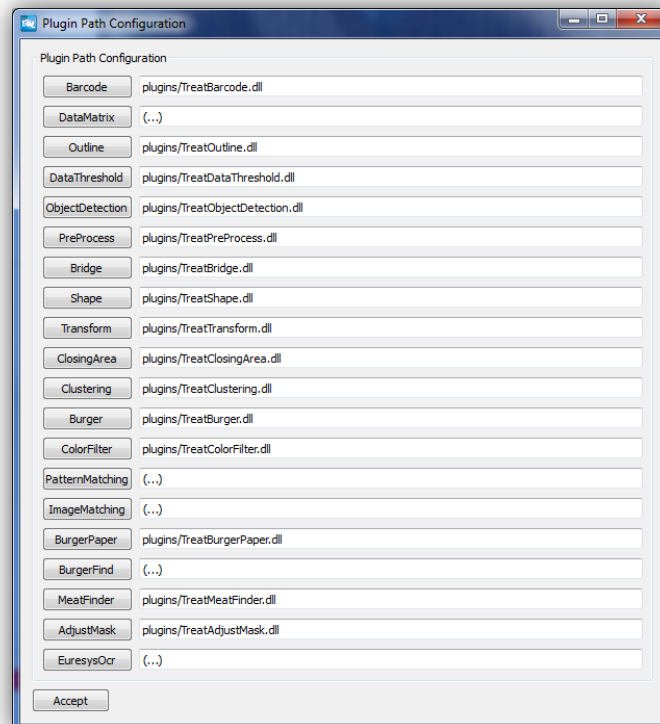
*WAcquireDialog: Visualitzant les característiques del mode d'adquisició amb nom "HD0".*

**Configuració de les anàlisis:**



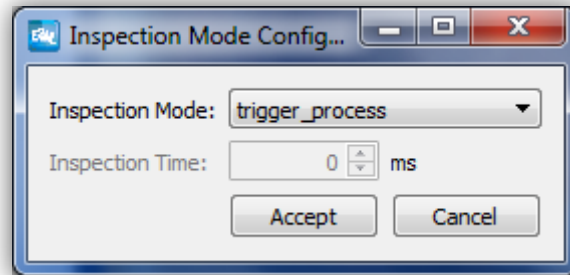
*WProcessManagerDialog: Visualitzant els detalls i tractaments del procés "ProcessIni"*

**Configuració de les rutes dels plugins:**



*WPluginPath*

Configuració del mode d'inspecció:



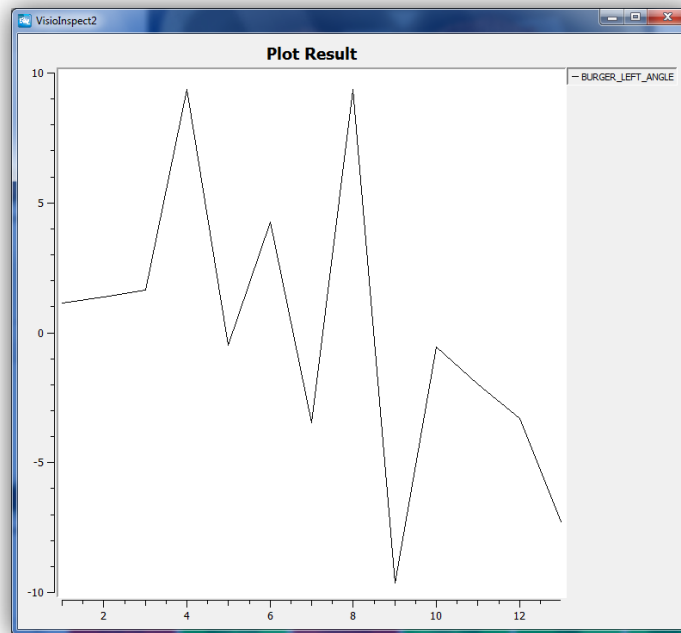
*WInspectionModeDialog*

Mostrar Resultats

	6	7	8	9	10	11	12	13
SHAPE_LOCAL_ERROR	1 31) D(1.76756)	D(0.788433)	D(1.74728)	D(1.09867)	D(1.8215)	D(1.37229)	D(2.10959)	
SHAPE_MEAN_ERROR	1 64) D(0.850109)	D(0.299953)	D(0.790843)	D(0.477783)	D(0.864952)	D(0.549797)	D(0.814107)	
BURGER_RIGHT_ANGLE	1 33) D(18.8655)	D(2.03504)	D(3.06221)	D(1.57643)	D(11.9292)	D(-1.03443)	D(-3.74231)	
BURGER_LEFT_ANGLE	1 2) D(8.37216)	D(-9.65065)	D(-0.533359)	D(-1.98717)	D(-3.30504)	D(-7.28428)	D(8.3313)	
BURGER_RIGHT_DAMAGE	1 53) D(252.443)	D(253.736)	D(255)	D(222.711)	D(255)	D(215.201)	D(212.529)	
BURGER_LEFT_DAMAGE	1 54) D(254.064)	D(255)	D(254.519)	D(250.933)	D(255)	D(234.511)	D(241.587)	
BURGER_RIGHT_DAMAGE_DESV	1 77) D(25.4088)	D(17.9094)	D(2.6974e-06)	D(84.8)	D(2.6974e-06)	D(92.5461)	D(95.0071)	
BURGER_LEFT_DAMAGE_DESV	1 99) D(15.4185)	D(2.6974e-06)	D(11.0685)	D(31.9449)	D(2.6974e-06)	D(69.3171)	D(56.9236)	

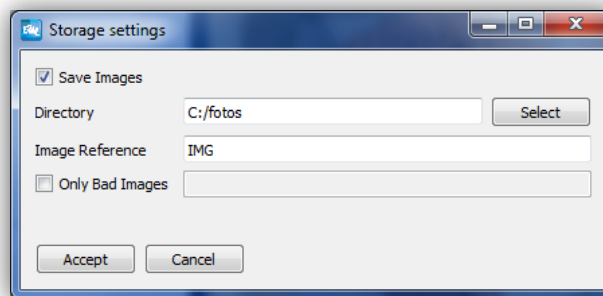
*WResultsObserver: Visualitzant en cada fila un resultat diferent. Cada columna representa una inspecció diferent*

**Mostrejar un sol resultat:**



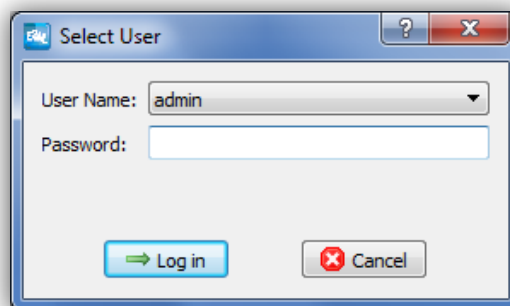
*WPlot: Visualitzant la progressió del resultat "BURGUER-LEFT-ANGLE"*

