

# Treball Final de Carrera – J2EE

## SegurGest

Aplicació J2EE per a la gestió d'assegurances i sinistres.



14 de Juny de 2016

Consultor: Vicenç Font Sagrista

SERGI RIBAS RUZ

E.T. Informàtica de Gestió

Universitat Oberta de Catalunya

# Introducció

L'objectiu d'aquest projecte és desenvolupar un eina per la gestió d'assegurances (pòlisses, sinistres, noves peticions) d'una empresa, i que alhora augmenti la comunicació entre l'empresa i tots els corredors/agents d'assegurances que aquesta tingui.

L'empresa introdueix totes les seves assegurances, i aquestes es vinculen a cada un dels corredors a qui pertany l'assegurança, perquè aquest puguin interactuar amb el client (empresa) podent oferir noves ofertes de renovació, modificant dades de la pòlissa actual, etc etc.

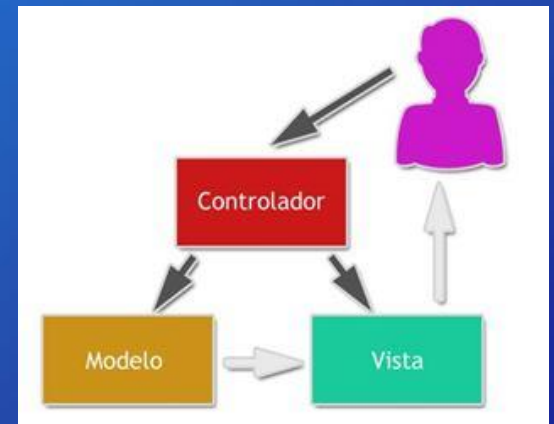
Es tracta d'un projecte real que m'ha demanat una empresa força gran del sector retail, i és qui m'ha transmès tots els requisits que necessiten.



# Tecnologia

S'ha seleccionat la tecnologia J2EE per al desenvolupament d'aquesta aplicació, ja que és el més extens actualment en quan a sistemes de serveis web.

S'aplicarà el patró MVC [(M)odel (DAO) – (V)ista (JSP) – (C)ontrolador (Servlets)], ja que ens permet crear un sistema més robust amb un cicle de vida més adequat, on es potencia la facilitat de manteniment, reutilització de codi i la separació de conceptes.



És una arquitectura que separa les dades i la lògica de negoci d'una aplicació, de la interface d'usuari i el mòdul encarregat de gestionar els events i les comunicacions.



# Tecnologia

La interfície està dissenyada de manera clara, amigable i intuïtiva. On el client sols tindrà la interfície d'usuari. S'utilitza el framework de Bootstrap per el disseny. Està configurada en un servidor web Apache, ja que té la capacitat de gestionar servlets de Java. I qualsevol que tingui accés i permisos podrà executar-la, sense necessitat d'instal·lar res en les màquines dels clients. Utilitza els estàndards del mercat per al software de servidor (JSP, J2EE, etc.)

Les dades s'emmagatzemen en MySQL, ja que es un gestor de base de dades OpenSource. I està molt utilitzada en entorns professionals.

S'utilitza la llibreria "taglibs" d'Struts, per la simple raó d'estalvi de costos (Servidor "bàsic" de cost 0, poc emmagatzematge, gran quantitat de peticions, estalvi d'ús de memòria).



# Arquitectura (MVC – Struts)

Struts implementa els components del patró MVC:

→ El controlador és un servlet... S'ha de configurar l'aplicació web perquè totes les sol·licituds d'usuari es redirigeixin a aquest servlet.

→ El controlador envia les sol·licituds de l'usuari a la classe adequada per executar l'acció.

→ La vista s'implantarà utilitzant pàgines JSP.

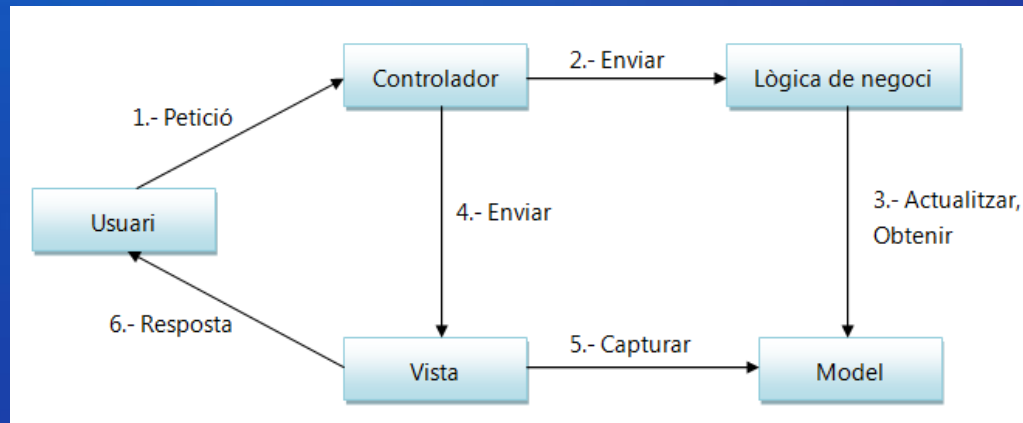
→ La implementació del model corre totalment a càrrec del programador, ja que és propi de la capa de negoci i no és dins de l'àmbit d'Struts.



