



Sistema Gestor de Vacances

Autor: Josep Cabruja Reig

Consultor: Vicenç Font Sagrista

Treball de fi de Carrera de ETIG
Gener 2011

ÍNDEX DE CONTINGUTS

RESUM.....	6
1. INTRODUCCIÓ.....	6
1.1. JUSTIFICACIÓ DEL TFC: PUNT DE PARTIDA I APORTACIÓ	6
1.2. OBJECTIUS DEL TFC	7
1.3. ENFOCAMENT I MÈTODE SEGUIT.....	7
1.4. PLANIFICACIÓ DEL PROJECTE	8
1.5. PRODUCTES OBTINGUTS	9
1.6. DESCRIPCIÓ DELS CAPÍTOLS DE LA MEMÒRIA	9
2. ANÀLISI DEL SISTEMA.....	10
2.1. DESCRIPCIÓ	10
2.2. REQUISITS FUNCIONALS	10
2.3. MODEL DE CASOS D' US I ACTORS.....	10
2.4. DIAGRAMA DE CASOS D' US	11
2.5. FITXES CASOS D' US	12
2.6. PROTOTIPS DE PANTALLES	18
3. DISSENY DE LA BASE DE DADES.....	25
3.1. DIAGRAMA MODEL ENTITAT/RELACIÓ.....	25
3.2. MODEL LÒGIC DE LA BASE DE DADES	26
3.3. SCRIPT PER LA CREACIÓ DE LES TAULES	26
4. DISSENY DE L' ARQUITECTURA DEL SISTEMA.....	29
4.1. INTRODUCCIÓ	29
4.1.1. JAVA EE	29
4.1.2. EJB.....	29
4.1.3. MVC	31
4.2. DISSENY DE L' ARQUITECTURA DE L' APLICACIÓ SGV.....	31
4.2.1. ARQUITECTURA DE SUPORT.....	31
4.2.2. ESPECIFICACIÓ DE L' ENTORN TECNOLÒGIC	32
4.2.3. CAPES DE L' ARQUITECTURA DEL SISTEMA.....	33
4.2.4. CAPA LÒGICA DE NEGOCI	33
4.2.4.1. DIAGRAMA CLASSES D' ENTITAT(EJB D' ENTITAT).....	33
4.2.4.2. DETALL DE LES CLASSES D' ENTITAT	34
4.2.4.3. RELACIONS ENTRE LES CLASSES ENTITAT	36
4.2.4.4. DIAGRAMES DELS EJB DE SESSIÓ	37

4.2.4.5. DETALL DELS EJB DE SESSIÓ.....	38
4.2.5. CAPA DE PRESENTACIÓ	38
4.2.5.1. PÀGINES JSF	38
4.2.5.2. MANAGED BEANS	39
4.2.6. DIAGRAMA DE RELACIÓ ENTRE PÀGINES JSF(XHTML),MANAGED BEANS I EJBS40	
5. PROVES REALITZADES.....	42
5.1.1. PROVES AMB LES DADES I ELS REQUISITS FUNCIONALS	42
5.1.2. PROVES AMB LES SESSIONS.....	42
5.1.3. PROVES DE VISUALITZACIÓ I FUNCIONAMENT	42
6. INSTAL·LACIÓ.....	42
6.1.1. INSTAL·LACIÓ DEL JDK.....	42
6.1.2. INSTAL·LAR EL SERVIDOR D' APLICACIONS GLASSFISH I LA BASE DE DADES APACHE DERBY	42
6.1.3. MODIFICACIÓ DE LES VARIABLES D'ENTORN	42
6.1.4. ENGEGAR EL SERVIDOR D' APLICACIONS GLASSFISH.....	43
6.1.5. ENGEGAR EL SERVIDOR DE BASE DE DADES APACHE DERBY	43
6.1.6. CREAR LA BASE DE DADES	44
6.1.7. CONFIGURAR EL SERVIDOR GLASSFISH	44
7. MANUAL UTILITZACIÓ	48
7.1.1. EXECUTAR L' APLICACIÓ	49
8. CONCLUSIONS.....	59
9. GLOSSARI.....	60
10. BIBLIOGRAFIA	61

INDEX D' IL·LUSTRACIONS

Il·lustració 1. Diagrama casos d' us	11
Il·lustració 2. prototip pantalla login	18
Il·lustració 3. prototip pantalla menú usuari empleat	18
Il·lustració 4. Prototip pantalla menú usuari Cap Sucursal	19
Il·lustració 5. Prototip pantalla menú usuari Responsable empresa	19
Il·lustració 6. Prototip pantalla alta usuari	20
Il·lustració 7. Prototip pantalla cercar usuaris	20
Il·lustració 8. Prototip pantalla consulta/modificació usuari	21
Il·lustració 9. Prototip pantalla alta sucursal	21
Il·lustració 10. Prototip pantalla cercar sucursals	22
Il·lustració 11. Prototip pantalla llistat sucursals	22
Il·lustració 12. Prototip pantalla consulta/modificació sucursals	23
Il·lustració 13. Prototip pantalla Alta sol·licitud vacances	23
Il·lustració 14. Prototip pantalla cercar sol·licituds vacances	24
Il·lustració 15. Prototip pantalla llistat de sol·licituds vacances cercades	24
Il·lustració 16. Prototip pantalla consulta/modificació Sol·licitud Vacances	25
Il·lustració 17. Diagrama model Entitat Relació	25
Il·lustració 18. Esquema EJB de sessió amb estat	30
Il·lustració 19. Esquema EJB de sessió sense estat	30
Il·lustració 20. Esquema patró de disseny model-vista-controlador	31
Il·lustració 21. Esquema arquitectura de suport	31
Il·lustració 22. Esquema especificació entorn tecnològic	32
Il·lustració 23. Esquema Model 3 capes	33
Il·lustració 24. Diagrama classes d' entitat	33
Il·lustració 25. Classe entitat Usuaris	34
Il·lustració 26. Classe entitat Sucursals	34
Il·lustració 27. Classe entitat Sol·licituds	35
Il·lustració 28. Classe entitat Sol·licituds	35
Il·lustració 29. Relació entre les classes entitat usuaris i sol·licituds	36
Il·lustració 30. Relació entre les classes entitat usuaris i sucursals	36
Il·lustració 31. Relació entre les classes entitat províncies i sucursals	37
Il·lustració 32. Relació entre les classes entitat províncies i usuaris	37
Il·lustració 33. Diagrama dels EJB de sessió	37
Il·lustració 34. Managed Beans	39
Il·lustració 35. Diagrama de relació entre pàgines Jsf(Xhtml), Managed Beans i Ejbs	41
Il·lustració 36. Engregar el servidor Glassfish	43
Il·lustració 37. Engregar el servidor Glassfish	43

II·lustració 38.Engegar el servidor de base de dades Apache Derby	43
II·lustració 39.Creació de la base de dades i execució de l' escript de creació de taules.....	44
II·lustració 40.Consola del servidor GlassFish	44
II·lustració 41.Creació pool de connexions	45
II·lustració 42.Creació pool de connexions	45
II·lustració 43.Propietats pool de connexions	45
II·lustració 44.Modificació de les propietats del pool de connexions	46
II·lustració 45.Pool de connexions SGV creat	46
II·lustració 46.Ping de connexió a la base de dades	46
II·lustració 47. Pantalla JDBC/Resources de la consola del servidor Glassfish.....	47
II·lustració 48.Creació del JNDI	47
II·lustració 49.JNDI un cop creat.....	47
II·lustració 50.Pantalla Applications de la consola del servidor Glassfish	48
II·lustració 51.Realitzar el Deploy	48
II·lustració 52.Deploy realitzat.....	48
II·lustració 53.Pantalla de login.....	49
II·lustració 54.Pantalla modificar contrasenya	50
II·lustració 55.Pantalla Menú de l' usuari Responsable Empresa	50
II·lustració 56.Pantalla menú altes de l' usuari Responsable Empresa.....	51
II·lustració 57.Pantalla alta usuari.....	51
II·lustració 58.Missatge conforme s' ha creat usuari.....	52
II·lustració 59.pantalla alta de sucursals.....	52
II·lustració 60.Exemple missatge d' error.....	53
II·lustració 61.Menú cercar de l' usuari Responsable empresa.....	53
II·lustració 62.Cerques d' usuaris.....	53
II·lustració 63.Exemple de cerca per sucursal assignada	54
II·lustració 64. Llistat d' usuaris.....	54
II·lustració 65.consulta d' usuari.....	55
II·lustració 66.pantalla modificació d' un usuari	55
II·lustració 67.Cerca de sol·licituds de vacances.....	56
II·lustració 68.Sel·lecció de data	56
II·lustració 69.Llistat de sol·licituds de vacances	57
II·lustració 70.Modificació de sol·licitud de vacances	57
II·lustració 71.Cerca de sucursal	57
II·lustració 72.Pantalla menú usuari empleat.....	58
II·lustració 73.Alta d' una sol·licitud de vacances	58
II·lustració 74.Pantalla menú usuari Cap Sucursal.....	59

RESUM

Aquest document és la memòria de com s'ha realitzat el programari Sistema Gestor de Vacances utilitzant la tecnologia Java, concretament, l'estàndard JAVA EE6.

Es tracta d'una aplicació web per gestionar les vacances dels treballadors d'una empresa composta per una seu central i varies sucursals distribuïdes per tot Catalunya.

Des de la seu central es vol portar un control de les vacances que fan tots els seus treballadors, tant de la seu central com de les sucursals.

De forma general, el Sistema Gestor de Vacances permetrà realitzar a un empleat sol·licituds de vacances que tindran que ser aprovades pel cap de la sucursal i un cop aprovades, tindran que ser notificades al responsable administratiu de l'empresa per a procedir a la seva validació i control.

Per desenvolupar aquesta aplicació s'han fet servir tots els coneixements adquirits en les diferents assignatures de la carrera d'ETIG relacionades amb l'anàlisi disseny i implantació de les aplicacions orientades a Objectes i de la gestió de les bases de dades. També ha sigut necessari l'estudi de la tecnologia Java i concretament de l'Estàndard JAVA EE6, el framework JSF 2.0, l'utilització de la llibreria de components Primefaces i del IDE NetBeans 7.0 M2, la qual cosa ha suposat una corba d'aprenentatge molt forta si tenim en compte el poc temps de que es disposa per a la realització del TFC.

- **Àrea del TFC:**J2EE
- **Paraules Clau:** JAVA EE6, EJB 3.1 (Enterprise Java Bean), JPA 2.0(Java Persistence API) , Framework JSF 2.0, llibreria de components Primefaces, Servidor d'aplicacions GlassFish v3, Gestor de bases de dades Apache Derby

1. INTRODUCCIÓ

Com a treball de fi de carrera s'ha realitzat l'anàlisi, disseny i implementació d'una aplicació utilitzant la tecnologia Java, concretament, l'estàndard JAVA EE6.

Es tracta d'una aplicació web per gestionar les vacances dels treballadors d'una empresa composta per una seu central i varies sucursals distribuïdes per tot Catalunya.

Des de la seu central es vol portar un control de les vacances que fan tots els seus treballadors, tant de la seu central com de les sucursals.

1.1. JUSTIFICACIÓ DEL TFC: PUNT DE PARTIDA I APORTACIÓ

El factor determinant que m'ha fet triar aquest Àrea del TFC, és que en la carrera d'ETIG no hi ha cap assignatura enfocada al desenvolupament d'aplicacions web i crec que actualment en el món empresarial i professional és de vital importància el seu coneixement ja que l'imposició d'Internet a nivell empresarial ha fet que les aplicacions d'empresa usin la xarxa com a vehicle de comunicació bàsic. A més, la possibilitat d'usar la web com a interfície està fent que moltes aplicacions antigues s'estiguin portant cap a aquest entorn.

Un altre factor que m'ha fet triar aquest àrea, és que Java com a llenguatge de desenvolupament, juntament amb l'arquitectura J2EE s'ha convertit en un estàndard en el món de la indústria per al desenvolupament distribuït d'aplicacions empresarials a Internet. Donat que l'orientació a objectes i

Java han constituït el puntal bàsic en totes les assignatures orientades a la programació que s' han cursat en la Carrera, em va semblar molt interessant veure, com, i fins a quin punt, es podien aplicar tots els coneixements adquirits al desenvolupament d' una aplicació web distribuïda.

El meu punt de partida per realitzar el TFC han sigut els coneixements adquirits durant tots aquesta anys en els que he cursat la carrera. Assignatures com Programació orientada a objectes, Enginyeria del programari, Tècniques de desenvolupament de programari, bases de dades I, bases de dades II i Sistemes de Gestió de bases de dades, entre altres, m' han sigut de gran utilitat. No obstant també haig de dir que la corba d' aprenentatge ha sigut molt forta, ja que s' han tingut que aprendre en un temps molt reduït molts conceptes (HTML, XHTML, XML, CSS, JAVASCRIPT, JSP, SERVLETS, EJB 3.1, JPA2.0, JSF2.0 , PRIMEFACES, GLASSFISH V3, APACHE DERBY.....).

La principal aportació d' aquest TFC ha sigut que m' ha permès introduir-me en l' arquitectura J2EE mitjançant l' anàlisi, disseny i implementació d' una aplicació distribuïda basada en aquesta arquitectura.

1.2.OBJECTIUS DEL TFC

El principal objectiu d' aquest TFC és donar a conèixer l' apassionant món del desenvolupament de programari per Internet usant la tecnologia Java.

El compliment d' aquest objectiu s' ha realitzat mitjançant l' anàlisi, disseny i implementació d' aquesta petita aplicació anomenada Sistema Gestor de Vacances, basada en l' estàndard JAVA EE6.

Aquesta aplicació es basa en el framework JSF2.0 que segueix el patró de disseny **Model Vista Controlador**

- **Model:** Conté la lògica de negoci, aquí he experimentat amb els EJB que s' encarreguen d' accedir a les bases de dades i amb el JPA2.0 com a mecanisme de mapatge objecte/Relacional per proveir d' una capa d' accés a la base de dades.
- **Vista:** Presenta la informació a l' usuari en un format adequat. Aquí he experimentat amb les pàgines JSF (XHTML) amb les fulles d' estil en cascada (CSS), amb PRIMEFACES, llibreria de components visuals open source per a JSF2.0, ens ofereix components com ara calendaris per els camps de data, barres de menús, botons, màscares per camps d' entrada, panells, suport natiu d' Ajax.....
També he experimentat una mica amb el JAVASCRIPT
- **Controlador:** Respon a esdeveniments, normalment accions de l' usuari e invoca canvis en el model i en la vista. Aquí he experimentat amb els Managed Beans que delegen tota la seva funcionalitat als EJBs de la capa de negoci per lo que les seves úniques funcions estan relacionades amb la gestió de les interaccions derivades de l' acció de l' usuari. Fan de intermediaris entre la vista i el model.

Per tant, el desenvolupament d' aquesta aplicació m' ha permès provar i experimentar amb les tecnologies concretes que són usades àmpliament en el món empresarial, la arquitectura J2EE i concretament l' estàndard JAVA EE6 i tots els seus components. **Això és el que constitueix el principal objectiu d' aquest TFC**

1.3.ENFOCAMENT I MÈTODE SEGUIT

L' enfocament i mètode seguit, es el determinat per les diferents PACS, consells i aclariments donats pel tutor i el contingut del Pla docent de l' assignatura.

No obstant per poder afrontar la PAC3, de forma paral·lela a les fites marcades en la planificació del TFC, des dels primers dies en que es va iniciar l' assignatura s' han dedicant moltes hores diàries a adquirir coneixements de l' arquitectura J2EE i concretament l' estandard JAVA EE6 i a realitzar petites proves practiques per anar assolint coneixements.

- **Pac1**
 - Decidir el programari que es vol desenvolupar acotat amb descripcions i requeriments.
 - Inclusió d' una planificació detallada de com es desenvoluparà el TFC amb fites de seguiment per consensuar el que estem fent amb el nostre tutor, fites finals indicat en tot moment les dates en las que s' han de produir aquestes fites
 - Decidir les eines i els frameworks a utilitzar
- **Pac2**
 - Anàlisi i disseny de l' aplicació respectant rigorosament les dates de les fites marcades en la planificació.
- **Pac3**
 - Implementació de l' aplicació. Començament amb un cas d' us senzill, sense tenir gaire en consideració la presentació de les dades i si el seu funcionament. Posteriorment, després d' implementar els casos d' us més importants i bàsics, es va tenir molt en compte la presentació amb la utilització de la llibreria de components Primefaces. Utilització de plantilles i de CSS.
 - Realització del jocs de proves per cada fita aconseguida i per a la prova de l' aplicació ja desenvolupada totalment.
- **Pac4**
 - Realització de la memòria de l' aplicació, que constituirà la part de documentació de l' aplicació, juntament amb la presentació.

Totes les fites s' han realitzat tenint en compte de forma rigorosa les dates marcades en la planificació de l' aplicació.

1.4. PLANIFICACIÓ DEL PROJECTE

La planificació del projecte es basa en la següent taula:

PAC2	Model de Casos d' us i actors	07/10/2010 a 14/10/2010
	Fitxes de casos d' us	07/10/2010 a 14/10/2010
	Model de pantalles	15/10/2010 a 21/10/2010
	Lliurament parcial de seguiment	21/10/2010
	Disseny relacional de la base de dades	22/10/2010 a 26/10/2010
	Diagrames de classes	27/10/2010 a 01/11/2010
	Lliurament parcial de seguiment	01/11/2010
	Diagrama d' arquitectura	02/11/2010 a 11/11/2010
	Lliurament definitius de la PAC	11/11/2010
PAC3	Construcció de la base de dades	12/11/2010 a 15/11/2010
	Integració de totes les eines i proves de funcionament (base de dades, IDE frameworks)	15/11/2010 a 17/11/2010

	Desenvolupament simple d' un cas d' us	18/11/2010 a 21/11/2010
	Lliurament parcial de seguiment	21/11/2010
	Completar el cas d' us (resta de camps, validacions etc..)	21/11/2010 a 25/11/2010
	Desenvolupament dels altres casos d' us	26/11/2010 a 20/12/2010
	Lliurament parcial de seguiment	04/12/2010
	Lliurament definitius de la PAC	20/12/2010
PAC4	Realització de la memòria i de la presentació	21/12/2010
	Lliurament parcial de seguiment	02/01/2011
	Lliurament parcial de seguiment	14/01/2011
	Lliurament definitius de la PAC	17/01/2011

1.5. PRODUCTES OBTINGUTS

- **Una aplicació web anomenada Sistema Gestor de Vacances** que permet gestionar les vacances dels treballadors d' una empresa composta per una seu central i varies sucursals distribuïdes per tot Catalunya.
- **Una memòria** de com s' ha realitzat el programari Sistema Gestor de Vacances utilitzant la tecnologia Java, concretament, l' estàndard JAVA EE6.
- **Una presentació Virtual** que sintetitza el treball realitzat al llarg del semestre i els resultats obtinguts

1.6. DESCRIPCIÓ DELS CAPÍTOLS DE LA MEMÒRIA

- **Anàlisi:** S' especifiquen les funcionalitats i els requisits funcionals de l' aplicació, es determina el model de casos d' us i Actors i s' especifiquen els diagrames de casos d' us i les seves fitxes.
- **Disseny de la Base de dades:** S' especifica el diagrama del model Entitat/Relació i el model lògic
- **Disseny de l' arquitectura del sistema:** S' especifica l' arquitectura de suport, l' entorn tecnològic, les capes de l' arquitectura dels sistema i el diagrama de relació entre pàgines JSF, Managed Beans i EJBs
- **Proves realitzades:** Breu descripció de les proves a les que s' ha sotmès l' aplicació
- **Manual d' instal·lació i d' us:** Breu explicació de com instal·lar l' aplicació i del seu funcionament

2. ANÀLISI DEL SISTEMA

2.1. DESCRIPCIÓ

De forma general, el sistema gestor de vacances permetrà realitzar a un empleat sol·licituds de vacances que tindran que ser aprovades pel cap de la sucursal i un cop aprovades, tindran que ser notificades al responsable administratiu de l'empresa per a procedir a la seva validació i control.