**Opcions d'emmagatzematge de les imatges resultants:**



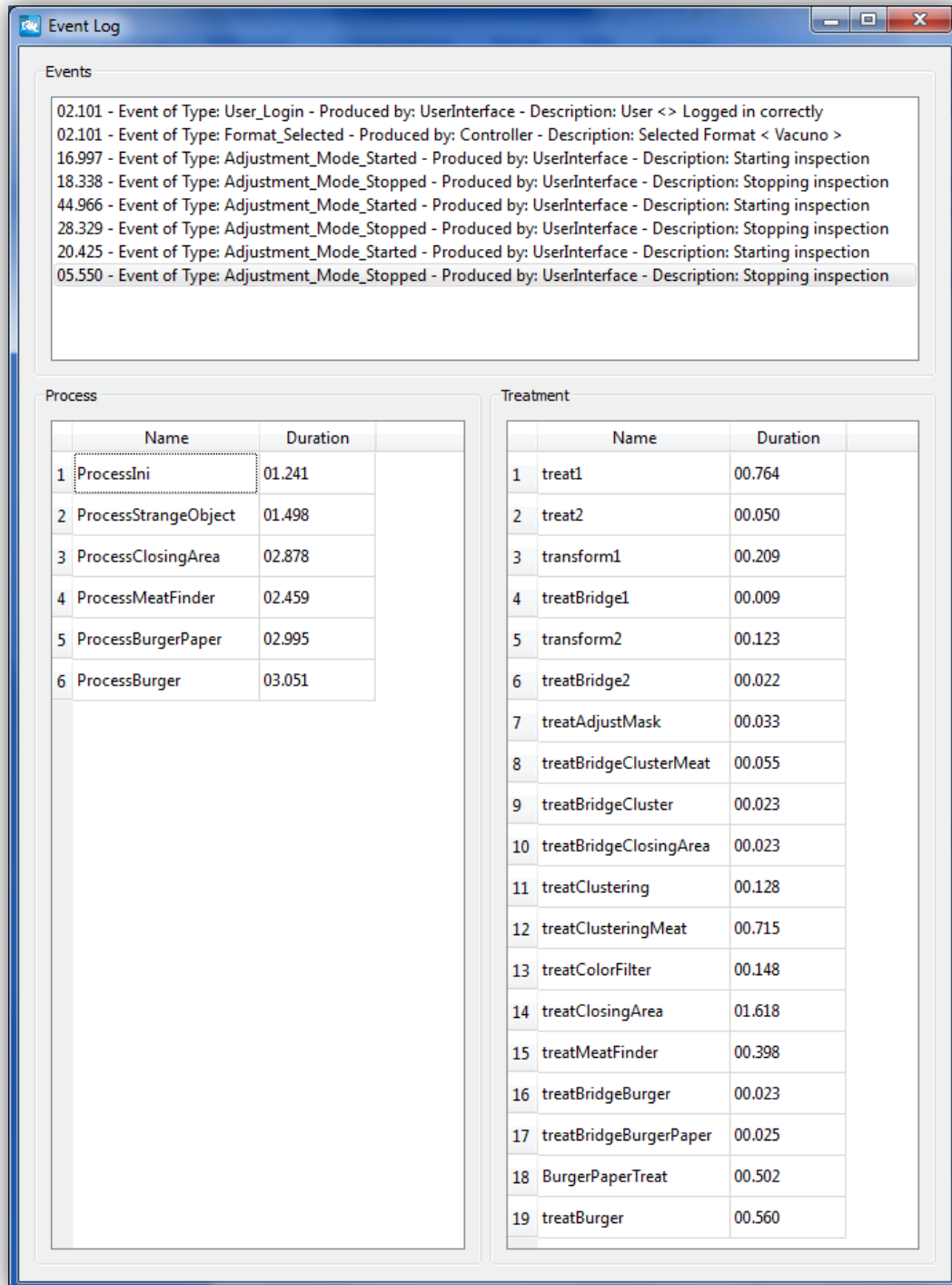
*WStorageDialog*

**Canviar d'usuari:**



*WUserSelector*

Mostrar eLL og d'events i temps de les anàlisis:

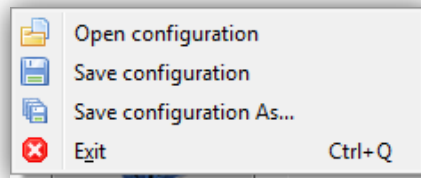


WEventLog: Visualitzant a la part superior el log del sistema, i a sota els temps dels diferents processos i tractaments.

## 7.3 Menús

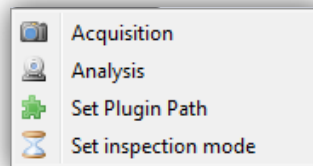
En aquest apartat es mostren els diferents menús que es despleguen de la barra de menús de la pantalla principal. Aquestes vistes ens permetran col·locar cada un dels diàlegs descrits a l'apartat anterior.

### Menú 1:



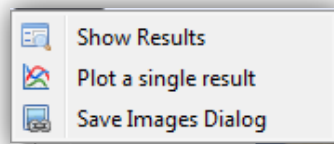
*Menú Arxiu*

### Menú 2:



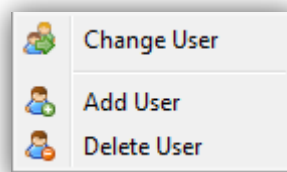
*Menú de configuració*

### Menú 3:



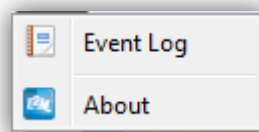
*Menú dels resultats*

### Menú 4:



*Menú dels usuaris*

### Menú 5:



*Menú de l'ajuda i informació*

## 7.4 Interfície gràfica Global

Aquesta captura de pantalla correspon a la interfície gràfica desenvolupada, on es poden veure tots els widgets i la barra de menús a la part superior. L'estructuració que s'ha seguit per a distribuir els widgets és la que es marca des de la direcció del projecte, i que ja ha estat mostrada a l'apartat 5.5 d'aquesta memòria.



