

# **Gestió del Manteniment**

**Joan Josep Martorell Umbert**  
ETIS

**Albert Grau Perisé**

10-11-2005

## **Resum del present document**

El document que esteu a punt de llegir conté una descripció exhaustiva del projecte de final de carrera que he desenvolupat com a estudiant d'Enginyeria Informàtica de Sistemes.

En el transcurs de les següents pàgines hi trobareu tota la informació necessària per conèixer el funcionament, l'ús i el disseny d'una aplicació orientada a la gestió i el control del manteniment de la maquinària instal·lada en una empresa fictícia via Internet.

El document està estructurat de manera lògica i intuïtiva a fi efecte d'anar introduïnt lentament al lector en el cos de l'aplicació i poder explicar pas per pas totes les singularitats i les decisions preses en el seu desenvolupament.

Seguint aquest precepte, es comença amb una extensa introducció per situar al lector dins el marc de l'aplicació, i es continua seguint les diverses etapes per les quals ha passat el desenvolupament del projecte: requisits, anàlisi, disseny.

La implementació es pot consultar en el document que conté tot el codi de l'aplicació.

Finalment, es presenten les conclusions que s'han després de la realització d'aquest projecte i es complementa la informació amb un glossari de termes, la bibliografia consultada i uns annexos sobre les sentències de creació de la base de dades.

## Índex de continguts

<a href="#">Resum</a> .....	2
<a href="#">Índex de continguts</a> .....	3
<a href="#">Índex de figures</a> .....	5
<a href="#">Capítol 1- Introducció</a> .....	6
1.1 <a href="#">Introducció</a> .....	6
1.2 <a href="#">Justificació i context</a> .....	7
1.3 <a href="#">Objectius</a> .....	8
1.4 <a href="#">Enfocament i mètode seguit</a> .....	9
1.5 <a href="#">Planificació</a> .....	10
1.6 <a href="#">Productes obtinguts</a> .....	13
1.7 <a href="#">Descripció específica del contingut</a> .....	14
<a href="#">Capítol 2 - Requisits</a> .....	15
2.1 <a href="#">Descripció funcional</a> .....	15
2.2 <a href="#">Tecnologia</a> .....	17
2.3 <a href="#">Actors</a> .....	18
2.4 <a href="#">Casos d'ús</a> .....	19
2.5 <a href="#">Requisits de la interfície d'usuari</a> .....	20
<a href="#">Capítol 3 - Anàlisi</a> .....	21
3.1 <a href="#">Paquets</a> .....	21
3.2 <a href="#">Descripció textual dels casos d'ús</a> .....	22
3.2.1 <a href="#">Cas d'ús nº1: Identificació</a> .....	22
3.2.2 <a href="#">Cas d'ús nº2: Introduir acció</a> .....	22
3.2.3 <a href="#">Cas d'ús nº3: Consulta d'accions pròpies</a> .....	23
3.2.4 <a href="#">Cas d'ús nº4: Consulta d'accions per màquina</a> .....	23
3.2.5 <a href="#">Cas d'ús nº5: Consulta d'accions per empleat</a> .....	24
3.2.6 <a href="#">Cas d'ús nº6: Resum</a> .....	24
3.2.7 <a href="#">Cas d'ús nº7: Afegir màquina</a> .....	25
3.2.8 <a href="#">Cas d'ús nº8: Esborrar màquina</a> .....	25
3.2.9 <a href="#">Cas d'ús nº9: Afegir empleat</a> .....	26
3.2.10 <a href="#">Cas d'ús nº10: Esborrar empleat</a> .....	26
3.2.11 <a href="#">Cas d'ús nº11: Modificar empleat</a> .....	27
3.3 <a href="#">Diagrama de casos d'ús</a> .....	28
3.4 <a href="#">Identificació de les classes d'entitat</a> .....	30
3.5 <a href="#">La interfície d'usuari</a> .....	33
<a href="#">Capítol 4 - Disseny</a> .....	39
4.1 <a href="#">Reutilització</a> .....	39
4.1.1 <a href="#">Model-Vista-Controlador</a> .....	39
4.1.2 <a href="#">Session Façade</a> .....	40
4.1.3 <a href="#">Domain Value Object</a> .....	41
4.1.4 <a href="#">Composite View</a> .....	42
4.1.5 <a href="#">Dispatcher View</a> .....	43
4.2 <a href="#">Gestió de la persistència</a> .....	45
4.3 <a href="#">Disseny dels casos d'ús</a> .....	46
4.3.1 <a href="#">Identificació</a> .....	46
4.3.2 <a href="#">Introduir acció</a> .....	47
4.3.3 <a href="#">Consulta d'accions pròpies</a> .....	49
4.3.4 <a href="#">Consulta d'accions per màquina</a> .....	50

4.3.5 <a href="#">Consulta d'accions per empleat</a> .....	50
4.3.6 <a href="#">Resum</a> .....	51
4.3.7 <a href="#">Afehir màquina</a> .....	52
4.3.8 <a href="#">Esborrar màquina</a> .....	53
4.3.9 <a href="#">Afehir empleat</a> .....	54
4.3.10 <a href="#">Esborrar empleat</a> .....	54
4.3.11 <a href="#">Modificar empleat</a> .....	54
<a href="#">Conclusions</a> .....	57
<a href="#">Glossari</a> .....	58
<a href="#">Bibliografia</a> .....	59
<a href="#">Annex 1: Sentències de creació de la BD</a> .....	60

## Índex de figures

<a href="#">Figura 1</a>	- Divisió del projecte en tasques i temporització.....	10
<a href="#">Figura 2</a>	- Diagrama temporal Setembre-October.....	11
<a href="#">Figura 3</a>	- Diagrama temporal Novembre .....	12
<a href="#">Figura 4</a>	- Diagrama temporal Desembre-Gener.....	12
<a href="#">Figura 5</a>	- Estructura bàsica .....	17
<a href="#">Figura 6</a>	- Actors i responsabilitats.....	18
<a href="#">Figura 7</a>	- Casos d'ús.....	19
<a href="#">Figura 8</a>	- Paquets de l'aplicació .....	21
<a href="#">Figura 9</a>	- Diagrama de casos d'ús.....	28
<a href="#">Figura 10</a>	- Diagrama de classes d'entitat .....	30
<a href="#">Figura 11</a>	- Atributs de les classes d'entitats .....	31
<a href="#">Figura 12</a>	- Identificació d'usuaris .....	33
<a href="#">Figura 13</a>	- Menú del Tècnic Responsable .....	34
<a href="#">Figura 14</a>	- Menú del Director de Fàbrica .....	34
<a href="#">Figura 15</a>	- Menú del Director General .....	35
<a href="#">Figura 16</a>	- Introducció d'accions.....	35
<a href="#">Figura 17</a>	- Consulta d'accions per empleat.....	36
<a href="#">Figura 18</a>	- Resultat de consulta d'accions per empleat.....	36
<a href="#">Figura 19</a>	- Resum.....	37
<a href="#">Figura 20</a>	- Afegir màquina .....	37
<a href="#">Figura 21</a>	- Afegir empleat.....	38
<a href="#">Figura 22</a>	- Model-Vista-Controlador .....	40
<a href="#">Figura 23</a>	- Session Façade.....	41
<a href="#">Figura 24</a>	- Domain Value Object.....	42
<a href="#">Figura 25</a>	- Composite View.....	43
<a href="#">Figura 26</a>	- Dispatcher View.....	44
<a href="#">Figura 27</a>	- Model relacional .....	45
<a href="#">Figura 28</a>	- Diagrama de seqüències del CU nº1 .....	47
<a href="#">Figura 29</a>	- Diagrama de seqüències del CU nº2 .....	48
<a href="#">Figura 30</a>	- Diagrama de seqüències del CU nº3 .....	49
<a href="#">Figura 31</a>	- Diagrama de seqüències del CU nº6 .....	51
<a href="#">Figura 32</a>	- Diagrama de seqüències del CU nº7 .....	52
<a href="#">Figura 33</a>	- Diagrama de seqüències del CU nº8 .....	53
<a href="#">Figura 34</a>	- Diagrama de seqüències del CU nº10 .....	55
<a href="#">Figura 35</a>	- Diagrama de seqüències del CU nº11 .....	56

## Capítol 1: Introducció

### 1.1 Introducció

Tot i ésser estudiant d'Enginyeria Informàtica de Sistemes, l'àmbit professional en el qual treballa diàriament dista enormement d'aquesta disciplina, doncs desenvolupa la tasca de tècnic electrònic responsable del manteniment d'una empresa del món del plàstic.

Del paràgraf anterior es desprèn immediatament una pregunta: perquè estudiar una carrera tan diferent de l'entorn laboral que hom exerceix? La resposta no és única, sinó que combina la il·lusió per aconseguir un títol universitari que sempre és útil com a eina de treball, amb l'afany de superació i les facilitats horàries que la Universitat Oberta de Catalunya proporciona als seus estudiants.

Això implica estar permanentment aprenent i esforçant-se per establir coneixements en dues branques de la tecnologia molt diferents i canviants: la informàtica i l'electrònica industrial. És precisament en aquest punt que sorgeix el moment crucial de la realització del treball de final de carrera.

Entre totes les possibilitats que ofereix la universitat per escollir el tema del projecte, una pren força entre les demés: el desenvolupament d'aplicacions per Internet amb la tecnologia J2EE. Aquest tema em resulta especialment interessant perquè se'n sent parlar molt, d'aquesta tecnologia, i a més, no en sabia absolutament res, exceptuant que es tracta d'una plataforma en Java per el desenvolupament d'aplicacions distribuïdes orientades a l'empresa.

El fet de no conèixer a priori res sobre el funcionament de la tecnologia utilitzada em va suposar un escull enorme, doncs havia d'aprendre a utilitzar J2EE i realitzar el projecte a la vegada, la qual cosa em féu patir diversos retrassos importants.

Per acabar-ho de complicar, vaig pensar en fer una aplicació que pogués utilitzar a l'empresa. En el món del manteniment industrial, un dels pilars que cal tenir ben present és la gestió i la manipulació de tota la informació que es deriva del manteniment de la maquinària: qui ha fet què sobre quina màquina i quan?

Tot i que existeix programari comercial orientat a solucionar aquesta gestió, sovint no respon a les necessitats reals de les empreses, per la qual cosa em vaig decidir a realitzar un experiment: fer una aplicació J2EE per Internet dedicada a la gestió i el control de la informació sobre el manteniment de la maquinària d'una empresa.

En el transcurs de les pàgines següents explicarem pas per pas el desenvolupament d'aquest projecte experimental.

## 1.2 Justificació i context

Aquest treball de final de carrera sorgeix de l'intent d'aprofitar la tecnologia Java J2EE per a la realització d'una aplicació dedicada a la gestió i el control del manteniment de la maquinària d'una empresa.

El món de la fabricació industrial és, com molts altres sectors, extremadament competitiu i això significa que l'empresari ha d'optimitzar al màxim el rendiment de la maquinària de fabricació que té sota el seu poder.

Qualsevol aturada injustificada d'una màquina o fins i tot, un rendiment per sota de les seves possibilitats, significa perdre diners i donar un pas enrera en la cursa per estar entre les empreses capdavanteres.

Al seu torn, la maquinària industrial és cada vegada més complexa i això provoca una enorme diversitat en el tipus d'avaries que poden produir-se, i també una dificultat major per solucionar aquests problemes.

Aquesta és la raó per la qual cal portar un control exhaustiu de les avaries o problemes que es produeixen en les màquines i també de les accions que s'efectuen a fi efecte de prevenir aquestes complicacions.

Tenir aquest tipus d'informació ben gestionada comporta molts beneficis:

- Si es produeix una avaria que ja s'havia produït anteriorment, es pot consultar de quina manera es va solucionar, i quins recursos es van utilitzar. Això permet repetir l'acció o provar una nova solució i comparar quina és més efectiva.
- Es pot dur una política d'accions preventives orientada a la minimització de les avaries.
- Es poden obtenir dades significatives sobre el rendiment a llarg termini tant de la maquinària de l'empresa com dels responsables de manteniment.

Els avantatges que acabem d'exposar justifiquen el present intent d'organitzar tota aquesta informació de manera efectiva. El context presentat, però, és molt ampli i en cap cas es pretén abastar totes les possibilitats, almenys en aquest treball, sinó fer un experiment per probar la viabilitat de la idea, com veurem més endavant.

### 1.3 Objectius

Tot seguit, esquematitzarem els objectius que ens proposem assolir amb la realització d'aquest treball de final de carrera:

- Aprendre el funcionament de la tecnologia J2EE per al desenvolupament d'aplicacions per Internet. Aquesta és una tecnologia nova i puntera en aplicacions d'aquest tipus i com a tal, suscita un elevat interès als estudiant d'informàtica, ja que sempre és necessari i interessant actualitzar els coneixements sobre una disciplina tant utilitzada i canviant.
- Aprendre a desenvolupar una aplicació nova començant des de zero: establir una planificació i anar creant un producte passant per diferents etapes: requisits, anàlisi, disseny, implementació, prova, manteniment, etc.
- Considerar i aplicar cadascuna de les diferents matèries estudiades per obtenir un producte final homogeni i funcional.
- Crear una aplicació experimental que respongui a una necessitat real de les empreses de fabricació com ho és la gestió del manteniment.



## 1.4 Enfocament i mètode seguit

En la creació d'un producte de programari, com passa també en altres tipus de producte, s'estableixen unes etapes a través de les quals es va perfilant la funcionalitat i el disseny que tindrà el producte acabat.

Hi ha diferents mètodes proposats per creadors i teòrics sobre com portar a terme la creació d'una nova aplicació de programari. D'entre tots els possibles, ens hem inclinat per utilitzar un cicle de vida del programari iteratiu i incremental basat en el cicle de vida en cascada.

Això significa que dividirem en petites unitats el programari a realitzar i treballarem cada unitat exhaustivament mitjançant l'aplicació de diferents etapes: recollida de requisits, anàlisi, disseny, implementació i prova.

Aquest mètode s'anomena iteratiu perquè es passa per aquestes etapes diverses vegades en un mateix projecte i incremental perquè es van considerant diferents parts per, al final, considerar el conjunt sencer.

El fet d'obrar d'aquesta manera implica que de vegades caldrà modificar parts que ja havíem realitzat en adonar-nos, mentre estudiem i construïm una unitat posterior, de certes mancances de les primeres que no s'havien contemplat.

L'explicació del desenvolupament del projecte que conté aquest document passarà també per aquestes mateixes etapes, tot i que en la descripció de cada etapa considerarem el conjunt de l'aplicació per raons òbvies (si no ho féssim així, aquest document seria molt extens i redundant).

En els capítols posteriors, doncs, es tractarà per aquest ordre:

- la recollida dels requisits
- l'anàlisi
- el disseny
- la implementació
- la prova

## 1.5 Planificació del projecte

La realització del projecte s'ha dividit en petites tasques que s'han repartit temporalment tenint en compte sobretot la quantitat de feina que comporta la tasca en qüestió, les entregues programades corresponents a les proves d'avaluació continuada i, en la mesura del possible, la disponibilitat laboral del moment.

Aquest últim factor ha estat molt difícil de preveure, doncs la meua ocupació acostuma a patir alts i baixos freqüents i imprevistos, per la qual cosa, algunes tasques no s'han dut a terme en els terminis fixats, cosa que ha jugat en detriment de les tasques posteriors.

L'aparició de certs problemes en l'aprenentatge de la tecnologia J2EE ha retrassat també algunes dates, sobretot pel que fa als servlets i a la gestió de la persistència.

Tot i això, la planificació s'ha anat seguint en l'ordre establert tenint molt en compte les tasques que s'havien de donar per finalitzades per poder passar a tasques posteriors. Cal dir que la planificació ha suposat una ajuda important per saber destinar temps i recursos a fi efecte de poder presentar el projecte acabat.

La planificació del projecte s'ha especificat amb l'ajut de diagrames temporals, els quals presentarem a continuació:





















Id		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1		Documentar-se J2EE	33 días	lu 20/09/04	ju 28/10/04		
2		Abstract	4 días	lu 20/09/04	ju 23/09/04		
3		Planificació	3 días	ju 23/09/04	do 26/09/04		
4		PAC1: Abstract i Planificació	1 día	lu 27/09/04	lu 27/09/04		
5		Identificació i descripció dels casos d'ús	4 días	lu 27/09/04	ju 30/09/04	3	
6		Diagrama de classes i de casos d'ús	3 días	ju 30/09/04	do 03/10/04		
7		Especificació de la interfície de l'aplicació	6 días	lu 04/10/04	do 10/10/04		
8		Disseny de l'estructura	11,5 días	lu 11/10/04	lu 01/11/04		
9		Viatge de feina	6 días	lu 18/10/04	sá 23/10/04		
10		PAC2: Anàlisi i Disseny	1 día	ma 02/11/04	ma 02/11/04	8	
11		Implementació de les classes d'entitat	8 días	mi 03/11/04	mi 10/11/04	10	
12		Prova de les classes d'entitat i persistència	6 días	do 07/11/04	vi 12/11/04		
13		Desenvolupament dels casos d'ús bàsics	19 días	sá 13/11/04	mi 08/12/04	12	
14		Proves dels casos d'ús bàsics	5 días	vi 03/12/04	ju 09/12/04		
15		PAC3: Primeres implementacions	1 día	vi 10/12/04	vi 10/12/04	14	
16		Desenvolupament dels casos d'ús restants	11 días	sá 11/12/04	vi 24/12/04	15	
17		Proves dels casos d'ús restants	5 días	lu 20/12/04	vi 24/12/04		
18		Memòria	11 días	lu 27/12/04	do 09/01/05	17	
19		Entrega final	1 día	lu 10/01/05	lu 10/01/05	18	

Figura 1 - Divisió del projecte en tasques i temporització

Tot seguit presentarem el desplegament d'aquestes tasques en diagrames temporals, en els quals s'aprecia l'evolució de les tasques així com quines d'aquestes s'havien de donar per finalitzades per poder continuar amb les següents:

