

DynBTS Dynamic Bug Tracker System

Eduard Calveras i Anguita
Enginyeria Tècnica d'Informàtica de Sistemes

Albert Grau i Perisé

25 de Juny de 2008

A la meva estimada Marta per la paciència que ha tingut aquests anys durant la meva carrera, i perquè en tot moment m'ha donat el suport necessari per continuar estudiant. I al meu fill Biel que no ha tingut per part meva l'atenció suficient que es mereix quan s'acaba d'arribar al món, tot i que ell sap que me l'estimo molt.



Continguts

Aquest projecte es basa en la tecnologia JEE i mostra una part de les capacitats que es pot aconseguir a partir d'ella.

Passarem per diferents eines d'aquesta tecnologia com són l'Apache Struts i l'Hibernate que són els pilars d'aquest projecte, però sense deixar de banda altres més conegudes com les JSP o XHTML que sense elles no tindriem interfície visual.

S'explicarà en detall l'aplicació del projecte en la vida real, la interacció amb l'usuari i també el procés que es realitza en el Servidor on estarà ubicada.

Finalment és farà un recorregut pel desenvolupament d'aquest projecte, des de l'especificació fins al funcionament, passant pels casos d'us, persistència, disseny, etc.

Espero que la lectura d'aquesta memòria, si més no, s'us faci amena i tanmateix interessant.

Índex General

Capítol I

1. Introducció	5
2. Objectius.	6
3. Mètode Seguit	7
4. Planificació.	8
5. Producte Obtingut.	9
6. Resta de la Memòria.	10

Capítol II

1. Especificació	11
2. Anàlisi	
Guions.	12
Casos d'ús	
<i>a. Diagrames.</i>	<i>14</i>
<i>b. Descripcions.</i>	<i>16</i>
Sistema funcional	25
Diagrames d'estat.	27
3. Disseny	
Persistència	28
Classes.	31
<i>a. Classes de Sistema.</i>	<i>33</i>
Diagrama de seqüència.	35
Interfícies.	37

Capítol III

1. Valoracions i Conclusions.	42
Glossari.	43
Bibliografia i Recursos Web.	44



Índex de figures

Casos d'ús

1. Grup Invitats.	14
2. Grup Depuradors.	14
3. Grup Desenvolupadors.	15
4. Grup Supervisors.	15
5. Administrador.	16

Sistema funcional	25
------------------------------------	----

Diagrames d'estat

1. Cicle de vida d'un Error.	27
--------------------------------------	----

Disseny

1. Persistència.	28
2. Classes	
<i>Entorn d'Usuari.</i>	31
<i>Entorn de Producte.</i>	31
<i>Entorn d'Incidència.</i>	32
<i>Entorn de Comentari</i>	32
<i>Classes Base.</i>	33
<i>Classes del Sistema.</i>	33
<i>Sistema de Paquets.</i>	34
3. Seqüències	
<i>Login.</i>	35
<i>Entrada d'un Error.</i>	36
4. Interfícies	
<i>Login.</i>	37
<i>Llista d'Usuaris.</i>	37
<i>Edició d'Usuari.</i>	38
<i>Llista de Grups.</i>	38
<i>Vista de Producte.</i>	39
<i>Edició de Component</i>	39
<i>Creació d'un Error.</i>	40

Capítol I

Introducció

De l'experiència en el meu treball diari, vaig veure necessària una aplicació que al igual que el BugZilla permetés la introducció d'errors de Programari general i el seu posterior manteniment, però no podia treballar amb una base de dades única, i tampoc amb un programa que treballés amb un únic idioma, i més sent una societat que es globalitza i cada cop és més multilingüe. A més a més hi havia la necessitat de que l'aplicació no només fes un enregistrament d'errors, addicionalment havia de poder registrar millores d'aquests programaris, ja que a la vida real normalment són aquestes millores la que fan evolucionar un producte.

En aquest projecte parlarem d'Incidències quan ens referim a Errors o a Millores. Parlarem de Productes quan ens referim a un Programari concret, i Components a les diferents parts de que consta un Producte.

DynBTS que és el nom que he donat al projecte prové de (Dynamic Bug Tracker System) Sistema Traçador d' Errors Dinàmic, que no és res més que la capacitat que té de:

- Treballar amb Diferents Base de Dades
- Tenir un entorn Multilingüe.
- Tenir una Interfície que sigui personalitzable.
- Tenir uns camps genèrics.

El tercer punt, és basa en el tipus d'interfície que rep l'usuari, que serà HTML i per tant modificant les pàgines i tenint diferents estils podem aconseguir diferents interfícies, tot i que en aquest projecte en farem servir només una.

Els Camps Genèrics proporcionen una clau per poder generar un entorn multilingüe, ja que moltes de les opcions o enumeracions que proporciona el sistema solen ser un número i a l'usuari se li mostra aquest amb una frase en el seu idioma. També està pensat per a que en un futur es puguin generar camps propis dins d'una Incidència, Producte, Usuari, etc.

DynBTS està pensada com una aplicació OpenSource, per respecte a les tecnologies i eines utilitzades com són L'Eclipse, l'Apache Struts, l'Hibernate, etc. i que gràcies a elles s'ha pogut realitzar aquest i altres projectes en l'entorn universitari.



Objectius

L'objectiu final d'aquest projecte és realitzar un Sistema Traçador d' Errors, però potser més important és l'objectiu d'aprendre noves tecnologies, que sovint estan a disposició de l'usuari però l'usuari les desconeix. Es per tant, des d'el meu punt de vista, una manera pràctica de realitzar una investigació sobre una tecnologia, aprendre a utilitzar-la i posar-la en funcionament.

D'altra banda no hem de restar importància al projecte en si. DynBTS pretén mostrar la capacitat de JEE per a realitzar aplicacions complexes, i al mateix temps modulars, utilitzant Struts2 per unir la capa d'Interfície amb la lògica de negoci, utilitzant Hibernate3 per unir la persistència de dades amb la lògica de negoci, i utilitzant Java com a llenguatge de programació; un orientat a objectes totalment modular.

DynBTS pretén resoldre les mancances d'alguns Traçadors de Bugs de codi obert, com són l'entorn multilingüe i la lliure elecció de Bases de Dades, entre d'altres que a continuació mostrem:

- Entorn multilingüe
- Suport per a múltiples Base de Dades.
- Suport Unicode
- Interfície gràfica personalitzable.
- Seguretat a partir de Grups de Treball.
- Camps genèrics i personalitzables.
- Capacitats de correu electrònic, (Mitjançant un client SMTP intern.)
- Llistats i Reports.
- LDAP* per a Usuaris (No implementat però es té en compte per futura ampliació.)

Mètode seguit

Per a realitzar el projecte m'he basat en algunes assignatures de la carrera, com per exemple Enginyeria del programari o Tècniques de desenvolupament de programari, tot i que s'hi apliquen els coneixements adquirits en moltes altres.

He pensat de seguir la mateixa metodologia en que es basen molts dels projectes orientat a objectes, Especificació, Anàlisi, Disseny i Codificació però no globalitzant tot el projecte, sinó realitzant les iteracions a les diferents parts d'aquest, el que anomenem un cicle de vida iteratiu i incremental.

Tampoc s'ha realitzat un prototipatge, he pensat que sempre es pot anant realitzat subsistemes incomplets que poden donar una visió del funcionament i es poden acabar de completar a posteriori, a més és una característica que el model orientat a objectes permet fer de forma fàcil.

Destacar que a la metodologia que normalment s'aplica a un projecte o aquest mateix, s'haurà d'incloure una part d'aprenentatge de la tecnologia aplicada, car aquesta serà a tot estirar posterior a l'Anàlisi de Requisits i englobaria bàsicament pel que fa aquest projecte l'estudi d'Struts2 i d'Hibernate3, els quals arrossegarien l'estudi de JSP, XHTML1.0, CSS i POJO's.



Planificació

De bon principi es va realitzar una planificació ajustada a les diferents fases del projecte.

Val a dir, que alguna complicació amb Struts2 va fer retardar les que seguien i potser és el resultat de % realitzat d'aquest projecte.

A continuació és mostra de manera cronològica la planificació que s'ha seguit:

- Especificació 16 de Març
- Anàlisi de requisits 14 d' Abril
 - JSP ~ 10 hores
 - Struts32 ~ 20 hores (Han resultat més)
 - Hibernate ~ 10 hores
 - Anàlisi ~ 6 hores
- Anàlisi funcional 14 d' Abril
- Disseny 21 d' Abril
- Codificació 09 de Juny
 - Estructura ~ 19 de Maig
 - Avançat ~ 02 de Juny
 - Testing ~ 09 de juny
- Memòria 25 de Juny

La disponibilitat d'hores a estat variada. En un principi tenia unes hores al migdia i altres a la tarda, posteriorment només a la tarda sobre unes 3h, però la principal font d'hores per a realitzar el projecte ha estat els caps de setmana i la comprensió familiar.

Producte obtingut

El producte obtingut l'anomenem DynBTS que està compost per Dynamic pel que respecte a les característiques adaptables com són la Base de Dades, l'Idioma de l'entorn i la Interfície en XHTML i CSS, i per les sigles BTS de "Bug Tracker System" que pròpiament dit es refereix a un Sistema Traçador i Gestor d'Errors, tot i que està pensat per a que sigui d'errors i de millores.

L'aplicació es depenent d'un Contenedor de Servlets per a la seva execució, i d'un Servidor Web per a la seva interacció amb l'usuari. Apache TomCat5.5 incorpora ambdós serveis.

També es necessari una Sistema de Gestió de Base de Dades(SGBD) per a l'emmagatzematge de la informació. La tecnologia Hibernate permet que sigui qualsevol que tingui un controlador per al propi Hibernate3 o bé per a JDBC (Java DataBase Connectivity) que està molt estès. Aquest projecte ha estat desenvolupat per exemple amb HSQLDB.

La interfície gràfica està realitzada amb XHTML i per tant visualitzable des de qualsevol navegador Web que compregui l'especificació XHTML1.0. En aquest projecte el client Web a estat un Firefox2. El framework Struts2 que hi ha implementat permet que la interacció de l'usuari i la lògica de negoci de l'aplicació quedin lligades mitjançant un fitxer de configuració XML.

Finalment la realització d'aquesta aplicació ha estat desenvolupada amb Eclipse 3.2 amb les llibreries de Java JDK5.0 sota les eines de JEE.



Resta de la memòria

A partir d'aquí trobarem tota la informació generada en el procés de realització d'aquest projecte. Això inclou:

- Especificació
- Casos d'ús
- El sistema funcional.
- Diagrames d'estat.
- Disseny persistent.
- Disseny de classes.
- Disseny d'interacció amb l'Usuari.

Capítol II

Especificació

Es vol realitzar una aplicació per a l'entrada i seguiment d'errors dels productes que pot tenir una empresa, a més a més també s'han de poder entrar i fer el seguiment de noves millores dels productes introduïts.

Es volen crear quatre grups per tal de especificar uns rols dins de l'aplicació que serien Invitats, Depuradors, Desenvolupadors i Supervisors.

El Depurador ha de poder entrar dins el sistema i obrir una nova Incidència, tan és si es tracta d'una Millora o d'un Error. Haurà d'introduir algunes dades addicionals, com per exemple la plataforma on es va produir, el tipus de sistema operatiu, versió del producte, tipus de base de dades i opcionalment l'origen de la Incidència. També serà important especificar la data límit per poder resoldre-la. Posteriorment quan s'hagi resolt l'ha de comprovar i tancar-la.

El Desenvolupador serà l'encarregat de resoldre la Incidència, que ha la vegada ha estat designat per un Supervisor per fer-ho. Indicarà el temps que estima necessari per resoldre-la, un % realitzat i un comptador d'hores efectives. Quan l'hagi realitzat ho indicarà apropiadament per que el depurador realitzi les proves pertinents i la tanqui.

Cada Supervisor tindrà assignat un Producte, doncs automàticament serà el primer que rebrà una notificació quan hi hagin de noves i també quan es finalitzin. Tindrà la capacitat d'assignar la Incidència a un Desenvolupador perquè la realitzi i serà notificat dels diferents estats d'aquesta. En qualsevol cas, aquest no deixa de ser un Desenvolupador i pot realitzar totes les tasques d'aquest si fos necessari.

Els Productes i el manteniment dels Components els realitzen els Supervisors, així com les diferents versions d'aquests. Addicionalment podran desactivar Productes i marcar Components com a obsolets.

Tots aquest Usuaris podran realitzar Comentaris de les diferents Incidències creades i realitzar consultes i Reports.

Els Invitats són usuaris que poden visualitzar l'estat i alguns camps d'Incidències, així com l'estat i els comentaris sobre aquesta, excepte els que siguin privats. També podran veure la llista de Productes en el sistema.

Finalment hi haurà un únic usuari Administrador que podrà realitzar les gestions d'alta, baixa i assignació de grup als usuaris del sistema, personalització de la interfície i manteniment de l'entorn en general.