*WMainWindow: Visualització de la pantalla principal de la interfície gràfica d'exemple requerida per l'empresa "Martinez Loriente S.A.", per a l'anàlisi i inspecció de barquetes amb hamburgueses.*

Interfície Gràfica per a software de control  
d'inspecció en temps real

**Interfície gràfica desenvolupada**



## 8. REVISIÓ DE LA PLANIFICACIÓ

8.1 Revisió de l'estudi temporal	155
8.2 Revisió de l'estudi econòmic	157



## 8.1 Revisió de l'estudi temporal

Un cop arribats al final del projecte, només manca realitzar una valoració sobre l'estudi temporal que es va fer al principi, per així poder observar i aprendre a valorar amb major exactitud en projectes futurs.

A continuació es mostra una taula que recull les diferents tasques descrites en la planificació, el temps que es va creure en un principi, el temps real destinat a la seva elaboració, i la desviació (variació) respecte la planificació inicial.

Tasca	Estimació inicial (Dies)	Temps Real (Dies)	Variació (Dies)
Estudi i definició del projecte	5	5	0
Anàlisi de requeriments	5	5	0
Especificació	10	12	+2
Disseny	20	18	-2
Implementació	100	110	+10
Proves	100	108	+8
Posada en funcionament	5	10	+5
Redacció de la memòria	30	37	+7
Documents de final d'etapa	0	0	0
<b>TOTAL</b>	<b>175</b> <b>(700 hores)</b>	<b>190</b> <b>(760 hores)</b>	<b>+15</b> <b>(80 hores)</b>

*Taula 1: Variacions temporals de les tasques del projecte  
(tenint en compte les tasques que es realitzen paral·lelament).*

Com es pot observar, l'estimació duta a terme difereix del temps real treballat. Això ha fet que el projecte finalitzés el dia 30 de desembre de 2009 en comptes del dia 4 de desembre de 2009, previst inicialment. Per a justificar aquesta desviació s'ha fet un petit estudi de cada una d'aquestes variacions per la tasca que li pertoca:

- Desviació de 2 dies en l'*Especificació*:
  - o Aquests dos dies de més són deguts a que a l'hora de començar a definir les diferents classes de la interfície gràfica ens vam donar compte que n'hi havia més de les previstes, sobretot pel que fa a les classes que defineixen tipus de missatges.

El fet d'intentar definir un patró a seguir per especificar cada classe (noms de les funcions, funcions comunes, variables globals de cada classe...) que representés un widget també va costar més del previst.

- Reducció de 2 dies en el *Disseny*:
  - o El fet de determinar una política de missatges molt clara i semblant per totes les accions que es poden dur a terme, ens va permetre reaprofitar molts diagrames de seqüència per a realitzar diferents accions que compartien el mateix esquema d'execució.

Al final vàrem haver de definir menys diagrames dels que ens havíem pensat en un principi, ja que moltes funcionalitats podien compartir el mateix. Això ens va permetre recuperar aquests 2 dies que s'havien perdut en l'especificació.

- Desviació de 10 dies en la *implementació*:
  - o Aquesta tasca, degut a la seva llargada i desconeixement de les dificultats en que ens podríem trobar, era molt difícil de preveure al realitzar la planificació inicial del projecte.

Inicialment es van predir que es necessitarien uns 100 dies per a dur-la a terme i al final n'han calgut 10 més. Les raons principals d'aquesta són els diferents riscos que ja van ser comentats en un apartat anterior [Apartat 1.6.3]. Durant l'execució d'aquesta etapa ens hem trobat amb diferents riscos d'aquests que han fet variar la planificació inicial.

- Desviació de 8 dies en les *proves*:
  - o Com que aquesta tasca es realitza paral·lelament amb la d'implementació, s'ha retardat 10 dies pel mateix motiu que aquesta.

Al final s'han hagut de fer les proves globals de funcionament només en 3 dies en comptes dels 5 que es tenien previstos. Per això la desviació ha estat de 8 dies i no de 10 com en la implementació, ja que ens hem hagut d'estalviar 2 dies de proves per tal de poder portar el sistema a l'entorn de producció quan el client ho ha requerit.

- Desviació de 5 dies en la *posada en funcionament*:
  - o Els terminis d'aquest etapa han variat degut a les exigències del client, ja que es va haver d'anar a provar el sistema a la línia de producció 2 dies abans del previst.