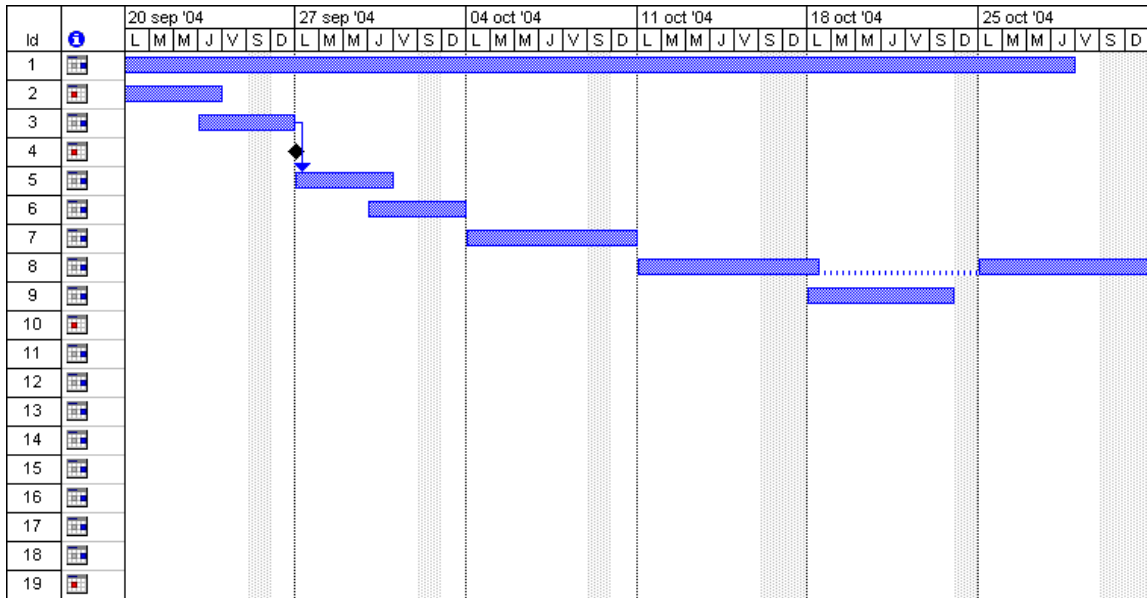


Figura 2 - Diagrama temporal Setembre-October

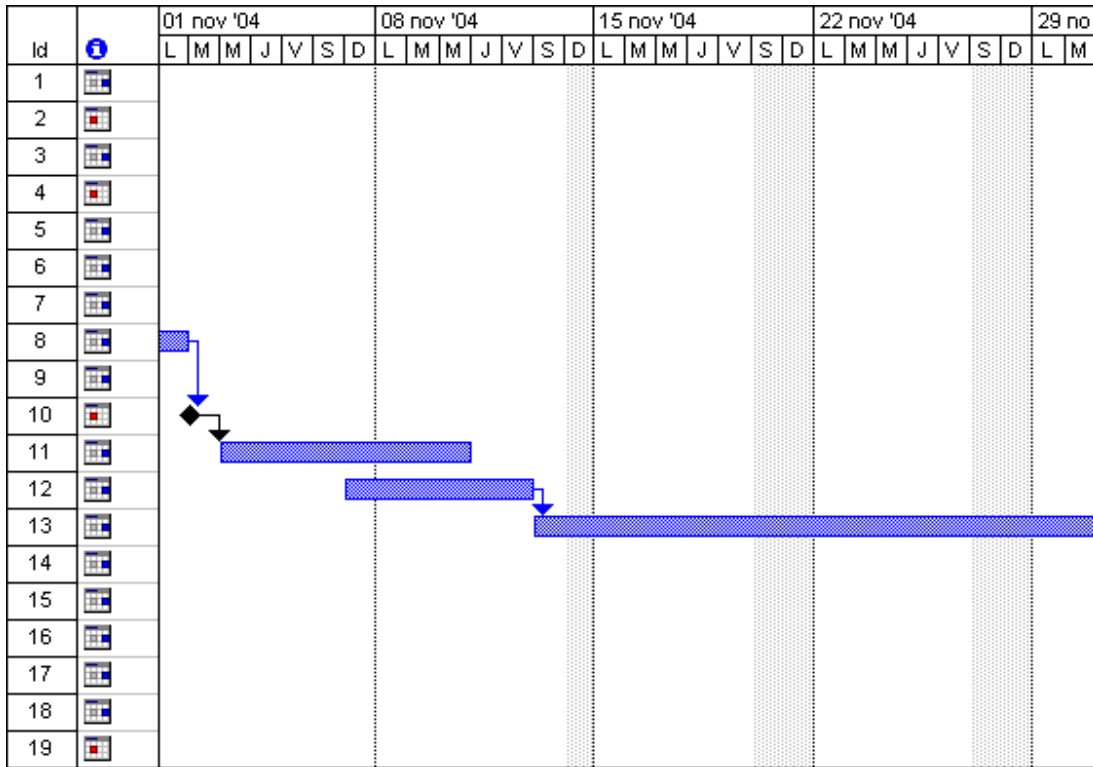


Figura 3 - Diagrama temporal Novembre

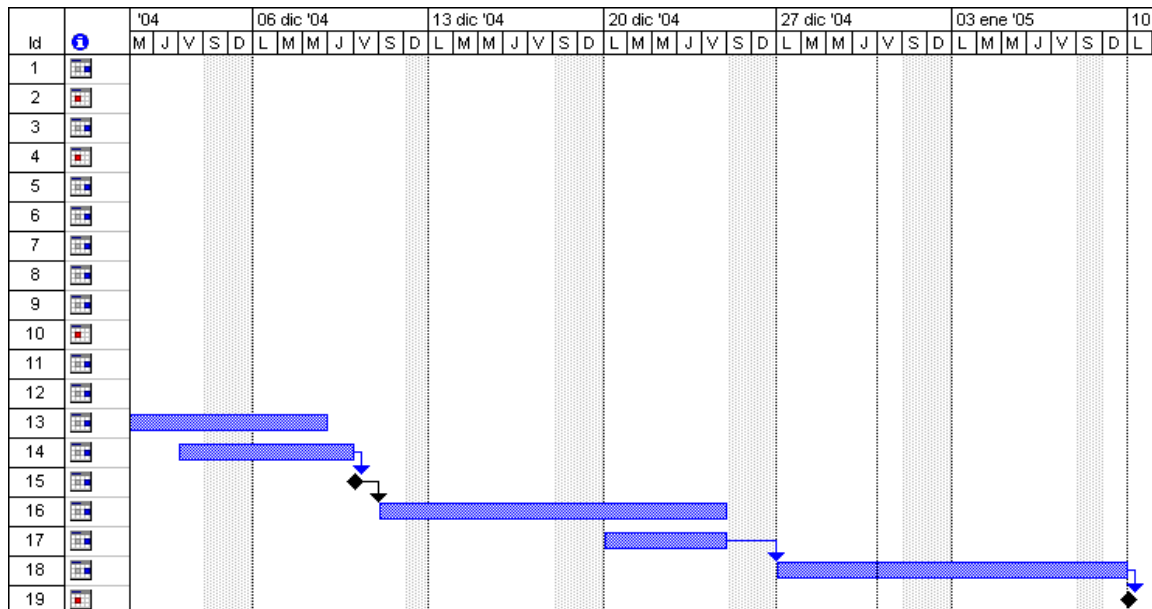


Figura 4 - Diagrama temporal Desembre-Gener

## 1.6 Productes obtinguts

La realització d'aquest treball de final de carrera dóna com a producte final una aplicació experimental destinada a gestionar la informació que es deriva del manteniment de la maquinària instal·lada en una empresa.

El nom general de l'aplicació és, doncs, Gestió del Manteniment.

L'aplicació contempla la funcionalitat bàsica d'un programari d'aquestes característiques i pretén donar validesa i determinar punts forts i punts febles a la idea de combinar la tecnologia J2EE amb la manipulació d'informació sobre les accions de manteniment que s'efectuen en la maquinària industrial.

L'aplicació permetrà, en funció del rol que desenvolupa l'usuari, introduir la informació més bàsica i necessària al sistema i/o consultar aquesta informació des de diferents punts de vista els quals estan destinats a emfasitzar certs aspectes del manteniment d'una instal·lació industrial. De la informació obtinguda serà possible treure'n conclusions molt determinants respecte al rendiment tant de la maquinària com del personal responsable i, a més a més, planificar estratègies o polítiques destinades a millorar aquest manteniment.

Donat que la raó d'ésser de l'aplicació rau en la gestió d'un tipus determinat d'informació, resulta obvi que cal desar tota aquesta informació d'una manera efectiva i flexible, per la qual cosa caldrà utilitzar una base de dades que estarà associada a l'aplicació. Així mateix, serà imprescindible disposar d'un servidor d'aplicacions que permeti el desplegament i l'execució del programari dissenyat. De les tecnologies i programaris emprats en el desenvolupament d'aquest treball en parlarem en el següent capítol.

## 1.7 Descripció específica del contingut

Els següents capítols desgranaran els detalls de l'aplicació de Gestió de Manteniment. Per a fer-ho, s'ha cregut oportú seguir paulatinament els mateixos passos que ha fet el producte, des de la idea inicial, exposada en aquesta introducció, fins a la implementació final.

Es començarà, per tant, per la documentació dels requisits de l'aplicació, una informació bàsica per a establir l'abast de l'aplicació i les seves funcions. Descriurem el context en el qual s'emmarca el programari i farem un esbós de les funcions principals i els objectius d'aquestes dins l'aplicació.

Un cop situats, podrem començar a analitzar les funcions del programari. Això es descriu en el capítol tercer. S'explicarà com es divideix l'aplicació en paquets, quins casos d'ús s'han detectat i com han d'operar i es presentaran diversos diagrames que facilitaran la comprensió d'aquests conceptes.

Després de l'anàlisi, passarem a explicitar el procés de disseny de l'aplicació. En aquest sentit, posarem de manifest que la reutilització forma part implícita de la tecnologia J2EE i de la programació orientada a objectes i exposarem quines formes de reutilització s'han considerat en el desenvolupament del projecte.

També tractarem la gestió de la persistència amb el recolzament, en aquest cas d'una base de dades relacional.

Acabarem el capítol amb el disseny dels casos d'ús. En aquest pas final, detallarem un per un totes les operacions dels casos d'ús i n'explicarem la raó de ser ajudant-nos de diagrames de seqüències.

Finalment, acabarem aquest document amb les conclusions que s'han després de la realització d'aquest projecte, un glossari i la bibliografia consultada.

## Capítol 2: Requisits

### 2.1 Descripció funcional

Aquesta aplicació sorgeix de la necessitat d'una empresa dedicada a la producció de films i bosses de plàstic de gestionar i supervisar el manteniment que el personal responsable realitza a la maquinària.

L'empresa centra la seva producció en la manufacturació de bobines de film bufat. També tracta la posterior transformació d'aquestes bobines en productes finals d'ús comú: bosses de plàstic, làmines, guants, etc.

L'empresa, de nom fictici JoanPlast S.A. té certa envergadura i per això la seva producció es realitza en diferents fàbriques (totes propietat de l'empresa, òbviament). Aquesta distribució geogràfica de la producció en punts relativament allunyats geogràficament els uns dels altres fa especialment adequat portar a terme el control del manteniment via Internet.

Concretament, l'empresa disposa de tres fàbriques:

- JoanPlast Girona és la primera que es fundà i és la que centralitza la gestió del manteniment. En ella hi treballa el Director General.
- JoanPlast Bilbao.
- JoanPlast Santander.

A cada fàbrica hi ha instal·lades un cert nombre de màquines, les quals són l'objecte de les accions de manteniment que volem gestionar.

Així mateix, a l'empresa hi treballa un cert nombre d'empleats, i tots tenen un càrrec assignat. Aquests treballadors es reparteixen en les diferents fàbriques de l'empresa i segons el seu càrrec podran accedir a diferents funcionalitats dins l'aplicació.

Inicialment es defineixen tres tipus d'usuari que poden accedir a l'aplicació:

- Tècnic Responsable.
- Director de Fàbrica.
- Director General.

A cada fàbrica hi treballa un seguit de Tècnics Responsables que són les persones que coordinen els recursos (personal, peces de recanvi, etc) i es responsabilitzen de la correcta realització de les accions de manteniment. Aquest usuari només pot introduir al sistema la informació sobre les accions de manteniment de les quals es responsabilitza i consultar les accions que ell mateix ha dut a terme.

A més a més, cada fàbrica compta amb un Director de Fàbrica. El Director de fàbrica és un usuari intermig a nivell de privilegis i, a més de les funcions pròpies del Tècnic Responsable, pot accedir a la informació relativa a les màquines i als empleats de la fàbrica que regenta, però no de les altres fàbriques de l'empresa.

Finalment, el Director General és l'usuari amb més privilegis de l'aplicació i té accés a la informació relativa a qualsevol punt de l'empresa (és a dir, relativa a qualsevol màquina o empleat de qualsevol de les fàbriques). A part d'això, també té accés a les funcions dels dos càrrecs anteriorment descrits. Aquest usuari és l'únic que pot donar d'alta noves màquines o introduir nous empleats (i esborrar-los). El Director General treballa a la fàbrica on es centralitza la informació sobre el manteniment. L'administrador del sistema, en el cas que existeixi, assumirà també aquest

rol, ja que és necessari que aquest empleat pugui navegar per tota l'aplicació a fi de complir amb la seva feina.

Qualsevol usuari de l'aplicació haurà d'identificar-se a l'inici de la sessió i el sistema obrarà en conseqüència, mostrant-li només les funcions que li són permeses.

L'aplicació haurà d'emmagatzemar informació sobre aquest empleats. Dels empleats, ens n'interessa la següent informació: el nom, els cognoms, l'antiguitat, el nom d'usuari i la contrasenya amb els quals s'identifica davant el sistema, el càrrec i la fàbrica on treballa.

Com s'ha dit anteriorment, cada fàbrica allotja al seu interior un cert nombre de màquines, de les quals també necessitem tenir-ne informació emmagatzemada. Aquesta informació es redueix a la referència, l'any d'instal·lació i una petita descripció sobre les seves característiques.

El punt més interessant de la informació que volem desar a la base de dades és la informació que es genera en la realització d'una acció de manteniment. Una acció de manteniment es pot definir com el conjunt de treballs efectuats sobre una màquina a fi de solucionar un problema o de prevenir-ne l'aparició. Més endavant es presenta un glossari de termes on es tracta aquesta i altres definicions importants de l'aplicació més a fons.

Una acció de manteniment es caracteritzarà, doncs, per la següent informació:

- Si es tracta d'una acció correctiva o preventiva (veure el glossari de termes).
- La data en què s'ha efectuat el treball.
- L'hora en què s'ha realitzat l'acció.
- El número d'hores destinades al treball.
- El nombre d'operaris que hi han intervingut.
- Una descripció del treball realitzat.
- El tipus d'acció.
- L'empleat responsable de dur-la a terme.
- Evidentment, la màquina objecte de l'acció.

El tipus de l'acció és una classificació general de les accions de manteniment. Exemples de tipus són: avaria mecànica, avaria elèctrica, etc.

Per simplificar una mica l'aplicació, es consideraran fixes les següents entitats. Això significa que el programari no en contemplarà la inserció, la modificació o l'esborrament:

- Les fàbriques. És ben poc usual el fet de construir una nova fàbrica o tancar-ne una. Per aquesta raó, l'aplicació les considerarà fixes i les inserirem directament mitjançant la base de dades (administrador del sistema).
- Els tipus de les accions. Els tipus són molt generals i clàssics del manteniment, i a més, hi ha estratègies que es basen en aquests tipus, per la qual cosa també són considerats fixes.
- Els càrrecs. En principi es defineixen tres càrrecs bàsics que contemplen les necessitats que ha de complir el programari.

Aquestes entitats es manipularan directament a nivell de base de dades.



## 2.2 Tecnologia

El projecte s'ha realitzat amb el recolzament de les següents tecnologies:

- Llenguatge de programació: Java. Dins del marc Java, s'utilitza J2EE, que defineix una plataforma per el desenvolupament d'aplicacions distribuïdes orientades a l'empresa. S'ha usat el J2SDK 1.4.1.
- Entorn de desenvolupament: Eclipse. Eclipse és un IDE de codi obert que permet treballar amb J2EE sense problemes. La versió utilitzada és la 2.1.
- Per al desenvolupament dels servlets s'ha inclòs el package corresponent a la versió J2SDK 2.1.
- Servidor d'aplicacions: JBoss. És un servidor d'aplicacions que implementa l'arquitectura J2EE i és de codi obert. Duu integrat un servidor Tomcat. S'ha utilitzat el JBoss 3.2.1, el qual porta el Tomcat 4.1.24.
- Plug-in d'Eclipse per utilitzar JBoss. Versió 1.0ea5.
- Eina per a desplegar l'aplicació: Ant, també de codi obert. La versió emprada és la 1.5.3.
- Base de dades: MySQL. És una base de dades de codi obert senzilla però suficient per a dur a terme el treball. S'ha fet servir la versió 1.4.

L'estructura bàsica de l'aplicació es pot esquematitzar en el següent model:

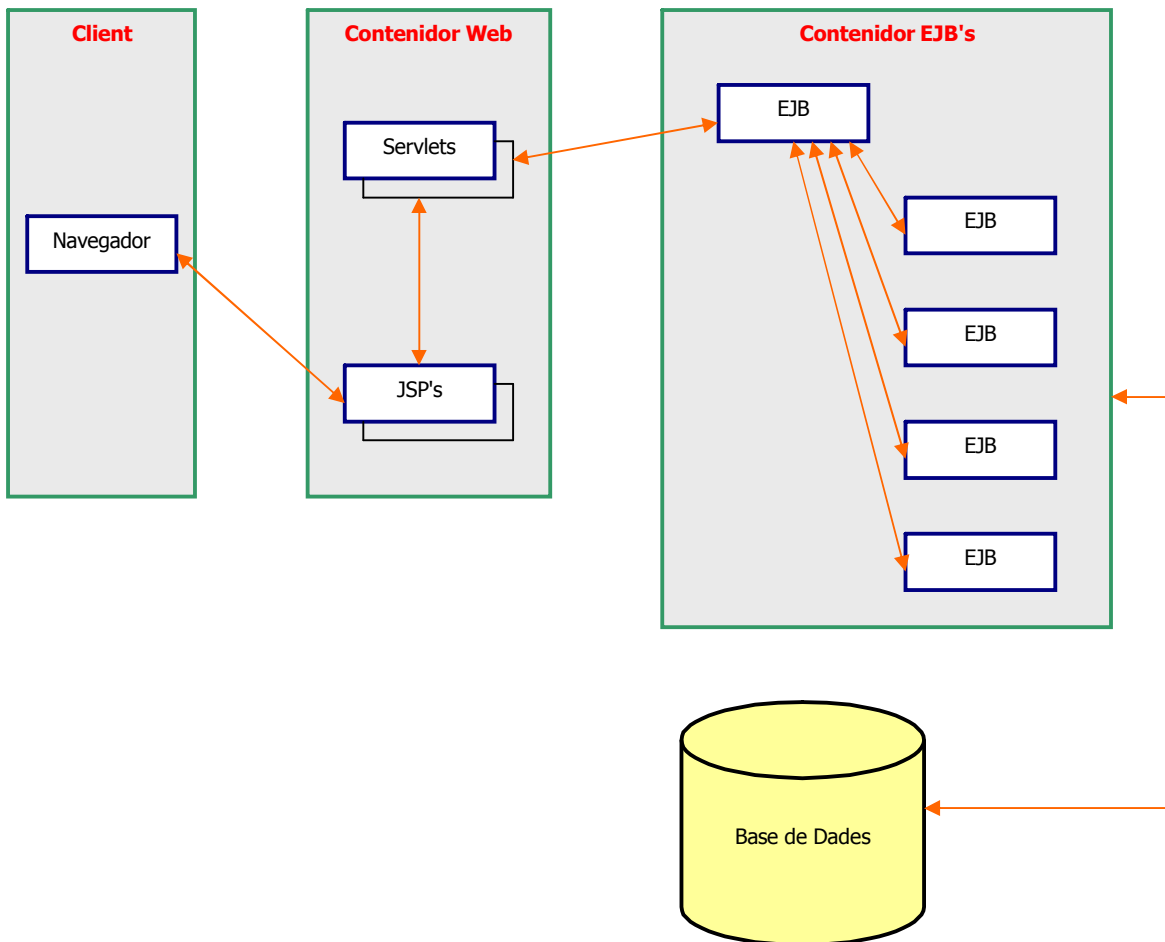


Figura 5 - Estructura bàsica

## 2.3 Actors

En l'aplicació, i segons els requisits que s'han descrit, s'han identificat clarament tres actors primaris (cal recordar, però, que l'administrador del sistema i el director general tenen els mateixos privilegis i, per tant, tots dos encaixaran amb un mateix càrrec pel que fa a l'aplicació). A continuació s'esquematitzen les responsabilitats que corresponen a cadascun dels actors identificats:

Actors	Responsabilitats
Tècnic Responsable	Introduir accions de manteniment efectuades sota la seva supervisió i de les quals n'és responsable.  Consultar les accions de manteniment que s'han realitzat amb el propi tècnic com a responsable.
Director de Fàbrica	Les anteriors responsabilitats.  Consultar les accions de manteniment que s'han realitzat a la fàbrica que l'actor regeix.
Director General	Les anteriors responsabilitats.  Consultar les accions de manteniment que s'han realitzat a qualsevol fàbrica de l'empresa.  Introduir noves màquines a l'aplicació.  Esborrar màquines.  Introduir nous usuaris.  Modificar nous usuaris.  Esborrar usuaris.

Figura 6 - Actors i responsabilitats

Qualsevol usuari de l'aplicació haurà d'identificar-se a l'inici de la sessió i el sistema obrarà en conseqüència, mostrant-li només les funcions que li són permeses.

## 2.4 Casos d'ús

Després del que s'ha explicat, estem en condicions de presentar una primera esquematització dels casos d'ús que l'aplicació contemplarà. Tot seguit en proporcionem el nom, els actors primaris de cadascun d'ells i una petita descripció:

Número	Cas d'ús	Actors	Descripció
1	Identificació	Tècnic Responsable Director de Fàbrica Director General	L'actor s'identifica davant el sistema per accedir a les funcions de l'aplicació que li són pròpies.
2	Introduir acció	Tècnic Responsable Director de fàbrica Director General	Inserció de la informació rellevant d'una acció de manteniment d'una màquina efectuada sota la responsabilitat de l'usuari.
3	Consulta d'accions pròpies	Tècnic Responsable Director de Fàbrica Director General	Consulta de les accions que l'actor ha efectuat com a responsable. S'especifica un marge temporal per a restringir les dades a voluntat.
4	Consulta d'accions per màquina	Director de Fàbrica Director General	Consulta de les accions realitzades en una màquina. El Director de Fàbrica només pot consultar aquelles accions que s'han efectuat en màquines de la fàbrica on treballa, mentres que el Director General no té cap restricció en aquest sentit. S'especifica un marge temporal per a restringir les dades a voluntat.
5	Consulta d'accions segons l'empleat	Director de Fàbrica Director General	Consulta de les accions realitzades sota la responsabilitat d'un empleat concret. El Director de Fàbrica només pot consultar aquelles accions que ha efectuat un empleat que treballa en la mateixa fàbrica que aquest, mentres que el Director General no té cap restricció en aquest aspecte. S'especifica un marge temporal per a restringir les dades a voluntat.
6	Resum	Director de Fàbrica Director General	Consulta resumida de la quantitat d'hores i intervencions efectuades en les màquines dins d'un període especificat. Si l'actor és un Director de Fàbrica, el resum contempla només les màquines de la mateixa fàbrica que aquest, mentres que si es tracta del Director General, el resum contempla totes les màquines de l'empresa.
7	Afegir màquina	Director General	Afegeix la informació sobre una màquina nova en una de les fàbriques de l'empresa.
8	Esborrar màquina	Director General	Esborra (dóna de baixa) una màquina d'una de les fàbriques de l'empresa.