Guions

Invitat

Un operador del servei Online de la companyia de desenvolupament rep una trucada que pregunta per un problema que li va succeir fa unes setmanes i encara no sap res, en la fitxa del servei online indica que es tracta d'un bug i realitza una consulta al traçador, l'Administrador va permetre que els invitats poguessin consultar l'estat i els comentaris d'un bug concret, així doncs obre el navegador, introdueix l'usuari i contrasenya i accedeix a una pàgina de benvinguda amb unes quantes opcions.

Una de les opcions és la de cerca, on introdueix el número de bug i li mostra la fitxa on veu que hi estan treballant però està pendent de resolució.

Tot usuari a més pot modificar el seu perfil, dins el possible, com l'idioma, notificacions per correu electrònic, etc.

Depurador.

Des del Servei online, li és notificat al Depurador que revisi un problema d'un client que té en un dels programes. El Depurador intenta reproduir l'error i una vegada trobat l'introdueix al sistema traçador, obre el navegador introdueix usuari i contrasenya i selecciona "nou bug", selecciona el producte i component, finalment omple la fitxa, versió o data, estat nou (hem de tenir en compte en cas de no haver pogut reproduir el bug, haurem de posar l'estat com a noconfirmat) i sempre assignarà el Supervisor relacionat al Producte.

Des del correu, el Depurador rep una notificació sobre un bug, l'ha de provar i tancar-lo o bé reobrir-lo si encara succeeix, el depurador té la capacitat de fer-fo. El Depurador va al navegador introdueix l'usuari i contrasenya i utilitza l'opció de cerca, la qual obre la fitxa del bug en qüestió, llegeix els comentaris i la resolució sobre aquest. Quan a comprovat si el bug s'ha resolt el Depurador tanca o bé reobre el bug si és el cas, i finalitza.

Desenvolupador.

El desenvolupador rep un correu indicant un nou bug per resoldre, accedeix al traçador mitjançant usuari i contrasenya i amb l'opció de cerca introdueix l'identificador del bug, on s'obre la fitxa i veu la descripció de l'error, severitat, prioritat i data límit. De fet el correu ja indicava aquestes coses més importants. el que el correu no indicava són els comentaris i camps com per exemple del client del qual prové, això és interessant, ja que certs clients tenen activades algunes opcions especials del programa i és necessari tenir-ho en compte.

Mentre està treballant amb el bug, rep un correu amb un comentari sobre aquest mateix bug. Per respondre'l és necessari que accedeixi dins el traçador ja que sinó el traçador no pot registrar-lo.

Supervisor.

El supervisor s'encarrega de repartir els bugs, que té assignats, als desenvolupadors. El Supervisor és el cap o encarregat d'un producte i té assignats tots els bugs sobre aquest, però com a Desenvolupador que és també se'ls pot assignar ell mateix. També pot rebre tots els comentaris, estats o canvis que hi hagin sobre els bugs assignats, és a dir, tots d'aquells del qual sigui el supervisor, mitjançant la configuració de les seves preferències.

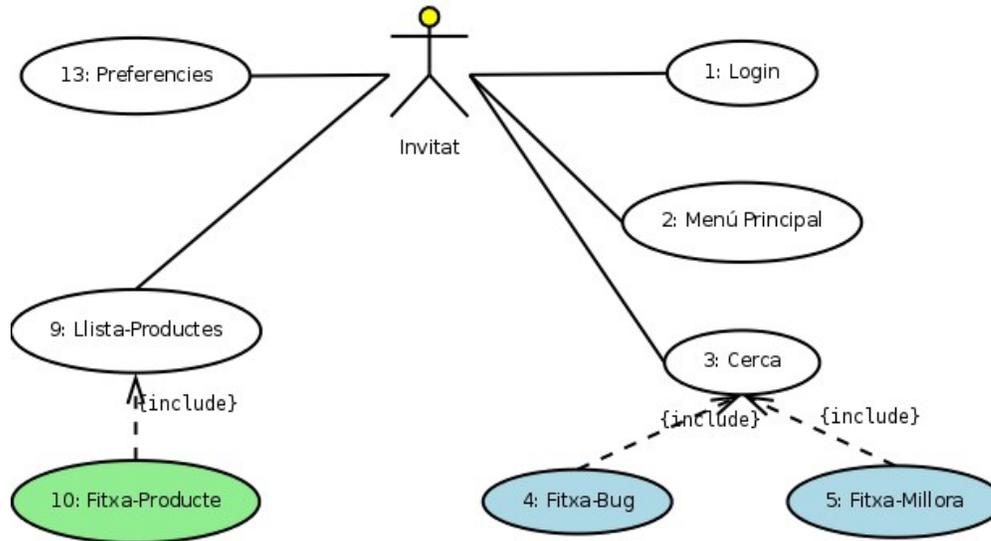
Administrador.

L'administrador estableix els permisos dels usuaris, com les consultes, reports. Així com algunes característiques per defecte quan per exemple es don d'alta un nou usuari, que també és propi de l'Administrador. Exemples de característiques són les notificacions per correu, idioma, restriccions, etc.

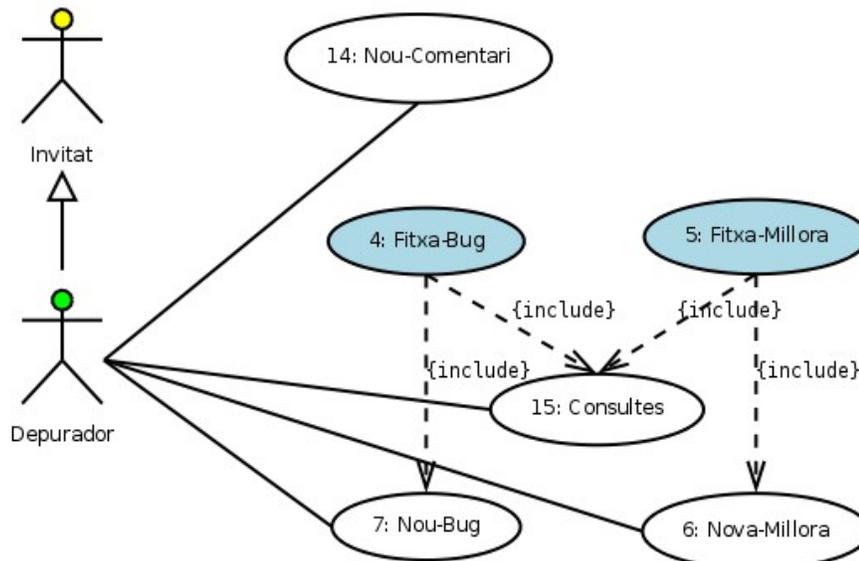


Casos d'ús

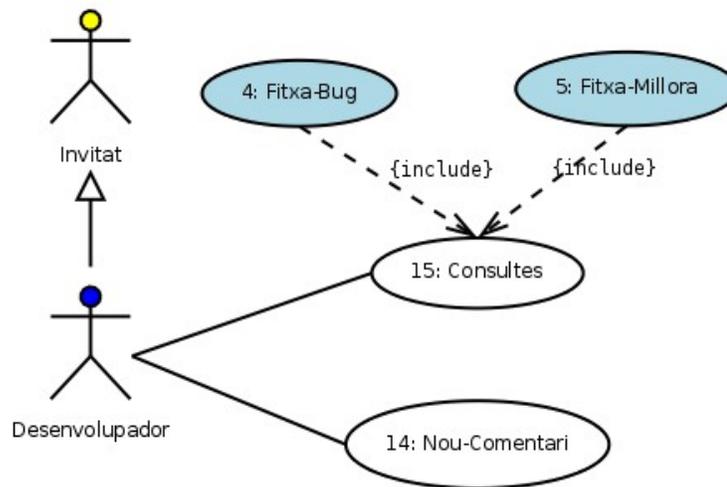
Casos d'ús referent al grup Invitats.



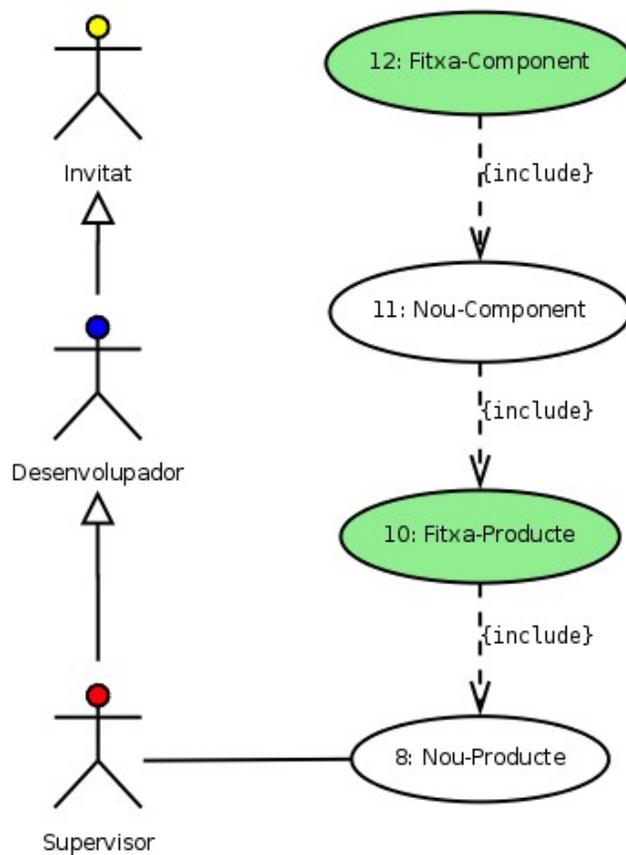
Casos d'ús referent al grup de Depuradors.



Casos d'ús referent al grup Desenvolupadors.

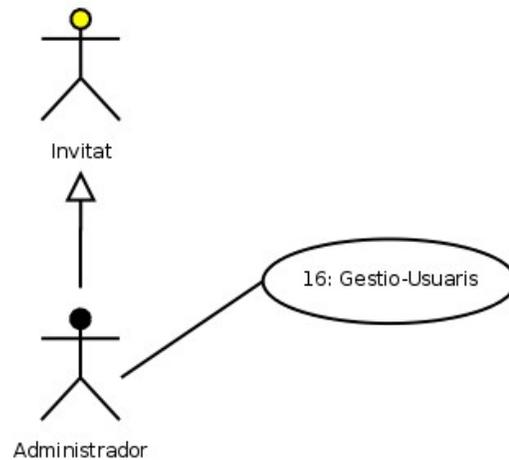


Casos d'ús referent al grups de Supervisors.





Casos d'ús referents a l'Administrador.



Cas d'ús número 1: "Login"

- Resum de la funcionalitat:
Accés al Sistema mitjançant usuari i contrasenya.
- Paper dins del treball de l'usuari:
Els actors descrits utilitzaran aquest cas d'us per accedir al Sistema. A part de ser una finestra de benvinguda és l'única via per accedir-hi.
- Actors:
Invitat, Depurador, Desenvolupador, Supervisor, Administrador.
- Casos d'ús relacionats:
Menú principal
- Precondició:
L'Usuari es troba fora del sistema.
- Postcondició:
L'Usuari és troba en dos situacions diferents:
 1. *Menú Principal si ha accedit correctament.*
 2. *Login amb l'error, si l'usuari o contrasenya no són correctes.*
- Alternatives del procés i excepcions:
Si l'usuari no introdueix l'usuari o contrasenya correcte es denega l'accés al sistema i és repeteix el procés.

Cas d'ús número 2: "Menú Principal"

- Resum de la funcionalitat:
Proporciona la distribució de les funcionalitats a través del Sistema.
- Paper dins del treball de l'usuari:
L'Usuari accedeix sempre després de validar-se amb l'usuari i contrasenya, i a partir d'aquí pot anar a altres llocs com per exemple els casos de [Cerca](#), [Nou Bug](#), [Nova Millora](#), [Preferencies](#) o [Consultes](#). I fins i tot si realitza un Logout accedeix al cas d'ús [Login](#).
- Actors:
Invitat, Depurador, Desenvolupador, Supervisor, Administrador.
- Casos d'ús relacionats:
[Login](#), [Cerca](#), [Nou Bug](#), [Nova Millora](#), [Preferencies](#) , [Consultes](#).
- Precondició:
L'usuari es pot trobar a qualsevol lloc del sistema, excepte si no s'ha autenticat.
- Postcondició:
L'Usuari es troba al Menú Principal
- Alternatives del procés i excepcions:
Cap.