A cada sucursal i a la seu central hi haurà un cap que aprovarà o refusarà les sol·licituds de vacances realitzades pels seus empleats,

A la seu Central de l'empresa el responsable administratiu s'encarregarà de validar i controlar totes les sol·licituds de vacances aprovades per els caps de les sucursals

2.2. REQUISITS FUNCIONALS

- Tots els usuaris accedeixen a l'aplicació a través d'una plana web, on es demana codi d'usuari i contrasenya per accedir al sistema, una vegada autenticats i segons el rol que tenen assignat, l'aplicació els hi dona accés a les funcionalitats relacionades amb el seu rol.
- Tots els usuaris podran realitzar el canvi de la seva contrasenya quan ells ho considerin oportú.
- Cada empleat disposarà de 20 dies laborables anuals de vacances. Aquests dies s'aniran reduint en funció de les sol·licituds presentades fins arribar a zero dies
- Cada sol·licitud de vacances que faci l'empleat com a mínim tindrà que ser de 5 dies.
- Un empleat només pot estar assignat a una sucursal.

2.3. MODEL DE CASOS D'US I ACTORS

En el sistema es distingeixen els següents actors amb les següents funcionalitats:

- **Funcionalitats del tipus empleat:**

El sistema ha de permetre les següents funcions a un usuari de tipus empleat:

- Creació de sol·licituds de vacances
- Cercar totes les sol·licituds que ha creat
- Llistar les sol·licituds de vacances realitzades per ell mateix amb l'objectiu de comprovar l'estat en que es troben les seves sol·licituds i poder fer consultes i modificacions. Només podrà fer modificacions quan l'estat de la sol·licitud sigui petició.

- **Funcionalitats del tipus Cap de sucursal:**

El sistema ha de permetre les següents funcions a un usuari de tipus Cap de sucursal:

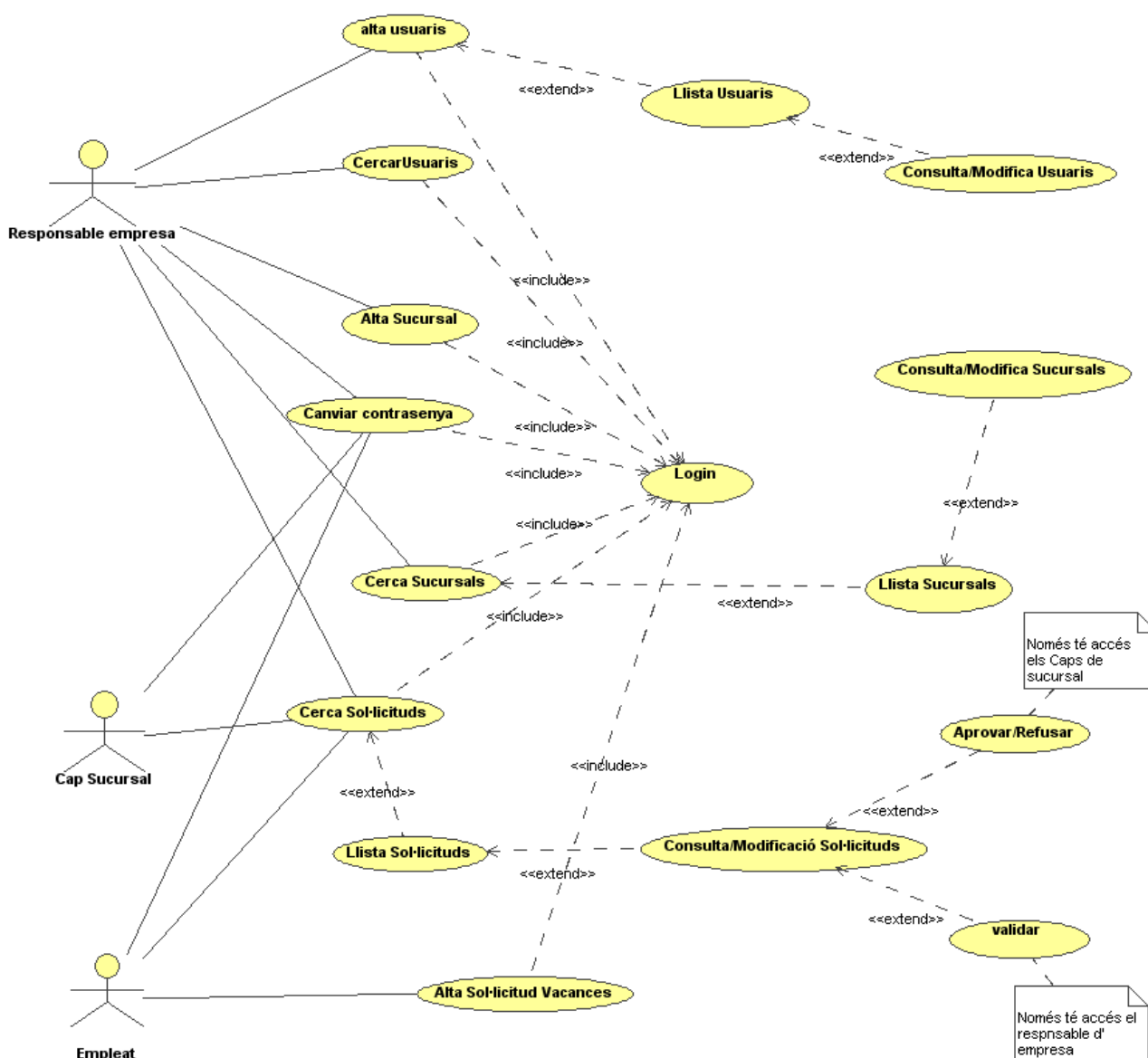
- Llistat de les sol·licituds de vacances realitzades pels empleats que pertanyen a la seva sucursal.
- Visualització de les dades de cada sol·licitud del personal que té al seu càrrec.
- Aprovar sol·licituds de vacances realitzades pels empleats que pertanyen a la sucursal.
- Refusar sol·licituds de vacances realitzades pels empleats que pertanyen a la sucursal.

- **Funcionalitats del tipus Responsable d'empresa**

El sistema ha de permetre les següents funcions a un usuari de tipus Responsable d'empresa:

- Donar d' alta als empleats, fer modificacions, consultes de les seves dades, donar-los de baixa i assignar-los a una sucursal.
- Llistat de totes les sol·licituds de vacances realitzades en cada sucursal
- Visualització de les dades de cada sol·licitud
- Validar sol·licituds de vacances prèviament acceptades pel responsable de cada sucursal.
- Manteniment de les sucursals, altes, baixes, consultes

2.4. DIAGRAMA DE CASOS D'US:



Il·lustració 21. Diagrama casos d' us

2.5. FITXES DE CASOS D' US

Cas d' us	Fer Login
Descripció:	Permet accedir al sistema a tots els usuaris.
Casos d' us relacionats:	
Actors	Empleat, Cap de sucursal, Responsable administratiu.
Precondició	L' usuari s'ha identificat en el sistema .
Postcondició	L' usuari queda identificat en el sistema . L' usuari té accés a totes les funcionalitats relacionades amb els seu Rol.
Flux bàsic	L' actor introdueix usuari i contrasenya. El sistema valida les dades. Si l' usuari es correcta se li dona accés a les funcionalitats assignades al seu rol.
Flux alternatiu	Si l' usuari introduït no es correcte, el sistema torna a demanar usuari i contrasenya.
Observacions	Quan es dona d' alta un usuari pel responsable de l'Empresa, se li assigna un usuari i contrasenya inicial. L' usuari podrà canviar en qualsevol moment la contrasenya inicial per una altre.

Cas d' us	Modificar Contrasenya
Descripció:	Permet als usuaris canviar la contrasenya inicial
Casos d' us relacionats:	
Actors	Empleat, Cap de sucursal
Precondició	L' usuari té assignat un usuari i una contrasenya L' usuari s'ha identificat en el sistema
Postcondició	L' usuari té una nova contrasenya
Flux bàsic	L' actor introdueix la nova contrasenya El sistema guarda la nova contrasenya.
Flux alternatiu	L' usuari cancel·la el canvi de contrasenya
Observacions	

Cas d' us	Donar d' alta usuaris
Descripció:	Permet donar d'alta als els usuaris del sistema.
Casos d' us relacionats:	
Actors	Responsable de l' empresa
Precondició	El Responsable de l' empresa té permisos per gestionar aquesta part de l' aplicació. El Responsable de l' empresa s'ha identificat en el sistema L' empleat no estava donat d' alta en el sistema
Postcondició	L' empleat queda donat d' alta al sistema L' empleat obtindrà un nom d' usuari i una contrasenya inicial per identificar-se en el sistema L' empleat quedarà assignat a una de les sucursals de l' empresa. L' empleat tindrà accés a totes les funcionalitats pròpies del rol d' empleat.
Flux bàsic	L' actor selecciona l' opció Alta Usuari L' actor introdueix tota la informació relativa al usuari El sistema valida les dades i les guarda a la base de dades
Flux alternatiu	L' empleat ja existeix a la base de dades
Observacions	La contrasenya que s' assigna a l' usuari es una contrasenya inicial amb la que l' usuari empleat podrà identificar-se per entrar al sistema em qualsevol moment l' empleat podrà fer el canvi de la contrasenya.

Cas d' us	Cercar/ordenar Usuaris
Descripció	Permet cercar o ordenar usuaris
Casos d' us relacionats	Consulta/modificació de sucursals
Actors	Responsable Administrador
Precondició	L' usuari té permisos per gestionar aquesta part de l' aplicació L' usuari s'ha identificat en el sistema
Postcondició	El sistema mostra tots els usuaris que compleixen el criteri de cerca
Flux bàsic	L' actor selecciona l' opció cercar usuaris L' actor cerca el usuari seleccionat un dels criteris de cerca El sistema respon amb un llistat amb totes les sucursals que compleixen el criteri de cerca.
Flux alternatiu	
Observacions	

Cas d' us	Consultar/modificar usuaris
Descripció:	Permet consultar o modificar la fitxa d'un usuari.
Casos d' us relacionats:	Cercar / ordenar Usuaris
Actors:	Administrador del Sistema
Precondició:	L' usuari té permisos per gestionar aquesta part de l' aplicació L' usuari s'ha identificat en el sistema
Postcondició:	Les dades de l' usuari queden modificades.
Flux bàsic:	L' actor selecciona l' usuari en el llistat d' usuaris trobats que compleixen el criteri de cerca i accedeix a la seva fitxa per visualitzar o modificar les dades.
Flux alternatiu:	
Observacions:	Modificant la situació d' usuari a baixa, aquest no tindrà accés al sistema

Cas d' us	Donar d' Alta Sucursals
Descripció	Permet donar d' alta les sucursals de l' empresa.
Casos d' us relacionats	Cercar / ordenar Sucursals.
Actors	Responsable de l' empresa
Precondició	L' usuari té permisos per gestionar aquesta part de l' aplicació. L' usuari s'ha identificat en el sistema . La sucursal no estava donat d' alta en el sistema.
Postcondició	La sucursal queda donat d' alta al sistema .
Flux bàsic	L' actor selecciona l' opció Alta Sucursal. L' actor introdueix tota la informació relativa a la sucursal. El sistema valida les dades i les guarda a la base de dades
Flux alternatiu	La sucursal ja existeix a la base de dades
Observacions	

Cas d' us	Cercar/ordenar Sucursals
Descripció	Permet cercar o ordenar sucursals
Casos d' us relacionats	Consulta/modificació de sucursals
Actors	Responsable de l' empresa
Precondició	L' usuari té permisos per gestionar aquesta part de l' aplicació L' usuari s'ha identificat en el sistema
Postcondició	El sistema mostra totes les sucursals que compleixen el criteri de cerca
Flux bàsic	L' actor selecciona l' opció cercar sucursals L' actor cerca la sucursal seleccionat un dels criteris de cerca El sistema respon amb un llistat amb totes les sucursals que compleixen el criteri de cerca.
Flux alternatiu	
Observacions	

Cas d' us	Consultar/modificar Sucursals
Descripció:	Permet consultar o modificar la fitxa d' una sucursal
Casos d' us relacionats:	Cercar / ordenar sucursals
Actors:	Responsable de l' empresa
Precondició:	L' usuari té permisos per gestionar aquesta part de l' aplicació L' usuari s'ha identificat en el sistema
Postcondició:	Les dades de la sucursal queden modificades.
Flux bàsic:	L' actor selecciona la sucursal en el llistat de sucursals trobades que compleixen el criteri de cerca i accedeix a la seva fitxa per visualitzar o modificar les dades.
Flux alternatiu:	
Observacions:	Es dona opció a marcar la sucursal com de baixa

Cas d' us	Alta sol·licitud de vacances
Descripció:	Permet als empleats crear una nova sol·licitud de vacances
Casos d' us relacionats:	
Actors	Empleat
Precondició	L' usuari té permisos per gestionar aquesta part de l' aplicació. L' usuari s'ha identificat en el sistema. L' usuari té un número de dies pendents de vacances superior o igual a 5
Postcondició	Queda gravada la sol·licitud de vacances en estat de pendent
Flux bàsic	L' actor accedeix a l' opció sol·licitar vacances L' actor omple les dades corresponents a la sol·licitud de vacances. El sistema guarda la nova contrasenya.
Flux alternatiu	
Observacions	

Cas d' us	Cercar sol·licituds de vacances
Descripció:	Permet cercar les sol·licituds de vacances
Casos d' us relacionats:	consultar /modificar sol·licituds
Actors:	Responsable de l'empresa, empleats, cap de la sucursal
Precondició:	L' usuari té permisos per gestionar aquesta part de l' aplicació. L' usuari s'ha identificat en el sistema.
Postcondició:	Existeixen sol·licituds que compleixen amb els criteris de cerca
Flux bàsic:	L' actor selecciona l' opció cercar sol·licituds L' actor selecciona els criteris de cerca de sol·licituds de vacances El sistema respon amb un llistat de sol·licituds de vacances que es corresponen amb els criteris de cerca
Flux alternatiu:	
Observacions:	<u>El Responsable administratiu</u> podrà cercar : Totes les sol·licituds de vacances dels empleats de qualsevol sucursal sigui quin sigui el seu estat (En estat de petició , aprovades, refusades, pendents de validar, validades) <u>El cap de Sucursal</u> podrà cercar : Totes les sol·licituds de vacances dels empleats de la seva sucursal sigui quin sigui el seu estat (En estat de petició , aprovades, refusades, pendents de validar, validades) <u>L' empleat</u> podrà cercar: Totes les sol·licituds que ha realitzat l' empleat sigui quin sigui el seu estat (En estat de petició, aprovades, refusades, pendents de validar, validades)

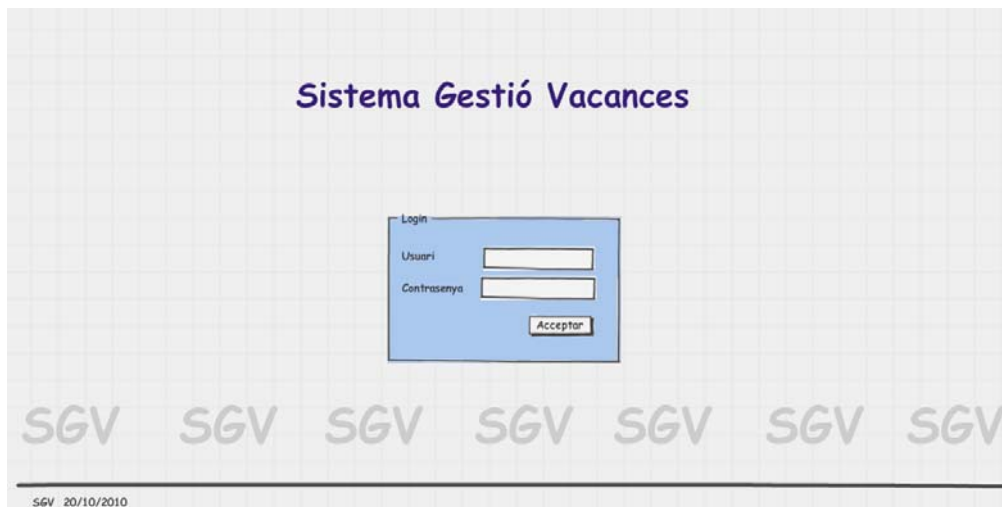
Cas d' us	Consultar/modificar Sol·licituds
Descripció:	Permet consultar o modificar la fitxa d'una sol·licitud
Casos d' us relacionats:	Cercar / ordenar Sol·licituds
Actors:	Responsable de l' empresa, empleat i Cap de sucursal
Precondició:	L' usuari té permisos per gestionar aquesta part de l' aplicació. L' usuari s'ha identificat en el sistema
Postcondició:	Les dades de la sol·licitud queden modificades.
Flux bàsic:	L' actor selecciona la sol·licitud en el llistat de sol·licituds trobades que compleixen el criteri de cerca i accedeix a la seva fitxa per visualitzar o modificar les dades.
Flux alternatiu:	
Observacions:	<u>L' empleat:</u> podrà modificar les dades corresponents a les seves sol·licituds sempre i quan el seu estat sigui el de pendent. <u>El cap de sucursal:</u> només podrà modificar l' estat de les sol·licituds dels empleats de la seva sucursal a refusada o aprovada i el camp d' observacions per explicar els motius de l' aprovació o rebuig de la sol·licitud. <u>El responsable de l' Empresa</u> podrà modificar l' estat de totes les sol·licitud aprovades pels caps de sucursal a validada i el camp d' observacions per explicar els motius de la validació o no de la sol·licitud aprovada pel cap de sucursal Modificant la situació d l' usuari a baixa, aquest no tindrà accés al sistema

Cas d' us	Validar sol·licituds de vacances
Descripció:	Permet validar les sol·licituds de vacances dels empleats prèviament acceptades pel responsable de cada sucursal
Casos d' us relacionats:	consultar /modificar sol·licituds
Actors:	Responsable de l' empresa
Precondició:	L' usuari té permisos per gestionar aquesta part de l' aplicació L' usuari s'ha identificat en el sistema. La sol·licitud està aprovada pel cap de la sucursal
Postcondició:	La situació de la Sol·licitud de vacances passa a ser validada pel responsable de l' empresa En el cas de ser validada el sistema resta els dies sol·licitats del total de vacances pendents de l' empleat
Flux bàsic:	L' actor selecciona la sol·licitud en el llistat de sol·licituds trobades que compleixen el criteri de cerca i que estan aprovades pels caps de sucursal i accedeix a la seva fitxa per modificar l' estat de la sol·licitud validada.
Flux alternatiu:	
Observacions:	

Cas d' us	Aprovar o refusar sol·licituds de vacances en estat de petició
Descripció:	Permet canviar l' estat de petició d' una sol·licitud de vacances a l' estat d' aprovada o refusada
Casos d' us relacionats:	
Actors:	Cap de Sucursal
Precondició:	L' usuari té permisos per gestionar aquesta part de l' aplicació L' usuari s'ha identificat en el sistema. La sol·licitud de vacances està en estat de petició per l' empleat
Postcondició:	La situació de la Sol·licitud de vacances passa a ser refusada o aprova pel cap de sucursal
Flux bàsic:	L' actor selecciona la sol·licitud en el llistat de sol·licituds trobades que compleixen el criteri de cerca i que estan en estat de pendent i accedeix a la seva fitxa per modificar l' estat de la sol·licitud a aprovada o refusada.
Flux alternatiu:	
Observacions:	En el cas que es refusi una sol·licitud, en el camp observacions es posarà la causa per la qual s' ha refusat la sol·licitud de vacances.