9	Inserir empleat	Director General	Afegeix un nou usuari a la base de dades del sistema.
10	Esborrar empleat	Director General	Esborra un usuari de la base de dades del sistema.
11	Modificar empleat	Director General	Modifica les dades d'un usuari de l'aplicació.

Figura 7 - Casos d'ús

## 2.5 Requisits de la interfície d'usuari

La interfície d'usuari és allò que els usuaris veuen del funcionament del programari. El disseny d'aquesta interfície ha de tenir en compte el tipus d'usuari que hi interaccionarà.

Els usuaris d'aquesta aplicació responen a rols diferents però en cap cas s'assumeix que aquests tinguin un bagatge informàtic important. Per exemple, un mecànic que pot ostentar perfectament el càrrec de Tècnic Responsable pot no tenir gens de coneixements sobre eines informàtiques i fins i tot, ser una mica destraler en aquest aspecte.

És per això que es fa necessari implementar una interfície d'usuari amb les següents característiques:

- Senzilla al màxim
- Homogènia
- Intuitiva

Aquesta aplicació serà utilitzada amb una periodicitat diària, però no intensiva. Com que l'entorn en el qual s'utilitzarà el programari pot ésser sorollós (els tècnics disposaran d'un petit punt de treball proper a la maquinària, els directors de fàbrica tindran un despatx que permeti supervisar l'entorn industrial al màxim i únicament el director general disposarà d'un espai més relaxat) es desaconsella cap mena d'avís acústic en la interfície d'usuari.

Tampoc s'utilitzaran recursos superflus com ara imatges o animacions que puguin fer la interfície més críptica a fi d'agilitzar les operacions.

Més endavant tornarem a insistir en aquests aspectes.

## Capítol 3: Anàlisi

### 3.1 Paquets

Inicialment s'havia previst dividir l'aplicació en dos paquets: un que contemplés els casos d'ús amb operacions bàsiques sobre les entitats, és a dir, inserció, modificació i esborrament, i un altre que agrupés els casos d'ús que realitzessin consultes sobre la informació emmagatzemada a la base de dades.

Un cop immersos en el desenvolupament del programari, però, s'ha cregut oportú dividir les classes de l'aplicació segons el seu funcionament dins el conjunt sencer, per la qual cosa s'ha optat per la definició d'un sol paquet que inclou la resta de paquets organitzats segons la funció que implementen.

En aquest sentit, podem establir el diagrama següent que representa aquesta situació:

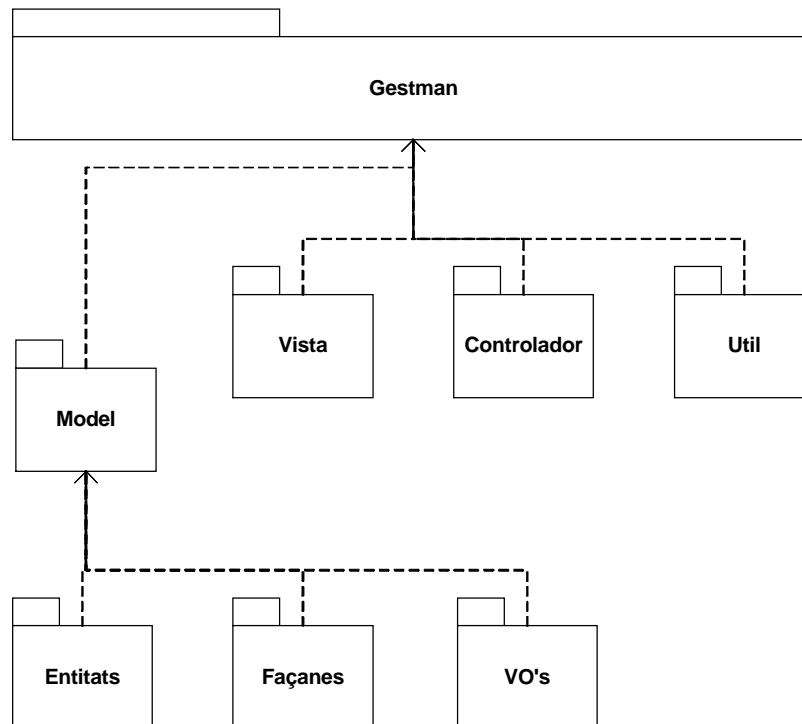


Figura 8 - Paquets de l'aplicació

En aquest diagrama podem veure el paquet principal que engloba l'aplicació (Gestman) i les inclusions d'altres paquets segons la seva utilització (Model, que al seu torn inclou els paquets Entitats, Façanes i VO's, Vista, Controlador i Util).

## 3.2 Descripció textual dels casos d'ús

### 3.2.1 - Cas d'ús número 1: "Identificació"

Resum de la funcionalitat: Permet a l'usuari identificar-se com a tal davant el sistema.

Paper dins el treball de l'usuari: Apareix sempre que s'accedeix a l'aplicació, sigui quin sigui el tipus d'usuari que hi accedeixi.

Actors: **Tècnic Responsable, Director de Fàbrica, Director General.**

Casos d'ús relacionats: Tota la resta.

Precondició: Cap.

Postcondició: el sistema reconeix l'usuari i, en funció del seu càrrec, es permet la interacció amb l'aplicació.

L'usuari introdueix el seu nom d'usuari i la seva contrasenya en un senzill formulari d'accés i pitja un botó, acció amb la qual el sistema procedeix a validar les dades. Si la informació és correcta, s'ofereix a l'usuari una pantalla inicial de benvinguda amb totes les funcions que li són permeses (sempre en funció del seu càrrec).

Alternatives de procés i excepcions: Si no es reconeix l'usuari s'ha d'informar de l'error.

### 3.2.2 - Cas d'ús número 2: "Introduir acció"

Resum de la funcionalitat: Afegeix una acció de manteniment a la base de dades del sistema.

Paper dins el treball de l'usuari: És el cas d'ús principal del treball dels **Tècnics Responsables.**

Actors: **Tècnic Responsable, Director de Fàbrica, Director General.**

Casos d'ús relacionats: Identificació.

Precondició: L'usuari s'ha identificat positivament davant el sistema i l'acció no existeix a la base de dades.

Postcondició: S'afegeix l'acció a la BD.

L'aplicació presenta un formulari amb la informació que s'ha d'omplir: la data i l'hora en què s'ha realitzat l'acció, un checkbox que indica si aquesta és preventiva, el número d'operaris, el número d'hores dedicades, la referència de la màquina i un camp de text on s'hi pot escriure una descripció de la tasca efectuada. El tipus general de l'acció es pot escollir d'entre una sèrie d'opcions que es presenten en una llista desplegable. El nom de l'empleat s'insereix automàticament sense que s'hagi de teclejar.

L'usuari introdueix les dades i, un cop fet això, prem un botó, el sistema les valida, i si són correctes les afegeix a la BD.

Alternatives de procés i excepcions: Si les dades són incorrectes, com per exemple, introduir una data incorrecta es notifica l'error a l'usuari.

En cas que la màquina objecte de l'acció de manteniment no estigui en la mateixa fàbrica que l'empleat que introdueix les dades, el sistema ha d'informar que aquesta informació no es pot introduir a la base de dades de l'aplicació.

### **3.2.3 - Cas d'ús número 3: "Consultar accions pròpies"**

Resum de la funcionalitat: Es poden visualitzar les accions que el propi usuari ha introduït a la BD.

Paper dins el treball de l'usuari: L'utilitza l'usuari de forma ocasional per comprovar quines accions de manteniment s'han realitzat sota la seva supervisió (i de les quals, per tant, és responsable).

Actors: **Tècnic Responsable, Director de Fàbrica, Director General.**

Casos d'ús relacionats: Identificació.

Precondició: L'usuari s'ha identificat positivament davant el sistema.

Postcondició: cap, en tractar-se d'una consulta

L'usuari selecciona el període de temps que desitja i el sistema retorna la informació referent a les accions de manteniment dutes a terme per l'usuari en el període especificat. La llista es presenta paginada per a una major comoditat.

Alternatives de procés i excepcions: En cas d'introducció errònia del període de consulta, cal avisar l'usuari perquè pugui tornar a especificar-lo.

### **3.2.4 - Cas d'ús número 4: "Consulta d'accions per màquina"**

Resum de la funcionalitat: Es poden visualitzar les accions efectuades sobre una màquina concreta.

Paper dins el treball de l'usuari: Emprat periòdicament per tal de programar altres accions i veure com es comporta la màquina en qüestió.

Actors: **Director de Fàbrica, Director General.**

Casos d'ús relacionats: Identificació.

Precondició: L'usuari s'ha identificat positivament davant el sistema.

Postcondició: cap, ja que es tracta d'una consulta.

L'usuari selecciona el període de temps que desitja contemplar i la màquina objecte de les accions i el sistema retorna la informació referent a les accions de manteniment relacionades en el període especificat.

Si l'usuari que realitza la petició és un **Director de Fàbrica**, només podrà consultar les accions relatives a les màquines que es troben en la seva mateixa fàbrica; si és **Director General**, pot consultar les accions relatives a qualsevol màquina de l'empresa.

Alternatives de procés i excepcions: Si com a **Director de Fàbrica** s'escull una màquina que no es troba a la fàbrica on es treballa, el sistema no presenta cap informació i informa de la violació comesa. En cas que les dades introduïdes (la referència de la màquina o les dates que defineixen el període de consulta) també s'informa de l'error.

### **3.2.5 - Cas d'ús número 5: "Consulta d'accions per empleat"**

Resum de la funcionalitat: Es poden visualitzar les accions realitzades per un empleat de la fàbrica.

Paper dins el treball de l'usuari: Emprat ocasionalment per tal de veure la feina realitzada per un empleat de la fàbrica.

Actors: **Director de Fàbrica, Director General.**

Casos d'ús relacionats: Identificació.

Precondició: L'usuari s'ha identificat positivament davant el sistema.

Postcondició: cap.

L'usuari selecciona el període de temps que desitja contemplar i l'empleat responsable de realitzar les accions i el sistema retorna la informació referent a les accions de manteniment relacionades en el període especificat.

Si l'usuari que realitza la petició és un **Director de Fàbrica**, només podrà consultar les accions relatives als empleats que treballen a la seva mateixa fàbrica; si és **Director General**, pot consultar les accions realitzades per qualsevol empleat de l'empresa.

Alternatives de procés i excepcions: Si com a **Director de Fàbrica** s'escull un empleat que no treballa a la mateixa fàbrica, el sistema informa de l'error.

En cas que les dades inserides en l'especificació de la consulta siguin errònies, cal informar d'aquesta situació a l'usuari.

### **3.2.6 - Cas d'ús número 6: "Resum"**

Resum de la funcionalitat: Es presenta una informació resumida de les accions de manteniment efectuades a la fàbrica o a l'empresa.

Paper dins el treball de l'usuari: S'hi accedeix periòdicament per tal de comprovar l'estat general de la fàbrica o empresa i treure'n conclusions.

Actors: **Director de Fàbrica, Director General.**



Casos d'ús relacionats: Identificació.

Precondició: L'usuari s'ha identificat positivament davant el sistema.

Postcondició: cap.

L'usuari selecciona el marge de temps que creu convenient i el sistema li retorna informació resumida sobre totes les accions que s'han efectuat durant aquest període. Si l'usuari que realitza la petició és un **Director de Fàbrica**, només podrà consultar la informació resumida que fa referència a les màquines de la seva mateixa fàbrica; si és **Director General**, en canvi, el resum inclourà totes les màquines de l'empresa.

### **3.2.7 - Cas d'ús número 7: "Afegir màquina"**

Resum de la funcionalitat: Afegeix una màquina nova a la BD.

Paper dins el treball de l'usuari: S'utilitza en cas d'instal.lació d'una nova màquina a l'empresa. La freqüència d'ús és, consegüentment, ocasional.

Actors: **Director General**.

Casos d'ús relacionats: Identificació.

Precondició: L'usuari s'ha identificat positivament davant el sistema. La màquina no existeix a la base de dades del sistema.

Postcondició: S'han afegit les dades de la màquina a la BD.

El sistema presenta un formulari amb tota la informació rellevant de la màquina. Aquesta informació es redueix a una referència textual única, l'any d'instal.lació de la màquina, la descripció de les seves característiques i la fàbrica on s'ha instal.lat, el nom de la qual es pot triar entre les opcions que es presenten en una llista desplegable. Un cop introduïda la informació, es prem un botó que permet a l'aplicació validar les dades i afegir-les a la BD si tot és correcte.

Alternatives de procés i excepcions: En cas d'informació incorrecta, el sistema presenta l'error a l'usuari.

### **3.2.8 - Cas d'ús número 8: "Esborrar màquina"**

Resum de la funcionalitat: Esborra una màquina de la BD.

Paper dins el treball de l'usuari: S'utilitza de forma molt ocasional, ja que el fet de donar de baixa una màquina (treure-la d'una fàbrica) resultar ser molt poc freqüent.

Actors: **Director General**.

Casos d'ús relacionats: Identificació.

Precondició: La màquina existeix dins l'aplicació.

Postcondició: La màquina s'esborra de la BD. Juntament amb ella, també s'esborren totes les accions de manteniment que hi feien referència.

L'usuari selecciona una de les màquines de l'empresa indicant la seva referència única i pitja un botó per esborrar-la. Després de la confirmació que sol·licita el sistema, s'esborra la màquina i totes les accions de manteniment que hi feien referència, doncs no és necessari tenir informació d'una màquina que ja no es posseeix i que, per tant, no cal dur-ne el control del manteniment.

Alternatives de procés i excepcions: Si durant el procés d'esborrament es genera algun tipus d'error, cal informar a l'usuari.

### **3.2.9 - Cas d'ús número 9: "Afegir empleat"**

Resum de la funcionalitat: Afegeix un nou empleat a la BD.

Paper dins el treball de l'usuari: Utilitzat ocasionalment en cas que un empleat nou formi part d'una de les fàbriques.

Actors: **Director General**.

Casos d'ús relacionats: Identificació.

Precondició: L'usuari s'ha identificat positivament davant el sistema. L'empleat no existeix a la base de dades del sistema.

Postcondició: S'han afegit les dades de l'empleat a la BD.

El sistema presenta un formulari amb tota la informació rellevant de l'empleat. Aquesta informació consisteix en el nom, els dos cognoms, l'antiguitat, el nom d'usuari i la contrasenya que s'usaran en el procés d'identificació de l'usuari davant l'aplicació, el càrrec que ocuparà i la fàbrica on treballarà. Aquests dos últims conceptes es podran escollir entre les opcions existents a la base de dades i que es presentaran mitjançant llistes desplegable. L'usuari omple degudament el formulari i després pitja un botó que permet a l'aplicació validar les dades i afegir-les a la BD si tot és correcte.

Alternatives de procés i excepcions: En cas d'informació incorrecta, el sistema presenta l'error a l'usuari.

### **3.2.10 - Cas d'ús número 10: "Esborrar empleat"**

Resum de la funcionalitat: Esborra un empleat de la BD.

Paper dins el treball de l'usuari: S'utilitza de forma ocasional en el cas en què un empleat deixi de treballar en l'empresa (o es canviï la secció on treballa).

Actors: **Director General**.

Casos d'ús relacionats: Identificació.

Precondició: L'empleat existeix dins l'aplicació.

Postcondició: L'empleat s'esborra de la BD.

L'usuari escriu el nom i els dos cognoms de l'empleat de l'empresa que vol donar de baixa i pitja un botó per esborrar-lo. Després de la confirmació que sol·licita el sistema, s'esborra l'empleat. Les accions de manteniment enregistrades en la base de dades que hagin estat efectuades per l'empleat, veuran actualitzada la clau forana que el referenciava a un valor nul.

Alternatives de procés i excepcions: Si hi ha algun error durant el procés cal informar-ne a l'usuari.

### **3.2.11 - Cas d'ús número 11: "Modificar empleat"**

Resum de la funcionalitat: Modifica les dades de l'empleat.

Paper dins el treball de l'usuari: S'empra ocasionalment, sobretot per canviar el càrrec de l'empleat.

Actors: **Director General**.

Casos d'ús relacionats: Identificació.

Precondició: L'empleat ja es troba dins la BD.

Postcondició: Les dades referents a l'empleat s'actualitzen a la BD del sistema.

L'usuari escull l'empleat teclejant el seu nom i els dos cognoms. Tot seguit es pitja un botó i es visualitzen les dades d'aquest empleat en un nou formulari, sempre i quan no s'hagi produït cap error. En aquesta pantalla, és possible modificar la informació relativa a aquest empleat. L'usuari canvia el que creu convenient i actualitza i confirma els canvis prement un botó.

Alternatives de procés i excepcions: En cas de generar-se algun error, s'informa a l'usuari perquè actuï en conseqüència.

### 3.3 Diagrama de casos d'ús

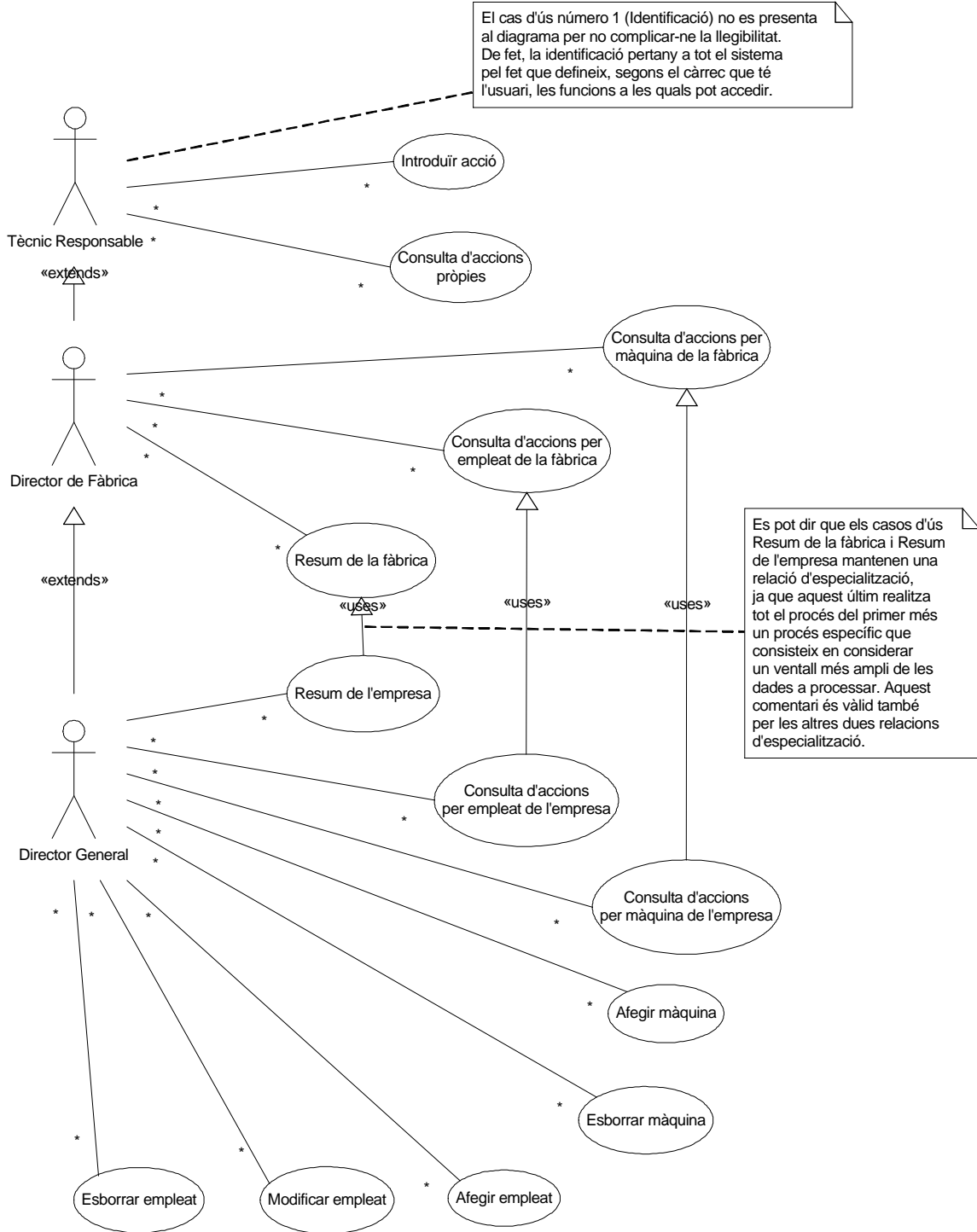


Figura 9 - Diagrama de casos d'ús

Els tres actors que s'han identificat en l'aplicació mantenen una relació d'especialització, com es desprèn del fet que el Director de Fàbrica pot desenvolupar tots els papers del Tècnic Responsable més algun de propi, i de forma anàloga, el Director General pot desenvolupar tots els papers del Director de Fàbrica més algun d'exclusiu d'aquest actor.