Cas d'ús número 3: "Cerca"

- Resum de la funcionalitat:
L'Usuari accedeix a aquest cas d'ús quan vol veure la fitxa d'un Bug o Millora.
- Paper dins del treball de l'usuari:
És el mitjà que té l'usuari per mostrar la informació sobre un Bug o Millora a través de la [Fitxa-Bug](#) o [Fitxa-Millora](#) respectivament..
- Actors:
Invitat, Depurador, Desenvolupador, Supervisor
- Casos d'ús relacionats:
[Fitxa-Bug](#), [Fitxa-Millora](#).
- Precondició:
L'Usuari es troba en el [Menú Principal](#).
- Postcondició:
La cerca correcta mostra la [Fitxa](#) del Bug o Millora.
- Alternatives del procés i excepcions:
Una manera alternativa d'obrir una fitxa seria utilitzar una [Consulta](#).



Cas d'ús número 4: "Fitxa-Bug"

- Resum de la funcionalitat:
Mostra la informació realacionada amb un Bug.
- Paper dins del treball de l'usuari:
En aquesta Fitxa podem veure i modificar els atributs d'un Bug, sempre que els privilegis ho permetin.
- Actors:
Invitat, Depurador, Desenvolupador, Supervisor
- Casos d'ús relacionats:
Nou-Bug, Cerca, Consultes.
- Precondició:
L'usuari està autenticat.
- Postcondició:
L'usuari veu la informació d' un Bug.
- Alternatives del procés i excepcions:
En principi els Invitats no poden modificar els atributs.

Cas d'ús número 5: "Fitxa-Millora"

- Resum de la funcionalitat:
Mostra la informació relacionada amb una Millora.
- Paper dins del treball de l'usuari:
En aquesta Fitxa podem veure i modificar els atributs d'una Millora, sempre que els privilegis ho permetin.
- Actors:
Invitat, Depurador, Desenvolupador, Supervisor
- Casos d'ús relacionats:
Nova-Millora, Cerca, Consultes
- Precondició:
L'usuari està autenticat.
- Postcondició:
L'usuari veu la informació de una Millora.
- Alternatives del procés i excepcions:
En principi els Invitats no poden modificar els atributs.

Cas d'ús número 6: "Nova-Millora"

- Resum de la funcionalitat:
Emplena els camps mínims per a la creació d'una millora.
- Paper dins del treball de l'usuari:
L'Usuari ha d'emplenar uns camps mínims com son un Producte, un Component i la descripció, per crear una nova Millora
- Actors:
Depurador, Supervisor
- Casos d'ús relacionats:
Fitxa-Millora, Nou-Producte, Menú Principal
- Precondició:
L'Usuari es troba en el Menú Principal
- Postcondició:
L'Usuari pot seleccionar Producte i Component, i continuar. També pot crear un Producte nou.
- Alternatives del procés i excepcions:
Només un Supervisor pot crear un Producte nou.

Cas d'ús número 7: "Nou-Bug"

- Resum de la funcionalitat:
Emplena els camps mínims per a la creació d'un Bug.
- Paper dins del treball de l'usuari:
L'Usuari ha d'emplenar uns camps mínims com son un Producte, un Component i la descripció, per crear un nou Bug
- Actors:
Depurador, Supervisor
- Casos d'ús relacionats:
Fitxa-Bug, Nou-Producte, Menú Principal
- Precondició:
L'Usuari es troba en el Menú Principal
- Postcondició:
L'Usuari pot seleccionar Producte i Component, i continuar. També pot crear un Producte nou.
- Alternatives del procés i excepcions:
Només un Supervisor pot crear un Producte nou.



Cas d'ús número 8: "Nou-Producte"

- Resum de la funcionalitat:
Crea un nou Producte en el Sistema.
- Paper dins del treball de l'usuari:
L'Usuari pot crear un nou producte assignant-hi un Supervisor i almenys tenir un Component.
- Actors:
Administrador, Supervisor
- Casos d'ús relacionats:
Nou-Component, Llista-Productes, Fitxa-Component, Fitxa-Producte.
- Precondició:
El producte no existeix al Sistema.
- Postcondició:
L'Usuari pot acabar de crear el Producte a partir de la Fitxa-Producte.
- Alternatives del procés i excepcions:
Cap.

Cas d'ús número 9: "Llista-Productes"

- Resum de la funcionalitat:
Mostra tots els productes del Sistema
- Paper dins del treball de l'usuari:
L'usuari pot visualitzar tots els productes definits i accedir a ells a través de la Fitxa-Producte. També pot crear-ne de nous a partir del cas d'ús Nou-Producte.
- Actors:
Invitat, Depurador, Desenvolupador, Supervisor, Administrador.
- Casos d'ús relacionats:
Fitxa-Producte, Nou-Producte, Menú Principal.
- Precondició:
L'usuari és troba al Menú Principal.
- Postcondició:
L'usuari veu tots els Productes definits.
- Alternatives del procés i excepcions:
Només el Supervisor i Administrador poden crear Nou-Producte.

Cas d'ús número 10: "Fitxa-Producte"

- Resum de la funcionalitat:
Mostra tots els atributs relacionats amb un producte.
- Paper dins del treball de l'usuari:
L'Usuari utilitza aquesta fitxa per veure o modificar un Producte. Fins i tot inhabilitar-lo.
- Actors:
Supervisor, Administrador.
- Casos d'ús relacionats:
Nou-Producte, Nou-Component, Llista-Productes.
- Precondició:
El producte està creat.
- Postcondició:
Es mostra la informació sobre el Producte.
- Alternatives del procés i excepcions:
Cap.

Cas d'ús número 11: "Nou-Component"

- Resum de la funcionalitat:
Crea un nou Component en un Producte.
- Paper dins del treball de l'usuari:
L'Usuari utilitza aquest cas d'ús per crear un nou Component en un Producte. Component que acabarà de realitzar a la Fitxa-Component.
- Actors:
Supervisor, Administrador.
- Casos d'ús relacionats:
Fitxa-Component, Fitxa-Producte.
- Precondició:
L'Usuari es troba a una Fitxa-Producte.
- Postcondició:
L'Usuari pot acabar de crear el Component a partir de la Fitxa-Component.
- Alternatives del procés i excepcions:
Sempre pot avortar. Només ho poden realitzar els Supervisor o Administrador.



Cas d'ús número 12: "Fitxa-Component"

- Resum de la funcionalitat:
Mostra la informació sobre un Component d'un Producte.
- Paper dins del treball de l'usuari:
L'Usuari pot veure o modificar la informació del Component.
- Actors:
Supervisor, Administrador.
- Casos d'ús relacionats:
Nou-Component, Fitxa-Producte.
- Precondició:
El Component està creat.
- Postcondició:
Es mostra la informació sobre el Component.
- Alternatives del procés i excepcions:
Cap.

Cas d'ús número 13: "Preferències"

- Resum de la funcionalitat:
Permet modificar les preferències de l'entorn de l'usuari.
- Paper dins del treball de l'usuari:
L'usuari pot modificar en major o menor mesura les preferències del seu entorn en el Sistema, com per exemple la llengua, notificacions, contrasenya, etc.
- Actors:
Invitat, Depurador, Desenvolupador, Supervisor, Administrador.
- Casos d'ús relacionats:
Menú Principal.
- Precondició:
L'usuari s'ha autenticat al Sistema.
- Postcondició:
L'usuari pot midificar les preferències.
- Alternatives del procés i excepcions:
L'Administrador a més a més pot modificar les preferències per defecte.

Cas d'ús número 14: "Nou-Comentari"

- Resum de la funcionalitat:
Permet respondre o afegir un comentari a un Bug/Millora.
- Paper dins del treball de l'usuari:
L'Usuari pot realitzar un Comentari sobre la resolució d'un Bug o Millora, quan aquest encara no estigui tancat/da, o bé quan s'hagi reobert.
- Actors:
Depurador, Desenvolupador, Supervisor.
- Casos d'ús relacionats:
[Fitxa-Bug](#), [Fitxa-Millora](#).
- Precondició:
El Bug o Millora estan creats.
- Postcondició:
Hi ha una nou Comentari,potser en replica d'un altre.
- Alternatives del procés i excepcions:
Els Invitats poden llegir certs comentaris.

Cas d'ús número 15: "Consultes"

- Resum de la funcionalitat:
Permet realitzar unes consultes per tal de trobar Bugs o Millores.
- Paper dins del treball de l'usuari:
L'Usuari pot realitzar unes consultes a través de paraules clau, dates, per Producte o Component, per Desenvolupador, etc. I els resultats retornats poden obrir les [Fitxa-Bug](#) o [Fitxa-Millora](#) relacionats.
- Actors:
Depurador, Desenvolupador, Supervisor, Administrador.
- Casos d'ús relacionats:
[Menú Principal](#), [Fitxa-Bug](#), [Fitxa-Millora](#).
- Precondició:
L'Usuari es troba al [Menú Principal](#).
- Postcondició:
Es mostren els Bugs o Millores segons la consulta realitzada.
- Alternatives del procés i excepcions:
[Es poden obrir directament els Bugs o Millores a través del cas d'ús Cerca.](#)



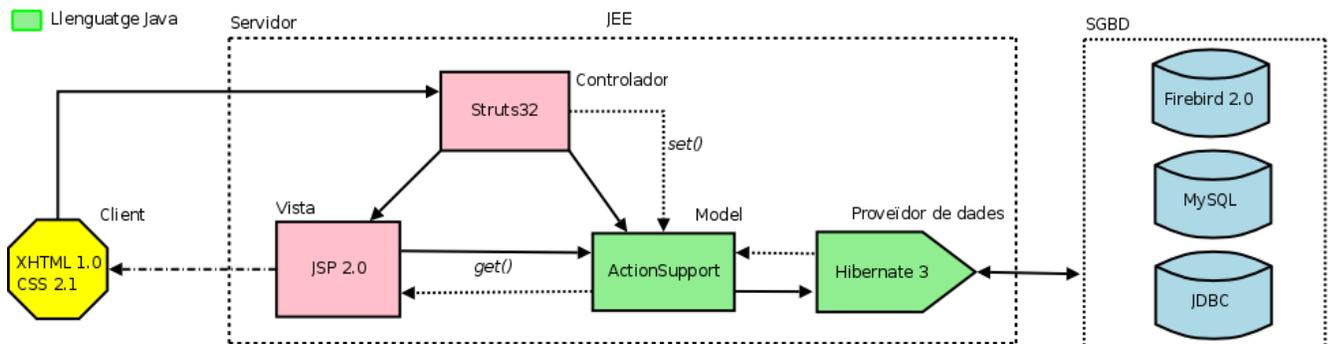
Cas d'ús número 16: "Gestio-Usuaris"

- Resum de la funcionalitat:
Permet el manteniment dels usuaris del sistema.
- Paper dins del treball de l'usuari:
L'Usuari pot consultar, modificar o crear Usuaris al sistema i assignar-los al grup de treball corresponent.
- Actors:
Administrador.
- Casos d'ús relacionats:
Menú Principal.
- Precondició:
L'Usuari es troba al Menú Principal.
- Postcondició:
Es mostra una fitxa d'usuari on mostrar o crear l'usuari, junt amb una opció per llistar els que hi han.
- Alternatives del procés i excepcions:
En cas d'activar l'opció amb LDAP la gestió és automàtica, i es realitza de forma externa a aquest sistema.

Sistema funcional

DynBTS és una Aplicació Web i per tant és necessari un servidor Web, més concretament un Contenedor de Servlets, que executi l'aplicació i la publiqui per que els usuaris amb els seus navegadors Web hi puguin interaccionar.

El servidor proporciona una resposta apropiada per a cadascun dels events que genera l'usuari com a client a partir de pàgines JSP.



El Servidor, tal com podem observar, es comunica amb la base de dades, que pot residir en la pròpia màquina física o en alguna altra.

Pel que fa al Servidor, independentment dels recursos externs, podem desglossar-ho en varies tecnologies, que en sí són la base del projecte:

JEE 5.0

Anteriorment anomenat J2EE, es la base de tot el desenvolupament de l'aplicació, a excepció de la part visual, i difereix en altres plataformes Java en que amplia el repositori amb altres tecnologies, com poden ser Struts i Hibernate. Al mateix temps Java és un llenguatge orientat a objectes amb el qual està realitzada aquesta aplicació, juntament amb llibreries java addicionals que formen part de la plataforma JEE, com són Struts2 i Hibernate3.

Hibernate 3

Proveïdor de dades del model de negoci. Es una tecnologia realitzada en java i orientada a objectes, en el sentit que abstrau les funcionalitats d'interacció amb els SGBD normalment relacionals, quedant la implementació independent del tipus de Base de Dades. Així doncs aquesta tecnologia esta pensada per a que durant la implementació de l'aplicació no pensem la manera de recuperar la informació de la base de dades, sinó en recuperar les classes persistents que s'han dissenyat.



Struts 2

Es un framework de programació que proporciona un aïllament entre la capa visual, de control i la de negoci d'una aplicació JEE, i que anomenem MVC (Model View Controller) compost separatament per la Vista, el Controlador i el Model:

Vista.

Es la part que l'aplicació construeix per a la interacció amb l'usuari. Aquesta com es pot observar en l'esquema, en Struts2 la crea o crida el Controlador davant dels events que l'usuari realitza, i obté la informació del Model que es on estan les dades de la sessió.