2.6. PROTOTIPS DE PANTALLES:

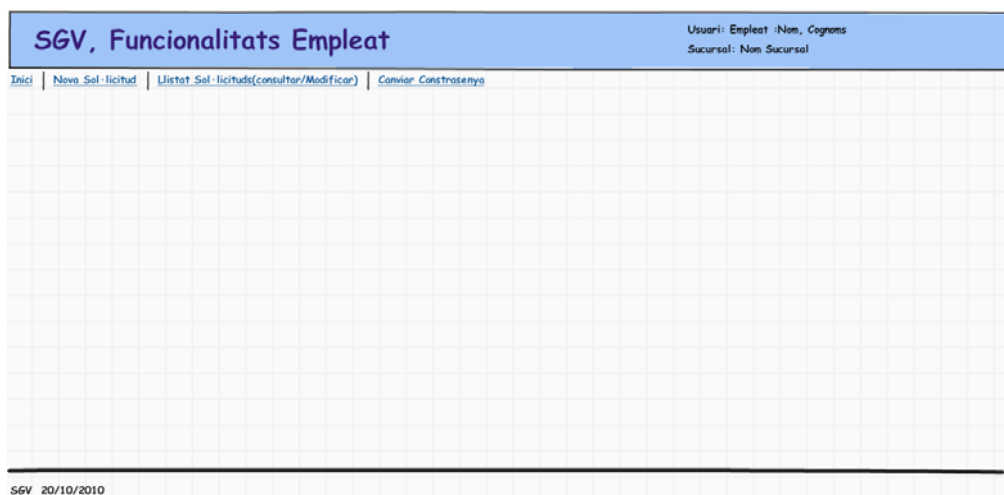
- Pantalla que permet fer login:



Il·lustració 22. prototip pantalla login

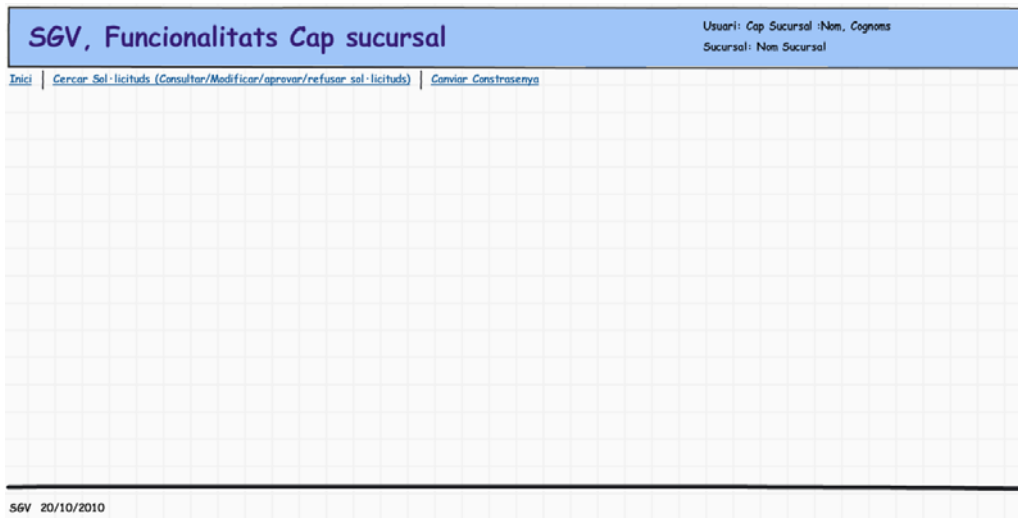
Un cop que l'usuari s'ha validat en el sistema, accedeix al menú de funcionalitats corresponent al seu rol

- Pantalla menú usuari empleat



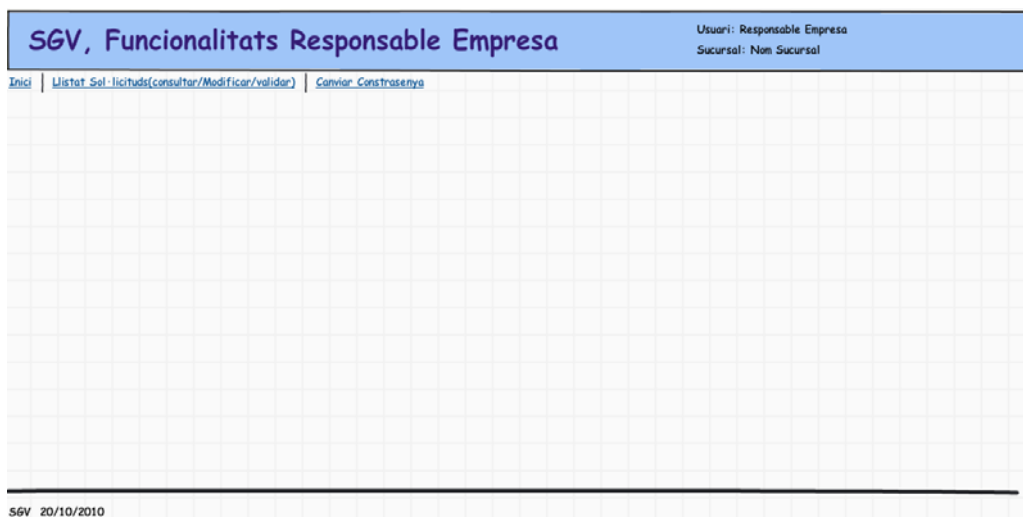
Il·lustració 23. prototip pantalla menú usuari empleat

- Pantalla menú usuari Cap Sucursal



Il·lustració 24. Prototip pantalla menú usuari Cap Sucursal

- Pantalla menú usuari Responsable empresa



Il·lustració 25. Prototip pantalla menú usuari Responsable empresa

- Pantalla que l'usuari Responsable Empresa farà servir per donar d'alta usuaris en el sistema. Quan es don d'alta un usuari se li assigna una contrasenya inicial que en qualsevol moment l'usuari la podrà canviar.

SGV: Alta Usuari

Responsable Administratiu: Nom, Cognoms

Sortir ...

Alta usuari

Cognoms:

Noms:

Dni:

Contrasenya inicial:

Ubicació:

Estat:

Adreça:

Codi Postal:

Població:

Telèfon:

Gravar Cancel·lar

SGV 20/10/2010

II-lustració 26. Prototip pantalla alta usuari

- Pantalla per cercar usuaris que el Responsable de l'Empresa farà servir

SGV: Cerca Usuaris

Re:

Sortir ...

Criteris Cerca

DNI

Cognoms i Nom

Sucursal

Acceptar

SGV 20/10/2010

II-lustració 27. Prototip pantalla cercar usuaris

- Pantalla Consulta/Modificació d' usuaris que el Responsable de l'Empresa farà servir. El camp estat indica si l' usuari està en situació d' alta o de baixa. Quan l' usuari esta en situació de baixa no podrà accedir al sistema. No es permet eliminar dades.

SGV: Consulta/modificació Usuari Resp

Consulta/Modificació usuari

Cognoms	<input type="text"/>
Noms	<input type="text"/>
Dni:	<input type="text"/>
Ubicació	<input type="text" value="Sucursal1"/>
Estat	<input type="text" value="Alta"/>
Adreça:	<input type="text"/>
Codi Postal:	<input type="text"/>
Població:	<input type="text"/>
Telèfon:	<input type="text"/>

SGV 20/10/2010

II-lustració 28-Prototip pantalla consulta/modificació usuari

- Pantalla donar d' alta sucursal que el Responsable de l'Empresa farà servir

SGV: Alta Sucursal Res

Alta Sucursal

Població:	<input type="text"/>
Codi Postal:	<input type="text"/>
Adreça:	<input type="text"/>
Telèfon:	<input type="text"/>

SGV 20/10/2010

II-lustració 29. Prototip pantalla alta sucursal

- Pantalla Cercar Sucursal que el Responsable de l'Empresa farà servir

SGV: Cerca Sucursals
Re

Creris Cerca

Codi

Denominació

Ubicació(Població)

SGV 20/10/2010

II-lustració 30. Prototip pantalla cercar sucursals

- Pantalla llistat sucursal segons criteri de cerca que s' ha sel·leccionat.

SGV: Llistat Sucursals
Responsable Administrati

CODI	DENOMINACIÓ	POBLACIÓ	C. Postal	ADREÇA	TELEFON	
1	Sucursal1	Girona	17006	C/costa Brava ,24	972 233658	<input type="button" value="Editar"/>
2	Sucursal2	Figueres	17600	C/Sant Rafael, 2	972 256589	<input type="button" value="Editar"/>
3	Sucursal3	Barcelona	08001	C/ Nou, 34	93 256398	<input type="button" value="Editar"/>
4	Sucursal4	Tarragona	46002	Avda. Josep Tarredellas, 11	99 9999999	<input type="button" value="Editar"/>
5	Sucursal5	Lleida	25653	C/ Del Sud, 45	98 9999999	<input type="button" value="Editar"/>

[Anteriors](#) 10 [Següents](#)

SGV 20/10/2010

II-lustració 31. Prototip pantalla llistat sucursals

- Pantalla per consultar i modificar les dades de les sucursals. El camp estat indica si la sucursal està en situació d' alta o de baixa. No es permet eliminar dades

SGV: Consulta/ Modificació dades Sucursal
Respo

Dades Sucursal

Codi:

Denominació:

Població:

Codi Postal:

Adreça:

Telèfon:

Estat:

SGV 20/10/2010

II-lustració 32. Prototip pantalla consulta/modificació sucursals

- Pantalla per donar de alta sol·licituds de vacances, aquesta pantalla la utilitzarà l' usuari empleat.

SGV: alta Sol·licitud vacances
Usuari: Empleat :
Sucursal: Nom St

Alta Sol·licitud de vacances

Codi Sol·licitud:

Data Inici:

Data fi:

Dies Sol·licitats:

Observacions:

SGV 20/10/2010

II-lustració 33. Prototip pantalla Alta sol·licitud vacances

- Pantalla per cercar sol·licituds, aquesta pantalla la faran servir tots els usuaris. No obstant l'usuari empleat només podrà veure les sol·licituds pròpies, el cap de sucursal veurà totes les sol·licituds que han realitzat els empleats de la seva sucursal i el responsable de l' Empresa podrà veure totes les sol·licituds, de qualsevol empleat i sucursal.

SGV: Cercar Sol·licituds
Usuari: Cap Sucursal :Nom, Cognoms
Sucursal: Nom Sucursal

Estat de la sol·licitud
Codi de sol·licitud

Des de: fins a:

Petició
 Aprovades
 Refusades
 Validades

SGV 20/10/2010

Il·lustració 34. Prototip pantalla cercar sol·licituds vacances

- Pantalla on es llisten les sol·licituds cercades. Aquesta pantalla la faran servir tots els usuaris. Fent clic en el botó editar, es pot consultar/modificar la sol·licitud sel·leccionada.

SGV: Llistat de Sol·licituds cercades
Usuari: Cap Sucursal :Nom, Cognoms
Sucursal: Nom Sucursal

CODI	NOM	COGNOMS	DATA SOL·LICITUD	DATA INICI	DATA FI	ESTAT	DIES SOL·LICITATS	DIES VACANCES	
1	Josep	Martinez Puig	01/06/2010	10/06/2010	15/06/2010	Validada	5	15	<input type="button" value="Editar"/>
5	Juan	Casas Pol	02/09/2010	10/09/2010	15/09/2010	Petició	5	20	<input type="button" value="Editar"/>
20	Pere	Miquel Arbos	03/09/2010	10/09/2010	16/09/2010	Petició	6	20	<input type="button" value="Editar"/>

[Anteriors](#) 10 [Següents](#)

SGV 20/10/2010

Il·lustració 35. Prototip pantalla llistat de sol·licituds vacances cercades

- Pantalla que serà utilitzada per tots els usuaris. L'usuari empleat només podrà fer modificacions mentre la sol·licitud estigui en estat de petició.

El cap de sucursal podrà modificar el camp Estat per refusar/aprovar la sol·licitud i el camp observacions per explicar la causa de refús o aprovació.
 El responsable d' empresa podrà modificar el camp Estat per validar la sol·licitud i el camp observacions per explicar la causa per la qual se valida o no.

SGV: consulta/modificació Sol·licitud
Usuari: Cap Sucursal :Nom, Cognoms
Sucursal: Nom Sucursal

Usuari Empleat

DNI usuari:

Cognoms i Noms:

Telèfon:

Dies Pendants:

Sol·licitud de l'Empleat

Codi Sol·licitud:

Data Sol·licitud:

Data Inici:

Data fi:

Dies Sol·licitats:

Data Estat Sol·licitud:

Estat Sol·licitud:

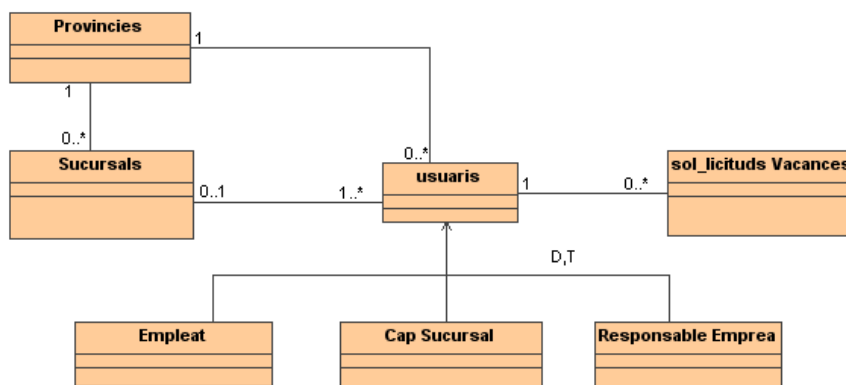
Observacions:

SGV 20/10/2010

Il·lustració 36. Prototip pantalla consulta/modificació Sol·licitud Vacances

3. DISENY DE LA BASE DE DADES

3.1. DIAGRAMA MODEL ENTITAT/RELACIÓ



Il·lustració 37. Diagrama model Entitat Relació

3.2. MODEL LÒGIC DE LA BASE DE DADES

- Les claus primàries es subratllen i les foranies es posen en cursiva
- **Relació províncies:**
Províncies(província,codProv)
Codprov es clau primària
- **Relació sucursal:**
Sucursals(adreca, codipostal, poblacio, estat, província, codsuc)
codsuc és clau primària i província és clau forana de Províncies
- **Relació Usuaris:**
La relació de generalització/especialització (Usuari, Empleat, Cap Sucursal i Responsable Empresa) es converteix amb la següent relació:

Usuaris(nom,cognoms,password,estat, adreca, codipostal, poblacio rol, tel,província, codsuc, codi,)

codi és clau primària, codsuc es clau forana de Sucursals i província és clau forana de Províncies
- **Relació Sol·licituds:**
Sol·licituds(estat, observacions, dataIni, dataFi, dataSol, dataEstat, diesSol, codSol, codPer)
CodSol és clau primària i codPer és clau forana d' usuaris

3.3. SCRIPT PER LA CREACIÓ DE LES TAULES

```

DROP TABLE sollicituds;
DROP TABLE usuaris;
DROP TABLE sucursals;
DROP TABLE províncies;

/*-----Taula províncies -----*/
create table províncies(
província  VARCHAR(70)  CONSTRAINT NN_provínciesProvíncia NOT NULL,
CODPROV    INTEGER     CONSTRAINT NN_provínciesCodprov NOT NULL
                                CONSTRAINT PK_províncies  PRIMARY KEY
);

/*-----Taula sucursals -----*/
create table sucursals(

adreca      VARCHAR(80)  CONSTRAINT NN_sucursalsAdreca NOT NULL,
codiPostal  VARCHAR(5)   CONSTRAINT NN_sucursalsCodiPostal NOT NULL,
poblacio    VARCHAR(80)  CONSTRAINT NN_sucursalsPoblacio NOT NULL,
estat       CHAR(1)      CONSTRAINT NN_sucursalsEstat NOT NULL
                                CONSTRAINT CH_sucursalsEstat CHECK(estat IN('A','B')),

província   INTEGER     CONSTRAINT FK4_sucursals REFERENCES províncies(CODPROV ),

```

```

CODSUC      INTEGER      NOT NULL Generated Always As identity (start with 1,Increment by 1)
              CONSTRAINT PK_sucursals PRIMARY KEY

);

/*-----Taula usuaris -----*/
create table usuaris(

nom          VARCHAR(50)  CONSTRAINT NN_usuarisNom NOT NULL,
cognoms      VARCHAR(50)  CONSTRAINT NN_usuarisCognoms NOT NULL,
password     VARCHAR(8)   CONSTRAINT NN_usuarispassword NOT NULL,
estat       CHAR(1)      CONSTRAINT NN_usuarisEstat NOT NULL
              CONSTRAINT CH_usuarisEstat CHECK(estat IN('A','B')),
Adreca       VARCHAR(80)  CONSTRAINT NN_usuarisAdreca NOT NULL,
codiPostal   VARCHAR(5)   CONSTRAINT NN_usuarisCodiPostal NOT NULL,
poblacio     VARCHAR(80)  CONSTRAINT NN_usuarisPoblacio NOT NULL,
tel          VARCHAR(10)  CONSTRAINT NN_usuarisTel NOT NULL,
rol          CHAR(1)      CONSTRAINT NN_usuarisRol NOT NULL
              CONSTRAINT CH_usuarisRol CHECK(rol IN('C','E','R')),
provincia    INTEGER      CONSTRAINT FK3_usuaris REFERENCES provincies(CODPROV ),
codsuc       INTEGER      CONSTRAINT FK2_usuaris REFERENCES sucursals(CODSUC),

CODI         VARCHAR(9)   CONSTRAINT PK_usuaris PRIMARY KEY

);

/*-----Taula sollicituds -----*/

create table sollicituds(

estat       CHAR(1)      CONSTRAINT NN_sollicitudsEstat NOT NULL
              CONSTRAINT CH_sollicitudsEstat CHECK(estat IN('P','A','R','V')),

Observacions VARCHAR(250),

dataIni     DATE          CONSTRAINT NN_sollicitudsDataIni NOT NULL,
dataFi      DATE          CONSTRAINT NN_sollicitudsDataFi NOT NULL,
dataSol     DATE          CONSTRAINT NN_sollicitudsDataSol NOT NULL,
dataEstat   DATE          CONSTRAINT NN_sollicitudsDataEstat NOT NULL,
diesSol     INTEGER      CONSTRAINT NN_sollicitudsDiesSol NOT NULL
              CONSTRAINT CH_sollicitudsDiesSol CHECK(diesSol BETWEEN 0 AND 20),
codPer      VARCHAR(9)   CONSTRAINT NN_sollicitudsCodPer NOT NULL
              CONSTRAINT FK1_sollicituds REFERENCES usuaris(CODI),

CODSOL      INTEGER      NOT NULL Generated Always As identity (start with 1,Increment by 1)
              CONSTRAINT PK_sollicituds PRIMARY KEY

);
```

```
);

/*-----Dades provincies-----*/
insert into provincies(provincia, CODPROV)
    values('Girona',17);

insert into provincies(provincia, CODPROV)
    values('Barcelona',8);

insert into provincies(provincia, CODPROV)
    values('Lleida',25);

insert into provincies(provincia, CODPROV)
    values('Tarragona',43);

/*-----Dades sucursals-----*/
insert into sucursals(adreca,codiPostal,poblacio, provincia, estat)
    values('c/Nou, 25','17001','Girona',17,'A');

insert into sucursals(adreca,codiPostal,poblacio, provincia, estat)
    values('c/Sant Rafael, 54','17600','Figueres',17,'A');

insert into sucursals(adreca,codiPostal,poblacio, provincia, estat)
    values('c/Abat Oliba, 32','08203','Sabadell',8,'A');

insert into sucursals(adreca,codiPostal,poblacio, provincia, estat)
    values('Plaça Sants','08028','Barcelona',8,'A');

insert into sucursals(adreca,codiPostal,poblacio, provincia, estat)
    values('c/Agustins,13','25001','Lleida',25,'A');

insert into sucursals(adreca,codiPostal,poblacio, provincia, estat)
    values('c/Canyelles,26','43001','Tarragona',43,'A');

/*-----Dades usuaris-----*/
insert into usuaris(nom, cognoms,password,estat,adreca,codiPostal,poblacio,provincia,tel,rol, codsuc,codi)
    values('Joan','Costa Martínez','Joan9','A', 'c/Lluis Pericot, 54','17600','Figueres',17, '972236589','C',2,'40108113T');
insert into usuaris(nom, cognoms,password,estat,adreca,codiPostal,poblacio,provincia,tel,rol, codsuc,codi)
    values('Jordi','Peradalta Gil','Jordi9','A', 'Sants, 21','08028','Barcelona',08, '934012568','R',4,'40308371C');

/*-----Dades sollicituds-----*/
insert into sollicituds(estat, observacions,dataIni,dataFi,dataSol,dataEstat,diesSol,codPer)
    values('P',' ', '2010-11-22', '2010-11-26', '2010-11-17', '2010-11-17',5, '40108113T');
```