En el diagrama anterior no s'hi ha representat el cas d'ús nº1 que correspon a la identificació dels usuaris davant l'aplicació, ja que aquest cas d'ús pertany més aviat al sistema, pel fet que és el que dóna accés a la resta de funcions.

Els casos d'ús nº4, 5 i 6 s'han desdoblats en dos a fi d'assenyalar la diferència de funcions existent quan l'actor primari resulta ser un Director de Fàbrica o un Director General. D'aquesta manera, el cas d'ús nº4 es divideix en Consulta d'accions per màquina de fàbrica i Consulta d'accions per màquina d'empresa, el cas d'ús nº5 es divideix en Consulta d'accions per empleat de fàbrica i Consulta d'accions per empleat d'empresa i el cas d'ús nº6 es divideix en Resum de fàbrica i Resum d'empresa.

En conseqüència, la Consulta d'accions per màquina de l'empresa manté una relació d'especialització amb la Consulta d'accions per màquina de la fàbrica, ja que realitza tot aquest procés sencer més algun d'específic que consisteix en considerar un ventall més ampli de les dades residents a la base de dades del sistema (anàlogament amb les altres dues relacions d'especialització entre casos d'ús).

### 3.4 Identificació de les classes d'entitats

La descripció de la funcionalitat de l'aplicació i del model del domini en permet deduir les classes d'entitats, les quals presentem a continuació en forma de diagrama:

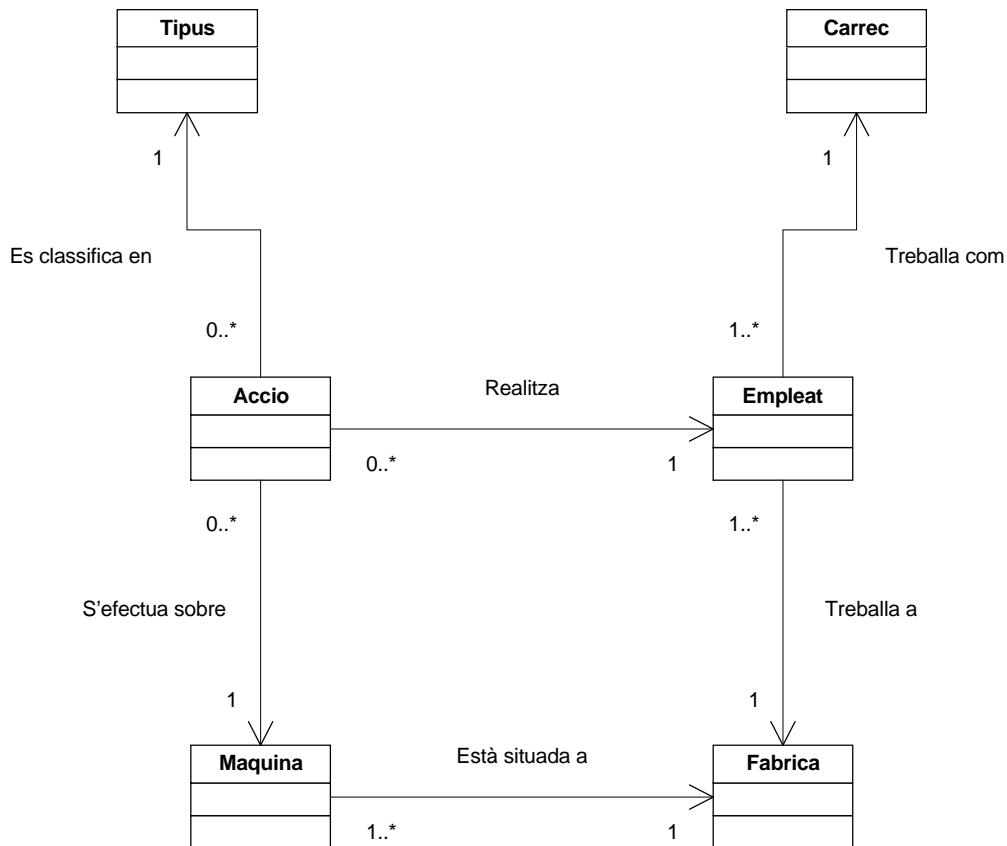


Figura 10 - Diagrama de classes d'entitat

Com es pot observar, s'han identificat sis classes d'entitats bàsiques: Tipus, Carrec, Accio, Empleat, Maquina i Fabrica. Les classes d'entitat constitueixen el pilar bàsic de tota aplicació, donat que són persistents, és a dir, requereixen un sistema d'emmagatzematge permanent perquè les dades siguin accessibles més enllà de la vida d'execució del programari. Tornarem a fer incís en aquests aspectes en el capítol de disseny.

Tot seguit especificarem els atributs d'aquestes classes d'entitats i comentarem algunes qüestions d'importància:

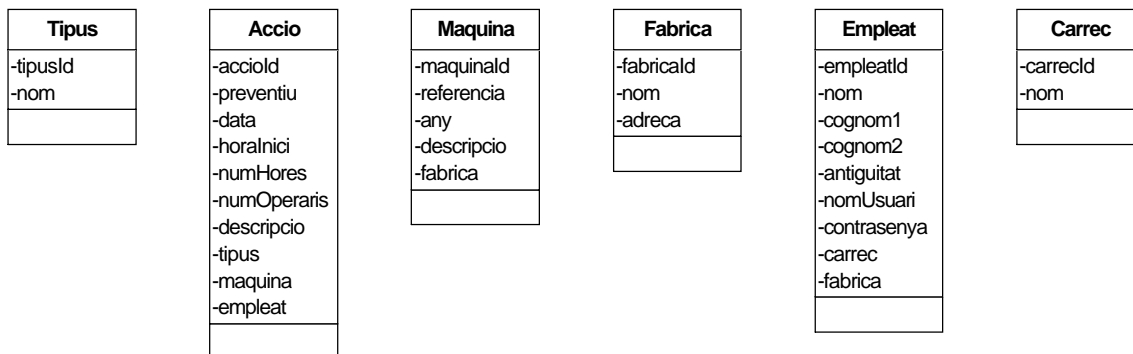


Figura 11 - Atributs de les classes d'entitat

Els atributs de les classes d'entitats són, com es pot comprovar, dades i conceptes que ja han aparegut en la descripció dels casos d'ús. A més a més, s'hi han inclòs uns atributs que identifiquen les classes d'entitat de forma única (identificadors o claus primàries).

Al diagrama s'observa que la classe Tipus només té un atribut -el nom- i podríem pensar que aquesta classe no és realment una classe de l'aplicació sinó un atribut d'Acció. Succeeix, però, que el tipus classifica de forma general l'acció de manteniment i, tot i que no s'implementa a l'aplicació per no complicar-la massa, això permetrà (en una posterior ampliació del sistema) crear-los i esborrar-los des de la mateixa aplicació i sobretot llistar les accions de manteniment segons el tipus de l'acció. Per aquesta raó, s'ha decidit considerar el Tipus com una classe d'entitat a tots els efectes.

Una cosa semblant succeeix amb Càrrec i, anàlogament al cas anterior, també s'ha decidit tractar-la de classe d'entitat. A més, és una pràctica comuna el fet de considerar els atributs amb valors repetitius, com és aquest cas, com una classe a fi de millorar-ne l'accés gràcies a les relacions i evitar confusions.

La figura anterior també especifica les relacions existents entre les diferents classes d'entitats que s'han definit. Aquestes relacions són totes associatives (és a dir, són associacions) i les justifiquem tot seguit:

- Relació "es classifica en" entre Tipus i Acció.  
La cardinalitat és de 1 a N, ja que una acció només pertany a un tipus i d'un tipus donat hi pot haver qualsevol nombre d'accions.
- Relació "treballa com" entre Empleat i Càrrec.  
La cardinalitat és de 1 a N perquè un empleat només pot treballar amb un càrrec i un càrrec pot ésser ostentat per un o diversos empleats.

- Relació "realitza" entre Acció i Empleat.  
Caracteritza el subjecte responsable de l'acció i té cardinalitat 1 a N. Només hi pot haver un empleat responsable d'una acció donada mentre que un empleat pot realitzar qualsevol nombre d'accions.
- Relació "s'efectua sobre" entre Acció i Màquina.  
Caracteritza l'objecte de l'acció de manteniment i la cardinalitat és 1 a N. Una acció donada només es realitza sobre una màquina, mentre que una màquina pot haver sofert qualsevol nombre d'accions.
- Relació "està situada a" entre Màquina i Fàbrica.  
Indica l'ubicació física de la màquina i la cardinalitat és 1 a N: una fàbrica té moltes màquines al seu interior, mentre que una màquina només pot estar a un sol lloc.
- Relació "Treballa a" entre Empleat i Fàbrica.  
Indica la fàbrica on té el lloc de treball l'empleat en qüestió. Cardinalitat 1 a N: una fàbrica té molts treballadors i un empleat només pot treballar per una fàbrica.

No s'han definit agregacions.

Tampoc hem especificat jerarquies d'herència ja que no resulten necessàries. Es pot pensar que és possible fer una jerarquia d'herència entre empleat i els diferents càrrecs que pot ostentar, però es desestima la idea perquè resulta interessant, tot i que no està implementat, que des de la mateixa aplicació es puguin definir nous càrrecs.



### 3.5 La interfície d'usuari

Ja hem mencionat les característiques bàsiques que haurà de tenir l'interfície d'aquesta aplicació: senzillesa, homogeneïtat i intuïció. Conseqüentment:

- S'ha decidit utilitzar una barra de navegació per les diferents funcions que l'usuari pot accedir segons el seu càrrec. Aquesta barra de navegació es situa en el marge esquerre, de manera que es separa clarament del contingut de la pàgina i dóna una visió de l'aplicació neta i efectiva.
- S'ha prescindit de qualsevol animació supèrflua a fi de no distreure l'usuari.
- S'ha prescindit d'indicacions sonores, doncs l'entorn físic en què s'executarà l'aplicació és un entorn sorollós.
- Totes les funcions tenen un títol descriptiu al començament de la pàgina que permet situar-se ràpidament en el mapa de la web.
- Els formularis tenen una aparença molt semblant els uns als altres i els controls també són els mateixos. Això afavoreix la homogeneïtat de l'aplicació i evita confusions.
- Les consultes es presenten en taules que faciliten la lectura i la comprensió de les dades.

Aquestes característiques encaixen en l'entorn industrial que emmarca el programari, com podem comprovar en les següents figures (algunes situacions no es presenten degut a la semblança entre elles):