Controlador

Bàsicament és la part que respon a les interaccions de l'usuari, i s'encarrega de proporcionar la Vista adient al respecte. Al mateix temps, si l'usuari ha introduït dades, el controlador envia aquestes dades al model per a que les processi i/o les desi.

Model

És on tenim la lògica de negoci, a la teoria en el Model es qui rep, processa i desa les dades de l'aplicació i per tant fa molta feina, potser massa; és per això que a la pràctica, pel que fa Struts2, el Model es subdivideix en Paquets més especialitzats. Per exemple en aquest projecte el model s'ha dividit en 2 capes més la persistència. Una interacciona amb el Controlador i les Vistes i l'altre amb la Base de Dades.

Al mateix temps, aquesta tecnologia, i concretament la Vista, depèn d'altres com:

XHTML 1.0

Es bàsicament el llenguatge amb la qual es compon cada una de les pàgines Web que l'usuari veu al seu navegador.

JSP 2.0

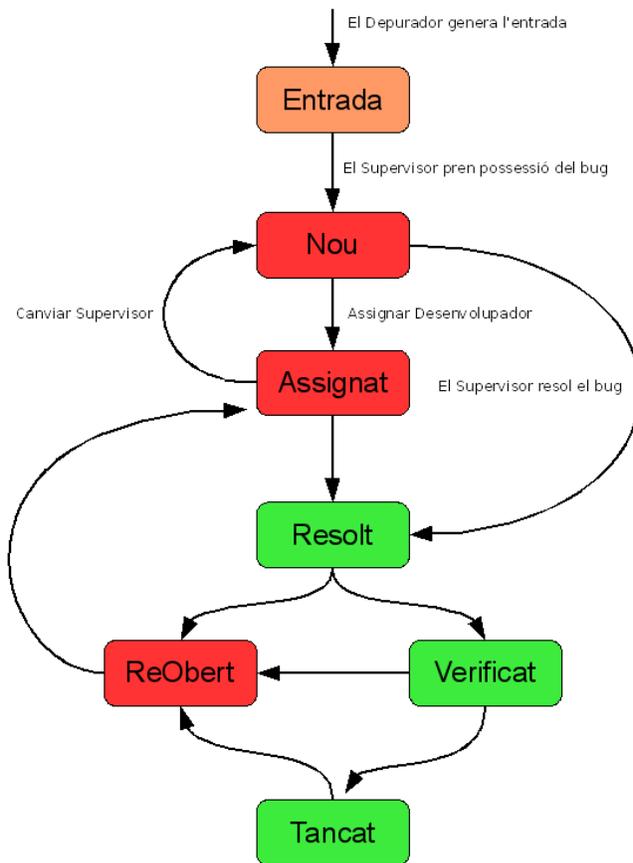
(Java Server Pages) és la tecnologia que utilitza el framework Struts2 per emplenar adientment les pàgines web amb els resultats provinents de la capa de negoci.

CSS 2.2

Les fulles d'estil són un element indispensable per tractar el "look and feel" de les pàgines web. En certa manera és el responsable de tenir varis temes visuals en una mateixa aplicació.

Diagrames d'estat

DynBTS és un traçador d'Errors i Millores, i per tant es fonamental saber l'evolució d'una Incidència dins el sistema. En el següent diagrama podem observar el cicle de vida d'un Error.



En aquest diagrama podem observar diferents actors, addicionalment especificarem d'altres igualment importants:

- Administrador*
- Supervisor.
- Desenvolupador
- Depurador
- Invitat*

Tal com s'ha comentat en un apartat anterior, el Depurador sempre entra i tanca les Incidències, ja que està implicat en els estats d'Entrada, Verificat, Tancat o Reobert.

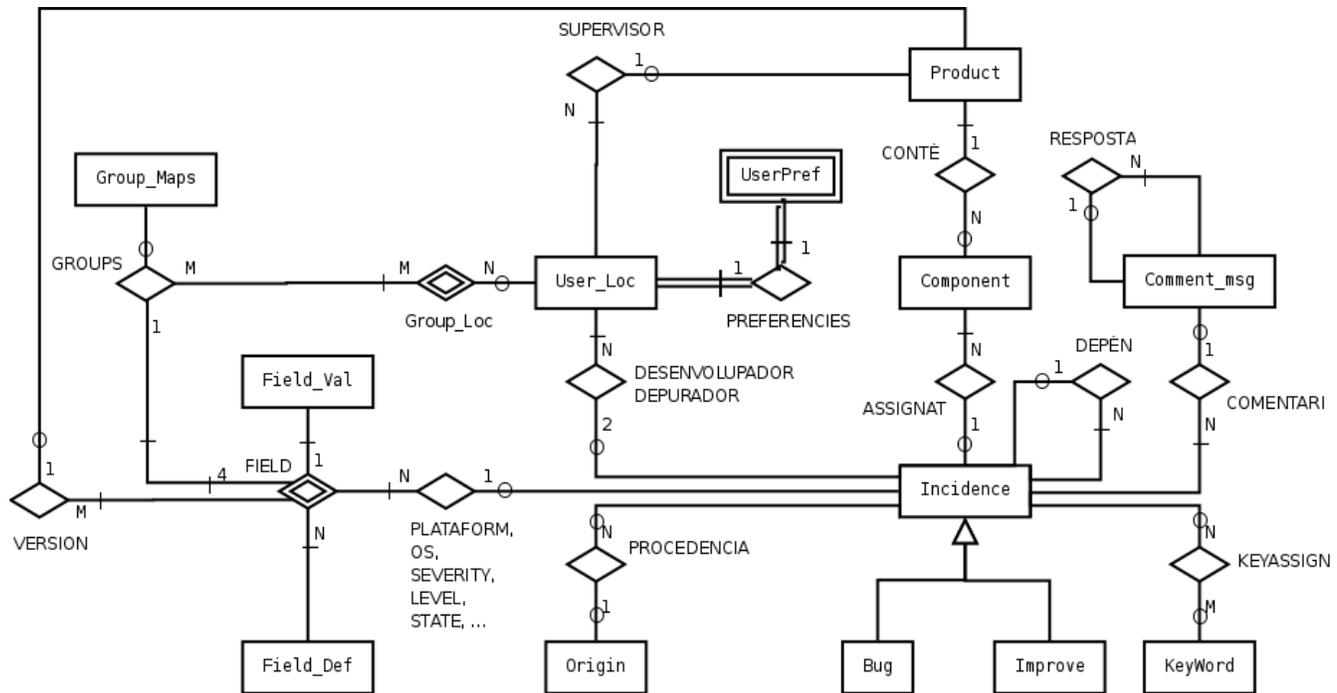
El Supervisor automàticament queda assignat quan la Incidència fa referència al Producte que té supervisat. El proper pas és Resoldre-la o bé assignar-la. Addicionalment mentre ja estigui assignada a un Desenvolupador pot canviar-lo i assignar la incidència a un altre.

Un Desenvolupador només pot Resoldre la Incidència, es per això que està.



Disseny de la persistència

A continuació veurem les classes persistents del sistema. Com que el projecte està desenvolupat utilitzant la tecnologia Hibernate3, aquestes classes formaran directament part del disseny del model de negoci:



A continuació es fa una breu descripció de cadascuna d'aquestes entitats:

User_Loc

Usuaris Locals del Sistema, informació bàsica.

Product

Productes gestionats. Sempre té un Supervisor encarregat. Es pot inhabilitar el producte.

Component

Components de cada Producte. Es pot marcar un component com a obsolet.

Field_Def

Definició de camps, també els que utilitza el sistema com l'Estat, Severitat, Prioritat o Resolució que tindran l'opció de reservats, i altres com Sistemes Operatius, Plataforma i Base de Dades entre d'altres que poden ser ampliables. "alias" representa el nom intern en el sistema, no visualitzable i que mitjançant l' Struts es mostrarà amb l'idioma corresponent.

Field_Val

Valors dels camps definits, representa la llista d'opcions associada a cada definició de camp, i també incorpora l' "alias" per representar els noms amb l'idioma corresponent. Per exemple la definició de "status" en Filed_Def l'associariem als següents registres amb alias: (st_unconfirmed, st_new, st_assigned, st_reopened, st_resolved, st_verified, st_closed)

Keyword

Conjunt de Paraules Clau, són útils quan realitzem "Consultes" sobre determinats Errors i Millores

Origin

Origen d'on prové la petició, pot ampliar-se si es requereix.

Incidence

Error(Bug) o Millora(Improvement), en qualsevol cas tota la informació relacionada es troba en aquesta classe. El que sigui una o una altra ho determina el camp "type_inc". Entre els camps definits hi ha la data d'entrada i la de verificació QA, percentatge de la Incidència realitzada, Data límit de finalització, El desenvolupador que ho arregla, i el depurador que ho verifica, també es pot observar una relació de dependència cap a una altra incidència, molt útil si aquesta necessita que es finalitzi una altra abans.

Comment_msg

Tots els comentaris de les incidències es troben aquí. També hi ha una relació de rèplica entre comentaris. Alguns comentaris poden ser privats, de manera que només ho pot llegir un mateix, el grup que el conforme o superior.

Group_Loc

Pertinença dels usuaris a un grup, pot succeir que un usuari pertanyi a més d'un grup. És important fixar-se que els grups o definició d'aquest es troben a Field_Val, per les raons d'idioma comentades.



Group_Maps

LDAP: Defineix un enllaç entre els grups locals i els definits en LDAP. Quan el Sistema vol cercar els usuaris utilitza aquesta classe per carregar-los i establir els privilegis.

User_Pref

Extensió de User_Loc, ja que aquesta última només conté la informació bàsica d'usuari. En aquesta altra classe hi tindrem la configuració d'Idioma, de correu electrònic, les opcions de notificació als canvis de les Incidències i Comentaris, etc.

Bug

Una subclasse d'Incidència que representa un Error. Quedarà determinada per un atribut de tipus en la classe genèrica.

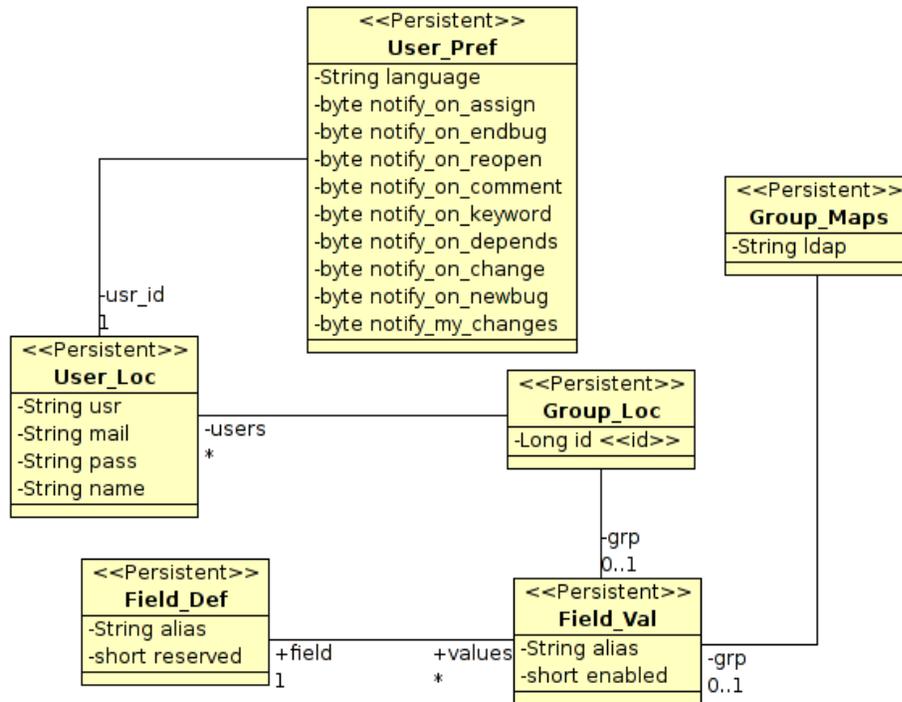
Improve

Una subclasse d'Incidència que representa una Millora. Quedarà determinada per un atribut de tipus en la classe genèrica.

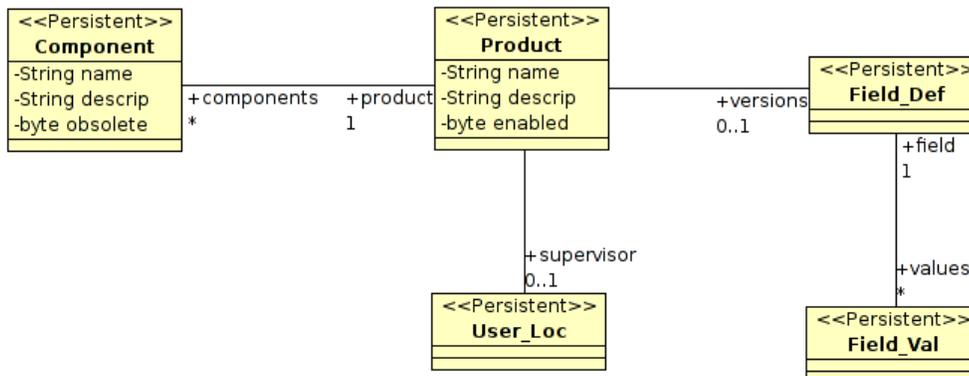
Disseny de Classes

A continuació és representa les entitats mostrades anteriorment com a Classes del sistema i que seran les que interaccionin amb Hibernate3 com a POJO's.