4. DISSENY DE L' ARQUITECTURA DEL SISTEMA

4.1. INTRODUCCIÓ

4.1.1. JAVA EE:

- JAVA EE (Java Enterprise Edition) . És una arquitectura que defineix el model de programació de aplicacions de n capes. Pensada per implementar aplicacions de tipus empresarial i aplicacions basades en web.
- Aquesta tecnologia suporta gran varietat de tipus d' aplicacions, des de les tipus web de gran escala a les petites de tipus client servidor.
- Està basat en pràctiques comuns i patrons de disseny que simplifiquen el desenvolupament d' aplicacions empresarials separant els components que formen cada capa
- Quan es realitzen aplicacions client/servidor el desenvolupament es divideix en un conjunt de capes destacant principalment les següents:
 - **Capa de presentació:** Es la que li serveix al client per interactuar amb la aplicació. En JEE se sol utilitzar Servlets o JSP per generar-la.
 - **Capa de lògica de negoci:** En ella s' implementen les principals funcionalitats de l' aplicació, dins trobaríem els EJB
 - **Capa de dades:** Encarregada del emmagatzemant i la gestió de les dades a través d' un sistema gestor de bases de dades.

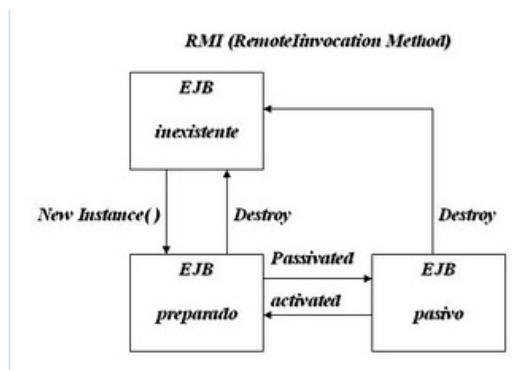
4.1.2. EJB:

- Els EJB són objectes distribuïts que s' executen dins d' uns contenidors d' EJB proporcionat per un servidor d' aplicacions
- Les aplicacions que incorporen EJB son distribuïdes , eficients, escalables, reutilitzables, configurables i portables. Fan us del servidor d' aplicacions pel suport transaccional, persistència , seguretat...
- Segons l' especificació EJB 3.1 hi ha tres tipus diferents d' EJB:
 - **EJB de Sessió:** Realitzen una tasca per un client. Són els encarregats d' interactuar amb el client ja que representen a aquest en el servidor.

Són EJB no persistents , es a dir només estan instanciats mentre dura la sessió amb el client en quant aquesta acaba es destrueixen.

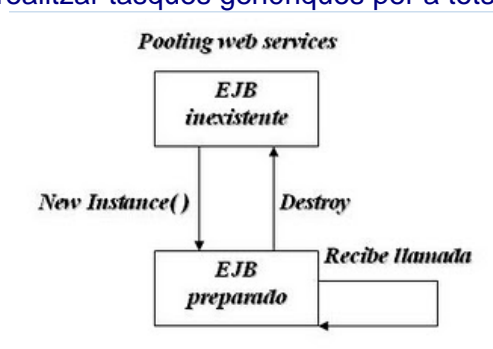
Hi ha dos tipus amb estat(stateful) i sense estat(stateless):

- **EJB de Sessió amb estat:** És aquell que permet mantenir l' estat del client mentre dura la sessió. Estan pensats per ser utilitzats per un sol client a la vegada, i per tant, utilitzar la mateixa instància del EJB en més d'una invocació. Quan un EJB de sessió amb estat porta molt temps sense utilitzar el contenidor de EJB serialitza l' objecte per alliberar memòria.



Il·lustració 38. Esquema EJB de sessió amb estat

- **EJB de Sessió sense estat:** És aquell que no guarda l'estat entre invocacions del mateix client (no guarda dades d'un client específic). Estan pensats per realitzar tasques genèriques per a tots els clients.

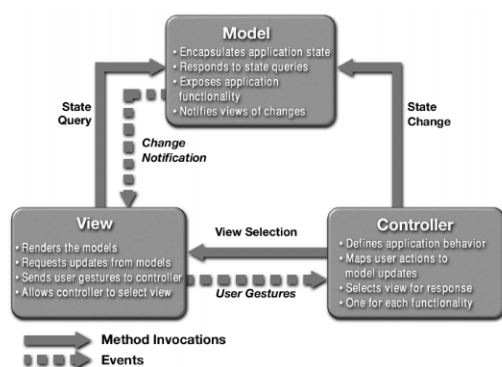


Il·lustració 39. Esquema EJB de sessió sense estat

- **Ejb d'Entitat (Classes d'entitat):** Representen un objecte de negoci i mantenen la seva persistència a través d'una base de dades
 - **EJB 3** permet delegar els serveis de persistència a la API de persistència de Java (JPA)
 - **Característiques de JPA:**
 - Permet integrar fàcilment la capa de lògica de negoci amb la capa de persistència.
 - L'ús d'anotacions permet definir les relacions existents entre entitats com: OneToOne, OneToMany, ManyToOne i ManyToMany.
 - Permet realitzar el mapatge des de les taules d'una base de dades existent creant les entitats necessàries i les seves corresponents relacions
 - Utilitza un EntityManager i un Persistence context per portar a cap les operacions d'inserció, actualització i eliminació
 - Utilitza JPQL (Java Persistence Query Language), un llenguatge de consulta similar al SQL.
- **Ejb de missatgeria:** Actua com un escoltador de JMS (Java Message Service) processant missatges de forma asíncrona

4.1.3. MVC:

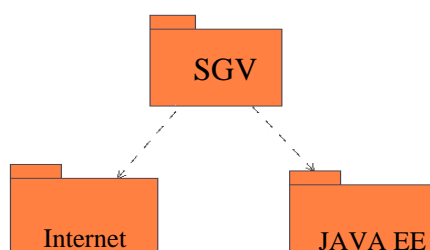
- **Model Vista Controlador(MVC)** és un patró d'arquitectura de software que separa les dades d'una aplicació, la interfície d'usuari, i la lògica de control en tres components diferents. El patró MVC es veu freqüentment en aplicacions web, a on la vista es la pàgina HTML i el codi que proveeix les dades dinàmiques a la pàgina; el model es el sistema de Gestió de Base de dades i la lògica de negoci; i el controlador es el responsable de rebre els esdeveniments d'entrada des de la vista .
- **En Java EE** la aplicació pot dividir-se en tres parts fonamentals:
 - **Model:** Conté la lògica de negoci de l'aplicació. Serà modelat per un conjunt de classes Java mitjançant EJB que s'encarregaran d'accedir a la base de dades.
 - **Vista:** Presenta la informació a l'usuari en u format adequat. L'usuari rebrà una sèrie de planes web que han sigut creades dinàmicament en el servidor mitjançant JSP
 - **Controlador:** Respon a esdeveniments, normalment accions de l'usuari e invoca canvis en el model i en la vista. En la plataforma JEE es desenvolupa mitjançant SERVLETS que fan de intermediaris entre la vista i el model.



Il·lustració 40. Esquema patró de disseny model-vista-controlador

4.2. DISSENY DE L' ARQUITECTURA DE L' APLICACIÓ SGV

4.2.1. ARQUITECTURA DE SUPORT



Il·lustració 21. Esquema arquitectura de suport

- La arquitectura de suport està composta pels següents elements:
 - **Internet Explorer 8**, permet la correcta visualització de l'aplicació.

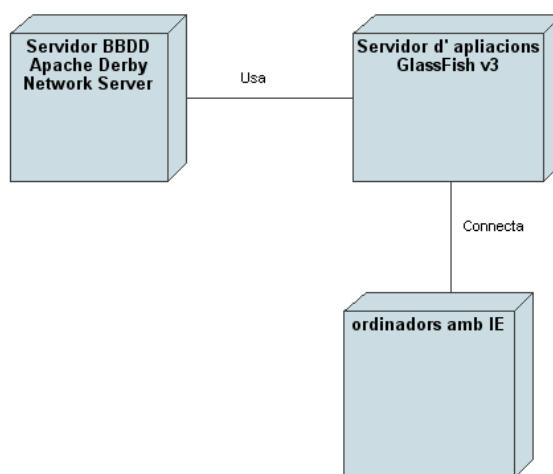
- **Java EE:**

- **IDE NetBeans 7.0 M2:** Integració total amb l' estàndard JAVA EE6: a més de incorporar suport per JSF2.0 també incorpora la llibreria de components Primefaces.
- **Servidor d' aplicacions GlassFish v3:** Servidor de software lliure desenvolupat per Sun Microsystems, companyia adquirida per Oracle Corporation que implementa totalment les tecnologies definides en la plataforma Java EE 6 i permet executar aplicacions que segueixin aquesta especificació. Es distribueix sota les llicències CDDL i GNU GLP.

En aquest servidor bàsicament es farà:

- **El deploy** de l' arxiu de distribució de l'aplicació SGV que ens genera el IDE Netbeans.
- **Crear el POOL de connexions a la base de dades**
- **Gestor de bases de dades Apache Derby :** És un sistema gestor de bases de dades relacional, escrit en Java. És un projecte open source amb llicència Apache 2.0 License

4.2.2. ESPECIFICACIÓ DE L' ENTORN TECNOLÒGIC



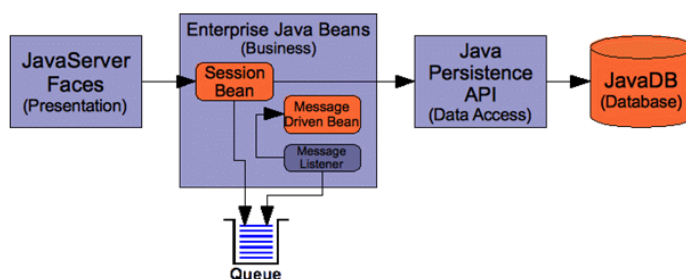
Il·lustració 22. Esquema especificació entorn tecnològic

- Es poden distingir tres nodes:
 - Node que representa els **ordinadors** dels usuaris que es connecten al servidor d' aplicacions
 - Node que representa al **servidor d' aplicacions**
 - Node que representa al **servidor de bases de dades**
- El servidor de bases de dades i el d' aplicacions podrien estar en nodes separats o en un mateix node

4.2.3. CAPES DE L' ARQUITECTURA DEL SISTEMA

- La arquitectura del sistema de l' aplicació Sistema Gestor de Vacances(SGV) estarà basada en la Arquitectura Standard JAVA EE 6 i serà un model de tres nivells o capes.
- **La capa de lògica de negoci** esta implementada amb components **EJB 3.1 (Enterprise Java Bean)** que fan us de **JPA 2.0 (Java Persistence API)** com a mecanisme de mapatge objecte/Relacional per proveir d' una capa d' accés a la base de dades.
- **La capa de presentació** es basa en **framework JSF 2.0**, pàgines JSF . S' utilitzen Facelets per la definició de les vistes, en lloc de pàgines Jsp per ser la tecnologia per defecte de JSF2.0 i per les facilitats que ofereix a l' hora de definir i utilitzar plantilles. També s' utilitza **Primefaces** que és una llibreria de components visuals open source per a JSF2.0, ens ofereix components com ara calendaris per els camps de data, barres de menús, botons, màscares per camps d' entrada, panells, suport natiu d' Ajax.....
- **La capa de dades** es basa en el gestor de bases de dades relacionals Apache Derby escrit en Java.

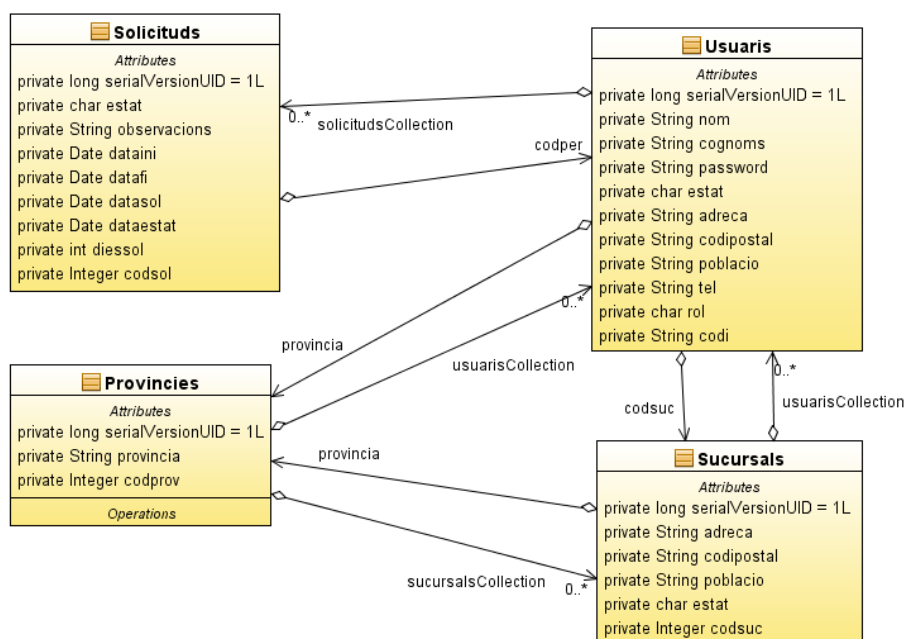
- **Representació de les tres capes:**



II-lustració 23. Esquema Model 3 capes

4.2.4. CAPA LÒGICA DE NEGOCI

4.2.4.1. DIAGRAMA CLASSES D' ENTITAT(EJB D' ENTITAT)



II-lustració 24. Diagrama classes d' entitat

4.2.4.2. DETALL DE LES CLASSES D' ENTITAT

- **Classe usuaris:** Aquesta classe conté i gestiona la informació dels usuaris.

Usuaris
<i>Attributes</i>
private long serialVersionUID = 1
private String nom
private String cognoms
private String password
private char estat
private String adreca
private String codipostal
private String poblacio
private String tel
private char rol
private String codi

Il·lustració 25. Classe entitat Usuaris

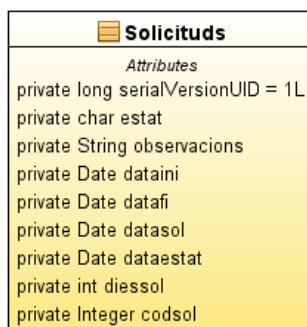
- **Atributs:**
 - **nom:** atribut tipus String que correspon amb el nom de l' usuari.
 - **cognoms:** Atribut de tipus String que correspon amb els cognoms de l' usuari
 - **password:** Atribut de tipus String que conté la contrasenya per l' usuari.
 - **estat:** Atribut de tipus Char que indica si l' usuari es troba en situació d' alta ('A') o de baixa ('B').
 - **adreca:** Atribut de tipus String que conté l' adreça de l' usuari
 - **codipostal:** Atribut de tipus String que conté el codi postal de l' adreça de l' usuari
 - **població:** atribut de tipus String que conté la població on viu l' usuari
 - **tel:** Atribut de tipus String que guarda el telèfon de contacte de l' usuari.
 - **rol:** Atribut de tipus Char que guarda el rol que té l' usuari que accedeix al sistema. El rol pot ser responsable de la empresa ('r'), cap de sucursal ('c') o empleat ('e').
 - **codi:** Atribut tipus String que conte el codi d' usuari.
- **Classe Sucursals:** Aquesta classe conté i gestiona la informació de les sucursals

Sucursals
<i>Attributes</i>
private long serialVersic
private String adreca
private String codiposta
private String poblacio
private char estat
private Integer codsuc

Il·lustració 26 Classe entitat Sucursals

- **Atributs**
 - **estat:** Atribut de tipus Char que indica si la' sucursal es troba en situació d' alta ('A') o de baixa ('B').
 - **adreca:** Atribut de tipus String que conté l' adreça de la sucursal
 - **codipostal:** Atribut de tipus String que conté el codi postal de l' adreça de la sucursal.
 - **població:** atribut de tipus String que conté la població on està ubicada la sucursal
 - **codsuc:** Atribut de tipus String que conté el codi de la sucursal.

- **Classe Sollicituds:**Aquesta classe conte i gestiona la informació de les sol·licituds de vacances dels usuaris.

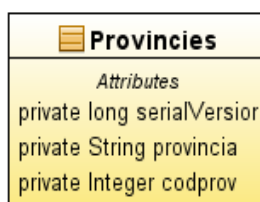


Il·lustració 27. Classe entitat Sol·licituds

- **Atributs:**

- **estat:** Atribut de tipus Char que guarda l' estat en que es troba la sol·licitud d' un usuari , els seus valors poden ser 'A' acceptada, 'R' refusada, 'P' petició, 'V' validada.
- **observacions:** Atribut de tipus String conté les observacions realitzades pel empleat o el cap de sucursal o el Responsable de l' Empresa.
- **dataIni:** Atribut de tipus Date que conté la data inicial del període de vacances sol·licitat.
- **dataFi:** Atribut de tipus Date que conté la data final del període de vacances sol·licitat.
- **dataSol:** Atribut de tipus date que guarda la da en que es realitza la sol·licitud per l' usuari.
- **dataEstat:** Atribut de tipus date que gurada la data en que es produeix una variació en l' estat de la sol·licitud de l' usuari (pas de petició a acceptada o refusada).
- **diesSol:** Atribut de tipus int que guarda el número de dies que compren el període sol·licitat de vancances.
- **codSol:** Atribut de tipus int que conté el codi de sol·licitud.