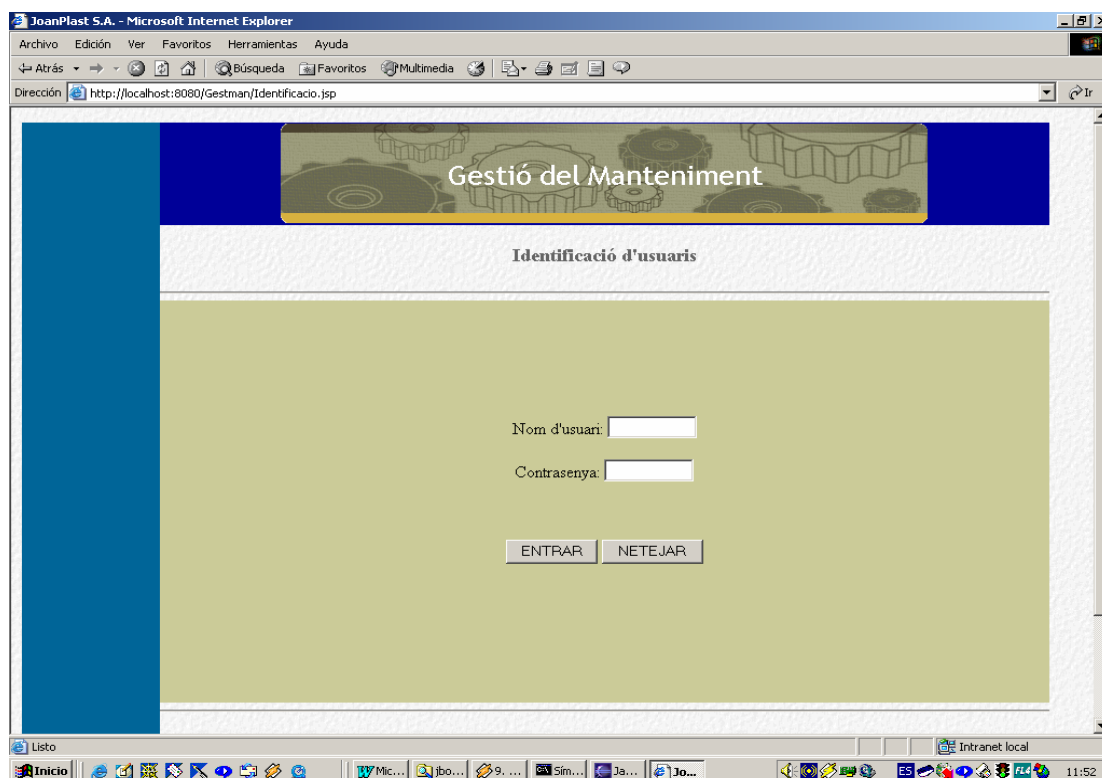


Figura 12 - Pàgina d'identificació



Figura 13 - Menú del Tècnic Responsable

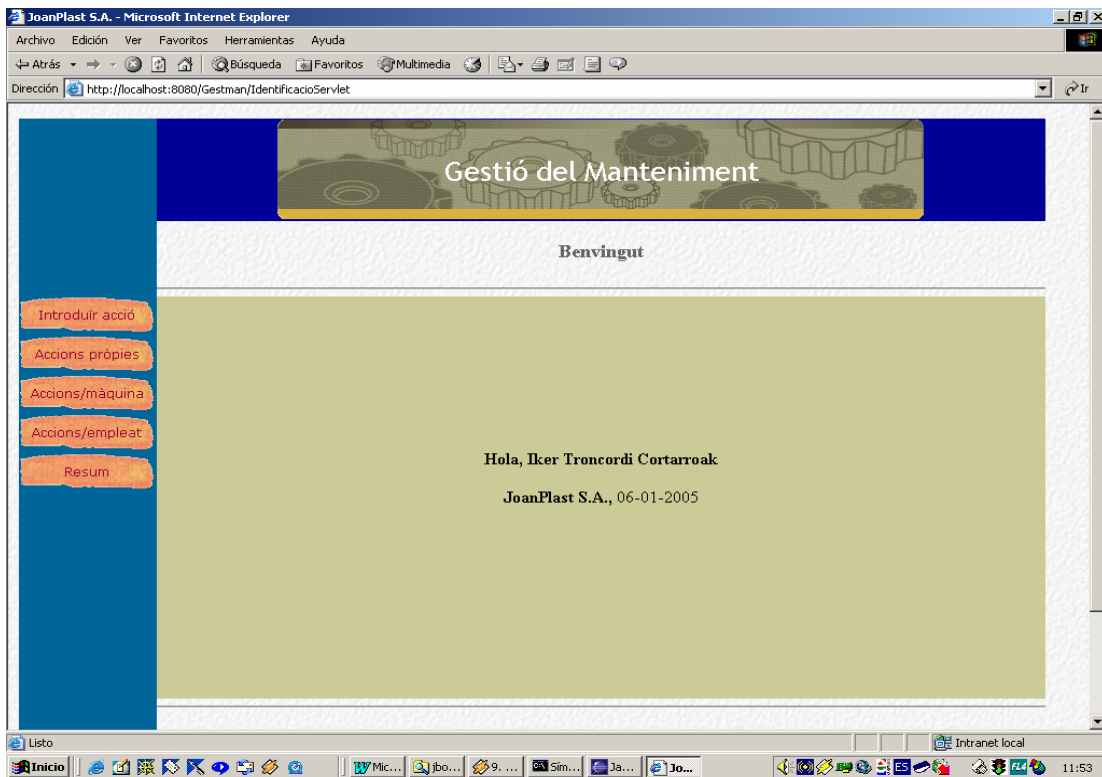


Figura 14 - Menú del Director de Fàbrica

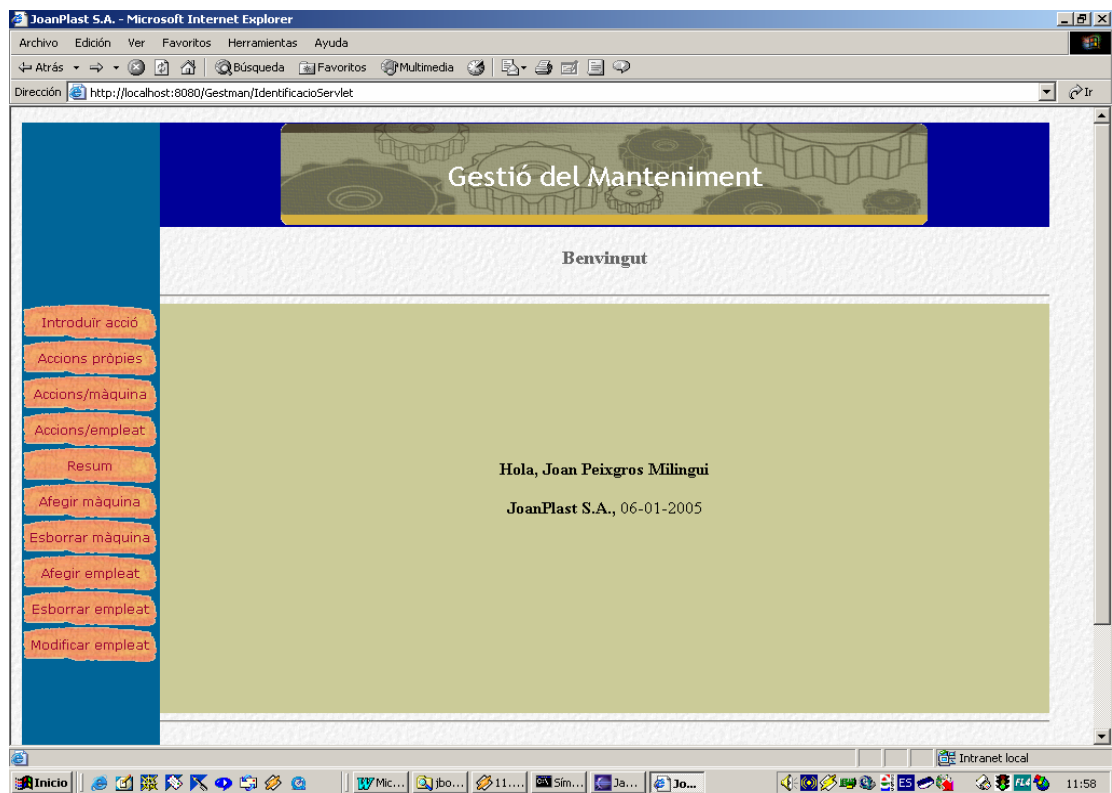


Figura 15 - Menú del Director General

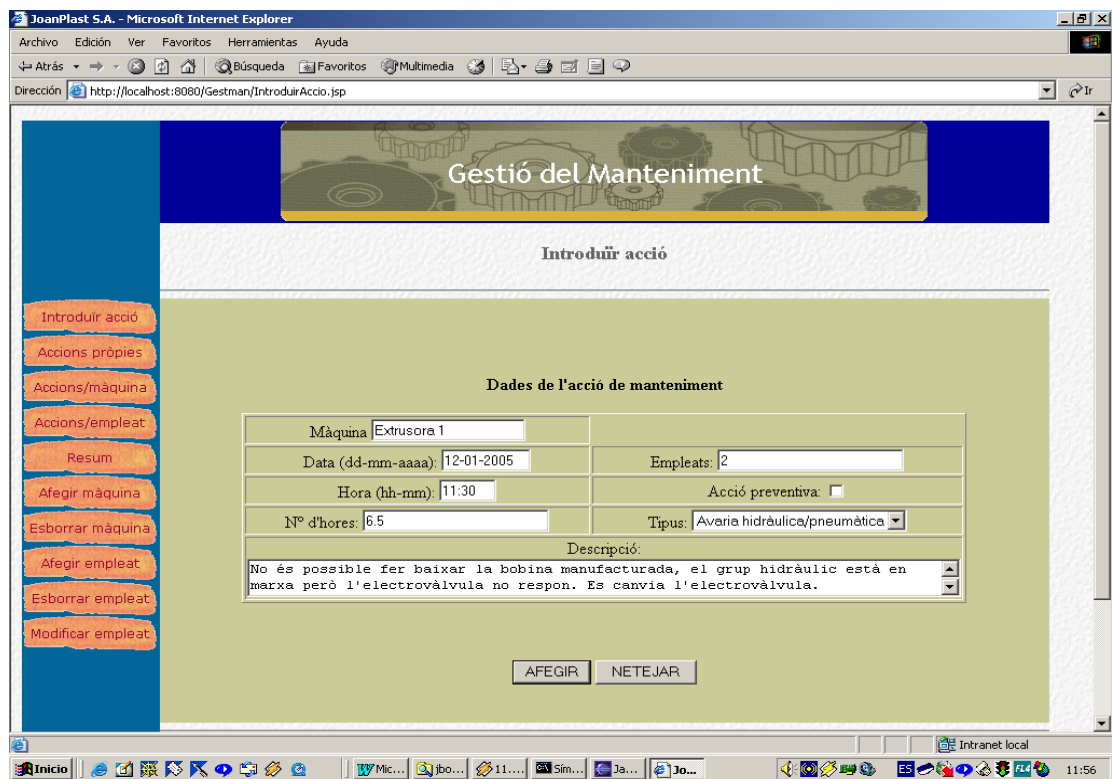


Figura 16 - Introducció d'accions

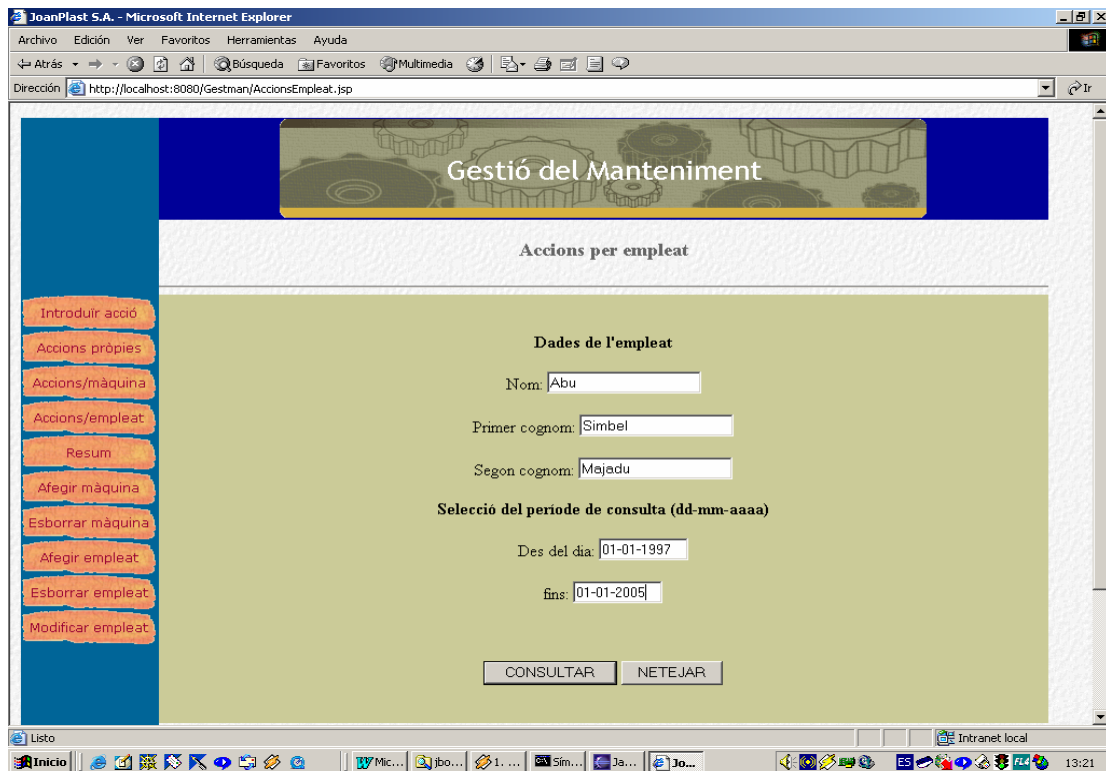


Figura 17 - Consulta d'accions per empleat

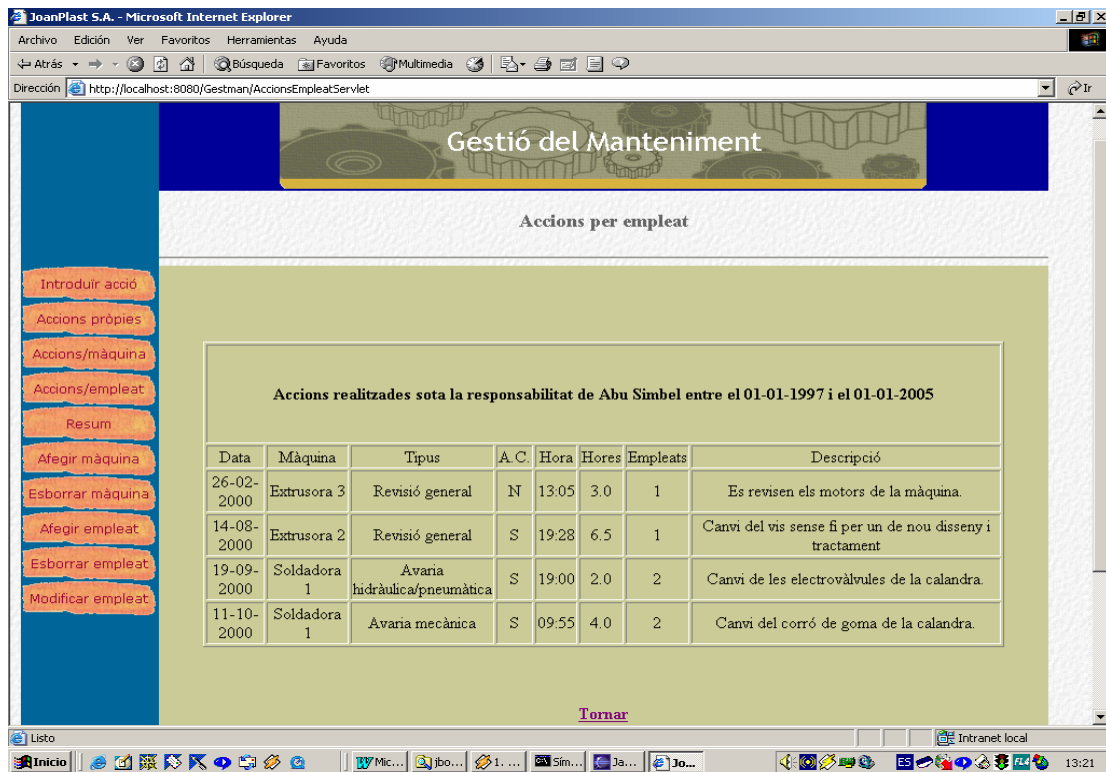


Figura 18 - Resultat de consulta d'accions per empleat

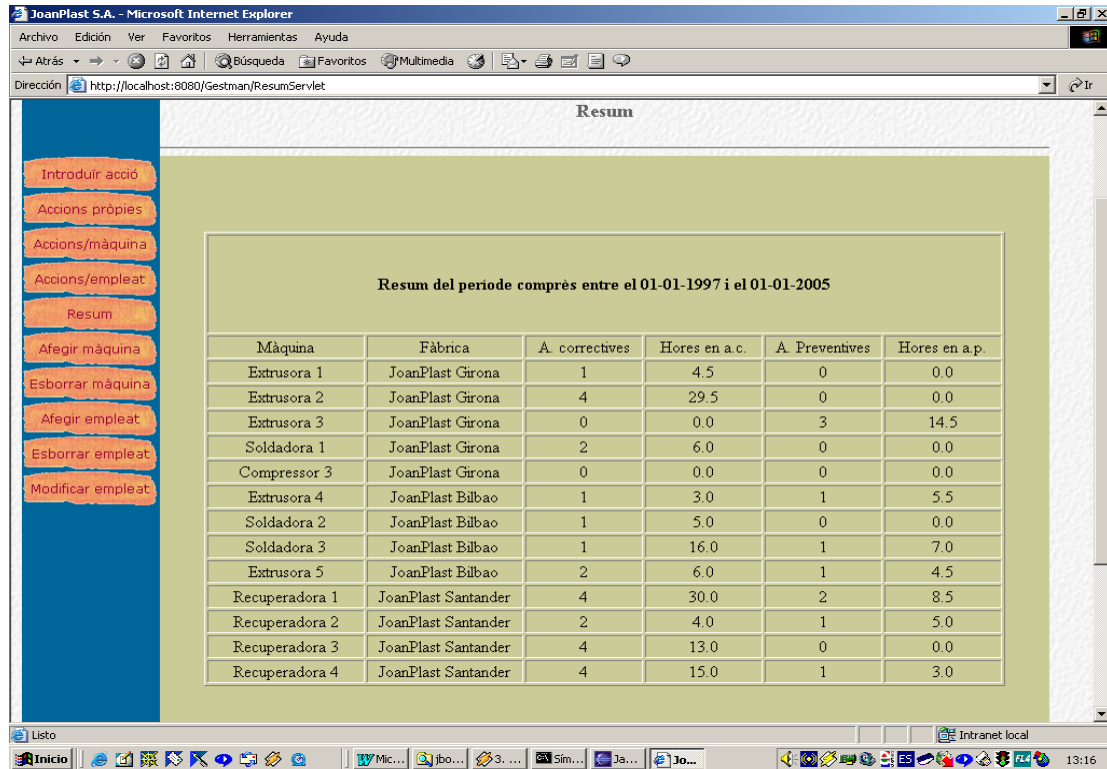


Figura 19 - Resum

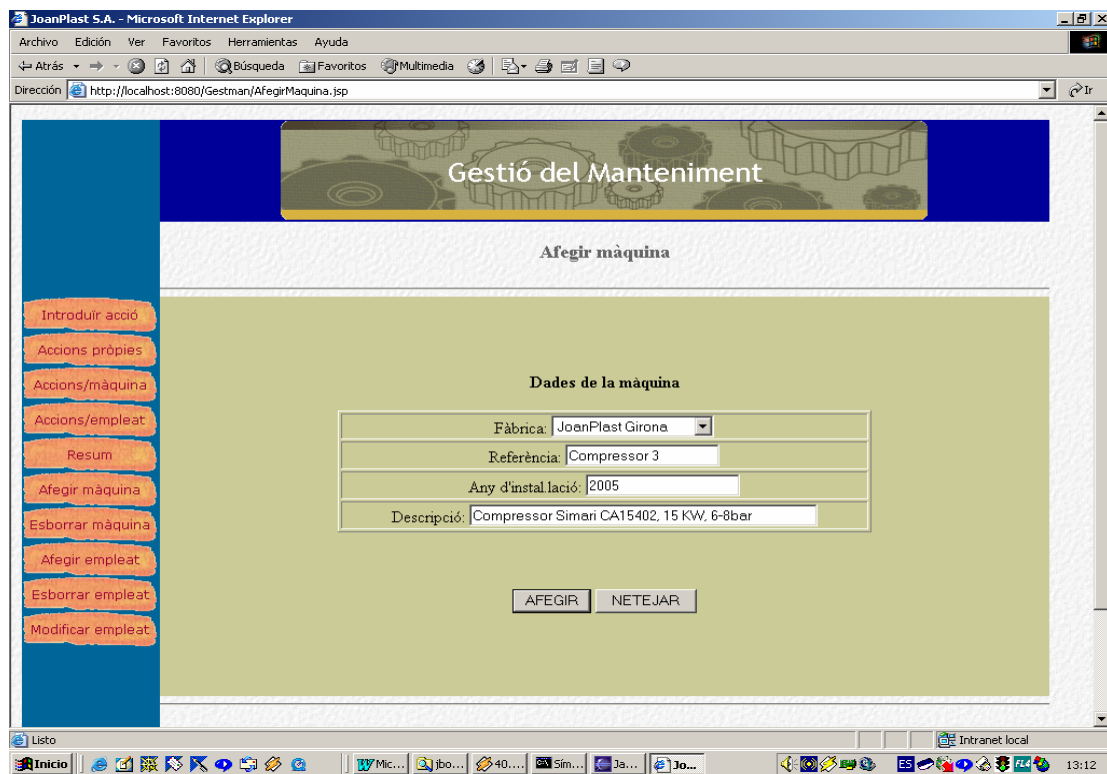


Figura 20 - Afegir màquina

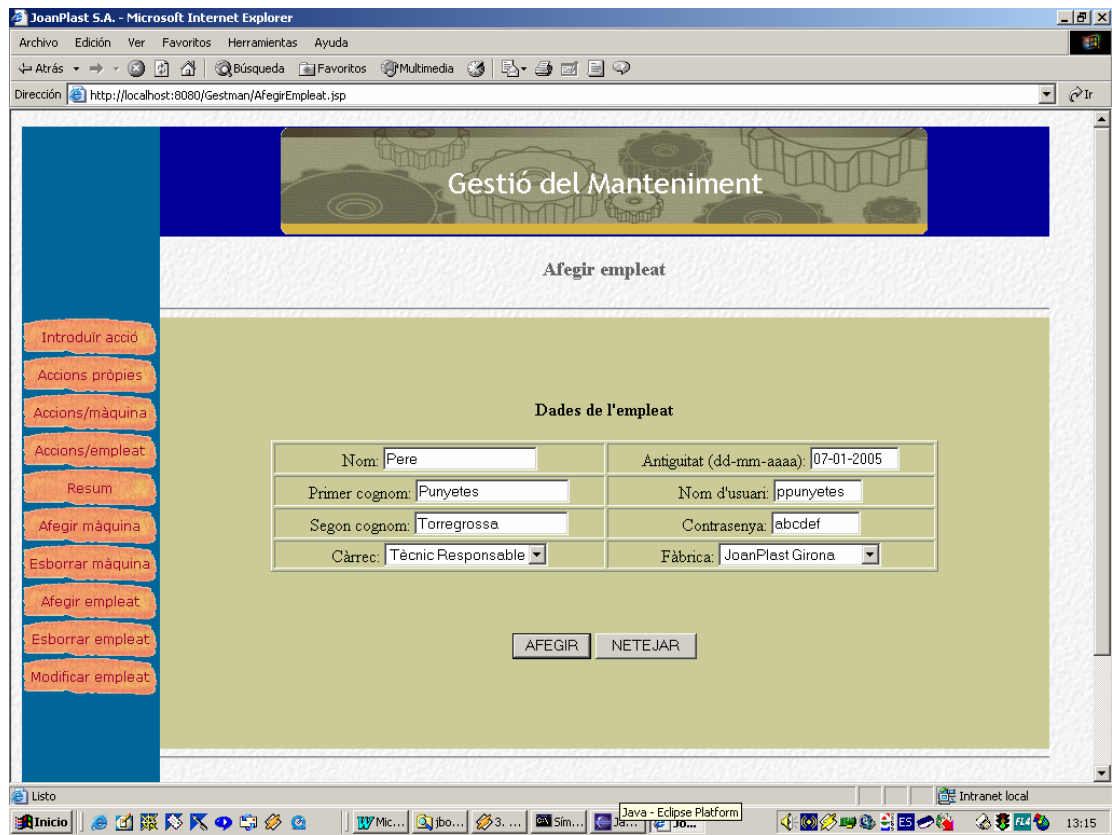


Figura 21 - Afegir empleat

## Capítol 4: Disseny

### 4.1 Reutilització

La reutilització és un dels pilars sobre els que es fonamenta la programació orientada a objectes i, lògicament, l'entorn J2EE segueix fidelment aquest precepte.

Els avantatges que implica la reutilització són molts i diversos: s'escurça el desenvolupament del programari, disminueixen les tasques de manteniment, millora la fiabilitat, es guanya en coherència, etc.

La reutilització es pot dur a terme de quatre formes: reutilització de classes, de components, patrons i bastiments. En la nostra aplicació, la reutilització apareix bàsicament en forma de patrons de disseny.

Un patró és una idea de disseny i implementació detallada que constitueix un esbós de la solució d'un problema que es presenta amb una certa freqüència.

Durant el desenvolupament de l'aplicació de Gestió del Manteniment, s'ha cregut oportú utilitzar les solucions d'alguns d'aquests patrons per fer més flexible i efectiu el codi generat. A continuació es descriuen els patrons utilitzats:

Nom: Model-Vista-Controlador

**Context:** Resulta molt interessant crear una separació de rols entre els diferents creadors d'una aplicació. A més, hi ha aplicacions que requereixen una interfície d'usuari flexible.

**Problema:** En aplicacions amb sistemes múltiples i sincronitzats de dades, l'esforç i la complexitat de la programació de les seves diferents vessants (interfície, model de dades, etc) és molt elevat i és aconsellable repartir-ne la implementació en diferents especialistes perquè cadascun pugui aportar al projecte la seva experiència en la vessant que li és encomanada.

**Solució:** Dividir l'aplicació en tres components separats i diferenciats que es podran implementar independentment els uns dels altres: el model, la vista i el controlador. El model és l'objecte que representa les dades del programa: manega les dades i en controla les transformacions. La vista és l'objecte que gestiona la presentació visual de les dades representades pel model. El controlador és l'objecte que proporciona significat a les ordres de l'usuari, actuant sobre les dades representades pel model i realitzant les peticions que calgui al model o a les vistes.

L'estructura d'aquest patró enfront l'aplicació que estem desenvolupant, pot representar-se de la manera següent:

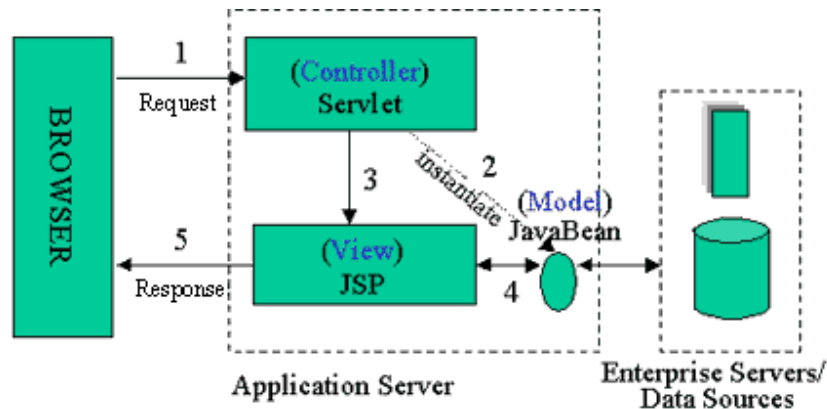


Figura 22 - Model-Vista-Controlador

En l'aplicació que ens duem entre les mans, considerarem les següents equivalències:

- El model estarà format pels Enterprise JavaBeans (tant d'entitat com de sessió).
- Les vistes es gestionaran mitjançant la tecnologia Java Server Pages.
- Els controladors que s'encarregaran de gestionar les peticions dels usuaris seran Servlets.

Nom: Session Facade.

Context: Els EJB's encapsulen lògica i dades de negoci i exposen les seves interfícies, i amb elles la complexitat dels serveis distribuïts, a la capa client.