A continuació es mostren les referents als Usuaris i el seu entorn:

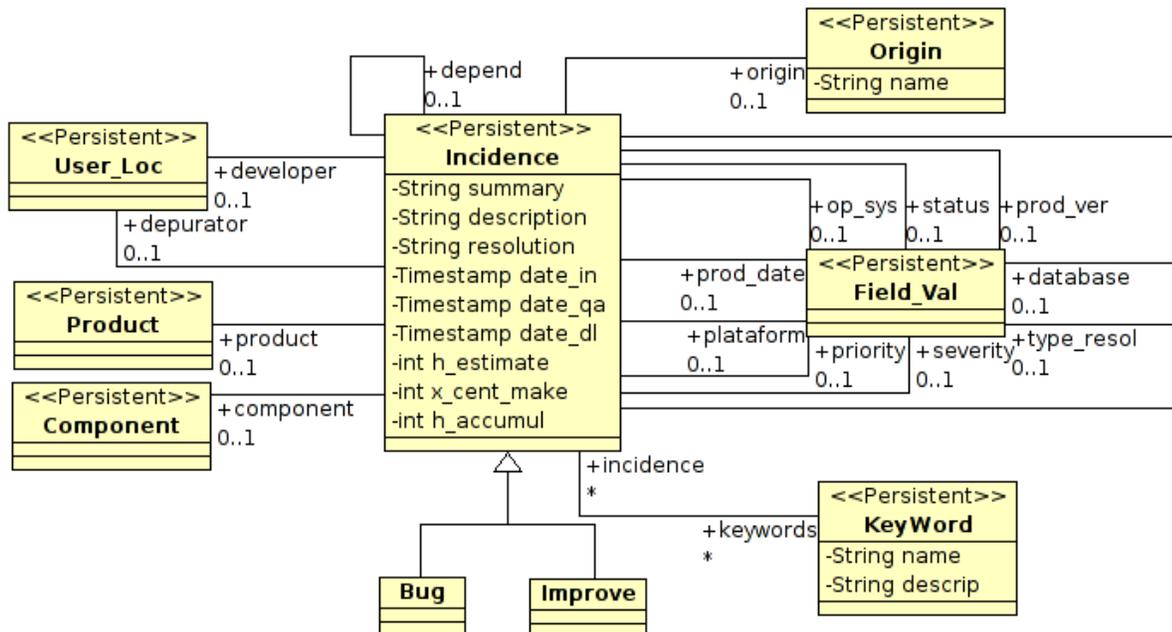


També mostrem el diagrama pel que l'entorn dels Productes. Les classes anteriorment tractades no mostren els atributs.

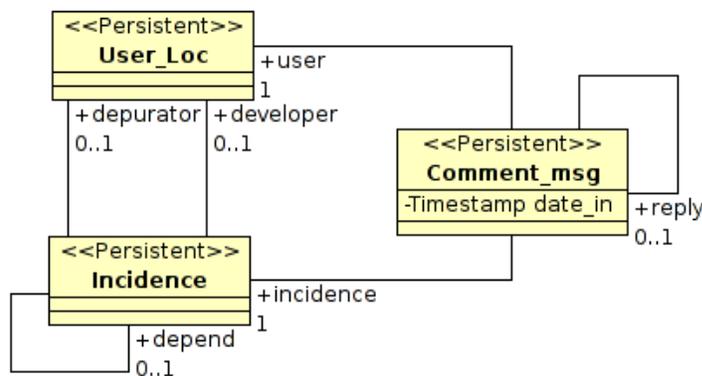




La classe potser més complexa del sistema és Incidence, que com en l'esquema de persistència ja es veia, enllaça gairebé totes les classes existents. Vegeu a més com actuen els camps genèrics Field_Def i Field_Val en aquest afer:

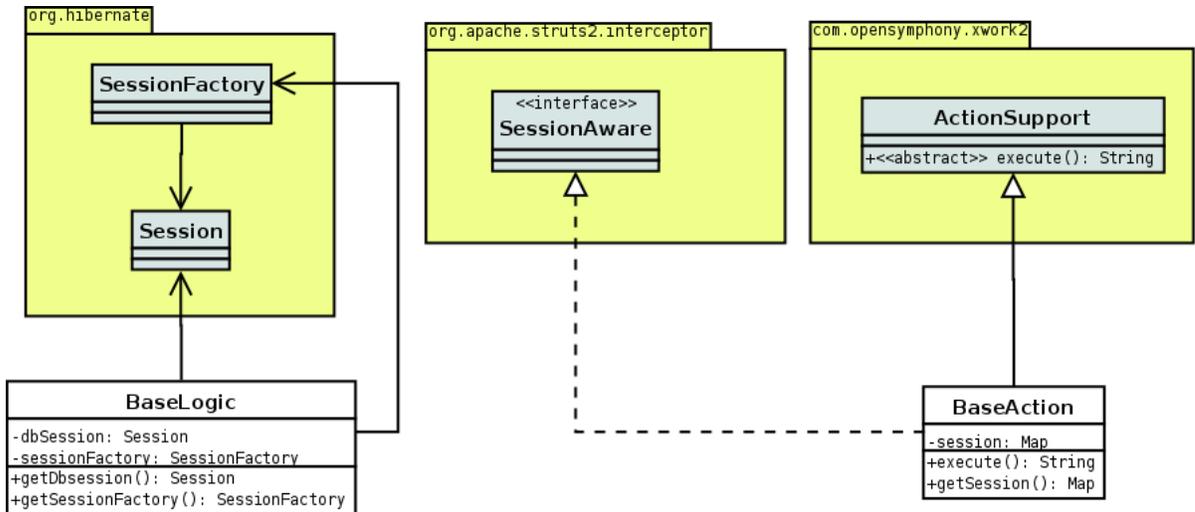


I finalment, pel que fa la persistència del sistema, ens queda mostrar l'entorn d'un Comentari sobre una Incidència:

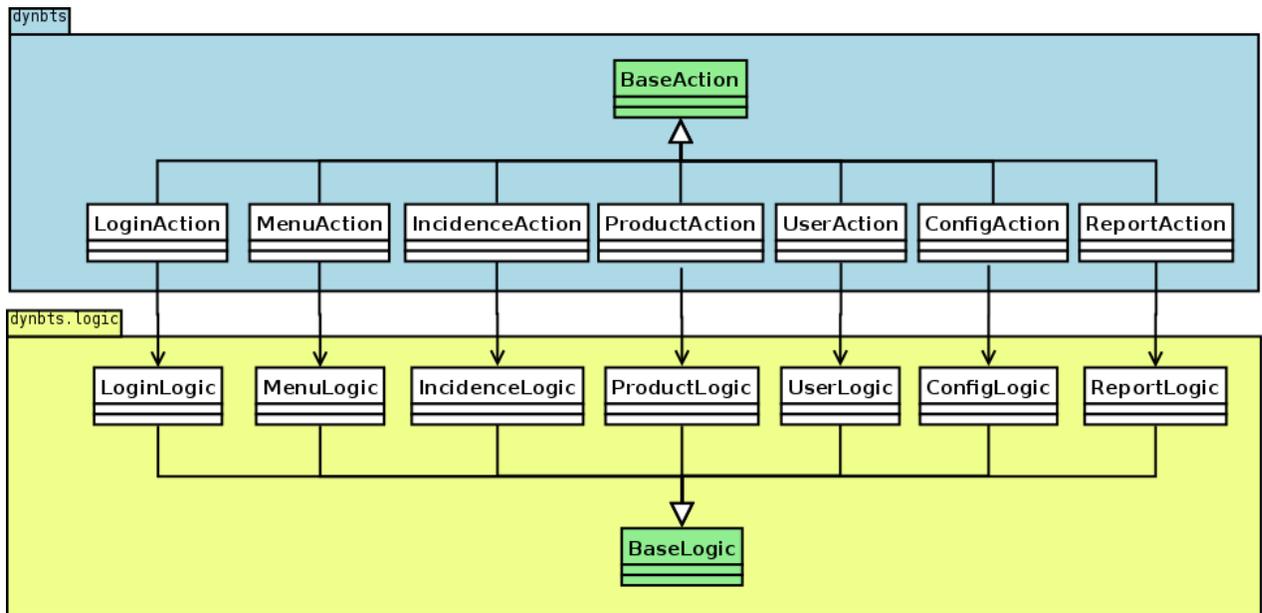


Classes del sistema

Pel que fa el disseny de classes del projecte s'han creat unes Bases per tal d'implementar un funcionalitat bàsica. El primer cas és la classe BaseAction que proporciona accés a les variables de sessió mentre un usuari interacciona amb l'aplicació. L'altra és una base per a la lògica de dades que interacciona amb Hibernate3 anomenada BaseLogic.

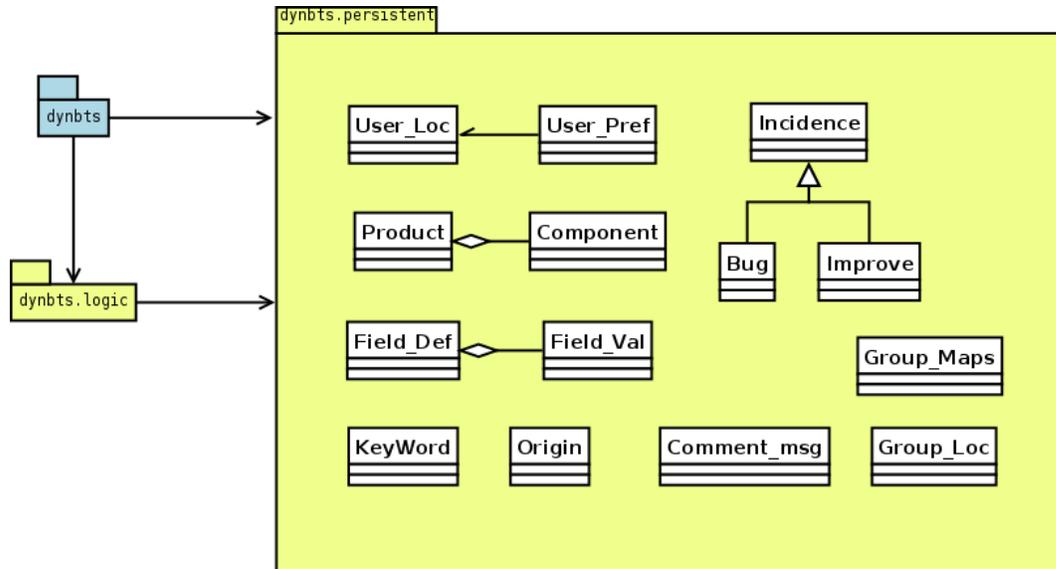


A continuació es mostren les subclasses i les relacions sense elles. Els principals mètodes no es posaran per facilitar la comprensió i visibilitat.



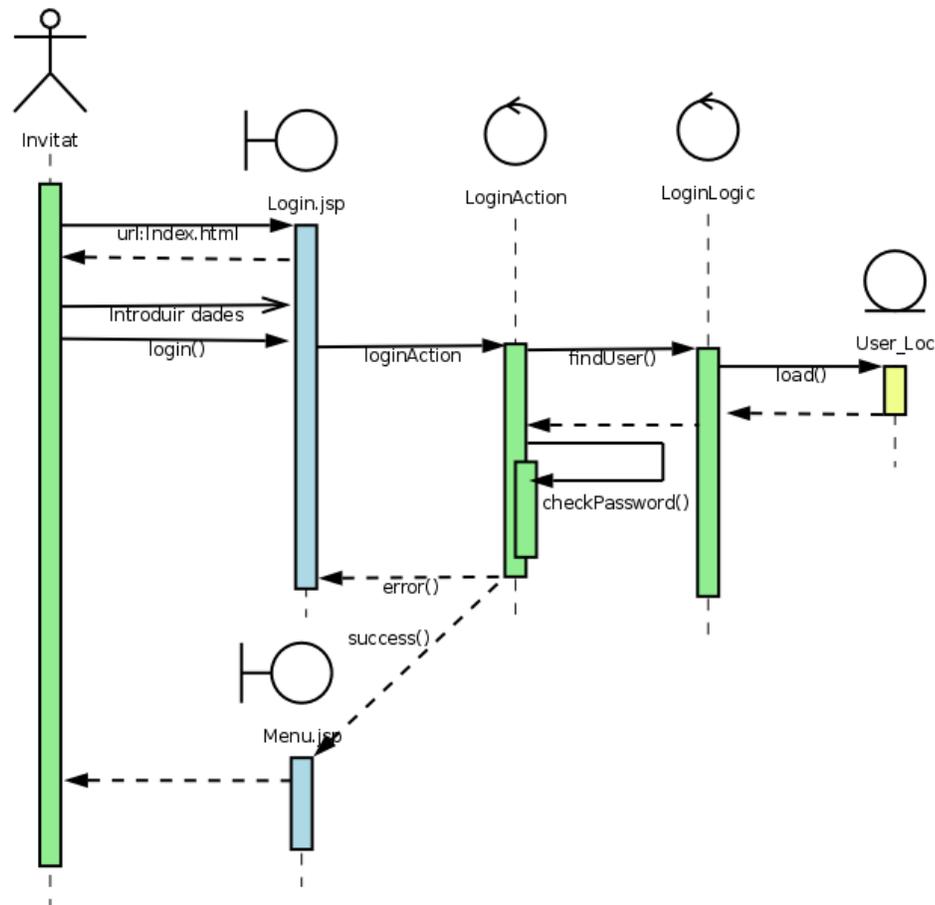


I finalment pel que fa la relació d'aquestes amb les persistents, també es mostra el sistema de paquets definitiu aplicat al projecte DynBTS en quan a Classes Java.