- **Classe Provincies:**Aquesta classe conte i gestiona la informació sobre les províncies on es troben ubicades les sucursal i a on viuen els usuaris.



Il·lustració 28. Classe entitat Sol·licituds

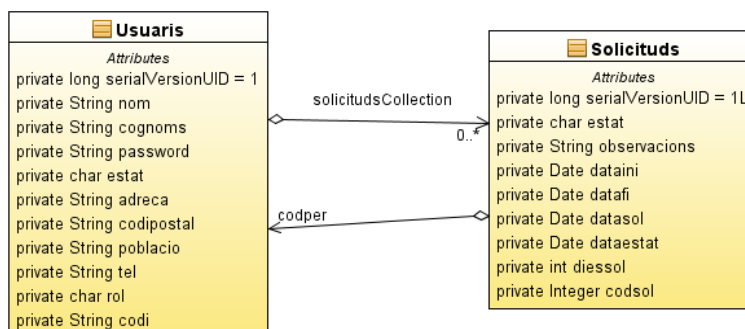
- **Atributs:**

- **provincia:** Atribut de tipus String que conté el nom de la província
- **codprov:** Atribut de tipus Integer conté el codi de província

4.2.4.3. RELACIONS ENTRE LES CLASSES ENTITAT

- **Usuaris amb sol·licituds**

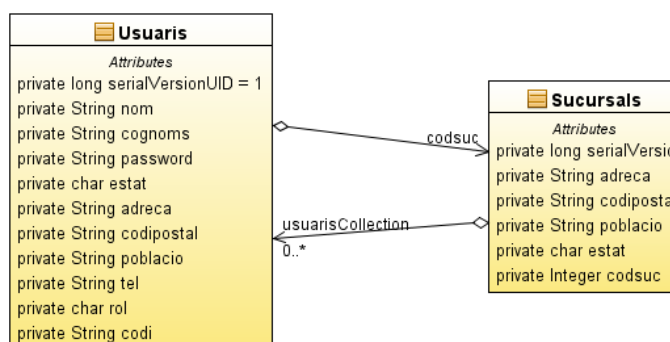
- **Relació 1 a varis:** tota sol·licitud pertany a un sol usuari i un usuari pot tenir varies sol·licituds.
La relació se estableix a través de l'atribut `codper` de la classe entitat sol·licituds que s'associa amb l'atribut `codi` de la classe entitat usuaris.



Il·lustració 29. Relació entre les classes entitat usuaris i sol·licituds

- **Usuaris amb sucursals.**

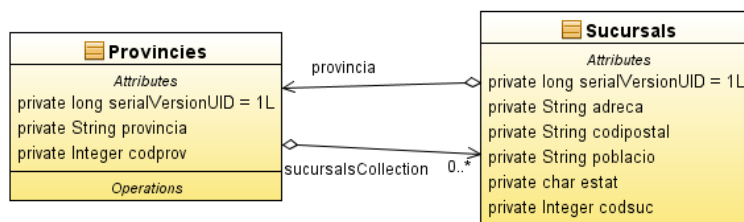
- **Relació 1 a vairs:** Tot usuari pertany a un sola sucursal i en una mateixa sucursal hi poden haver varis usuaris.
La relació s'estableix a través de l'atribut `codSucursal` de la classe entitat sucursal que s'associa amb l'atribut `codSucursal` de la classe entitat usuaris.



Il·lustració 30. Relació entre les classes entitat usuaris i sucursals

- **Províncies amb sucursals**

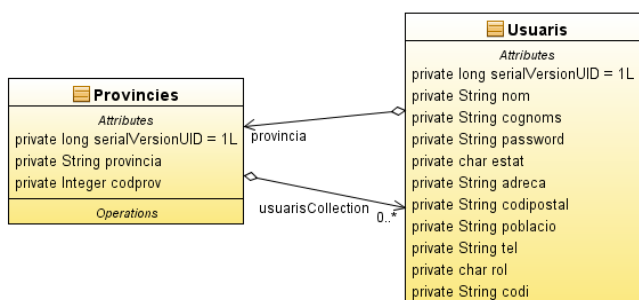
- **Relació 1 a vairs:** Tota sucursal pertany a un sola província i en una mateixa província hi poden haver varis sucursals.
La relació s'estableix a través de l'atribut `provincial` de la classe entitat sucursal que s'associa amb l'atribut `codprov` de la classe entitat províncies.



Il·lustració 31. Relació entre les classes entitat provincies i sucursals

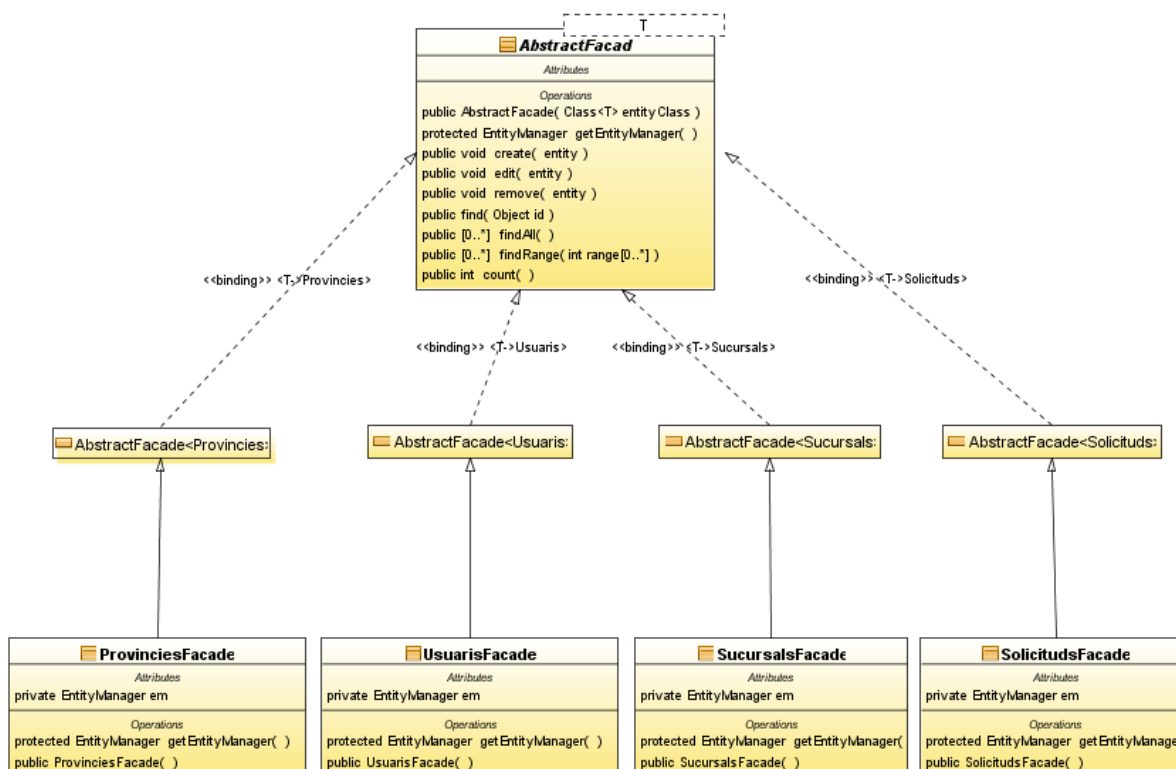
• Provincies amb usuaris

- **Relació 1 a vairs:** Tot usuari pertany a un sola província i en una mateixa província hi poden haver varis usuaris.
La relació s' estableix a través de l' atribut provincia de la classe entitat usuaris que s' associa amb l'atribut codprov de la classe entitat provincies.



il·lustració 32. Relació entre les classes entitat provincies i usuaris

4.2.4.4. DIAGRAMES DELS EJB DE SESSIÓ



Il·lustració 33. Diagrama dels EJB de sessió

4.2.4.5. DETALL DELS EJB DE SESSIÓ

- Les classes Ejb de sessió **SucursalsFacade**, **UsuarisFacade**, **SolicitudsFacade** i **ProvinciesFacade** són subclasses de la superclasse abstracta **Abstractfacade** de tal manera que existeix una relació d' herència entre les subclasses i la superclasse.
- Cada classe Ejb de sessió hereta de abstract facade els mètodes per fer diversos tipus de cerques i operacions com ara crear noves sucursals, usuaris o sol·licituds, editar, validar usuaris, eliminar....
- Tots els Ejb de sessió són del tipus sense estat, es a dir no guarden l' estat entre invocacions del mateix client, no guarden dades d' un client específic.

4.2.5. CAPA DE PRESENTACIÓ:

Les responsabilitats de la capa de presentació bada en JSF es distribueixen en tres components:

- **Pàgines JSF:** Fitxers XHTML a on es defineix la disposició i propietats dels components JSF de la presentació WEB
- **Managed Beans:** Classes Java que proveeixen les dades a presentar en les pàgines JSF i els mètodes invocats per les accions desencadenades pels esdeveniments de la pagina JSF.
- **Llibreria de Components Primeface:** dona opció a poder utilitzar en les pàgines jsf una gran multitud de components, com ara calendaris per la inserció de dades de tipus data, barres de menús, botons, panells, fitxes tabulars.....




4.2.5.1. PÀGINES JSF

- Totes les pàgines JSF que formen part de l' aplicació comparteixen la mateixa plantilla(definida en el fitxer **template.xhtml** conforme a la sintaxis dels Facelets), de tal manera que totes les pàgines tenen una capçalera, un cos, i un peu.
- Cada una de les pàgines JSF defineixen la seva pròpia zona de contingut:
 - **Login.xhtml:** Presenta un formulari per fer login.
 - **MenúEmpleat. xhtml:** Presenta un formulari amb una barra de menús amb les funcionalitats bàsiques del usuari amb rol d' empleat.
 - **MenúCapSucursal.xhtml:** Presenta un formulari amb una barra de menús amb les funcionalitats bàsiques del usuari amb rol de cap de sucursal.
 - **MenuResponsable.xhtml:** Presenta un formulari amb una barra de menús amb les funcionalitats bàsiques del usuari amb rol de Responsable d' empresa.
 - **AltaUsuari.xhtml:** Presenta un formulari per donar d'alta un usuari.
 - **consultaModUsuari.xhtml:**Presenta un formulari per fer les consultes i modificacions d' usuaris.
 - **AltaSucursal.xhtml:** Presenta un formulari per donar d' alta sucursals.
 - **ConsultaModSucursal.xhtml:** Presenta un formulari per consultar i modificar sucursals
 - **AltaSolicituds.xhtml:** Presenta un formulari per donar d' alta sol·licituds de vacances
 - **ConsultaModSolicituds.xhtml:** Presenta un formulari per consultar i modificar sol·licituds de vacances.
 - **CercaUsuaris.xhtml:** Presenta un formulari per fer cerques d' usuaris per diferents criteris.

- **CercaSucursals.xhtml**: Presenta un formulari per fer cerques de sucursals per diferents criteris.
- **CercaSolicituds.xhtml**: Presenta un formulari per fer cerques de sol·licituds per diferents criteris.
- **Llistatusuaris.xhtml**: Presenta un llistat d' usuaris que compleixen un criteri de cerca.
- **LlistaSucursals.xhtml**: Presenta un llistat d' sucursals que compleixen un criteri de cerca.
- **LlistaSolicituds.xhtml**: Presenta un llistat de sol·licituds que compleixen un criteri de cerca.

4.2.5.2. MANAGED BEANS

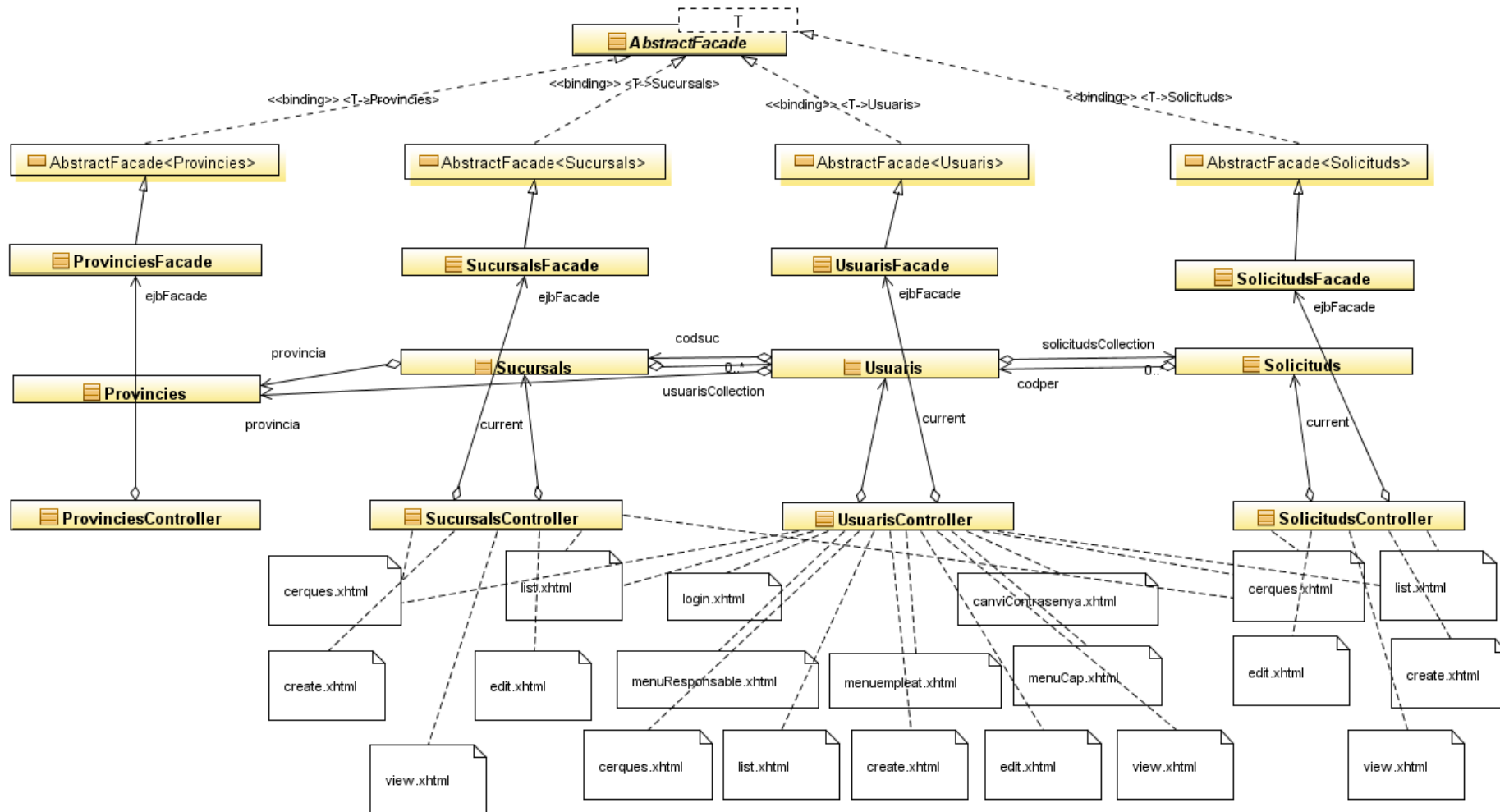
- A la aplicació es defineixen 4 Managed Beans:

 SucursalsController	 SolicitudsController	 UsuarisController
<p><i>Attributes</i></p> <pre>private DataModel items = null private int selectedRowIndex private String SucursalSel private String tipusLlistat = "" private ArrayList parametres = new ArrayList() private int codiSucursal = 0</pre> <p><i>Operations</i></p> <pre>public SucursalsController() public Sucursals getSelected() private SucursalsFacade getFacade() public PaginationHelper getPagination() public String BuscarSucursalPerCodi(String pantall public String BuscarSucursalsPerPoblacio() public int getCodiSucursal() public void setCodiSucursal(int codiSucursal) public String getPaginaOrigen() public String prepareList() public String prepareView() public String prepareCreate() public String create() public String prepareEdit() public String update() private void updateCurrentItem() public DataModel getItems() private void recreateModel() public String next() public String previous() public SelectItem[0..*] getItemsAvailableSelectOne public SelectItem[0..*] getItemsAvailableSelectOne</pre>	<p><i>Attributes</i></p> <pre>private int TotalDiesVacances = 20 private DataModel items = null private int selectedRowIndex private String tipusLlistat = "" private ArrayList parametres = new ArrayList() private String opcionsEstat[0..*] package Calendar c1 = Calendar.getInstance() private Date dataIniciCerca = c1.getTime() private Date dataFiCerca = c1.getTime() private int codiSolicitud</pre> <p><i>Operations</i></p> <pre>public SolicitudsController() public Solicituds getSelected() private SolicitudsFacade getFacade() public PaginationHelper getPagination() public String prepareList() public String prepareView() public String prepareCreate() public String create() public String prepareEdit() public String update() private void updateCurrentItem() public DataModel getItems() private void recreateModel() public int getDiesPendents() public int getDiesPendentsEdit() public int getDiesSolicitud() public String BuscarSolicitudPerDates() public String BuscarSolicitudPerCodi(String p public Date getDataIniciCerca() public void setDataIniciCerca(Date dataIniciC public Date getDataFiCerca() public void setDataFiCerca(Date dataFiCerca public int getCodiSolicitud() public void setCodiSolicitud(int codiSolicitud public String[0..*] getOpcionsEstat() public void setOpcionsEstat(String opcionsEst public String getPaginaOrigen() public String next() public String previous()</pre>	<p><i>Attributes</i></p> <pre>private DataModel items = null private int selectedRowIndex private String tipusLlistat = "" private ArrayList parametres = new ArrayList()</pre> <p><i>Operations</i></p> <pre>public UsuarisController() public Usuaris getSelected() private UsuarisFacade getFacade() public PaginationHelper getPagination() public String prepareList() public String prepareView() public String prepareCreate() public String create() public String prepareEdit() public String update() public void updateContrasenya() private void updateCurrentItem() public DataModel getItems() private void recreateModel() public String next() public String previous() public String getUsuarioval() public String getRolval() public String getSucval() public String getCarrecval() public String getPaginaOrigen() public void CancelarUsuari() public void IdSessioUsuari() public String BuscarUsuari() public String BuscarUsuariPerCodi(String pantalla public String BuscarUsuariPerCodSuc() public String BuscarUsuariPerCognoms() public String getdataSistema()</pre>
<p> ProvinciesController</p> <p><i>Attributes</i></p> <p><i>Operations</i></p> <pre>public ProvinciesController() public SelectItem[0..*] getItemsAvailableSelectOne(</pre>		

Il·lustració 34. Managed Beans

- Tots els Managed Beans tenen l' abast de sessió (**@SessionScoped**), per lo que els seus atributs estaran disponibles mentre duri la sessió de l' usuari(o fins que caduqui). El seguiment de la sessió es responsabilitat dels servlet JSF, el programador només ha de declarar el tipus d' abast.
- Tots els Managed Beans delegen tota la seva funcionalitat als EJBs de la capa de negoci per lo que les seves úniques funcions estan relacionades amb la gestió de les interaccions derivades de l' acció de l' usuari:
 - Fer disponibles les dades a mostrar, que a la vegada s' obtindran dels EJBs.
 - Mantenir les dades introduïdes pel usuari durant la seva interacció amb l' aplicació.
 - Oferir els mètodes encarregats de gestionar els esdeveniments generats pels Components JSF de l' aplicació.
 - Delegen en els EJBs les operacions d' accés a dades i les responsabilitat d' implementar els casos de us de l' aplicació.
 - Determinar el flux entre pàgines JSF mitjançant els valors de retorn (String) dels manejadors d' accions(links i botons) desencadenades per l' usuari.
 - Inserció dels missatges d'error en l' arbre de components de la pàgina JSF(mitjançant el FacesContext)

4.2.6. DIAGRAMA DE RELACIÓ ENTRE PÀGINES JSF(XHTML), MANAGED BEANS I EJBs



il·lustració 35. Diagrama de relació entre pàgines Jsf(Xhtml), Managed Beans i Ejbs

5. PROVES REALITZADES

5.1.1. PROVES AMB LES DADES I ELS REQUISITS FUNCIONALS

- S' han fet proves amb usuaris de diferents rols verificant les seves funcionalitats i els requisits funcionals de l' aplicació
- S' han verificat que les dades es guarden correctament a la base de dades

5.1.2. PROVES AMB LES SESSIONS

- S' ha verificat que un cop validat un usuari, si copiem en el navegador una adreça corresponen a una plana xhtml i l' enganxem a una altre sessió del navegador, l' aplicació ens mostra la pantalla de login perquè detecta que es vol accedir a una plana xhtml sense haver fet login prèviament.