A part també s'han dedicat 3 dies més a recollir imatges des de l'entorn de producció, i a resoldre diferents problemes d'enfocament de les càmeres, configuració del sistema i adaptar el software a les necessitats del client.

- Desviació de 7 dies en la *redacció de la memòria*:
  - o Al principi es van definir 30 dies per redactar la memòria. Al final però, n'han calgut uns quants més degut sobretot al fet d'haver de passar a net i modificar alguns dels diagrames UML, per tal d'introduir-los al document d'una manera més llegible i formal.

Els últims dies de redacció d'aquesta memòria han estat una mica més intensius, dedicant més de 4 hores al dia per tal de poder acabar de confeccionar-la. Per això la variació de dies no coincideix amb el número d'hores dedicades, ja que alguns dels dies finals s'ha treballat més de 4 hores.

## 8.2 Revisió de l'estudi econòmic

A l'hora de realitzar la revisió econòmica del projecte ens vàrem donar compte que hi havia alguns canvis respecte la planificació inicial. Aquesta variació no suposa una diferència molt significativa en el cost total del projecte, i des de l'empresa s'ha considerat acceptable i raonable després de les diferents incidències que hi ha hagut durant el transcurs del projecte.

A continuació es mostra una taula explicativa dels costos previstos en la planificació inicial, els costos finals, i la seva variació.

Tipus de cost	Estimació inicial	Cost Final	Variació
Hardware	308 €	308 €	0 €
Software	285 €	210 €	-75 €
Recursos Humans	16.750 €	19.130 €	+ 2.380 €
<b>TOTAL</b>	<b>17.343 €</b>	<b>19.648 €</b>	<b>+2.305 €</b>

Taula 2: Variacions dels diferents tipus de costos del projecte.

Com es pot observar, l'estimació duta a terme a l'inici difereix en els costos de software del projecte i de recursos humans, deixant el cost de hardware tal i com es va preveure en un principi. Igualment a continuació es justifiquen cada un d'aquests costos finals juntament amb els seves variacions:

- Manteniment del cost previst en el *hardware*:
  - o Al principi del projecte es va realitzar una aproximació del cost del hardware basada en l'aprofitament que es tindria de cada equip. Aquesta planificació es va fer utilitzant un raonament lògic, sense cap estudi en profunditat (ja que no es va creure oportú).

El fet de no haver necessitat cap equip extra ni haver hagut de canviar-ne cap, ha fet que aquest cost no variés en el transcurs del projecte.

- Reducció de 75 € del cost de *software*:
  - o Aquests 75 € són els corresponents a la llicència que es creia que s'havia de demanar del *Microsoft Office 2007*. Al final s'ha fet ús d'una llicència d'estudiant per al mateix producte, ja que només s'ha instal·lat al computador on es treballava, no es va posar en l'equip de prova ni en el servidor.

S'ha fet servir algun altre software durant el projecte, però sempre s'ha buscat que fos de lliure distribució, que no comportés un augment del cost.

- Augment de 2.580 € del cost de *recursos humans*:
  - o Aquest augment de cost es deu al fet que s'han necessitat més hores de les previstes a l'inici, per a la realització del projecte. Això ha fet que el cost de cada recurs implicat en alguna de les tasques que s'ha modificat, variés respecte a l'inicial.

Els recursos que han variat el seu cost són:

- Client:
  - Ha invertit unes 4 hores més de les previstes a l'inici.
  - El seu cost no el tenim en compte en aquesta anàlisi, ja que no s'ha d'incloure en el projecte.
- Director del projecte:
  - Ha invertit 8 hores més en la tasca de "Posada en funcionament", de les previstes inicialment.
  - Això ha provocat un augment de 280 € en el cost d'aquest recurs.
- Analista programador:
  - Ha invertit 60 hores més entre la tasca d' "Implementació" i la de "Posada en funcionament", de les previstes inicialment.
  - Això ha provocat un augment de 1.500 € en el cost d'aquest recurs.
- Programador
  - Ha invertit 80 hores més en la tasca de "Implementació" i la de "Redacció de la memòria", de les previstes inicialment.
  - Això ha provocat un augment de 600 € en el cost d'aquest recurs. No es comptabilitzen totes les hores de més que ha treballat, ja que algunes d'aquestes hores han estat fora de l'horari laboral, per tal de poder acabar la redacció de la memòria.