Diagrames de Seqüència

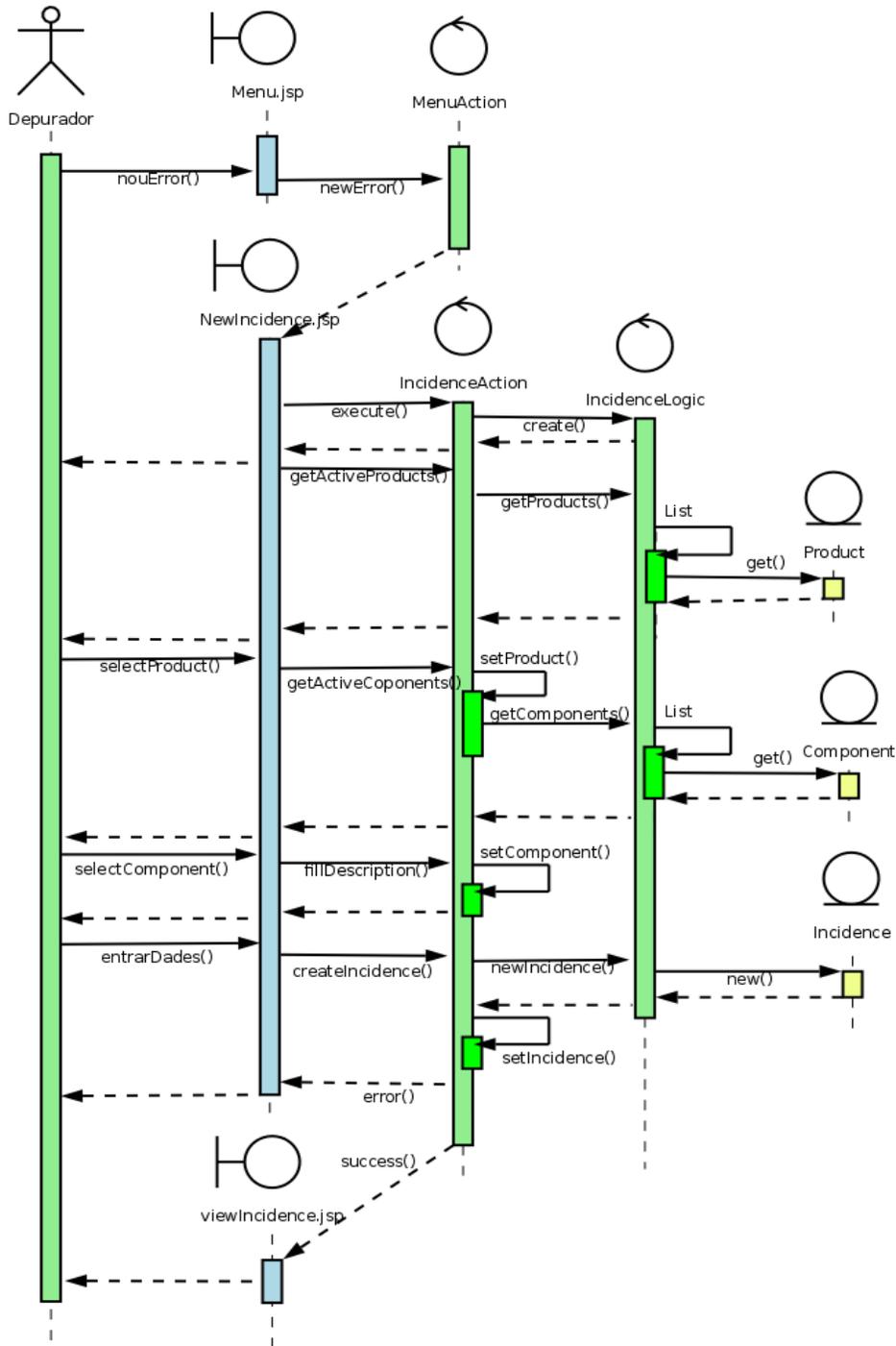
A continuació es mostra un diagrama genèric, car tots els usuaris hereten de l'actor Invitat. Aquest és el relacionat amb l'entrada al sistema.



Es pot observar com carrega l'Usuari de la base de dades, a partir del compte o correu electrònic, i després verifica que la contrasenya sigui la mateixa. En l'aplicació la funció `checkPassword()` compara les contrasenyes una vegada encriptades.

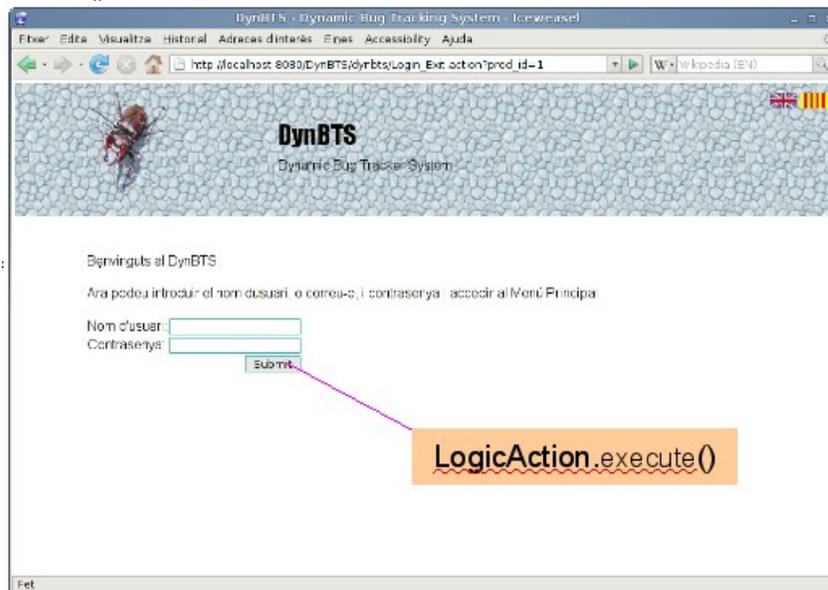


Ara es mostra la seqüència de creació d'un Error per part d'un Depurador. Observarem que l'acció per part de l'Usuari és provinent d'una pàgina Menu.jsp , i que una vegada finalitzada se'n va a una altra editIncidence.jsp on s'acabarà d'omplir la Incidència.

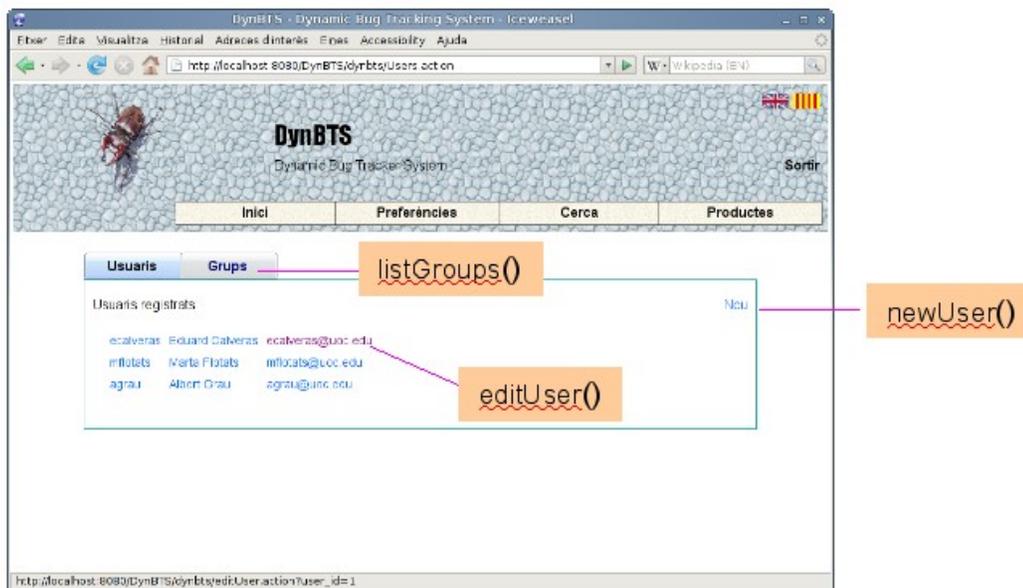


Interfície

A continuació es mostren algunes captures de les diferents Vistes de l'aplicació amb els seus corresponents mètodes d'acció. Aquests mètodes executen la lògica de negoci (Model) que captura el Controlador. Comencem per la vista Login.



Saltem el MenuPrincipal i anem a la Gestió d'Usuaris, concretament a la Llista d'Usuaris.





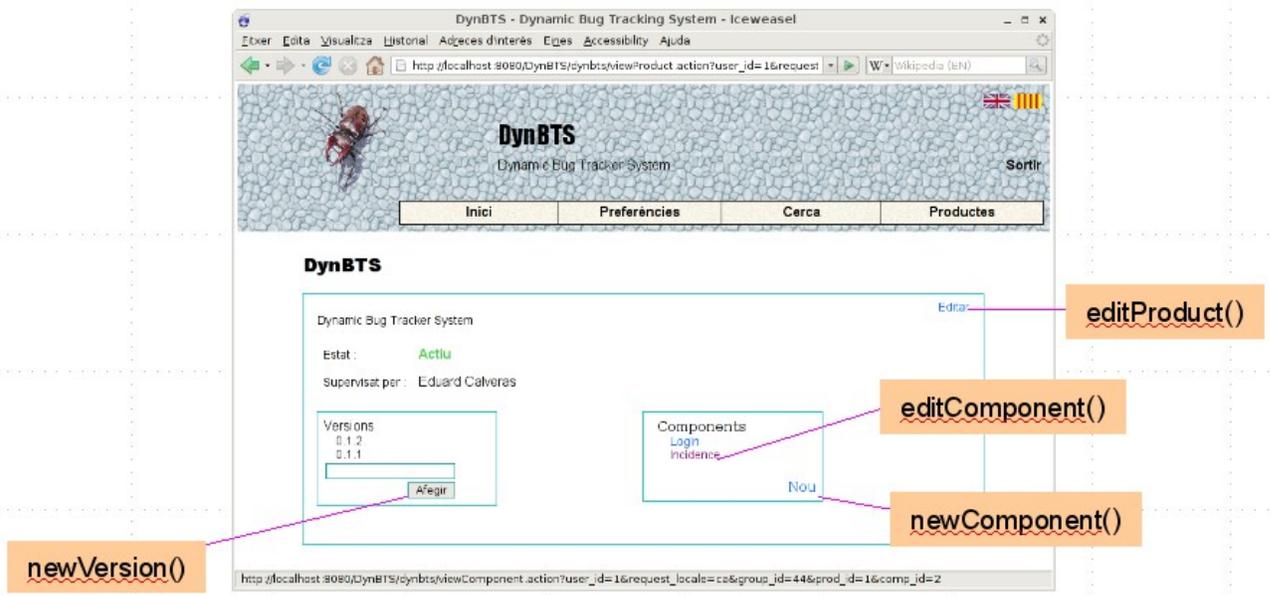
Si piquem sobre d'un usuari el Controlador en visualitzarà la pàgina d'edició d'aquell Usuari.

The screenshot shows the 'Eduard Calveras' user profile. The 'Grups' tab is active, showing a list of groups: Supervisors (checked), Depuradors, Desenvolupadors, and Invitats. The 'Eliminar' button is located at the top right of the user details section. The 'Desar' button is at the bottom of the user details form.