5.1.3. PROVES DE VISUALITZACIÓ I FUNCIONAMENT EN VARIS NAVEGADORS

S' ha provat l' aplicació utilitzant els següents navegadors

- Internet Explorer 8
- Mozilla Firefox
- Google Chrome

En tots els navegador l' aplicació funciona i es visualitza correctament.

6. INSTAL·LACIÓ

6.1.1. INSTAL·LACIÓ DEL JDK

- Descarregar el fitxer jdk-6u14-windows-i586.exe o versió posterior de l' adreça: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Instal·lar amb les opcions per defecte

6.1.2. INSTAL·LAR EL SERVIDOR D' APLICACIONS GLASSFISH I LA BASE DE DADES APACHE DERBY

- Descarregar el fitxer glassfish-3.0.1-windows.exe de la següent adreça:
- <http://glassfish.java.net/downloads/3.0.1-final.html>

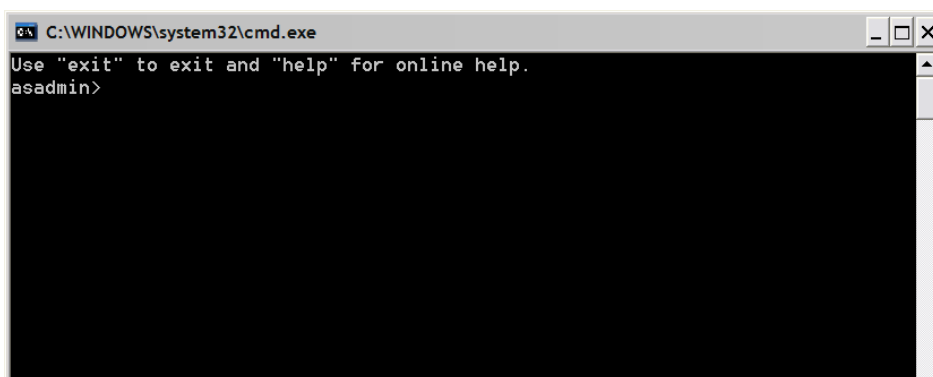
6.1.3. MODIFICACIÓ DE LES VARIABLES D'ENTORN

- Valors de les variables per una instal·lació del servidors **Glassfish i Apache Derby** en la carpeta **C:\glassfishv3**
 - **Crear la variable d' usuari CLASSPATH amb el següent valor:**
%DERBY_HOME%;%JAVA_HOME%;C:\Archivos de programa\Java\jdk1.6.0_14\lib;
 - **Crear la variable DERBY_HOME amb el següent valor:**
C:\glassfishv3\javadb
 - **Crear la variable d' usuari JAVA_HOME amb el següent valor:**
C:\ Archivos de programa \Java\jdk1.6.0_14
 - **Modificar la variable PATH, amb el següent:**

C:\Archivos de programa\jdk1.6.0_14\bin\C:\glassfishv3\javadb\bin

6.1.4. ENEGGAR EL SERVIDOR D' APLICACIONS GLASSFISH

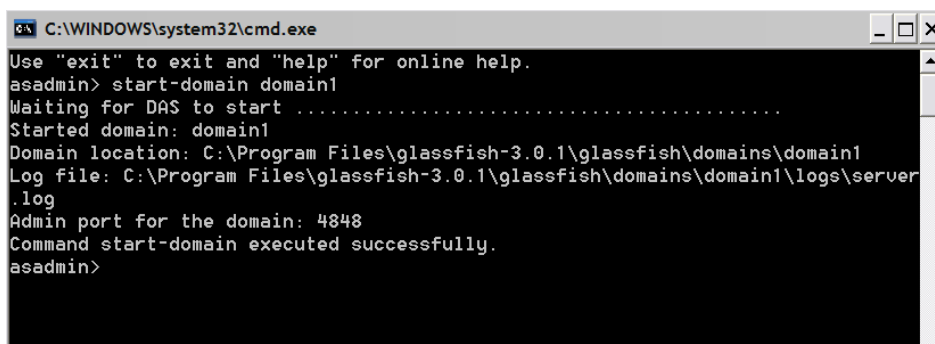
- Per una instal·lació del servidors **Glassfish i Apache Derby** en la carpeta **C:\glassfishv3**
 - Executar l' arxiu **c:\glassfishv3\bin\asadmin.bat**



```
C:\WINDOWS\system32\cmd.exe
Use "exit" to exit and "help" for online help.
asadmin>
```

Il·lustració 36. Engegar el servidor Glassfish

- Escriure en el prompt la següent instrucció: **start-domain domain1**
Al fer intro ens sortirà la següent pantalla que ens indica que els servidor Glassfish està en funcionament.

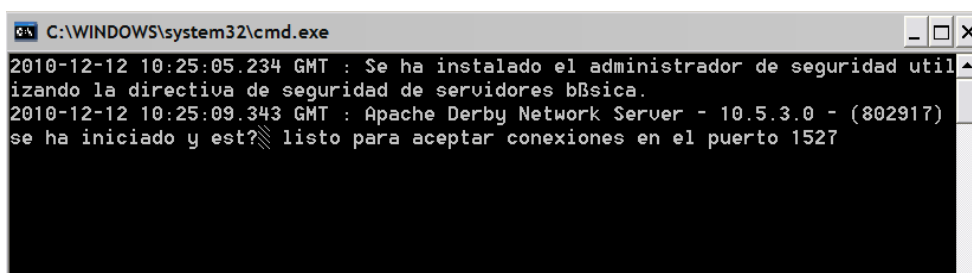


```
C:\WINDOWS\system32\cmd.exe
Use "exit" to exit and "help" for online help.
asadmin> start-domain domain1
Waiting for DAS to start .....
Started domain: domain1
Domain location: C:\Program Files\glassfish-3.0.1\glassfish\domains\domain1
Log file: C:\Program Files\glassfish-3.0.1\glassfish\domains\domain1\logs\server
.log
Admin port for the domain: 4848
Command start-domain executed successfully.
asadmin>
```

Il·lustració 37. Engegar el servidor Glassfish

6.1.5. ENEGGAR EL SERVIDOR DE BASE DE DADES APACHE DERBY

- Executar l' arxiu **C:\glassfishv3\javadb\bin\startNetworkServer.bat**
- Al fer intro ens sortirà la següent pantalla que ens indica que els servidor Apache Derby està en funcionament.



```
C:\WINDOWS\system32\cmd.exe
2010-12-12 10:25:05.234 GMT : Se ha instalado el administrador de seguridad utilizando la directiva de seguridad de servidores básica.
2010-12-12 10:25:09.343 GMT : Apache Derby Network Server - 10.5.3.0 - (802917) se ha iniciado y está listo para aceptar conexiones en el puerto 1527
```

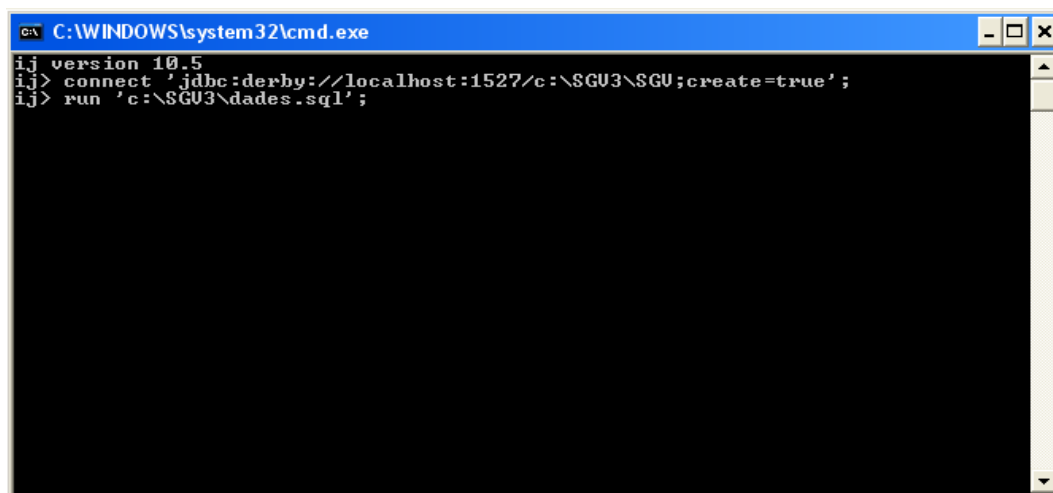
Il·lustració 38. Engegar el servidor de base de dades Apache Derby

6.1.6. CREAR LA BASE DE DADES

- Executar l' arxiu: **C:\glassfishv3\javadb\bin\ij.bat**
- Per crear la base de dades anomenada SGV a la carpeta c:\sgv3, executar el comandament:
Connect 'jdbc:derby://localhost:1527/C:\SGV3\SGV;create=true';

Per executar el script de creació de taules **dades.sql** ubicat a la carpeta c:\sgv3, executar el comandament:

run 'c:\sgv3\dades.sql';



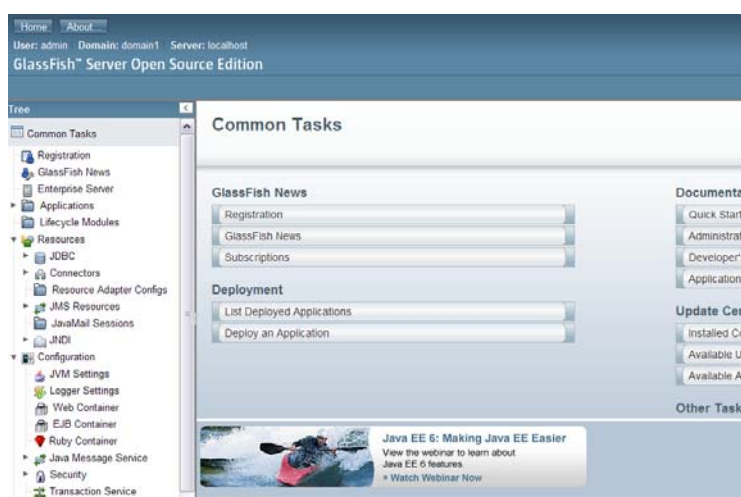
```

C:\WINDOWS\system32\cmd.exe
ij version 10.5
ij> connect 'jdbc:derby://localhost:1527/c:\SGU3\SGU;create=true';
ij> run 'c:\SGU3\dades.sql';
  
```

Il·lustració 39.Creació de la base de dades i execució de l' escrit de creació de taules

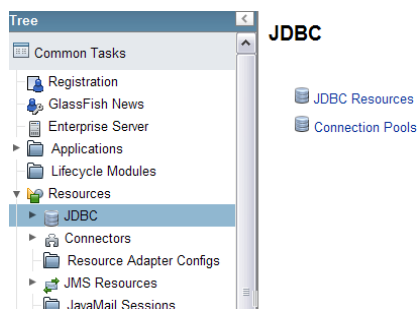
6.1.7. CONFIGURAR EL SERVIDOR GLASSFISH

- **Entrar a la Consola del Servidor**
 - Escriure la següent adreça al navegador per entrar a la consola del servidor:
<http://localhost:4848>
 - Fer login amb l' usuari: admin i password: admin



Il·lustració 40.Consola del servidor GlassFish

- **Crear el pool de connexions:**
 - Fer clic sobre Resources->JDBC:



Il·lustració 41. Creació pool de connexions

- Fer clic sobre Connection Pools i seguidament sobre el botó new:



Il·lustració 42. Creació pool de connexions

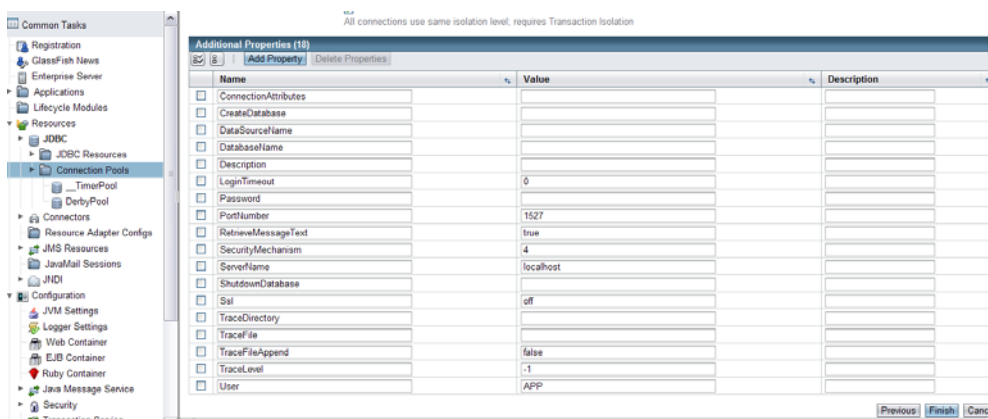
- Omplir els camps amb les següents dades

Name: **SGV**

Resource Type : **javax.Sql.DataSource**

Database Vendor: **Derby**

- Fer clic sobre el botó next per accedir a la següent pantalla:



Il·lustració 43. Propietats pool de connexions

- Modificar les propietats addicionals de manera que quedin les que es mostren a Continuació.

Edit JDBC Connection Pool Properties

Save Cancel

Modify properties of an existing JDBC connection pool.

Additional Properties (7)			
Name	Value	Description	
<input type="checkbox"/> DatabaseName	C:\SGV3\SGV		
<input type="checkbox"/> User	SGV		
<input type="checkbox"/> Password	SGV		
<input type="checkbox"/> driverClass	org.apache.derby.clientDriver		
<input type="checkbox"/> URL	jdbc:derby://localhost:1527/SGV		
<input type="checkbox"/> ServerName	localhost		
<input type="checkbox"/> PortNumber	1527		

Il·lustració 44. Modificació de les propietats del pool de connexions

- Un cop fet això, fer clic sobre el botó finish perquè quedi creat el pool de connexions SGV:

Pool Name	Resource Type	Classname
<input type="checkbox"/> TimerPool	javax.sql.XADataSource	org.apache.derby.jdbc.EmbeddedXADataSource
<input type="checkbox"/> DerbyPool	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource
<input checked="" type="checkbox"/> SGV	javax.sql.DataSource	org.apache.derby.jdbc.ClientDataSource

Il·lustració 45. Pool de connexions SGV creat

- Comprovar la connexió amb la base de dades:
A la pantalla anterior fer clic sobre el pool SGV per accedir a la següent pantalla on farem clic al botó ping.

Ping Succeeded

Edit JDBC Connection Pool

Modify an existing JDBC connection pool. A JDBC connection pool is

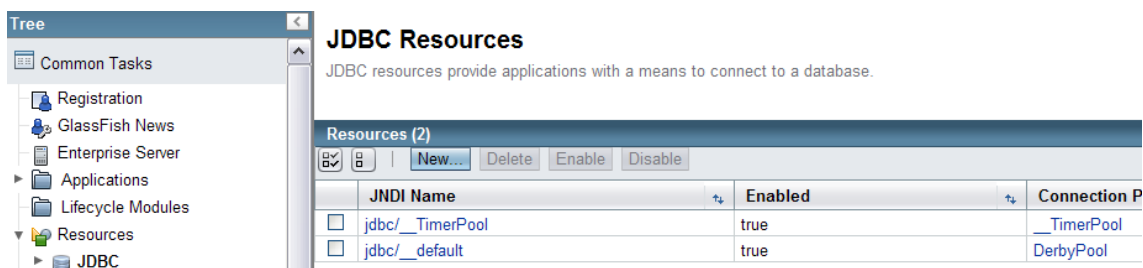
Load Defaults Flush Ping

Il·lustració 46. Ping de connexió a la base de dades

En cas de no fer ping s'han de revisar les propietats del pool de connexions i comprovar les variable de entorn i que tant el servidor Derby com el Glassfish estan engegats

- **Crear el JNDI**

- Fem clic a JDBC/Resources



Il·lustració 47. Pantalla JDBC/Resources de la consola del servidor Glassfish

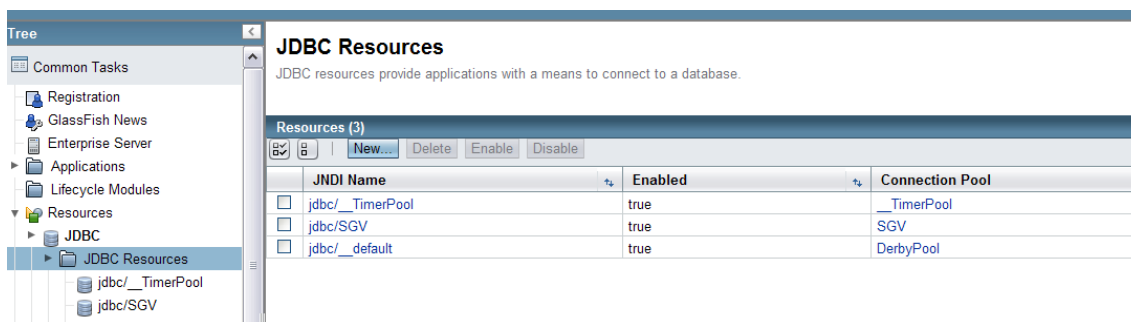
- A la pantalla anterior fer clic sobre el botó new i omplir les següents dades:



Il·lustració 48. Creació del JNDI

JNDI Name: **jdbc/SGV**
Pool Name: **SGV**

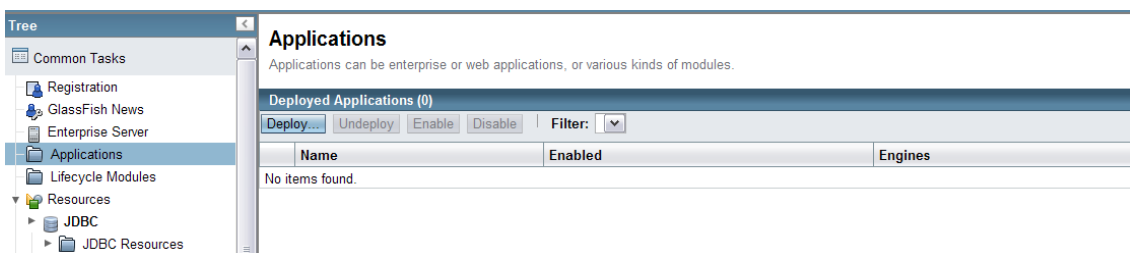
- Al fer clic en el botó ok el JNDI queda creat:



Il·lustració 49.. JNDI un cop creat

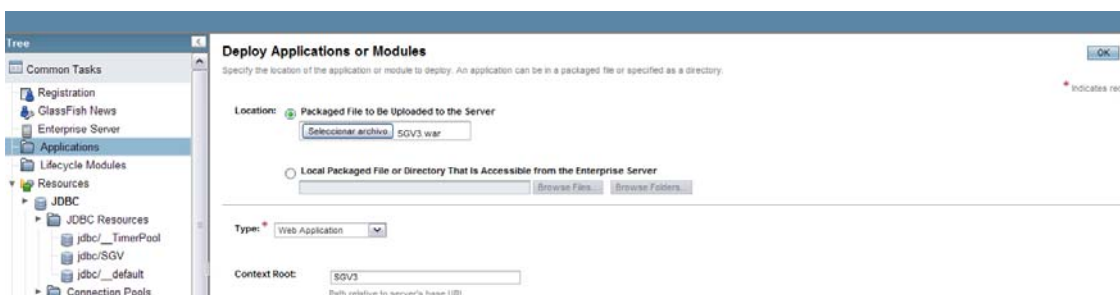
- **Deploy de l' aplicació**

- Fer clic sobre Applications i després sobre el botó deploy



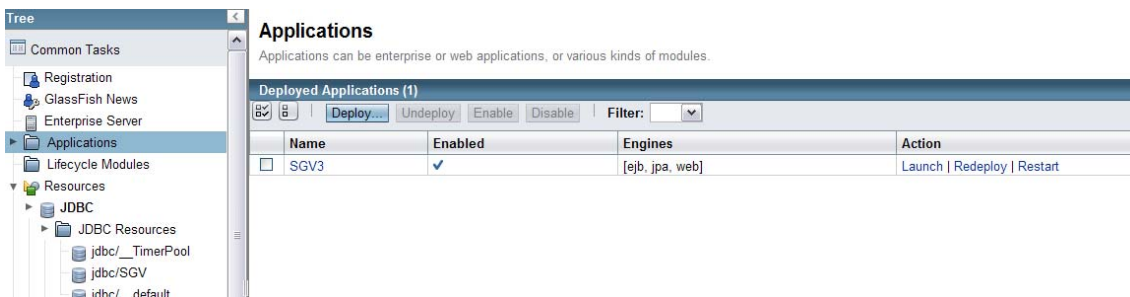
II-lustració 50.Pantalla Applications de la consola del servidor Glassfish

- Seleccionar l' arxiu de l' aplicació



II-lustració 51.Realitzar el Deploy

- Al fer clic sobre el botó ok es realitza el deploy



II-lustració 52.Deploy realitzat

7. Manual utilització:

- Amb l' script "dades.sql" també s' incorporen les següents dades corresponents als 3 rols d' usuaris de l' aplicació:

Usuaris per provar:

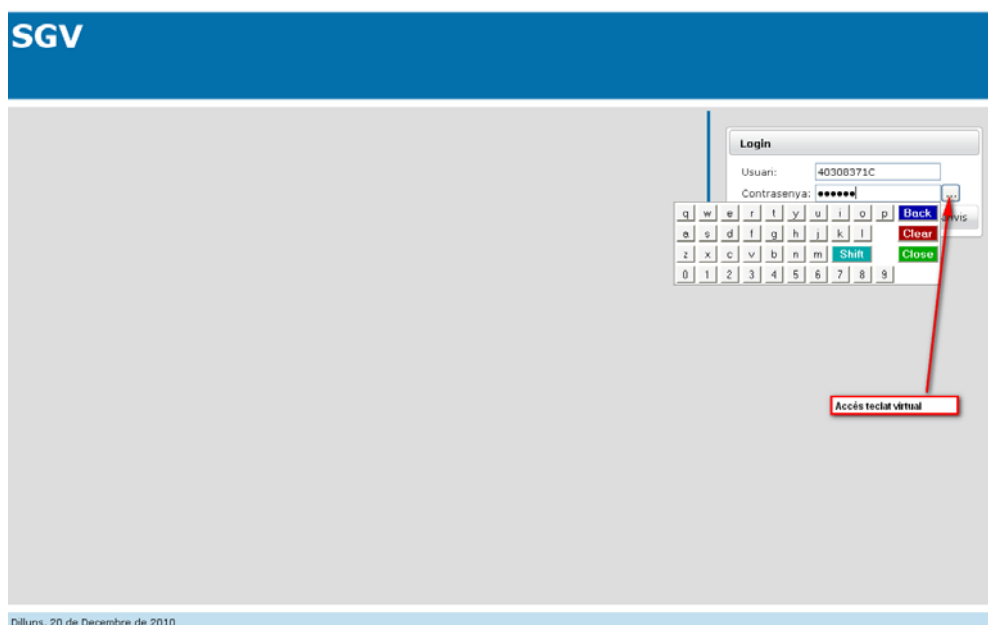
Rol	Usuari	Contrasenya
Responsable d' Empresa	40308371C	JORDI9
Empleat	40308374C	JOSEP9
Cap Sucursal	40308372C	MARGA9

7.1.1. EXECUTAR L' APLICACIÓ

Introduir la següent adreça en el navegador:: <http://localhost:8080/SGV3/>

- **Pantalla per fer Login:**

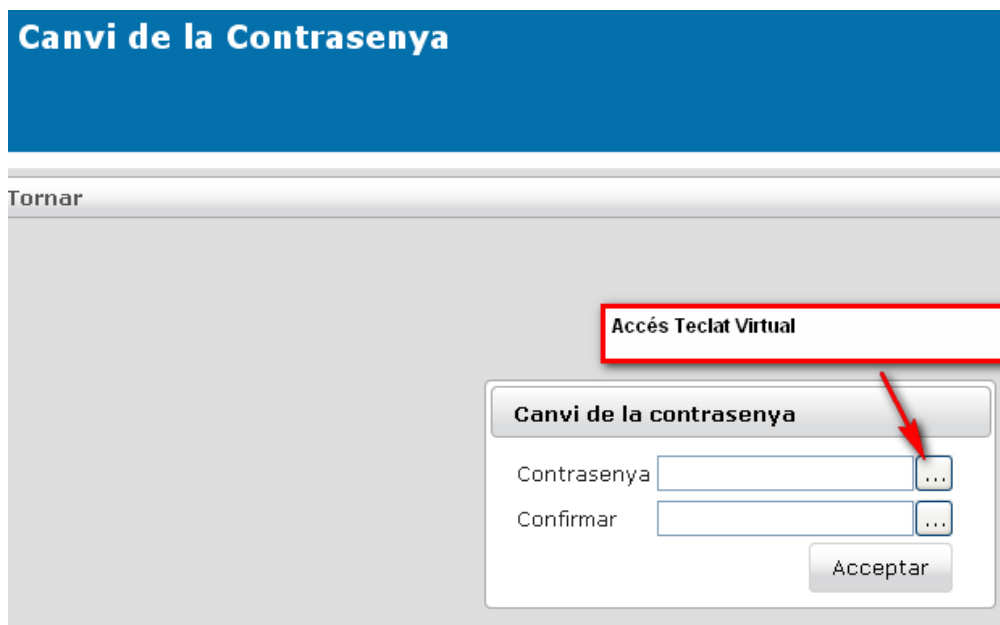
- La contrasenya es pot introduir directament teclejant, o fent clic en el teclat virtual
- Els camps no distingeixen majúscules de minúscules a excepció del camp contrasenya
- Quan es fa clic en el botó acceptar s' accedeix a la pantalla de menú que li correspon al tipus d' usuari que s' ha validat
- Els usuaris que estan en situació de baixa no poden entrar en el sistema, surt un missatge d' usuari en situació de baixa.



Il·lustració 53.Pantalla de login

- **Pantalla per modificar contrasenya**

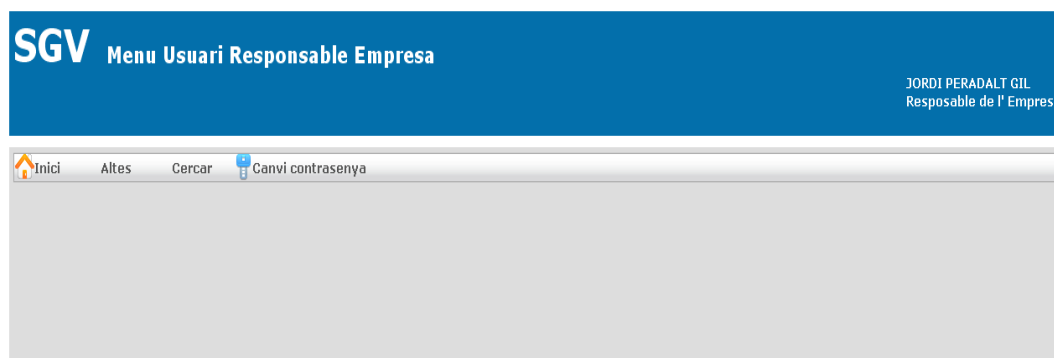
- Tots els usuaris poden canviar la contrasenya quan ho creguin oportú.



Il·lustració 54. Pantalla modificar contrasenya

- **Pantalla menú usuari Responsable empresa**

- Dona accés a totes les funcionalitats d' aquest tipus d' usuari
- A la capçalera sempre s' indica en quina opció es troba l' usuari, qui s' ha validat i el seu rol.
- Al peu de pàgina sempre hi figura la data del sistema



Il·lustració 55. Pantalla Menú de l' usuari Responsable Empresa

- **Menu altes (Responsable empresa)**

- Per el tipus d' usuari amb rol Responsable d' Empresa permet accedir a altes d' usuaris i altes de sucursals



Il·lustració 56. Pantalla menú altes de l'usuari Responsable Empresa

- **Pantalla Alta Usuaris (Responsable empresa)**
 - Permet donar d'alta un usuari
 - No permet entrar un codi d'usuari duplicat

Il·lustració 57. Pantalla alta usuari

- El camp codi postal té una mascara d' entrada que només permet entrar números
- El camp tel també té una mascara d' entrada amb el format de telèfon
- El botó desfer canvis serveix per esborrar el valor de les variables que hem escrit fins el moment.

Password:
 Adreça:
 Codipostal:
 Població:
 Província: ---
 Tel:
 Rol: EMPLEAT
 Codi:
 Sucursal Assignada: ---

L'usuari ha sigut creat.

II-lustració 58.Missatge conforme s' ha creat usuari

- Quan s' han gravat les dades ,el sistema avisa (això ho fa per totes les pantalles, altes sucursals, usuaris, sol·licituds...)
- Quan es produeix algun error, el sistema també avisa (això ho fa per totes les pantalles, altes sucursals, usuaris, sol·licituds...)

- **Pantalla Alta Sucursals (Responsable empresa)**

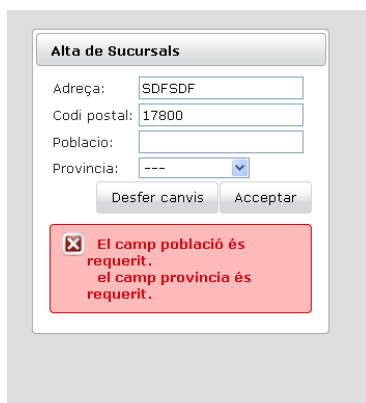
- Permet donar d' alta una sucursal
- No permet entrar un codi de sucursal duplicat

SGV Alta de sucursals

Alta de Sucursals
 Adreça:
 Codi postal:
 Població:
 Província: ---

II-lustració 59.pantalla alta de sucursals

- Exemple d'avis d'error, al fer acceptar sense omplir tots els camps requerits



The screenshot shows a web form titled "Alta de Sucursals". It contains four input fields: "Adreça:" with the value "SDFSDF", "Codi postal:" with the value "17800", "Població:" which is empty, and "Província:" which is a dropdown menu showing "---". Below the fields are two buttons: "Desfer canvis" and "Acceptar". A red error message box is displayed below the form, containing the text: "El camp població és requerit. el camp província és requerit."

Il·lustració 60.Exemple missatge d'error

- **Menú cerques (Responsable empresa)**

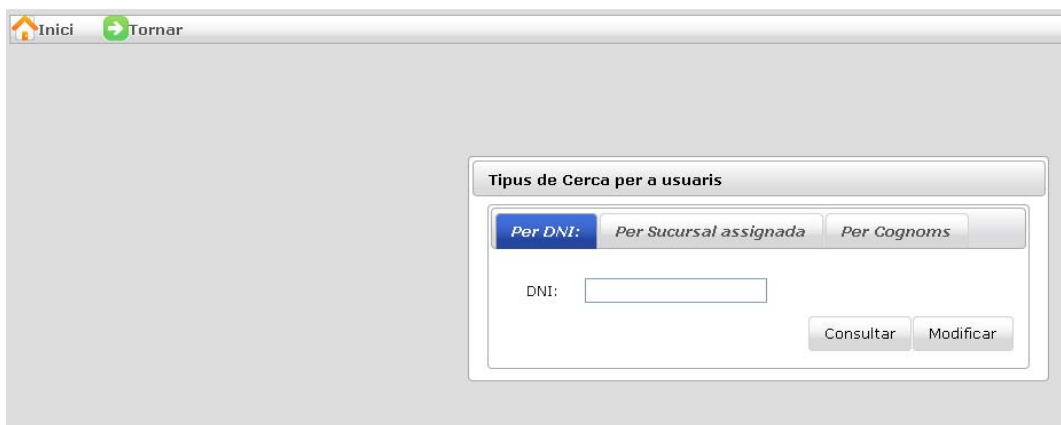
- Permet accedir a les cerques d'usuaris, sucursals i de les sol·licituds de vacances que han fet els usuaris empleats de qualsevol sucursal



Il·lustració 61.Menú cercar de l'usuari Responsable empresa

- **Pantalla cerques d'usuaris (Responsable empresa)**

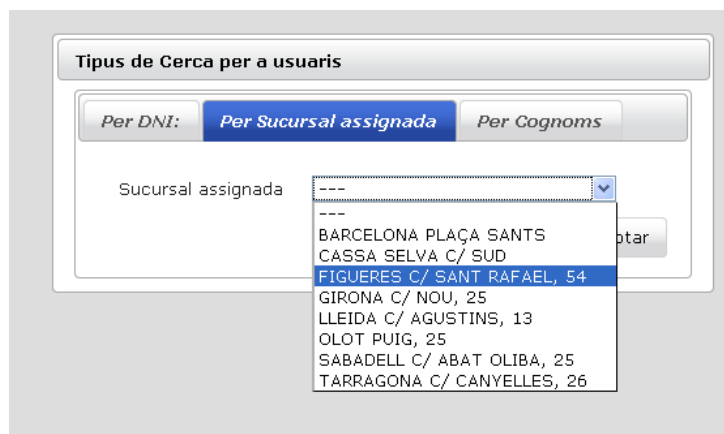
- Fent clic a les diverses pestanyes permet cercar usuaris segons tres criteris: Per DNI, per Sucursal assignada i per cognoms



The screenshot shows a search interface titled "Tipus de Cerca per a usuaris". It features three tabs: "Per DNI:" (selected), "Per Sucursal assignada", and "Per Cognoms". Below the tabs is a text input field labeled "DNI:". At the bottom right of the search area are two buttons: "Consultar" and "Modificar". The interface also includes "Inici" and "Tornar" navigation links at the top left.

Il·lustració 62.Cerques d'usuaris

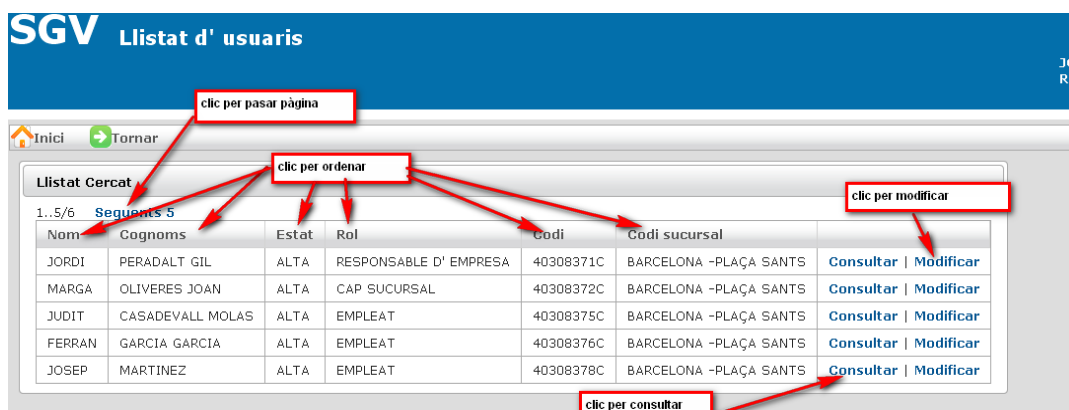
o Exemple de cerca per sucursal assignada



Il·lustració 63. Exemple de cerca per sucursal assignada

o Exemple de llistat d' usuris trobats en la cerca per sucursal assignada

- Aquest llistat es pot ordenar per qualsevol de les columnes
- Cada fila dona accés a la pantalla de consulta o Modificació de les dades del usuari
- Si no es visualitzen tots els usuaris en una sola pantalla es mostra un panell de navegació per consultar anteriors i següents.