## 9. TREBALL FUTUR

9.1 Immediat	161
9.2 A llarg termini	161





## 9.1 Immediat

Hi ha diverses funcionalitats del sistema que no han estat incloses dins dels objectius d'aquest projecte, ja que els terminis que es van marcar des d'un principi, no ho permetien. Per això en aquest apartat es mostren les funcionalitats que es podrien considerar necessàries en a un futur immediat:

- Permetre visualitzar més d'una imatge a la pantalla principal, permetent així poder veure la imatge original al mateix temps que la imatge resultant d'alguna de les anàlisis.
- Millorar la usabilitat d'algunes de les funcions implementades, fent-les més vistoses i fàcils de manipular.
- Facilitar nous menús i diàlegs per poder configurar més aspectes de la configuració de les màquines.
- Millorar l'estructuració de la pantalla principal, estructurant de manera més lògica i visualment agradable els diferents widgets.
- Un "wizard" de configuració inicial del sistema, on s'hi pot afegir el número de càmeres, l'assignació de cada càmera a un procés i l'assignació dels tractaments a cada procés.

## 9.2 A llarg termini

A part de les futures funcionalitats que s'haurien de tractar en un període no molt llunyà, també n'hi ha d'altres que, ja sigui degut a la seva dificultat de desenvolupament o a la seva poca importància actualment, no resulten ser uns objectius molt immediats.

Malgrat això val la pena tenir-los en compte per quan hi hagi temps i recursos suficients per a dur-los a terme, ja que suposarien millores importants al sistema:

- Oferir la possibilitat de poder configurar la interfície gràfica directament des d'algun diàleg d'algun menú, sense haver-ho de fer modificant directament l'arxiu de configuració XML (com es fa actualment).
- Permetre la possibilitat de tenir una interfície gràfica diferent per a cada un dels formats. Així quan es canviés de format, es canviaria també la interfície gràfica, sense necessitat de reiniciar l'aplicació.
- Validació de la configuració de la interfície gràfica, perquè no hi hagi widgets sobreposats i estiguin correctament distribuïts.



## 10. CONCLUSIONS

10.1	Punt de vista professional	165
10.2	Punt de vista del departament de I+D	165
10.3	Punt de vista personal	165



## 10.1 Punt de vista professional

Després d'haver desenvolupat el projecte i d'haver realitzat totes les proves, tot l'equip de l'empresa E2M confia que aquest nou sistema millori les seves funcionalitats i capacitats empresarials, així com la seva imatge corporativa. Aportarà un valor afegit al servei que s'ofereix als clients. Amb el temps es podrà quantificar el benefici i seguir millorant a partir de l'experiència obtinguda.

Per a l'empresa E2M aquest projecte representa un salt qualitatiu en els seus serveis. Forma part d'una sèrie d'inversions en tecnologia, destinades a créixer, adaptar-se al mercat, a les noves necessitats, a la situació econòmica actual i a ser més competitiu davant la competència.

A través del projecte **VisioInspect2**, Estudis Electro-Mecànics ha establert les bases de la seva prosperitat tecnològica a mitjà termini, rejuvenint el software per tal de poder córrer en els sistemes operatius actuals i en màquines més potents (QuadCore per exemple).

A més, amb la separació de les diferents capes, s'ha desenvolupat un codi que permet la refactorització, podent incorporar canvis futurs d'una forma senzilla i ràpida.

## 10.2 Punt de vista del departament de I+D

Aquest projecte ens ha servit de model per a futurs projectes, ja que la metodologia de treball utilitzada ha estat satisfactòria.

El projecte ha servit per millorar els coneixements i mètodes de treball del departament a partir de l'experiència obtinguda, de la documentació que s'ha creat de cadascun dels processos del projecte, de la manera de realitzar la presa de requeriments, i de la planificació de tot el desenvolupament.

La idea és anar millorant tots aquests procediments a partir de les experiències. Com a primer pas, el projecte ens ha estat de gran ajuda, ja que era un projecte complet i que s'havia de realitzar de principi a fi.

## 10.3 Punt de vista personal

Des del punt de vista personal, el projecte m'ha aportat molts valors i coneixements que des de la universitat es tracten, però no es poden experimentar en profunditat. He conegut noves eines que segurament em seran de gran utilitat en el futur, així com noves formes de treballar en un projecte de desenvolupament de software.