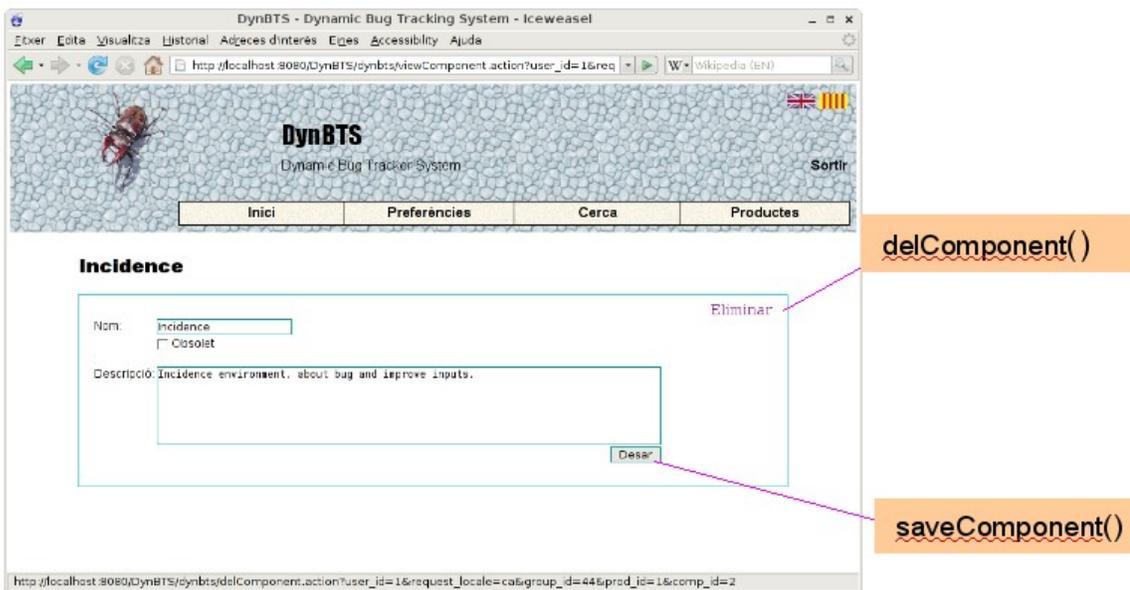
Podem canviar la pertinença dels grups de seguretat per aquí o bé anar directament a visualitzar els Grups per canviar-ne varis usuaris de cop.

The screenshot shows the 'Groups' management interface. The 'Users' tab is selected, and the 'Supervisors' group is highlighted. The 'editGroup()' annotation points to the group name.

Seleccionem els Depuradors, si us heu fixat hem realitzat un canvi d'idioma i ara els camps sense sortir de l'aplicació s'adapten al nou entorn. Passem, però a visualitzar una altra pàgina, en aquest cas la d'un Producte. L'accés és fàcil, doncs sempre el tenim en una pastilla del Menú Superior.



Si volguéssim editar un Component només caldrà seleccionar-lo:





Per acabar una pàgina en la creació d'un Error d'un Producte, encara que no s'observi, els components només es visualitzen quan es selecciona el Producte, en aquest cas ja hem seleccionat un d'ells:

The screenshot shows a web browser window titled "DynBTS - Dynamic Bug Tracking System - Iceweasel". The URL is "http://localhost:8080/DynBTS/dynbts/fillDescription.action?productId=2&cc=...". The page has a header with a beetle logo and the text "DynBTS Dynamic Bug Tracker System". Below the header is a navigation bar with buttons for "Inici", "Preferencies", "Cerca", and "Productes". The main content area is titled "Nou Error" and contains the following text: "Per poder crear un nou {BUG} heu de primer emplenar els següents camps:". There are two columns of links: "Seleccioneu un Producte:" with links for "DynBTS", "Eclipse", and "Now"; and "Seleccioneu un Component:" with links for "Editor" and "Parser XML". Below these is a text input field with the placeholder "Introduïu una petita descripció del problema:" and the text "Hi han problemes!". A "Crear" button is located to the right of the input field. Three orange callout boxes with purple arrows point to: "getActiveComponents()" pointing to the "Eclipse" link; "fillDescription()" pointing to the "Parser XML" link; and "createIncidence()" pointing to the "Crear" button.

Capítol III

Valoracions i Conclusions

Les noves tecnologies constantment canvien, sobretot en la informàtica, es per tant un requeriment constant el fet d'anar-se actualitzant. En certa manera es la filosofia que vaig seguir quan vaig començar la carrera, l'aprenentatge és quelcom lo més important per un Informàtic Tècnic, Programador o Enginyer.

Aquest projecte potser a estat com pujar al cim d'una muntanya. Les assignatures que es realitzen durant l'inici de la carrera es fan centrades amb unes tecnologies que tens a l'abast, o si més no sempre hi ha algú que et don un cop de mà. A mida que progressem en la carrera veiem un objectiu clar, les assignatures cada cop són més concretes per a la Informàtica i comencem a veure noves tecnologies i maneres d'aplicar-les, tot i que continuem tenint un entorn amb el suport de moltes persones. Finalment en el Projecte comencem a veure tanta quantitat de tecnologia com si fossin els pics d'altres cims que ens semblen inacabables, i a més a més amb continua evolució.

En aquest projecte he treballat amb dos tecnologies que desconeixia completament, una és Struts2 i l'altre és Hibernate3, apart la JEE si que havia sentit parlar però no sabia com s'utilitzava.

Struts2 tot i que al principi les proves que vaig realitzar em van funcionar correctament, a l'hora de treballar amb entitats no em van funcionar, i vaig perdre molt de temps investigant el problema, que finalment vaig resoldre implementant tants accessors com camps hi havia a la pàgina JSP, però vaig tenir que reimplementar bastantes unitats.

Hibernate3 va ser més fàcil, ja que no hi havia canvis substancials respecte a les proves que vaig realitzar. L'únic que potser he apreciat d'aquesta tecnologia es que has de fer un disseny pensant de la manera que treballa, i per tant s'ha de conèixer amb antelació per a poder dissenyar la persistència correctament.

Tot i això la valoració que hi faig és positiva, perquè, encara que inacabada, coneixent les dues tecnologies a aquestes alçades tenia un bon ritme de producció. Potser m'hagués fet falta encara alguna altra tecnologia per poder realitzar les pàgines JSP més modulars i eficaces, i potser més ràpidament.

En general estic content per la feina realitzada, tot i que m'hagués agradat molt més enllestir-la completament. Però la part enriquidora que me'n porto del Projecte i de la Carrera crec que és molt més important.



Glossari

- Bug** : Error o Incidència
- Bugzilla** : Sistema Traçador d'Errors de Codi Obert de l'organització Mozilla, que permet realitzar la gestió d'incidències via Web.
- Component** : Part d'un Programa, normalment són unitats o mòduls que faciliten la ubicació específica d'un procediment.
- Css**: (Cascade Style Sheets) S'utilitza per al disseny de les pàgines Web. Vegeu pàg.26
- Depurador**: Usuari que prova i realitza els test per aconseguir una versió de Component o Producte final.
- Desenvolupador** : Usuari que codifica un Programa. En el manteniment del Programa també repara els errors que vagin comunicant.
- Eclipse**: Framework de desenvolupament majoritàriament del llenguatge Java, que permet fer aplicacions o Productes.
- Event**: És una paraula en informàtica que podríem donar-li el significat d'interrupció que espera un servei determinat.
- Firefox**: Navegador lleuger OpenSource de l'organització Mozilla.
- Framework**: Bastiment de Programació.
- Hibernate** : Entorn de programació per a la persistència de les classes Java. Vegeu pàg.25
- HTML**: (Hypertext Markup Language). És un llenguatge d'etiquetes amb el qual es programen les pàgines Web. Vegeu també XHTML.
- Improve**: Millora d'un Programa o Producte.
- JDBC**: (Java DataBase Connectivity)
- JDK**: (Java Development Kit)
- JEE**: (Java Enterprise Edition)
- JSP**: (Java Server Pages)
- LDAP**: (Light Directory Access Protocol)
- Login**: Normalment l'entrada a un Sistema segur es realitza amb una autenticació.
- Navegador**: Eina per poder visualitzar pàgines HTML de la Web.
- OnLine**: Quan un Programa, Producte o aplicació és troba connectat a la Internet o a alguna xarxa externa.
- OpenSource**: Codi Obert, normalment d'aquestes aplicacions en podem obtenir el codi font.
- Paquet**: En aquest projecte es fa referència com a agrupació de classes Java que tenen alguna cosa comuna.
- Perfil**: D'un Usuari conté informació sobre l'entorn que vol sobre una aplicació perquè aquesta s'adapti a ell lo millor possible.
- POJO**: (Plain Old Java Object)
- Producte**: Aplicació o Programa.
- Servlet**: Programa escrit en Java que s'executa en un Servidor Contenidor. Normalment per ser visualitzat o donar servei via Web.
- Sistema**: Aplicació i part que l'envolta.
- Struts**: Bastiment per programar seguint el model MVC, vegeu pàg. 26
- TomCat**: Servidor Contenidor de Servlets de l'organització Apache, també incorpora un servidor Web.
- Unicode**: Estandard que permet representar el text en qualsevol idioma.
- Web**: (World Wide Web) La xarxa d'Internet sobre pàgines HTML i derivades.
- XHTML**: (Extensible Hypertext Markup Language) Vegeu pàg. 26.



Bibliografia i Recursos Web.

Hibernate Quickly

Autor: Patrick Peak & Nick Heudecker

Editorial: Manning Publications Co.

Hibernate in Action

Autor: CHRISTIAN BAUER & GAVIN KING

Editorial: Manning Publications Co.

Starting Struts2

Autor: Ian Roughley

Editorial: C4Media Inc.

JavaServer Pages™ Specification

Autor: Pierre Delisle, Jan Luehe, Mark Roth

Editorial: Sun Microsystems, Inc.

THE J2EE™ Tutorial

Autor: Stephanie Bodoff, Dale Green, Kim Haase, Eric Jendrock, Monica Pawlan, Beth Stearns

Editorial: Sun Microsystems, Inc.

Hibernate Reference Documentation Version: 3.2.2

Referencia Oficial d'Hibernate3.

<http://struts.apache.org/2.x>

Pàgina Oficial d'Struts2.

<http://www.hibernate.org>

Pàgina Oficial d'Hibernate3.



DynBTS

Dynamic Bug Tracker System

 **UOC**

DynBTS

Dynamic Bug Tracker System

Projecte amb JEE
Sistema de Control d' Errors i Millores

Eduard Calveras i Anguita

Enginyeria Tècnica d'Informàtica de Sistemes.

Consultor: Albert Grau i Perisé

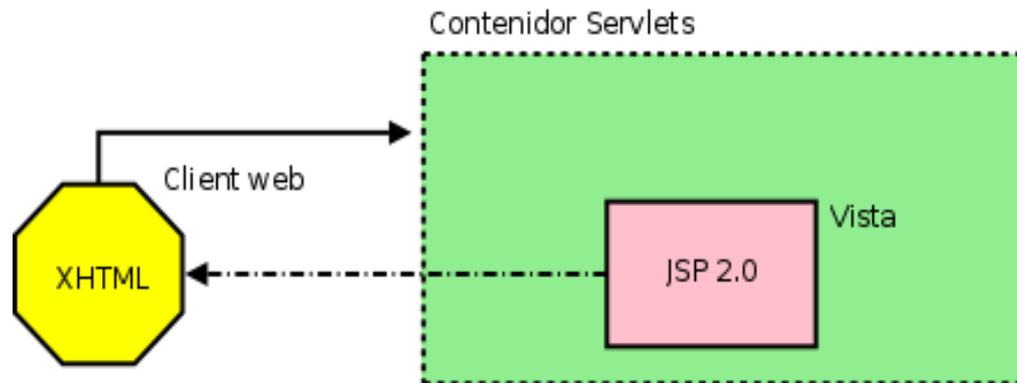
Juny de 2008
Universitat Oberta de Catalunya



Introducció

DynBTS és una Aplicació Web que permet realitzar entrades d'errors i millores de varis Programes externs a aquest, Tant l'entrada com el seguiment es pot fer mitjançant un navegador Web que compleixi els estàndards establerts per la W3C per a XHTML1.0 i CSS2.1.

El servidor proporciona una resposta apropiada per a cadascun dels events que genera l'usuari a partir de pàgines JSP.



Tecnologia aplicada

JEE 5.0

- Anteriorment anomenat J2EE, es la base de tot el desenvolupament de l'aplicació, a excepció de la part visual, i difereix en altres plataformes en que amplia el repositori amb altres tecnologies, com poden ser Struts i Hibernate.

Struts 2

- Es un framework de programació que proporciona un aïllament entre la capa visual, de control i la de negoci d'una aplicació JEE, i que anomenem MVC (Model View Controler). Al mateix temps, aquesta depèn d' altres tecnologies:
 - *XHTML 1.0*
 - *CSS 2.2*
 - *JSP 2.0*

Tecnologia aplicada

XHTML 1.0

- Es bàsicament el llenguatge amb la qual es compon cada una de les pàgines Web que l'usuari veu al seu navegador.