Il·lustració 64. Llistat d' usuaris

- Exemple de **pantalla de consulta d' un usuari**

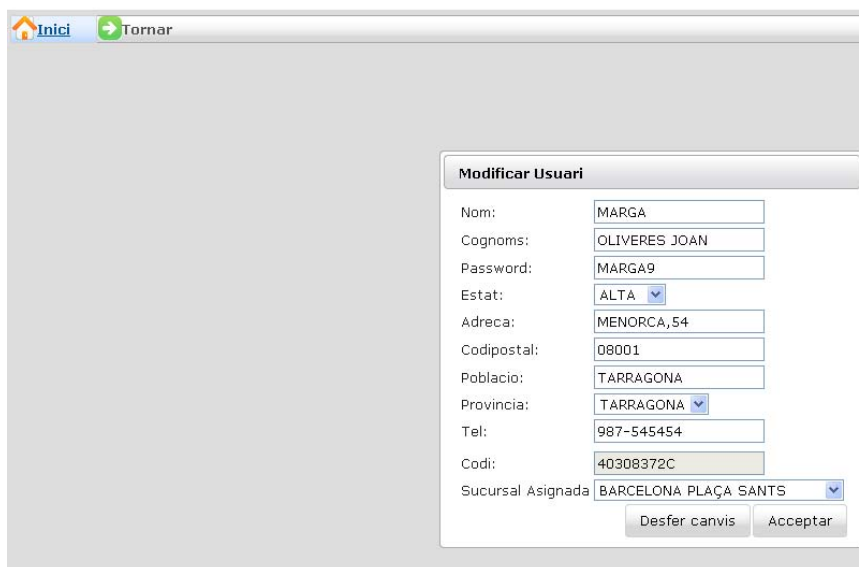


The screenshot shows a web application interface with a blue header bar. Below the header, there are two buttons: 'Inici' (with a home icon) and 'Tornar' (with a right arrow icon). The main content area is light gray. A white box with a gray border and title 'Consulta Usuari' is positioned on the right side. It contains the following user information:

Nom:	JORDI
Cognoms:	PERADALT GIL
Estat:	ALTA
Adreça:	C/SANTS, 21
Codi postal:	08028
Poblacio:	GIRONA
Provincia:	17
Tel:	934-012568
Rol:	R
Codi:	40308371C
Codisucursal:	4
Sucursal assignada:	BARCELONA -PLAÇA SANTS

Il·lustració 65.consulta d' usuari

- Exemple de **pantalla de modificació d' un usuari**
 - Canviant l' estat a baixa donem l' usuari de baixa



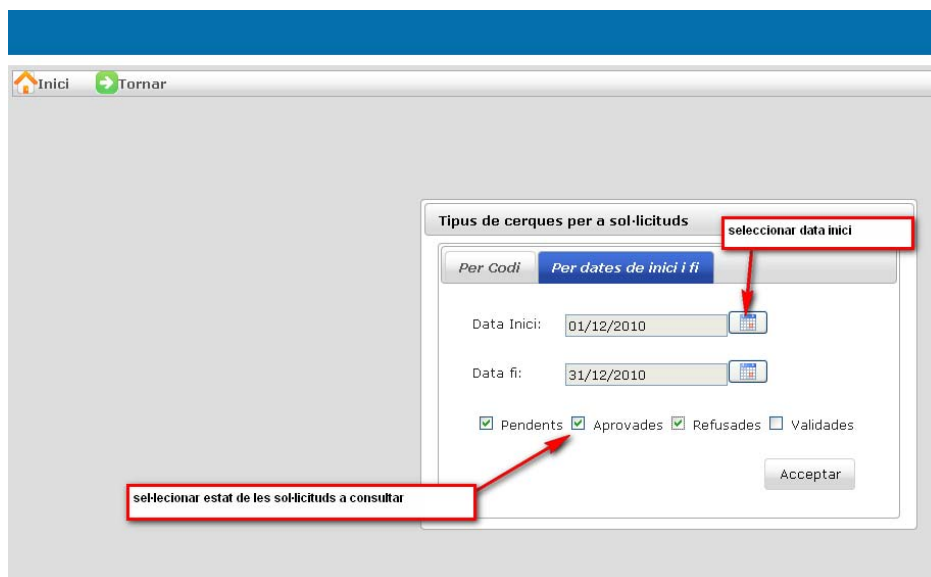
The screenshot shows the same web application interface as above. A white box with a gray border and title 'Modificar Usuari' is positioned on the right side. It contains the following user information, with input fields for modification:

Nom:	<input type="text" value="MARGA"/>
Cognoms:	<input type="text" value="OLIVERES JOAN"/>
Password:	<input type="text" value="MARGA9"/>
Estat:	<input type="text" value="ALTA"/>
Adreça:	<input type="text" value="MENORCA,54"/>
Codipostal:	<input type="text" value="08001"/>
Poblacio:	<input type="text" value="TARRAGONA"/>
Provincia:	<input type="text" value="TARRAGONA"/>
Tel:	<input type="text" value="987-545454"/>
Codi:	<input type="text" value="40308372C"/>
Sucursal Assignada:	<input type="text" value="BARCELONA PLAÇA SANTS"/>

At the bottom of the form are two buttons: 'Desfer canvis' and 'Acceptar'.

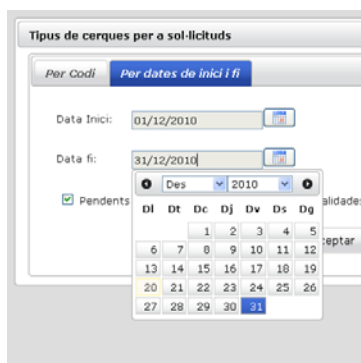
Il·lustració 66.pantalla modificació d' un usuari

- **Pantalla cerques de sol·licituds de vacances (Responsable empresa)**
 - Permet cercar les sol·licituds que han fet els usuaris dins un rang de dates i segons el estat en que es troba la sol·licitud (Pendent, aprovada, refusada, validada)



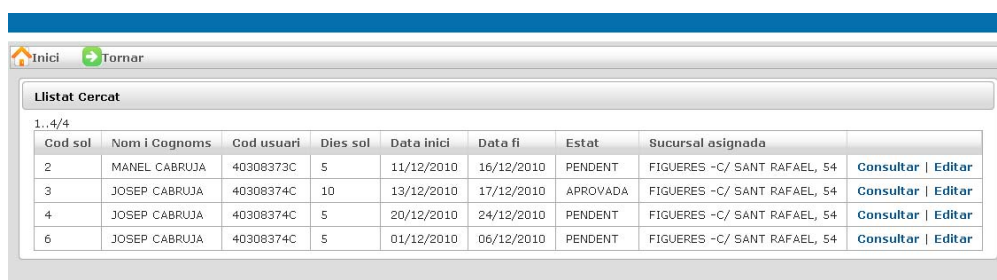
Il·lustració 67. Cerca de sol·licituds de vacances

- o Exemple de selecció de data



Il·lustració 68. Selecció de data

- o Exemple de llistat d' sol·licituds trobades en la cerca de sol·licitud de vacances
 - Aquest llistat es pot ordenar per qualsevol de les columnes
 - Cada fila dona accés a la pantalla de consulta o Modificació de les dades de la sol·licitud
 - Si no es visualitzen totes les sol·licituds en una sola pantalla es mostra un panell de navegació per consultar anteriors i següents.



1. 4/4

Cod sol	Nom i Cognoms	Cod usuari	Dies sol	Data inici	Data fi	Estat	Sucursal assignada	
2	MANEL CABRUJA	40308373C	5	11/12/2010	16/12/2010	PENDENT	FIGUERES -C/ SANT RAFAEL, 54	Consultar Editor
3	JOSEP CABRUJA	40308374C	10	13/12/2010	17/12/2010	APROVADA	FIGUERES -C/ SANT RAFAEL, 54	Consultar Editor
4	JOSEP CABRUJA	40308374C	5	20/12/2010	24/12/2010	PENDENT	FIGUERES -C/ SANT RAFAEL, 54	Consultar Editor
6	JOSEP CABRUJA	40308374C	5	01/12/2010	06/12/2010	PENDENT	FIGUERES -C/ SANT RAFAEL, 54	Consultar Editor

II-lustració 69.Llistat de sol·licituds de vacances

- Exemple de **pantalla de modificació d' una sol·licitud de vacances**
 - El camp dies pendents: indica el dies pendent de vacances que té l' usuari
 - L' usuari Responsable empresa, només pot modificar els camps estat amb el valor de validat i el camp d' observacions



Modificació de sol·licituds

Cod sol:

Estat:

Data estat:

Data sol:

Data inici:

Data fi:

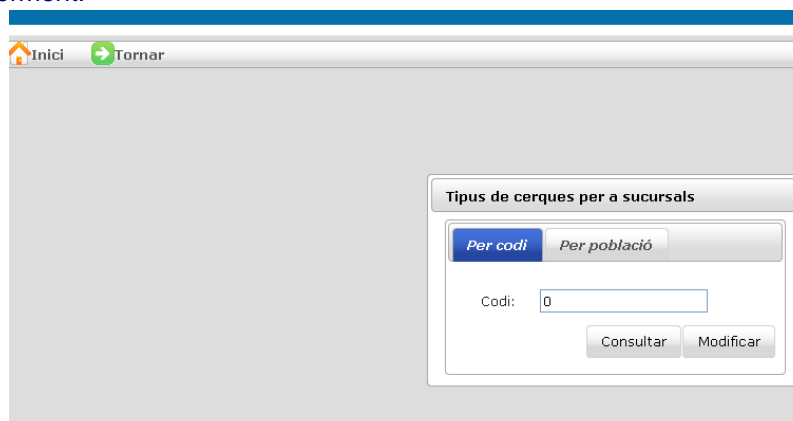
Dies pendents:

Dies sol:

Observacions:

II-lustració 70.Modificació de sol·licitud de vacances

- **Pantalla cerques de sucursals (Responsable empresa)**
 - Permet fer cerques de sucursals
 - Les pantalles de llistat de sucursals, modificació i consulta són similars a les descrites anteriorment.



Tipus de cerques per a sucursals

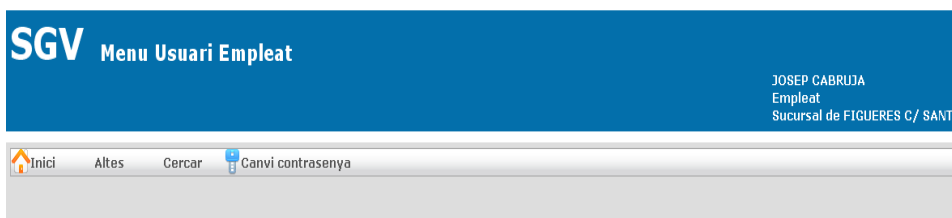
Per codi Per població

Codi:

II-lustració 71.Cerca de sucursal

- **Pantalla de menú Usuari Empleat**

- Dona accés a totes les funcionalitats d' aquest tipus d' usuari
- A la capçalera sempre s' indica en quina opció es troba l' usuari, qui s' ha validat i el seu rol.
- Al peu de pàgina sempre hi figura la data del sistema



Il·lustració 72. Pantalla menú usuari empleat

- **Pantalla Alta de sol·licituds (Usuari Empleat)**

- Permet donar d' alta sol·licituds de vacances als usuaris amb rol d' empleat
- Si dies pendents es zero el sistema no deixarà fer cap més sol·licitud de vacances
- El número de dies sol·licitats ha de ser com a mínim 5 i com a màxim 20
- Les dates s' entren fen clic en el botó que dona accés al calendari.
- Si la data de inici és posterior a la data de fi, el programa avisa, també avisa de qualsevol altre error.

Il·lustració 73. Alta d' una sol·licitud de vacances

- **Resta pantalles (Usuari Empleat)**

- Són similars a les ja descrites, només indicar que les funcionalitat d' aquest usuari són les següents:
 - Donar d' alta sol·licituds de vacances ,
 - Cercar només les seves sol·licituds de vacances
 - Modificar només les seves sol·licituds mentre el seu estat sigui pendent

- Consultar només les seves sol·licituds
- Canviar de contrasenya

- **Pantalla de menú Cap sucursal**

- Dona accés a totes les funcionalitats d' aquest tipus d' usuari
- A la capcelera sempre s' indica en quina opció es troba l' usuari, qui s' ha validat i el seu rol.
- Al peu de pàgina sempre hi figura la data del sistema



Il·lustració 74. Pantalla menú usuari Cap Sucursal

- Totes les pantalles són similars a les descrites.
- Aquest usuari només pot visualitzar les sol·licituds de vacances que han fet els usuaris de la seva sucursal.
- Aquest usuari només pot modificar les sol·licituds de vacances per canviar el seu estat a aprovat o refusat

8. CONCLUSIONS

La realització d' aquest Treball de fi de carrera, ha sigut una experiència molt interessant i enriquidora.

Al començar aquest TFC el principal objectiu que tenia era el de poder aconseguir tenir una idea de com es realitza el desenvolupament d' aplicacions Web usant la tecnologia java. En la carrera d' ETIG no hi ha assignatures enfocades específicament al desenvolupament d' aquest tipus d' aplicacions i avui en dia és de vital importància en el món empresarial i professional ja que l' imposició d' Internet a nivell empresarial ha fet que les aplicacions d' empreses usin la xarxa como a vehicle de comunicació bàsic. També moltes de les aplicacions antigues s' estan portant cap aquest entorn per la possibilitat d' usar la web como a interfície. Crec que aquest objectiu ha estat aconseguit amb la realització d' aquest treball.

Una de les dificultats que he tingut ha estat la gran quantitat de conceptes nous, especificacions, components, frameworks i entorns de desenvolupament que hi ha entorn al món J2EE. Això comporta una corba d' aprenentatge molt forta. Per a mi, el primer va ser llegir molt sobre el tema, tot tipus d' articles que anava trobant per Internet sobre conceptes bàsics como el patró de disseny MVC i els conceptes relacionats amb les tres capes que especifica aquest patró com els EJB d' entitat, de sessió, servlets, pàgines JSP..., petits exemple d' aplicacions.

Un cop vaig tenir una idea dels conceptes bàsics, una altre dificultat que vaig trobar va ser la de decidir quina tecnologia utilitzar sense tenir molt clar les avantatges i desavantatges i les dificultats que podria torbar-me derivades de l' elecció.

Al final l'elecció va ser l'utilització del IDE Netbeans 7.0 M2, EJBs, el framework JSF2.0, la llibreria de components Primefaces, el servidor d'aplicacions Glassfish i el gestor de base de dades Apache Derby. El motiu d'aquesta elecció va ser que l'IDE Netbeans està molt ben integrat amb tota aquesta tecnologia i disposa de molta ajuda per el desenvolupament de aplicacions web, como ara assistents que et generen l'esquelet d'una aplicació, la qual cosa és de gran ajuda per comprendre el seu funcionament.

El que he trobat també molt interessant és la llibreria de components Primefaces, la manera tan fàcil d'implementar en les planes xhtml components como ara taules amb tot tipus de funcionalitats, calendaris per la introducció de dates, barres de menús. etc. He trobat que aquesta llibreria et facilita molt la feina i que dona un aspecte molt professional a l'aplicació.

Una altra cosa que també voldria destacar és la flexibilitat, i potencia que suposa l'utilització del framework JSF2.0, els Managed Beans, les planes xhtml i la navegació entre elles.

També he trobat molt sencilla la instal·lació i configuració del servidor d'aplicacions Glassfish i del gestor de bases de dades Apache Derby. Amb l'execució del fitxer d'instal·lació del servidor Glassfish també s'instal·la l'Apache Derby. La consola del servidor és relativament sencilla, al menys per la configuració bàsica d'un pool de connexions a la base de dades i per fer el deploy de l'aplicació.

Amb a realització d'aquest TFC he vist i he utilitzat una part del que representa el món J2EE i m'ha donat peu a aprofundir sobre els temes vistos i a interessar-me en l'aprenentatge de altres tecnologies como ara els Serveis Web, altres components, frameworks...

Como a conclusió final voldria dir que la realització d'aquest treball ha sigut una experiència molt bona i que he aconseguit els objectius que m'havia proposat en un principi.

9. GLOSSARI

Paraules molt utilitzades en aquesta memòria:

- **JAVA EE (Java Enterprise Edition):** És una arquitectura que defineix el model de programació de aplicacions de n capes. Pensada per implementar aplicacions de tipus empresarial i aplicacions basades en web.
- **Els EJB:** Són objectes distribuïts que s'executen dins d'un contenidor d'EJB proporcionat per un servidor d'aplicacions.
- **EJB de Sessió:** Realitzen una tasca per un client. Són els encarregats d'interactuar amb el client ja que representen a aquest en el servidor.
- **EJB d'Entitat (Classes d'entitat):** Representen un objecte de negoci i mantenen la seva persistència a través d'una base de dades.
- **JPA (Java Persistence API):** És un framework del llenguatge de programació Java que maneja dades relacionades en aplicacions usant la Plataforma Java EE i Java SE
- **JPQL (Java Persistence Query Language):** un llenguatge de consulta similar al SQL.
- **Patró de disseny:** És una solució a un problema que es repeteix sovint. La solució ha de ser fàcilment aplicable sempre que sorgeix el mateix problema. Són solucions basades en l'experiència i que s'ha demostrat que funcionen.

- **MVC (Model Vista Controlador)** és un patró d'arquitectura de software que separa les dades d'una aplicació, la interfície d'usuari, i la lògica de control en tres components diferents. El patró MVC es veu freqüentment en aplicacions web, a on la vista es la pàgina HTML i el codi que proveeix les dades dinàmiques a la pàgina; el model és el sistema de Gestió de Base de dades i la lògica de negoci; i el controlador és el responsable de rebre els esdeveniments d'entrada des de la vista .
- **Framework:** És un conjunt de biblioteques, eines i normes a seguir que ajuden a desenvolupar aplicacions. Permeten la reutilització de codi, l'estandardització del desenvolupament i l'utilització del cicle de desenvolupament de tipus interactiu - incremental (especificació, codificació, manteniment i evolució).
- **JSF 2.0:** És una tecnologia i framework per aplicacions Java basades en web que simplifica el desenvolupament d'interfícies d'usuari en aplicacions Java EE.
- **Facelets:** és un framework simplificat de presentació, on es possible dissenyar de forma lliure una pàgina web i després associar-li els components JSF específics. és un framework per plantilles (templates) centrat en la tecnologia JSF. Utilitza el format XHTML.
- **Managed Beans:** Classes Java que proveeixen les dades a presentar en les pàgines JSF i els mètodes invocats per les accions desencadenades pels esdeveniments de la pàgina JSF.
- **Primefaces** És una llibreria de components visuals Open Source per a JSF2.0, ens ofereix components com ara calendaris per els camps de data, barres de menús, botons, màscares per camps d'entrada, panells, suport natiu d'Ajax.....
- **GlassFish v3:** Servidor de software lliure desenvolupat per Sun Microsystems, companyia adquirida per Oracle Corporation que implementa totalment les tecnologies definides en la plataforma Java EE 6 i permet executar aplicacions que segueixin aquesta especificació. Es distribueix sota les llicències CDDL i GNU GPL.
- **Apache Derby :** És un sistema gestor de bases de dades relacional, escrit en Java. És un projecte open source amb llicència Apache 2.0 License

10. BIBLIOGRAFIA

- **Alejandro Pèrez García.** JSF 2 ya está aquí !!! The JSF Return, ahora más sencillo que nunca !!!
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jsf2Return>
- **Cristóbal González Almirón.** Introducción a JSF Java
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava>
- **Junta de Andalucía.** JavaServer Faces(JSF)
<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/JSF>

- **José Manuel Sánchez Suárez.** Facelets en JSF 2: sistema de plantillas y componentes por composición.
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jsf2FaceletsTemplatesAndCompositeComponents>
- **Cay Horstmann.** JavaServer Faces 2.0.
<http://jdurrett.ba.ttu.edu/Resources/Programming/JavaServerFaces-Horstmann.pdf>
- **Suport de JSF 2.0 en l'IDE NetBeans 6.8**
http://netbeans.org/kb/docs/web/jsf20-support_ca.html#managedBean
- **Beatriz Bonilla Sánchez.** VALIDACIONES Y CONVERSIONES EN JSF
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=convValidJSF>
- **Customized J2EE Training.**JSF 2.0 Tutorials
<http://www.coreservlets.com/JSF-Tutorial/jsf2/>
- **Tutorial de JavaServer Faces**
<http://www.sicuma.uma.es/sicuma/Formacion/documentacion/JSF.pdf>
- **Cédric Simon.** Curso JavaServer Faces y Ajax
<http://www.scribd.com/doc/28079982/Curso-de-JSF-2-0>
- **Eric Jendrock, Ian Evans, Kim Haase, Chinmaye Srivathsa.** The Java EE6 Tutorial
<http://download.oracle.com/javasee/6/tutorial/doc/>
- **Eric Jendrock, Jennifer Ball, Debbie Carson, Ian Evans, Scott Fordin, Kim Haase.** El tutorial de java EE 5
<http://java.cabezudo.net/trabajos/JEE5/manual/jee5.v0.01.00/index.html>
- **Epidata consulting** Bloque de Conocimiento de EJB
<http://www.epidataconsulting.com/tikiwiki/tiki-index.php?page=EJB#JNDI>
- **Enrique Viñé Lerma.** Introducción a Primefaces
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=introduccionPrimefaces>
- **Primefaces**
<http://www.primefaces.org/showcase/ui/keyboard.jsf>
- **Oracle** JPQL language Reference
http://download.oracle.com/docs/cd/E16764_01/apirefs.1111/e13046/ejb3_langref.html