El fet de desenvolupar el projecte juntament amb un grup de persones, m'ha permès aprendre noves metodologies de treball que encara no havia experimentat.

A part d'aquestes vivències en el camp de l'enginyeria del software també he anat recopilant coneixements en altres tipus de camps, com podria ser el d'administració de sistemes i el de sistemes electrònics. Això segurament ho he d'agrair a les diferents feines paral·leles que s'han anat realitzant dins del departament mentre es desenvolupava el projecte, i que m'han servit per ampliar aquests coneixements.

En definitiva, cal remarcar que aquesta ha estat una bona experiència que, a part de servir-me per a poder presentar el projecte final de carrera, m'ha servit per a introduir-me una mica més en el món laboral, adquirint nous coneixements i experiències.

## 11. BIBLIOGRAFIA

11.1	Libres	167
11.2	Referències Web	169





## 11.1 Llibres

- [SOF01] Software Engineering: A Practitioner's Approach. Roger S Pressman. McGraw-Hill.
- [PHAR02] Handbook of Computer and Computerized System Validation for the Pharmaceutical Industry. Stephen Robert Goldman. 1st Books Library.
- [DES03] Design patterns : elements of reusable object-oriented software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison Wesley.
- [IMP04] Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Scott Meyers. Addison Wesley.
  - o Item 36: Differentiate between inheritance of interface and inheritance of implementation.
- [GUIQT05] C++ GUI Programming with Qt4. Jasmin Blanchette, Mark Summerfield. Prentice Hall.
- [OCV06] Learning OpenCV: Computer Vision with the OpenCV Library. Gary Bradski, Adrian Kaehler, Bradski Gary, Kaehler Adrian. O'Reilly Media.
- [ES07] Enginyeria del software especificació: especificació de sistemes orientats a objectes amb la notació UML. Dolors Costal, M. Ribera Sancho, Ernest Teniente. Edicions UPC.
- [ES08] Enginyeria del software Disseny I: disseny de sistemes orientats a objectes amb notació UML. Cristina Gómez, Enric Mayol, Antoni Olivé, Ernest Teniente. Edicions UPC.

## 11.2 Referències Web

\*Enllaços vàlids fins el dia 30 de desembre de 2009

- [WWW01] "Explicació Acoblament"
  - o [http://en.wikipedia.org/wiki/Coupling\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Coupling_(computer_science))
- [WWW02] "Teoria de Patrons de disseny"
  - o [http://es.wikipedia.org/wiki/Patrones\\_de\\_diseño](http://es.wikipedia.org/wiki/Patrones_de_diseño)
- [WWW03] "Explicació *OpenCV*"
  - o <http://es.wikipedia.org/wiki/OpenCV>
- [WWW04] "Pàgina web de l'aplicació *Gimp*"
  - o <http://www.gimp.org.es/>

- [WWW05] "Pàgina web de l'aplicació *Google Chrome*"
  - o <http://www.google.com/chrome>
- [WWW06] "Pàgina web del sistema operatiu *Ubuntu Server Edition*"
  - o <http://www.ubuntu.com/products/whatisubuntu/serveredition>
- [WWW07] "Pàgina web de l'aplicació *MySQL*"
  - o <http://www.mysql.com/>
- [WWW08] "Pàgina web de l'aplicació *MediaWiki*"
  - o <http://www.mediawiki.org/wiki/MediaWiki/ca>
- [WWW09] "Pàgina web de l'aplicació *Tortoise SVN*"
  - o <http://tortoisesvn.net/>
- [WWW10] "Pàgina web de l'aplicació *BugZilla*"
  - o <http://www.bugzilla.org/>
- [WWW11] "Pàgina web de l'aplicació *Coppermine Photo Gallery*"
  - o <http://coppermine-gallery.net/>
- [WWW12] "Pàgina web de l'aplicació *Doxygen*"
  - o <http://www.doxygen.org/>
- [WWW13] "Pàgina web de les llibreries, aplicacions i documentació de *Qt*"
  - o <http://qt.nokia.com/>
- [WWW14] "Pàgina web de descàrrega de les llibreries de *OpenCV*"
  - o <http://sourceforge.net/projects/opencvlibrary/>
- [WWW15] "Pàgina web de l'aplicació *StarUML*"
  - o <http://staruml.sourceforge.net/en/>