Problema: Els clients han de cridar molts mètodes dels EJB's per a implementar els casos d'ús d'una aplicació. Això comporta els següents problemes:

- Fort acoplament, que provoca la dependència directa entre els clients i els objectes de negoci.
- Un nombre massa elevat de crides a mètodes entre el client i el servidor, cosa que provoca una disminució del rendiment de la xarxa.
- Falta d'una estratègia d'accés uniforme dels clients, exposant els objectes de negoci a un mal ús.

Solució: Utilitzar un bean de sessió com una façana a fi d'encapsular la complexitat de les interaccions entre els objectes de negoci participants en un fluxe de treball. El Session Facade manega els objectes de negoci i proporciona un servei d'accés uniforme als clients.

Resulta interessant aplicar aquest patró mitjançant beans de sessió sense estat i intentar agrupar casos d'ús similars en diversos beans de sessió a fi d'evitar un Session Facade excessivament extens i complexe.

Estructuralment, el patró Session Facade presenta el següent aspecte:



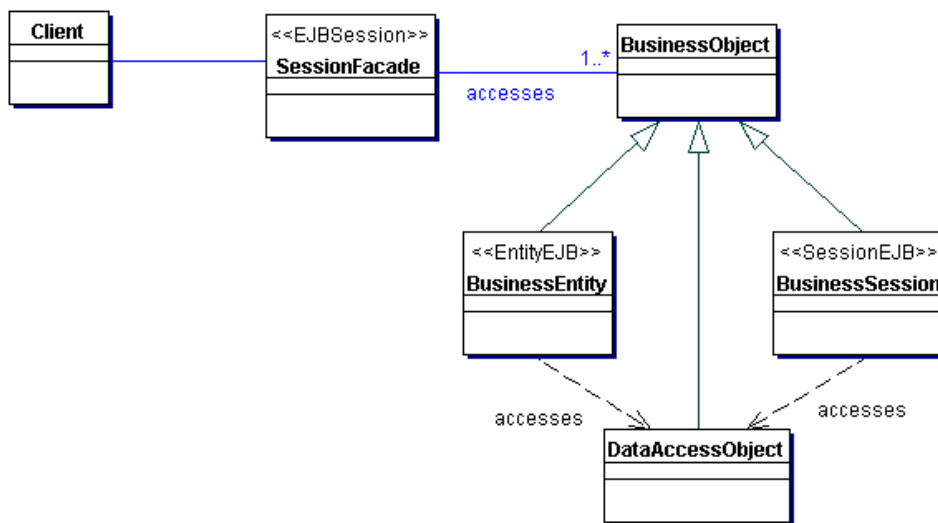


Figura 23 - Session Façade

En l'aplicació que estem considerant implementarem tres beans de sessió com a Session Facade, els quals contindran la interfície pels clients dels casos d'ús relatiu a cadascun dels actors identificats de l'aplicació.

Això, per tant, generarà tres noves classes: TecnicFacade, DFabricaFacade i DGeneralFacade. Com hem esmentat, cada façana contindrà els mètodes adequats per dur a terme els casos d'ús relatiu a l'actor en qüestió. A més a més, les façanes corresponents a actors que també poden accedir a la funcionalitat d'altres actors amb menys privilegis, heretaran de les façanes d'aquests últims i gaudiran, doncs, d'aquestes funcions (per exemple, DFabricaFacade heretarà la funcionalitat de TecnicFacade i, al seu torn, DGeneralFacade heretarà la funcionalitat de DFabricaFacade)

Dins aquest marc, serà possible sobrecriure alguns mètodes que tenen diferent comportament segons l'actor, com per exemple el mètode que obté la informació resumida de les màquines de la fàbrica o empresa, segons si l'actor és Director de Fàbrica o Director General. En aquest sentit, l'ús de l'herència i el polimorfisme, pilars importantíssims dins del paradigma de l'orientació a objectes, proporciona un mecanisme eficient i poderós per a solucionar aquests problemes.

Nom: Domain Value Object.

Context: Les aplicacions client necessiten comunicar, transportar i intercanviar dades de negoci amb els EJB's i altres objectes.

Problema: Cada crida als mètodes set i get d'un objecte remot implica una invocació remota, cosa que afecta directament a l'escalabilitat d'una aplicació (masses crides que poden afectar també el rendiment de la xarxa). A més, el pas dels atributs d'un objecte del domini a un mètode pot conduir a mètodes amb molts arguments, i cada modificació d'un atribut de l'objecte del domini afectarà la signatura de tots els mètodes que utilitzin aquest objecte, amb la qual cosa es creen problemes d'acoplament i mantenibilitat.

Solució: Es pot utilitzar una classe que implementi la interfície Serializable, amb tots els atributs de la classe d'entitat que es pretén encapsular, de manera que serà aquest objecte el que s'usarà per les lectures i escriptures remotes, com es desprèn de la següent il·lustració:

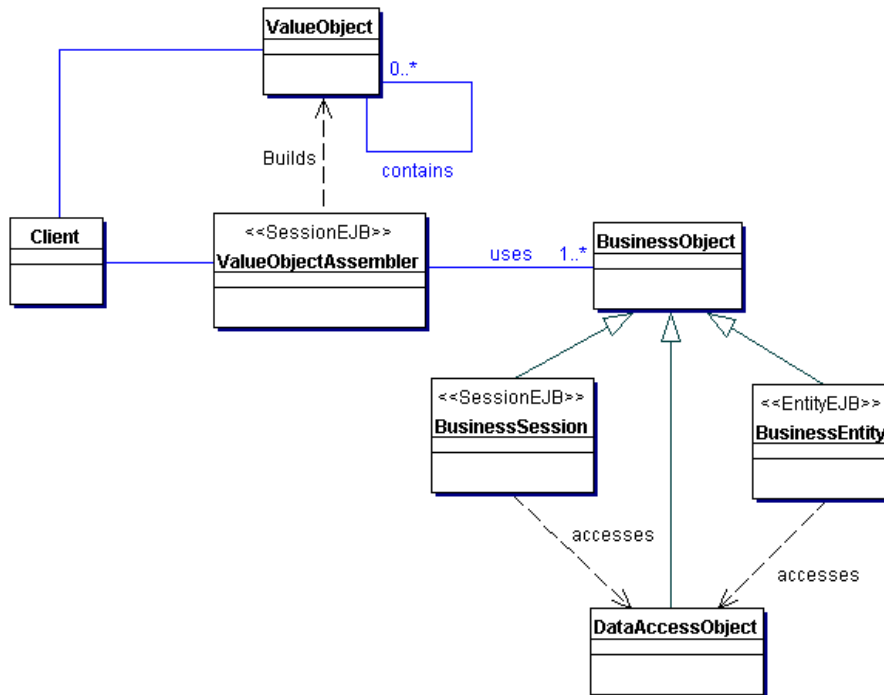


Figura 24 - Domain Value Object

Donat que hem identificat sis classes d'entitat, l'aplicació d'aquest patró generarà sis noves classes: TipusVO, CarrecVO, FabricaVO, MaquinaVO, EmpleatVO i AccioVO. Això implica que les crides a mètodes que accedeixin a informació sobre les classes d'entitat, ja sigui de lectura o d'escriptura, tindran com a retorn o paràmetre respectivament un objecte d'aquest tipus.

Nom: Composite View.

Context: Les pàgines web amb un mínim de sofisticació presenten contingut de diverses fonts de dades, utilitzant diverses subvistes que completen una sola pàgina.

Problema: En lloc de proporcionar un mecanisme per a combinar modularment les subvistes, les pàgines es construeixen introduint codi relatiu al format del contingut directament a cada vista. La modificació de la distribució de múltiples vies és difícil i propensa a errors, degut a la duplicació de codi.

Solució: Utilitzar vies compostes que es componen de diverses subvistes atòmiques. Cada component de la plantilla es pot incloure dinàmicament dins el total i la distribució de la pàgina es gestiona independentment del contingut. Aquesta solució s'esquematitza tot seguit:

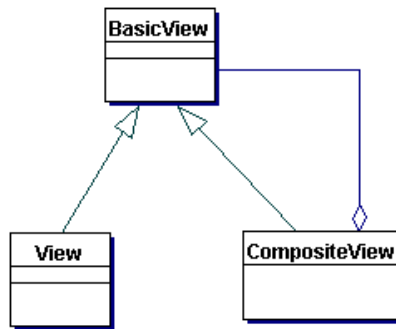


Figura 25 - Composite View

Aquest patró pot implementar-se mitjançant diferents estratègies. De totes elles, hem triat la composició a través de Tags estàndard (sobretot del tipus `<%@ include>`). Aquesta estratègia és molt fàcil d'implementar però no proporciona el poder i la flexibilitat d'altres estratègies com l'ús d'etiquetes personalitzades. No obstant això, hem cregut convenient agilitzar la creació de les pàgines i la senzillesa d'aquest patró ha contribuït a aquesta causa.

L'aplicació, doncs, utilitzarà diverses subvistes (porcions de pàgines jsp) atòmiques que ens serviran com a peces bàsiques per a construir les vistes finals.

Nom: Dispatcher View

Context: El sistema controla el fluxe de l'execució i accedeix al procés de presentació, que és el responsable de general el contingut dinàmic.

Problema: No hi ha un component centralitzat per a gestionar el control d'accés, la recuperació del contingut i la presentació de la vista i hi ha codi de control duplicat per diverses vistes, la qual cosa redueix la modularitat i proporciona una pobre separació dels rols entre els diferents especialistes responsables de l'aplicació.

Solució: Combinar un controlador i un dispatcher amb vistes i helpers per a gestionar les peticions dels clients i preparar una presentació dinàmica com a resposta. Els controladors poden delegar la recuperació del contingut als helpers. El dispatcher és el responsable del control de la vista i la navegació i poden encapsular-se dins un controlador o d'un altre component separat. El funcionament es pot esquematitzar de la següent manera:

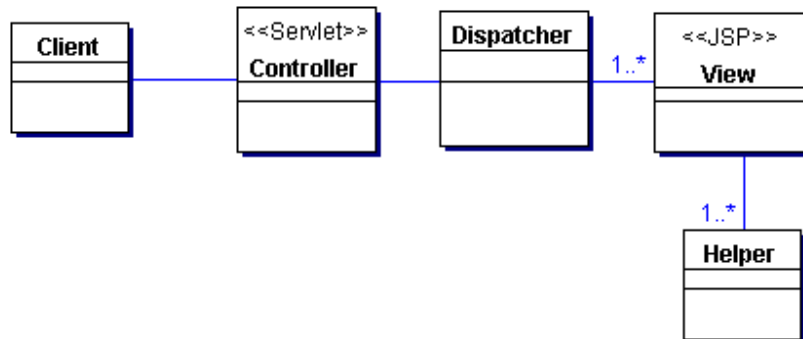


Figura 26 - Dispatcher View

Dins de les possibilitats d'implementació que proporciona aquest patró, hem triat l'opció Dispatcher in Controller, és a dir, que el component dispatcher està integrat dins del controlador (servlet en el nostre cas). Pel que fa als components d'ajuda o helpers, en aquesta aplicació no s'han utilitzat, sinó que s'ha integrat aquesta funció dins els beans de sessió que constitueixen el patró SessionFaçade.

## 4.2 Gestió de la persistència

La gestió de la persistència és molt important en el disseny d'una aplicació. És imprescindible desar la informació manipulada de forma permanent, en cas contrari l'aplicació no tindria sentit. Aquesta informació que cal desar resideix a les classes d'entitat de l'aplicació, ja que una classe d'entitat modela (de fet, n'és una abstracció) un objecte o concepte del món real que hi està implicat de manera que caldrà usar-lo en moltes ocasions. Com hem vist anteriorment, hem definit sis classes d'entitat: tipus, càrrec, acció, empleat, màquina i fàbrica.

Per a realitzar aquesta operació necessitarem un sistema d'emmagatzematge permanent, i en el nostre cas ens decantarem per l'ús d'una base de dades relacional, en concret MySQL. Aquesta decisió s'ha pres considerant la gran presència d'aquest tipus de bases de dades en el món informàtic i les enormes prestacions que proporcionen a l'usuari.

Per utilitzar una base de dades relacional com a sistema d'emmagatzematge permanent caldrà transformar el diagrama estàtic de disseny que considera les classes d'entitat (persistents) en un model relacional.

L'absència d'herència en el nostre diagrama fa que aquest pas sigui encara més senzill. El model relacional obtingut és el següent:

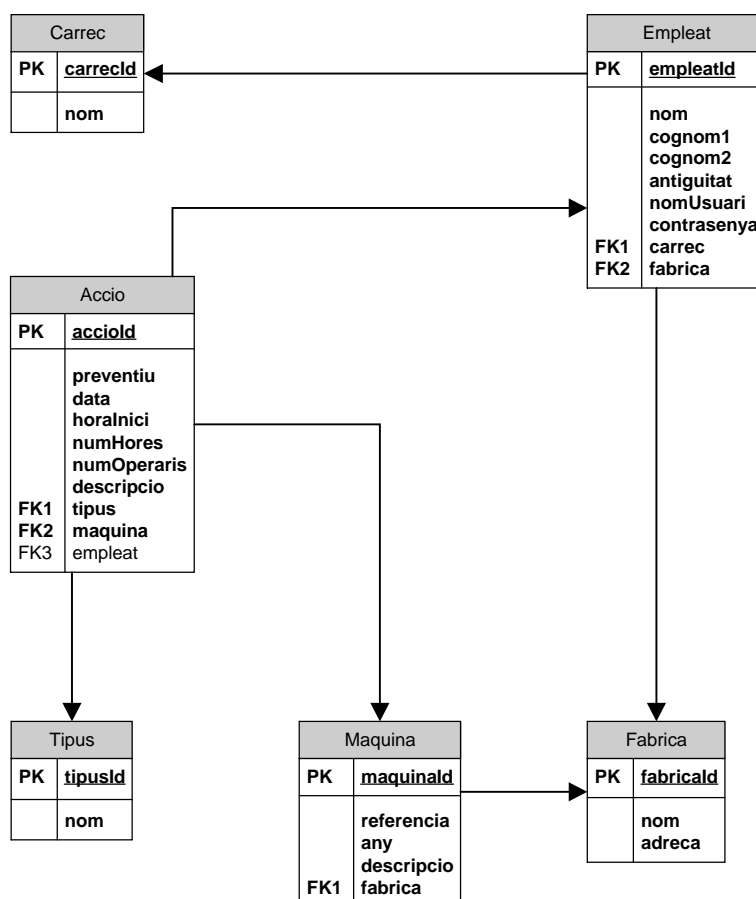


Figura 27 - Model relacional

Les sentències de creació de la base de dades es poden veure en un dels annexos del present document, així com també diverses sentències més dedicades a inserir uns registres inicials a la base de dades i de netejar-la completament.

Un cop hem triat el sistema d'emmagatzematge, caldrà escollir la manera com interacciona l'aplicació amb la base de dades associada. La tecnologia J2EE proposa dues maneres oposades de tractar la persistència: Persistència gestionada pel contenidor (CMP) i Persistència gestionada pel propi bean (BMP). No correspon aquí tractar l'eterna discussió sobre els avantatges i els inconvenients d'utilitzar cadascun dels dos mètodes, així que simplement ens centrarem en la solució proposada en l'aplicació:

Utilitzarem persistència gestionada pel bean. Aquesta solució implica l'escriptura de més quantitat de codi per part del programador però permet optimitzar-lo millor. Caldrà utilitzar, per tant, la interfície JDBC per afavorir la connectivitat dels beans amb la informació present a la base de dades. En aquest sentit, s'ha desenvolupat una classe d'utilitat que gestiona la connexió i la desconexió de la base de dades, així com altres processos útils en la interacció amb la BD. Les transaccions es deixen en mans del contenidor d'EJB's.

### **4.3 El disseny dels casos d'ús**

En les properes pàgines s'anirà explicant, un per un, el disseny dels casos d'ús, i completarem aquesta informació amb diagrames de seqüències, els quals il·lustraran perfectament com es desenvolupa cada procés.

#### **4.3.1 Identificació**

El procés d'identificació és l'entrada de l'usuari al sistema i té com a objectiu validar les dades d'aquest i presentar-li les funcions que li són pròpies en funció del seu càrrec.

L'usuari ha d'entrar el seu nom d'usuari i la contrasenya a l'aplicació mitjançant un senzill formulari. Un cop fet això, el sistema rep les dades i primer accedeix a la Session Façade més senzilla (TècnicFaçade) per contrastar aquestes dades amb la base de dades mitjançant un mètode anomenat VerificaUsuari. Si l'autenticació és correcta, es realitzen les següents operacions:

- Obtenim un vector dels càrrecs enregistrats a la base de dades, un dels tipus i un de les fàbriques i els desarem com a variables de sessió. Això ens permetrà posteriorment (en les vistes) accedir a aquestes dades i presentar-les en els formularis on siguin necessàries en forma de llistes desplegable.
- Associem les dades de l'empleat com a variable de sessió perquè altres processos de la mateixa sessió en puguin fer ús.
- Finalment, realitzem una cerca pels càrrecs de l'aplicació a fi de trobar quin és el que ostenta l'usuari. Un cop trobat, instanciem la Session Façade que correspon a la funcionalitat de l'usuari (si és un Tècnic Responsable això no farà falta perquè ja ho hem fet a l'inici) i inserim tota aquesta informació com a variables de sessió. D'aquesta manera, els controladors podran cridar els mètodes necessaris de la Session Façade sense preocupar-se del càrrec ja que l'autenticació ja s'ha realitzat.

El diagrama de seqüències -simplificat- corresponent a aquest procés es presenta tot seguit:

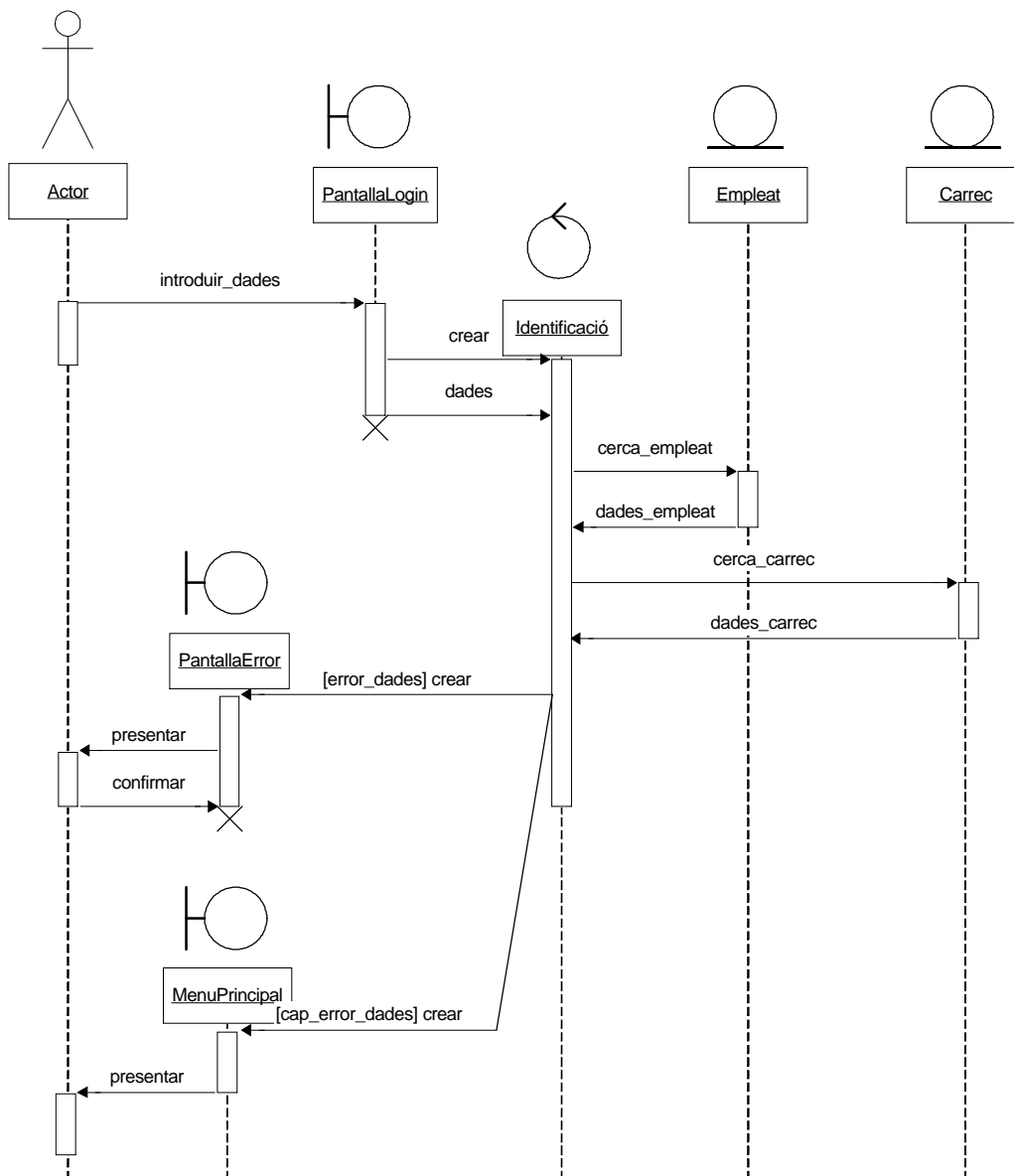


Figura 28 - Diagrama de seqüències del CU n°1

### 4.3.2 Introduir acció

Bàsicament, aquest cas d'ús realitza una validació de les dades inserides i les afegeix a la base de dades si tot és correcte, com podem observar en el diagrama:

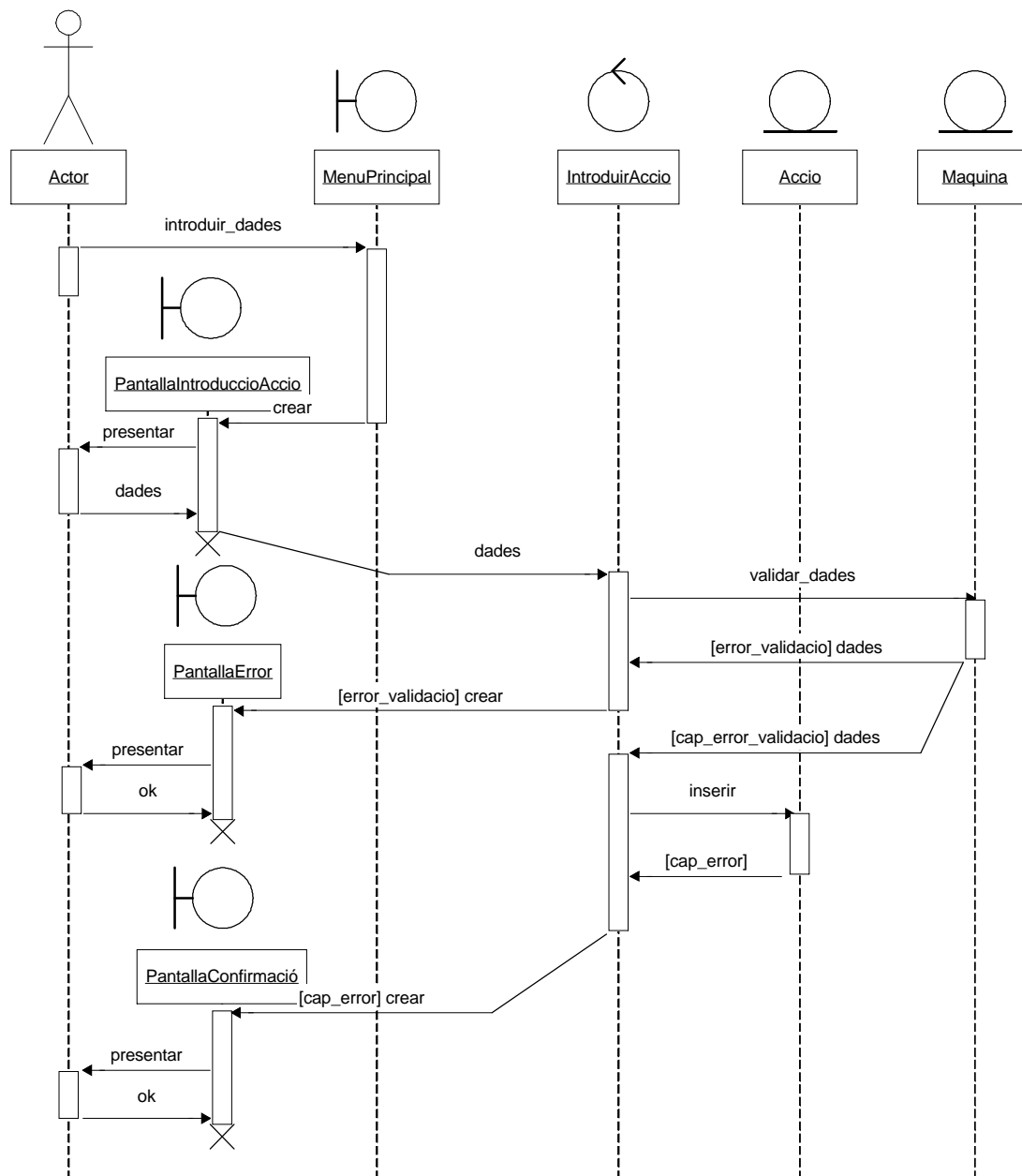


Figura 29 - Diagrama de seqüències del CU n°2

L'aplicació no permet introduir una acció de manteniment d'una màquina que no es troba instal·lada a la mateixa fàbrica que l'usuari per raons evidents. El mètode principal que s'utilitza s'anomena `afegeixAccio` i està definit a `TecnicFacade`. A la façana s'hi accedeix a l'inici del procés. Donat que les façanes dels Directors de Fàbrica i els Directors Generals són subclasses de `TecnicFacade`, el procés es pot realitzar sense complicacions (en aquest cas el resultat és el mateix sigui qui sigui l'usuari).



### 4.3.3 Consulta d'accions pròpies

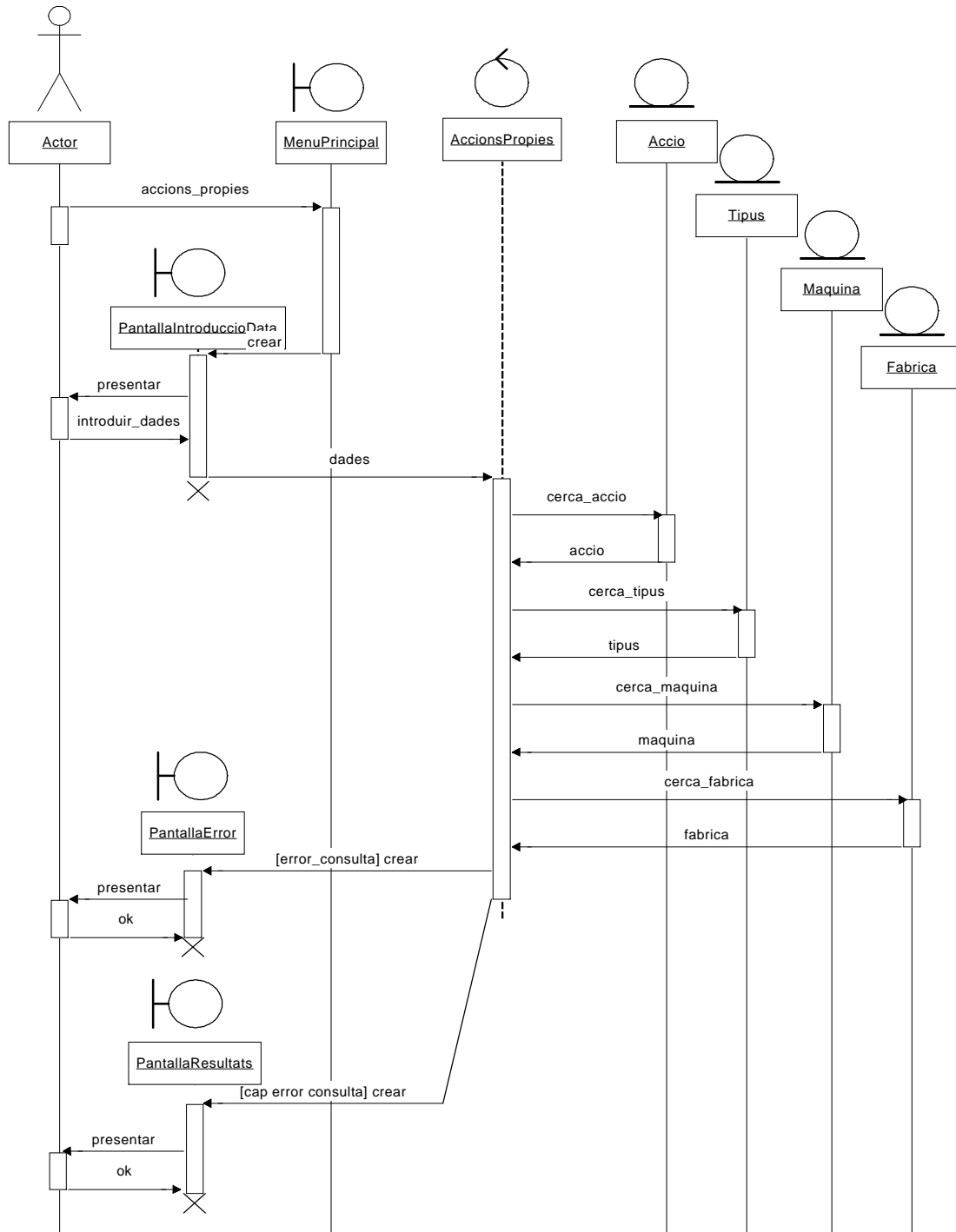


Figura 30 - Diagrama de seqüències del CU n°3

Aquest procés demana i filtra les dades sobre el període a consultar. Si les dades són correctes, crida al mètode `accionsPropies` definit a la façana `TecnicFacade` i passa la informació obtinguda a la vista encarregada de presentar els resultats.

#### **4.3.4 Consulta d'accions per màquina**

Primer de tot, es demanen les dades sobre el període a consultar i la referència de la màquina objecte de la consulta. Un cop tot és correcte es procedeix de la següent forma:

- S'accedeix a la façana de sessió. En aquesta ocasió si que hi ha diferència de funcions segons l'usuari que es tracta.
- Es realitza la consulta a la façana mitjançant el mètode `accionsMaquina` definit a `DFabricaFacade` i sobreescrit (herència + polimorfisme) a `DGeneralFacade`. En cas que l'usuari sigui `Director de Fàbrica` i la màquina consultada no estigui en la seva mateixa fàbrica, la façana retorna un vector nul, cosa que el controlador detecta i informa a l'usuari d'aquesta situació. Si l'usuari és `Director General`, la façana no comprova on pertany la màquina i, per tant, realitza la consulta sense problemes.
- Un cop s'ha obtingut el resultat, es passa el fluxe de l'aplicació a la vista que ha de presentar els resultats.

El diagrama de seqüències d'aquest cas d'ús és força similar a l'anterior, per la qual cosa no l'hem inclòs en aquest document.

#### **4.3.5 Consulta d'accions per empleat**

Anàlogament al cas anterior, es demanen les dades sobre el període a consultar i el nom i els cognoms de l'empleat objecte de la consulta. Un cop tot és correcte es procedeix de la següent forma:

- S'accedeix a la façana de sessió. També aquest cop hi ha diferència de funcions segons l'usuari que es tracta.
- Es realitza la consulta a la façana mitjançant el mètode `accionsEmpleat` definit a `DFabricaFacade` i sobreescrit (herència + polimorfisme) a `DGeneralFacade`. En cas que l'usuari sigui `Director de Fàbrica` i l'empleat consultat no treballi a la seva mateixa fàbrica, la façana retorna un vector nul, cosa que el controlador detecta i informa a l'usuari d'aquesta situació. Si l'usuari és `Director General`, la façana no comprova on treballa l'empleat i, per tant, realitza la consulta sense problemes.
- Un cop s'ha obtingut el resultat, es passa el fluxe de l'aplicació a la vista que ha de presentar els resultats.

Tampoc s'inclou el diagrama de seqüències d'aquest cas d'ús degut a la similitud entre els dos casos anteriors.

### 4.3.6 Resum

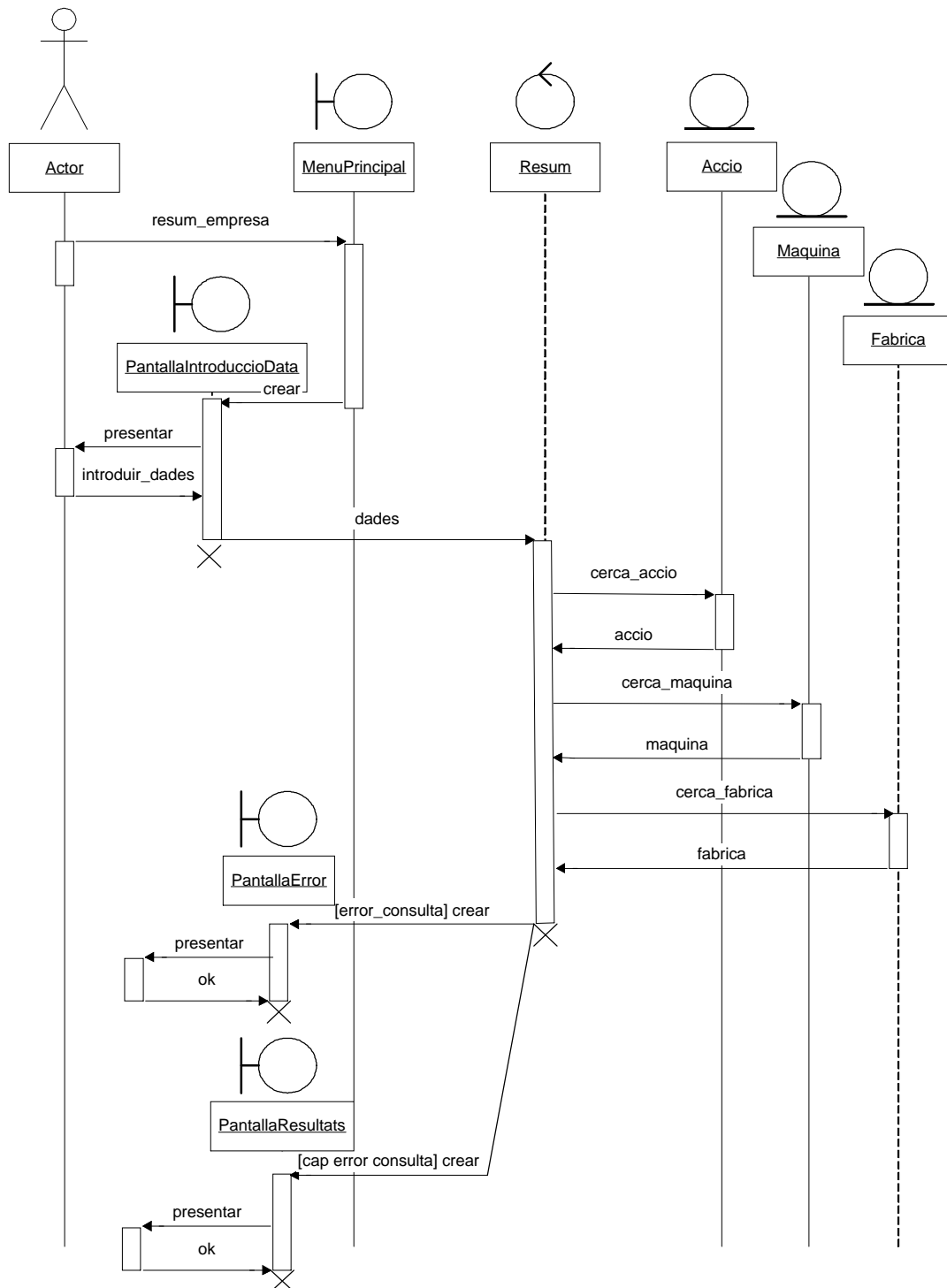


Figura 31 - Diagrama de seqüències del cas d'ús n°6

Aquest cas d'ús també té diferent funció segons l'usuari. És per això que torna a ser important l'ús de l'herència que s'ha fet en el disseny dels beans de sessió. Un Director de Fàbrica rebrà un seguit de dades -encapsulades per major eficiència- que consideren només la fàbrica que regeix mentres que un Director General obtindrà dades globals de l'empresa. El controlador simplement comprova les dades, crida al mètode resumFàbrica de la façana (recordem que la façana s'escriu com a variable de sessió i per tant és accessible des de tots els controladors) i passa el fluxe de control a la vista corresponent.

#### 4.3.7 Afegir màquina

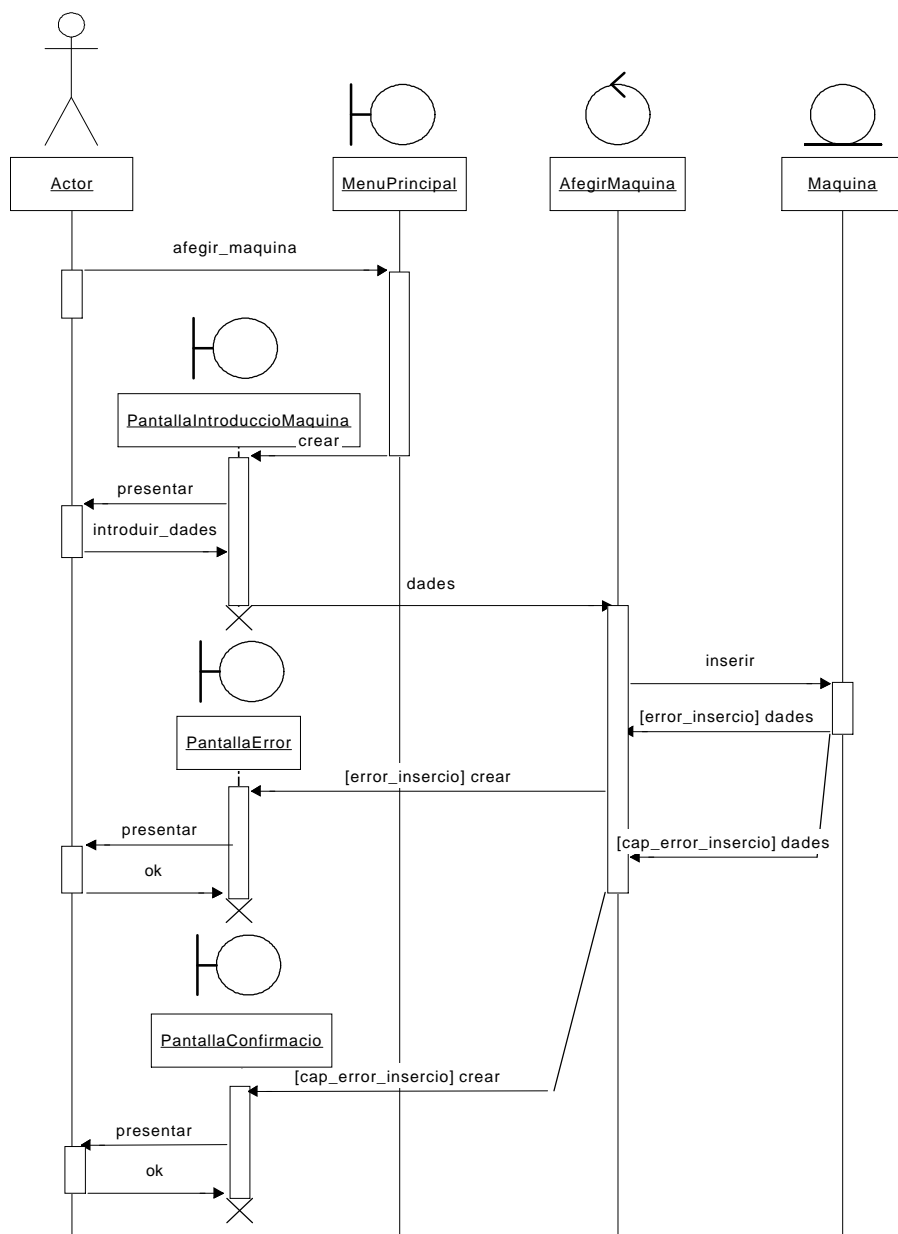


Figura 32 - Diagrama de seqüències del CU n°7

Aquest cas d'ús no té cap complicació especial, i a més, és exclusiu dels usuaris amb rol de Director General:

- S'accedeix a la façana de sessió
- Es comproven les dades
- S'insereix el registre
- S'informa l'usuari que el procés s'ha executat