JSP 2.0

- (Java Server Pages) és la tecnologia que utilitza el framework Struts2 per emplenar adientment les pàgines web amb els resultats provinents de la capa de negoci.

CSS 2.2

- Les fulles d'estil són un element indispensable per tractar el "*look and feel*" de les pàgines web. En certa manera és el responsable de tenir varis temes visuals en una mateixa aplicació.

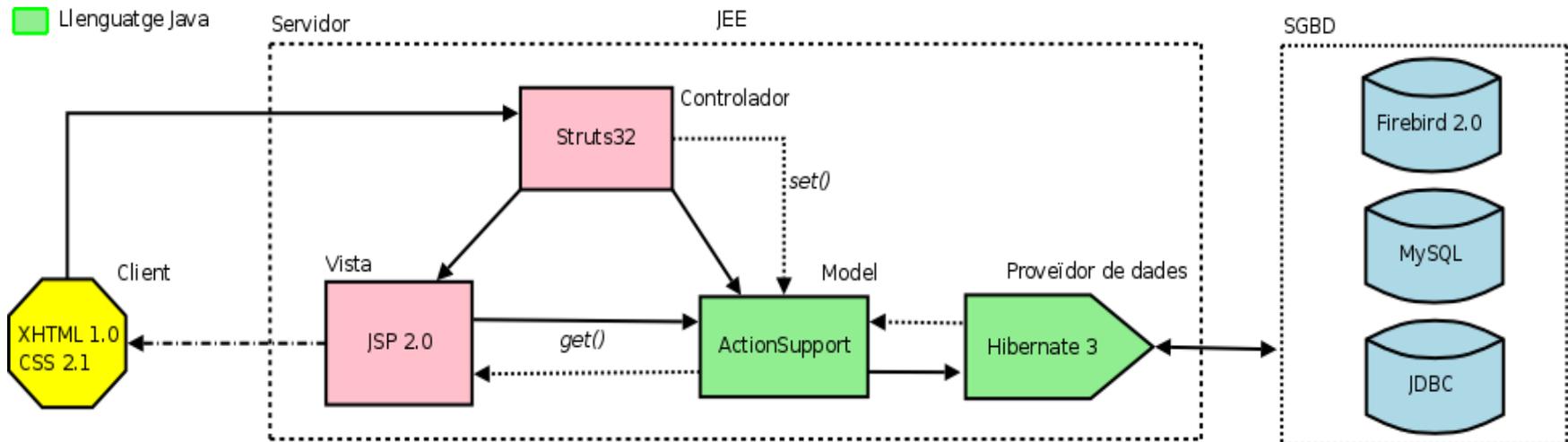


Tecnologia aplicada

Hibernate 3

- Proveïdor de dades del model de negoci. Es una tecnologia orientada a objectes que abstrau les funcionalitats d'interacció amb els SGBD, quedant la implementació independent del tipus de Base de Dades.

El sistema funcional d'aquestes tecnologies el podríem representar en el següent esquema:



Entorn de desenvolupament

No sense menystenir, per a realitzar aquest projecte s'han utilitzat unes eines indispensables, tot i que podrien ser unes altres.

Eclipse 3.2.2

- Entorn de desenvolupament per a Java, i altres llenguatges que suporta extensions, moltes del JEE, per desenvolupar coses més específiques.

Apache Tomcat 5.5

- Servidor contenidor de Servlets, per a executar el nostre projecte realitzat amb Java. Implementa les especificacions per a JSP 2.0 necessàries per al nostre projecte. També incorpora un propi servidor HTML i per tant no es necessària la instal·lació d'un servidor Web addicional.

Entorn de desenvolupament

Firefox 2

- Aquesta aplicació pot ser visualitzada amb qualsevol navegador, ja que compleix els estàndards de la W3C, però en les proves s'ha utilitzat aquest.

GNU/Linux Debian Etch 2.6.23

- El sistema operatiu en el qual s'ha realitzat i provat el projecte DynBTS. Sobre el gestor de finestres Gnome. Funcionant sobre un ordinador portàtil Intel Core2 @ 1,88GHz i 2GB de RAM.

Resum funcional

Els usuaris i grups

En l'aplicació DynBTS s'han creat uns grups de treball, cadascun té el seu rol, però un usuari pot pertànyer a qualsevol d'ells:

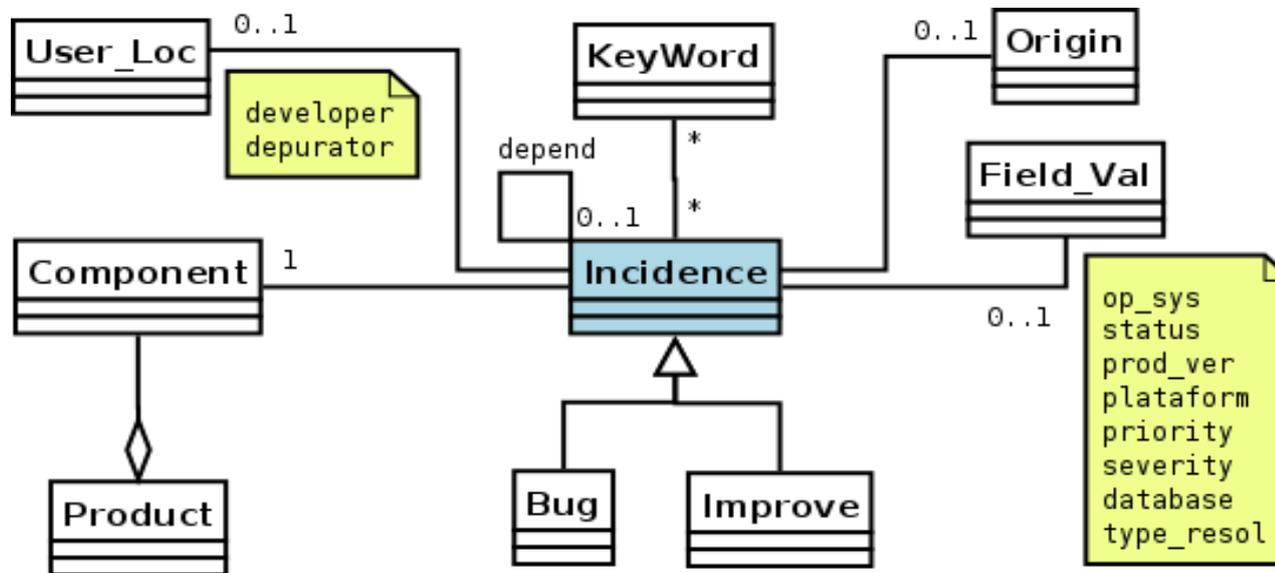
- **Invitats:** Visualitzar l'estat d'una incidència i els comentaris.
- **Depuradors:** Introdueixen noves incidències, i també les corregeixen i tanquen.
- **Desenvolupadors:** S'encarreguen de reparar o resoldre la incidència
- **Supervisors:** Són desenvolupadors encarregats d'un producte, i assignaran les noves incidències als desenvolupadors que creguin adient.



Resum funcional

Classes principals

El projecte es basa entorn a la classe persistent *Incidence*, que representa tant errors com millores. Tot i que hi han d'altres que no es presenten en aquest diagrama:

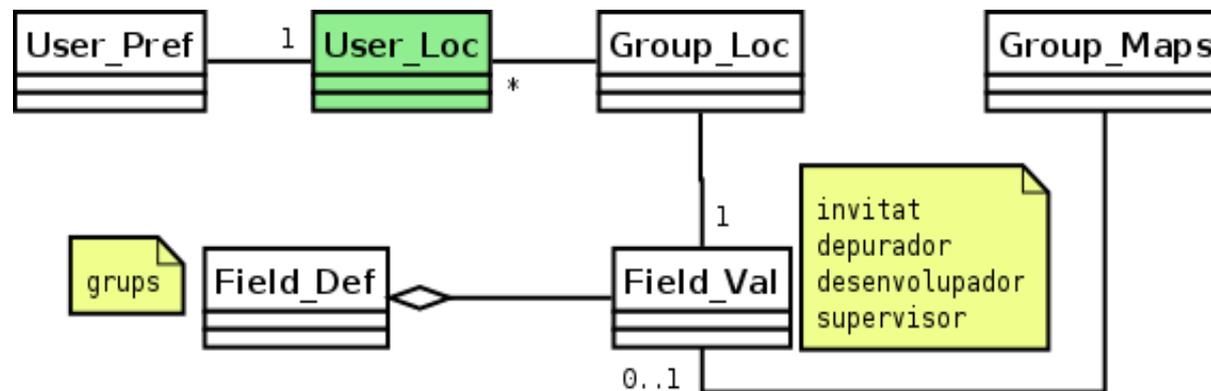




Resum funcional

Classes principals

També es important l'entorn d'usuari, la seguretat del sistema passa per una bona política de privilegis. La definició de grups, com en l'anterior diagrama, es recupera d'un camp genèric:



Com a comentari sobre la seguretat, aquest projecte desa un *Hash* de la contrasenya de cada usuari, amb la qual no es possible el seu desxifratge.

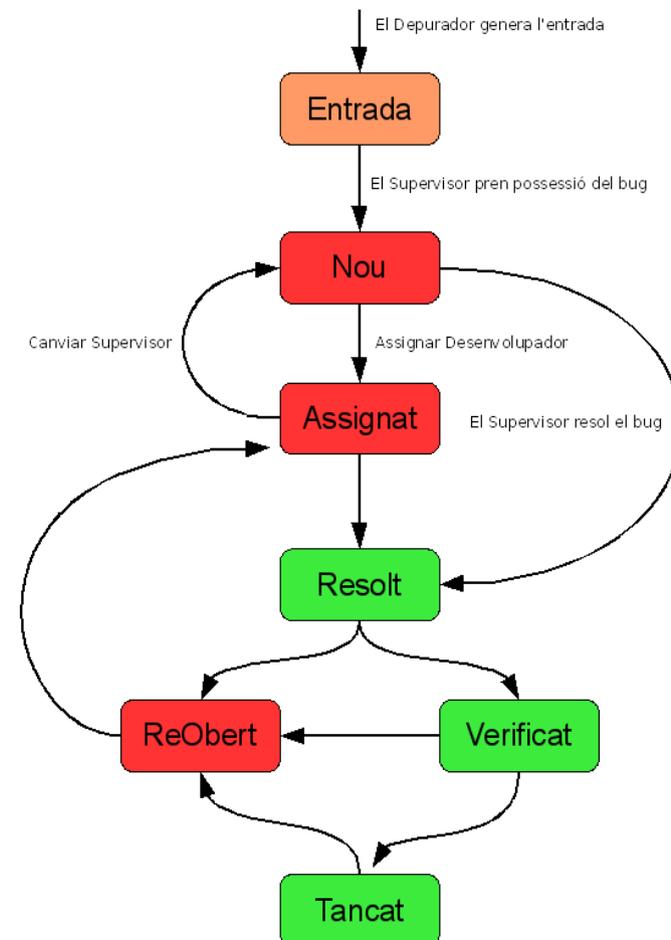


Resum funcional

La vida d'un bug

Des de que s'obre fins que es finalitza una Incidència passen unes etapes, entre elles hi intervenen varis usuaris, ie:

- El *Depurador* l'obre,
- El *Supervisor* l'assigna,
- El *Desenvolupador* la resol.
- El *Depurador* la verifica i la tanca, o bé la pot reobrir.



Resum funcional

Estructura del projecte.

DynBTS vista per un programador està estructurada d'una manera entenedora, si més no organitzada per nivells, paquets o directoris:

- Pel que fa les classes de Java:

-  dynbts Classes de comunicació amb les JSP i Struts2.
-  dynbts.logic Classes amb la lògica on intervé Hibernate3.
-  dynbts.persistent Classes persistents o del món real.

- Pel que fa l'entorn Web, dins de *WebContent*:

-  index.html Pàgina d'entrada a l'aplicació.
-  dynbts Conté totes les pàgines JSP / XHTML i CSS
-  dynbts/img Conté totes les imatges que utilitza l'aplicació.



Conclusions

DynBTS és un projecte ambiciós, que mira de tenir un dinamisme en quan a l'aparença visual, en quan al llenguatge, en quan al tipus de base de dades, i a més a més, com està realitzat amb Java és multiplataforma.

L'aplicació no està totalment acabada, però els pilars principals com és l'estructura de persistència i la majoria de lògica de negoci assoleixen el 75% de funcionalitat actualment en funcionament. L'aparença de l'aplicació és una part important de l'aplicació i en aquest projecte s'ha endut bona part del temps dedicat per les dificultats del HTML amb entorns RAD.

Però el fet d'haver pogut dedicar aquest temps al JEE m'ha fet veure el munt d'aplicacions que es poden arribar a fer, i a més a més, com utilitzar altres tecnologies que no s'apliquen en aquest projecte però que he trobat útil de cara a posteriors.



Gràcies per l'atenció.

FI