#### 4.3.8 Esborrar màquina

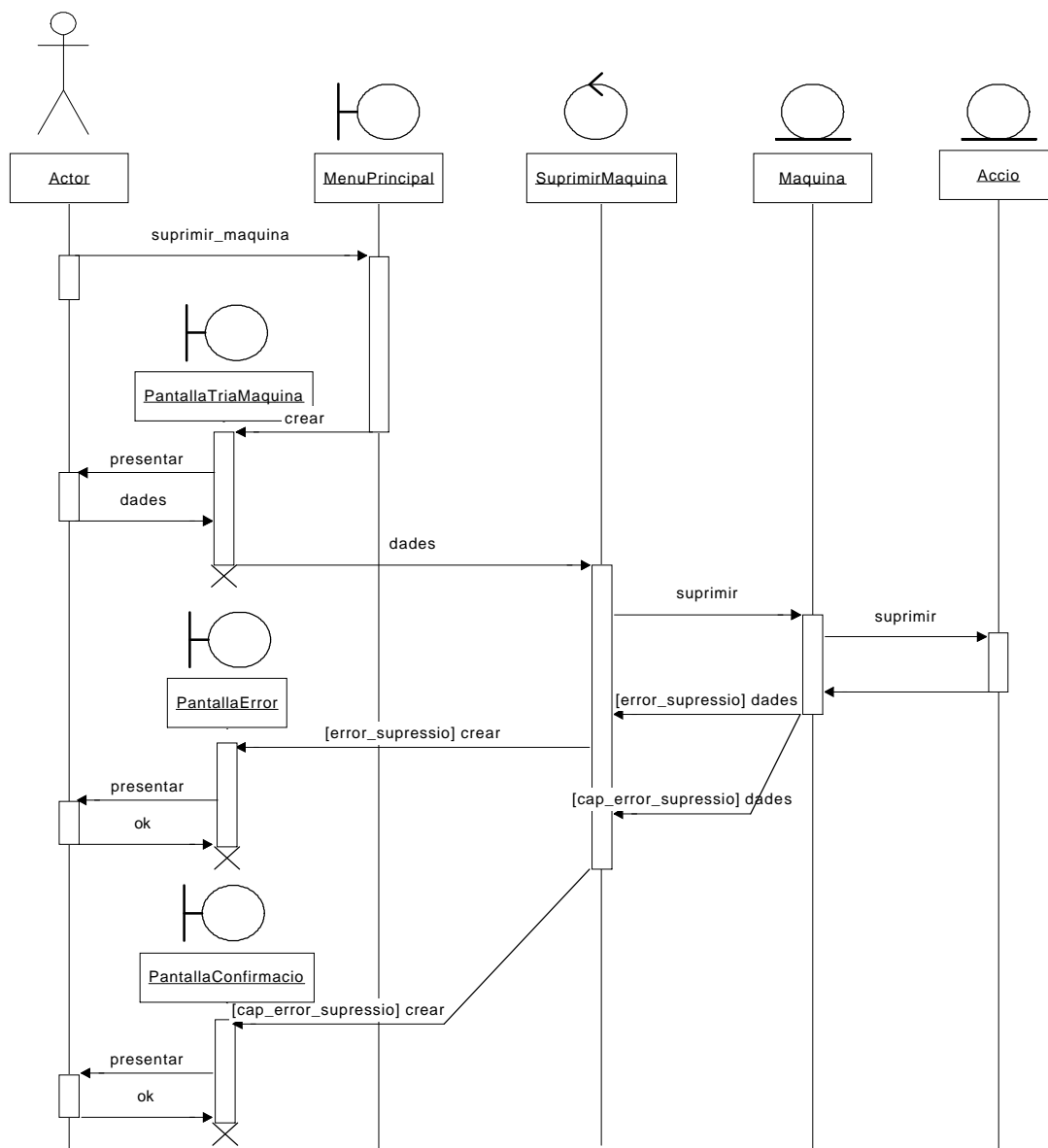


Figura 33 - Diagrama de seqüències del CU n°8

El procés d'esborrar una màquina només el pot realitzar el Director General i consta dels passos següents:

- S'accedeix a la façana de sessió.
- Cridem al mètode de la façana que permet esborrar una màquina (esborrarMaquina, amb la referència d'aquesta com a paràmetre). Aquest mètode invoca el mètode remove de l'ejb d'entitat corresponent i és aquest mètode el que s'encarrega de suprimir totes les accions de manteniment relatives a la màquina que es vol esborrar i, un cop feta aquesta operació, esborra la pròpia màquina de la base de dades.
- Finalment s'informa l'usuari de l'èxit o el fracàs del procés.

#### **4.3.9 Afegir empleat**

Aquest procés és molt semblant al d'afegir màquina, així que ens estalviarem el diagrama de seqüències. Els passos que segueix passen per accedir a la façana de sessió, recuperar les dades del formulari d'inserció, verificar-les, inserir el registre i confirmar l'èxit del procés.

#### **4.3.10 Esborrar empleat**

El procés d'esborrar un empleat implica una mica més de feina. En primer lloc, cal accedir, com de costum, a la façana de sessió (el cas d'ús és exclusiu dels Directors Generals). Després cal recuperar les dades de l'empleat que es vol esborrar provinents del formulari, s'encapsulen i es passen al mètode suprimeixEmpleat de la façana. Aquest mètode cridarà al mètode remove de l'EJB d'entitat. Aquest mètode s'encarrega per si sol d'actualitzar les accions de manteniment que tenien com a responsable l'empleat que es vol esborrar tot escrivint un zero en la clau forana corresponent. Un cop s'ha realitzat aquesta operació, esborrarà l'empleat de la base de dades.

El diagrama de seqüències corresponent a aquest cas d'ús es presenta a la pàgina següent.

#### **4.2.11 Modificar empleat**

La modificació d'un empleat comporta la realització dels següents processos:

- Accés a la façana
- Recuperar les dades del formulari i cercar la resta de les dades de l'empleat a la base de dades
- Presentar totes les dades de l'empleat a fi que puguin ser modificades.
- Recuperar les dades modificades del formulari corresponent i actualitzar-les a la base de dades
- Informar l'usuari.

El diagrama de seqüències es presenta més endavant.

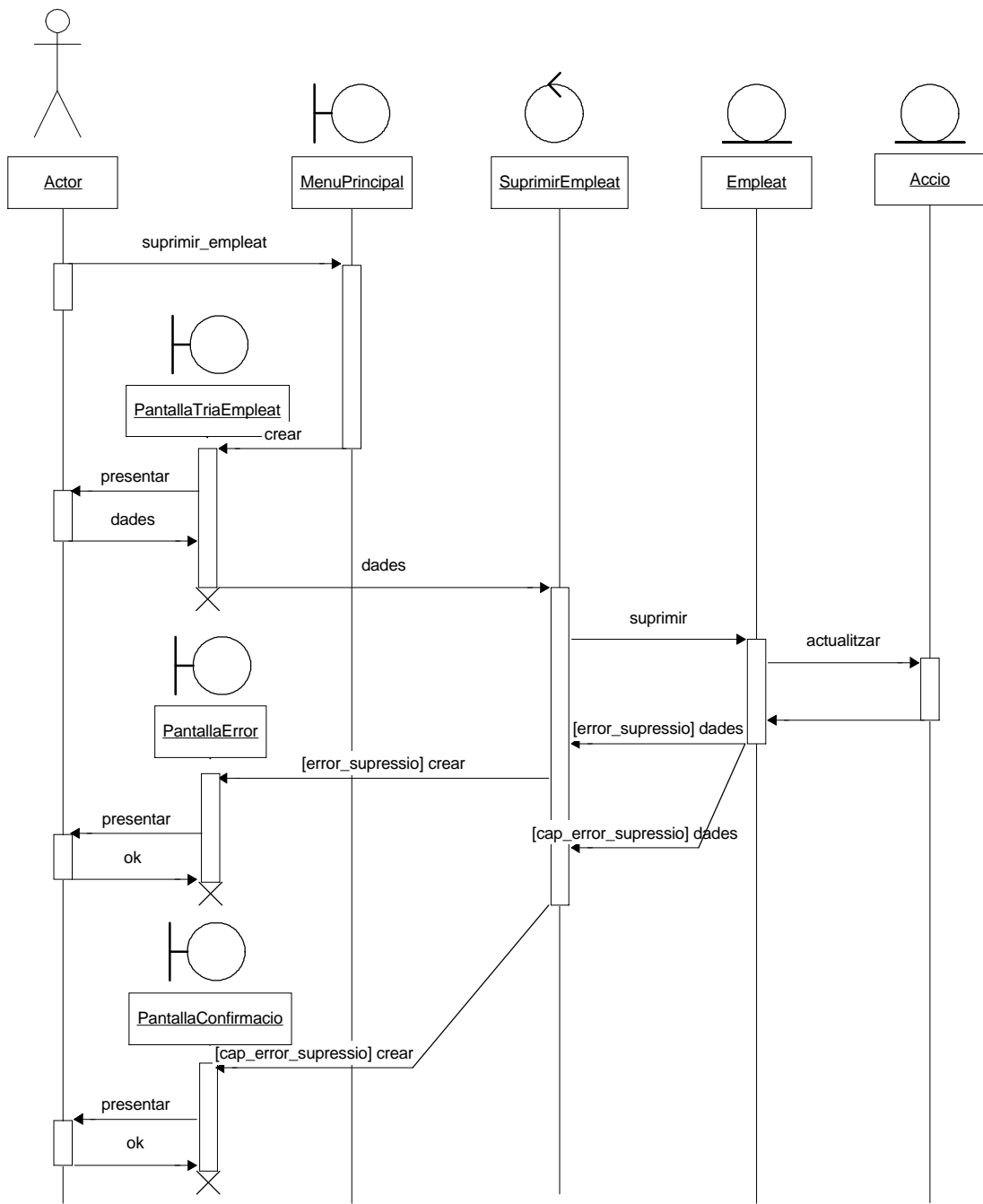


Figura 34 - Diagrama de seqüències del CU n°10

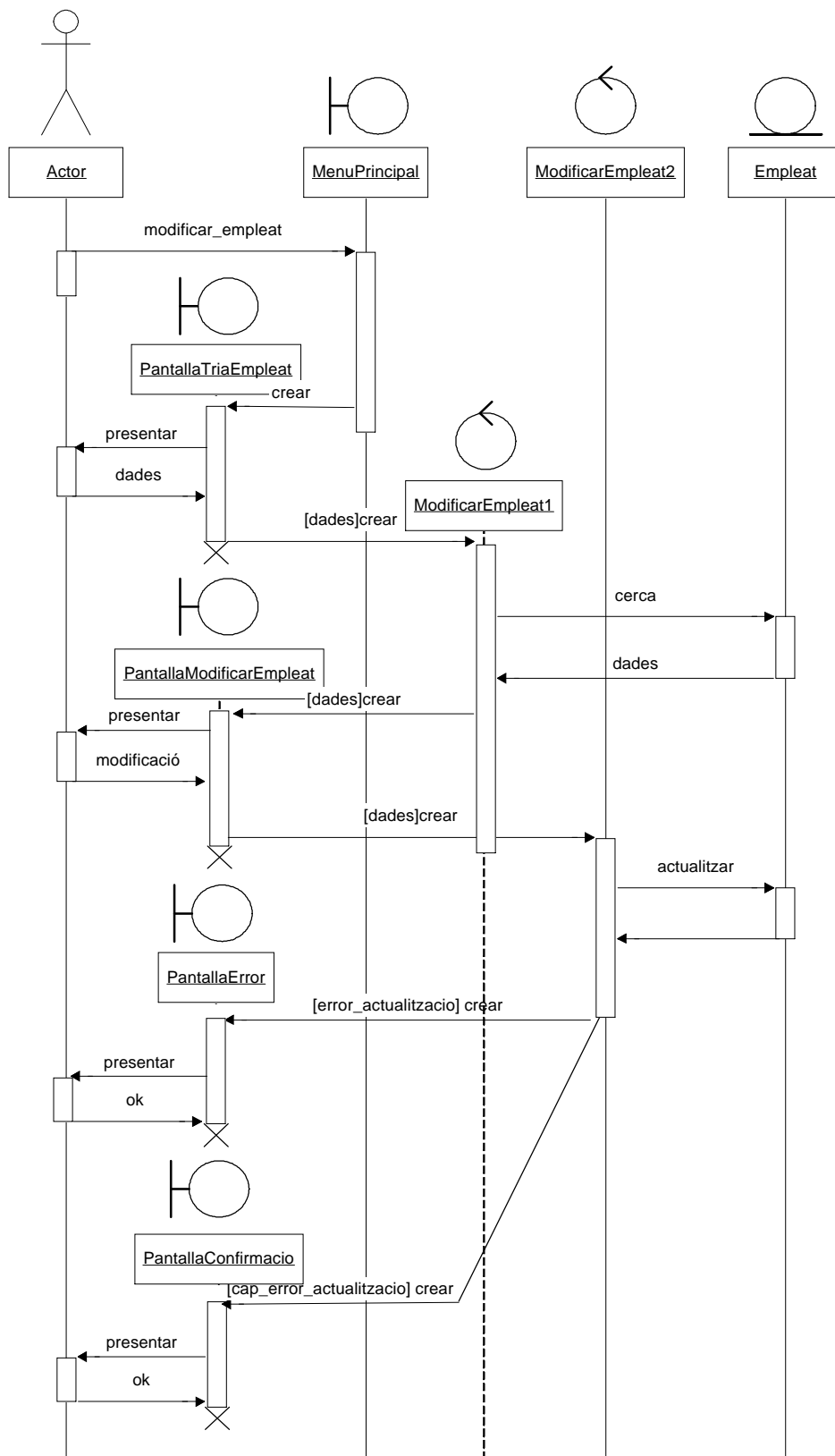


Figura 35 - Diagrama de seqüències del CU n°11



## Conclusions

La realització d'aquest projecte ha suposat un gran repte del qual n'he eixit satisfet: el projecte s'ha acabat en el temps previst i ha acomplert les expectatives sobre el seu ús i la seva funcionalitat. Si bé totes les funcions previstes s'han desenvolupat amb èxit, cal dir que he patit nombrosos imprevistos que han retrassat algunes fases del projecte.

Això ha anat en detriment d'algunes parts d'aquest treball, sobretot pel que fa a la interacció amb l'usuari: falta polir una mica més el filtratge de les dades que s'insereixen mitjançant els formularis de l'aplicació i informar dels errors d'una forma més amigable a l'usuari.

Aquests retrassos s'han concentrat sobretot en dos aspectes: la gestió de la persistència i els servlets. Inicialment havia pensat en utilitzar la persistència gestionada pel contenidor (CMP) a fi d'aprendre'n el seu funcionament, però en veure que la gran quantitat d'informació trobada i de certs problemes amb els fitxers xml em retrassaven massa del pla previst, vaig decidir canviar la persistència perquè fos gestionada pels propis beans, cosa que ja havia vist en les assignatures de bases de dades. El segon retràs important fou a conseqüència de l'aprenentatge dels servlets i dels seus descriptors de desplegament.

No obstant, crec que l'experiència ha estat molt positiva, doncs:

- He après el funcionament de la tecnologia J2EE, de la qual no en sabia res abans de començar. Penso que és una plataforma molt interessant ja que dóna una solució (unes més bones que altres) a cadascun dels aspectes a considerar d'una aplicació. L'enorme informació present sobre el tema ajuda també forjar-se un lloc de renom entre les tecnologies d'aquest tipus.
- He après a desenvolupar aplicacions reals començant des de zero i seguint un cicle de vida del projecte iteratiu i incremental. En aquest sentit, penso que cal posar especial èmfasi en les primeres fases del projecte: recollir i documentar bé el conjunt de funcions que cal implementar i resoldre els dubtes que això comporta amb el client abans de començar a picar el codi proporciona una visió més sòlida del que s'ha de fer i evita molts errors.
- He après a planificar un projecte informàtic en un marge definit de temps, com administrar els recursos disponibles i com superar els imprevistos que, inevitablement, sorgeixen.
- He comprovat els beneficis de l'ús de les tècniques de desenvolupament de programari orientat a objectes: patrons, interfícies, etc. Cal obtenir informació de primera mà d'aquests recursos i utilitzar-los segons els preceptes indicats per aconseguir els objectius que pretenen.
- He desenvolupant una eina experimental que pot ser d'utilitat real en el món laboral en què estic immers, això sí, amb noves funcionalitats.

Per tot això en conjunt, doncs, crec que ha valgut la pena l'esforç realitzat, al qual he dedicat un bon grapat d'hores, moltíssimes més de les preceptives segons els 7.5 crèdits establerts pel TFC.

- **Glossari**

**Acció de manteniment:** Les màquines responsables de la producció necessiten un manteniment: cal canviar certs elements de tant en tant, cal lubricar-les, de vegades s'espantllen i cal reparar-les, etc. Tota la informació derivada d'aquests treballs és el que es pretén tenir emmagatzemat en una base de dades a fi de poder controlar mitjançant diversos tipus de consulta quines màquines s'espantllen més, quina fàbrica té més aturades de producció, etc.

**Consulta:** Recuperació de les dades existents a la base de dades del sistema sobre les accions de manteniment en màquines de l'empresa, presentades de tal manera que se'n puguin deduir altres informacions.

**Director de fàbrica:** Càrrec dels empleats responsables d'una de les fàbriques de l'empresa. Com a tals, han de tenir sota el seu control (i per tant, han de poder supervisar) les accions de manteniment efectuades dins la fàbrica que regeixen.

**Director general:** Càrrec dels empleats responsables de tota l'empresa. Han de poder consultar la informació relativa a qualsevol fàbrica. També són responsables d'afegir, modificar i esborrar les entitats que intervenen en l'aplicació. L'administrador del sistema té també el mateix perfil.

**Empleat:** Persona que treballa a una de les fàbriques de l'empresa. Cada empleat té assignat un càrrec, el qual determinarà la interacció que pot tenir l'empleat amb l'aplicació de gestió del manteniment.

**Empresa:** Entitat dedicada a la producció de productes del ram del plàstic. Consta d'un cert nombre de fàbriques ubicades a diferents localitats i/o països.

**Fàbrica:** Edifici que pertany a l'empresa on es realitza part de la seva producció. Una fàbrica té un número determinat de màquines i hi treballa un cert nombre de treballadors.

**Màquina:** Màquina de qualsevol tipus que estigui involucrada en la producció dels productes de l'empresa. Està ubicada en una fàbrica de l'empresa i és l'objecte de les accions de manteniment.

**Tècnic responsable:** Càrrec dels empleats tècnics que supervisen i executen les accions de manteniment i per tant, en són responsables. Són els empleats que, majoritàriament, introduïran a l'aplicació la descripció de les accions de manteniment que s'han dut a terme sota la seva responsabilitat.

**Tipus de l'acció:** Classificació general de les accions de manteniment. És una descripció petita que pot englobar un gran nombre d'accions diferents. Per exemple: avaria elèctrica, avaria pneumàtica, etc.

## **Bibliografia**

El material consultat en el desenvolupament del projecte és:

Material en paper de la Universitat Oberta de Catalunya:

- Enginyeria del Programari I
- Bases de Dades I
- Bases de Dades II
- Interacció Humana amb els Ordinadors

Material consultat a Internet:

- [www.javaworld.com](http://www.javaworld.com)
- [www.sun.com](http://www.sun.com)
- [www.adictosaltrabajo.com](http://www.adictosaltrabajo.com)
- [www.programacion.com](http://www.programacion.com)

## Annex 1

-- Sentències de creació de la base de dades associada al projecte de Gestió del Manteniment

```
create database gestman;
use gestman;

create table tipus (
    tipusId      int          primary key,
    nom          varchar (50) not null unique
);
create table carrec (
    carrecId     int          primary key,
    nom          varchar(50)  not null unique
);
create table fabrica (
    fabricaId    int          primary key,
    nom          varchar(50)  not null unique,
    adreca       varchar(50) not null
);
create table maquina (
    maquinaId   int          primary key,
    referencia   varchar(50)  not null unique,
    any          int          not null,
    descripcio  text         not null,
    fabrica      int          not null references fabrica (fabricaId)
);
create table empleat (
    empleatId   int          primary key,
    nom          varchar(50)  not null,
    cognom1     varchar(50)  not null,
    cognom2     varchar(50)  not null,
    antiguitat  date         not null,
    nomUsuari   varchar(10)  not null unique,
    contrasenya varchar(10)  not null,
    carrec      int          not null references carrec(carrecId),
    fabrica     int          not null references fabrica(fabricaId)
);
create table accio (
    accioId     int          primary key,
    preventiu   bool         not null,
    data        date         not null,
    horaInici   time         not null,
    numHores    float        not null,
    numOperaris int          not null,
    descripcio  text         not null,
    tipus       int          not null references tipus(tipusId),
    maquina     int          not null references maquina(maquinaId),
    empleat     int          references empleat(empleatId)
);
```