# Arquitectura (MVC – Struts)

**El procés d'una petició amb Struts és el següent:**

- 1.- El client fa la petició, que rep el controlador Struts.
- 2.- El controlador envia la petició, identificant l'acció i executant la lògica de negoci adequat.
- 3.- La lògica de negoci actualitza el model i obté dades de la mateixa, emmagatzemant-los en beans.
- 4.- En funció del valor retornat per la lògica de negoci, el controlador escull la següent vista a mostrar.
- 5.- La vista captura les dades obtingudes per la lògica de negoci.
- 6.- La vista mostra les dades al client.



# Resum funcional de SegurGest

L'aplicació permet gestionar les assegurances, i tot el que comporta, com: sinistres, petició de noves assegurances; sigui quina sigui la seva companyia asseguradora i sigui quin sigui el seu corredor.

Els usuaris tenen uns rols assignats que els hi dona accés a les diferents funcionalitats de l'aplicació.

Hi ha 3 tipus d'usuaris:

- Usuari Administrador (de la web)
- Usuari Manager (corredor d'assegurances)
- Usuari civil (poden ser de 3 tipus)
  - Gestors d'empresa
  - Empleats
  - Persona física / autònom



# Resum funcional de SegurGest

**Usuari Administrador:** Serà l'encarregat d'assignar les diferents assegurances i peticions a un corredor. I de modificar les dades dels usuaris en cas de tenir que desactivar el compte o canviar el e-mail d'accés etc etc. i a més tindrà permís absolut en tota l'aplicació.

**Usuari Manager:** És la figura del corredor/agent d'assegurances. Aquest podrà gestionar les assegurances, les peticions i els sinistres que tingui assignats.

**Usuari Civil:** Són els usuaris d'empresa i/o empleats de l'empresa que podran afegir les assegurances al sistema, demanar peticions per a noves assegurances i gestionar els sinistres que hi hagi. N'hi ha de 3 tipus...





# Resum funcional de SegurGest

## **Usuari civil – Gestor d'empreses:**

*Un Grup d'empreses (Account), que està formada per varies empreses (Company). Aquest "account" té varis usuaris gestors (User) que poden gestionar qualsevol assegurança i/o sinistre de qualsevol de les empreses del grup. Excepte si, tot i formar part del grup, només està assignat a una empresa d'aquest grup (camp "company\_administrator" de la taula "common\_user").*

## **Usuari civil – Persona física/autònom:**

*Una única "account", que està formada per només una "company". I on tenim també només un "user" que podrà gestionar qualsevol assegurança i/o sinistre que pertanyi a ell mateix.*

## **Usuari civil – Empleat:**

*Ídem al tipus "Persona física/autònom" però que a més està vinculat a una empresa a través del camp "departament" de la taula d"usuaris".*



# Resum funcional de SegurGest

## Gestió d'assegurances i petició de nous projectes:

*L"usuari civil" pot crear, modificar i eliminar (borrat lògic) assegurances i peticions.*

*Aquestes assegurances són assignades per l"usuari administrador" a un "usuari manager" (corredor).*

*Un cop assignada, el manager ja pot modificar l'assegurança, establint així una comunicació directe entre corredor i prenedor de l'assegurança. On el corredor pot, per exemple, oferir noves condicions de renovació abans del final de termini, sense la necessitat de que el client estigui pendent d'aquest venciment.*



# Resum funcional de SegurGest

## Gestió d'assegurances i petició de nous projectes:

*Possibles estats d'una assegurança:*

- *Pendent d'assignar (per defecte)*
- *Assignat a manager.*
- *Tramitant*
- *Petició modificació*
- *Petició cancel·lació*
- *Cancel·lat*
- *Validat*

*Possibles estats d'una petició d'assegurança:*

- *Pendent d'assignar (per defecte)*
- *Assignat a manager.*
- *Tramitant*
- *Aprovat*
- *Rebutjat.*



# Resum funcional de SegurGest

## Gestió de Sinistres:

L'“usuari civil” pot crear, modificar i eliminar (borrat lògic) els seus sinistres. Aquest s'assignarà al “usuari manager” que estigui vinculat amb la pòlissa.

Amb el sinistre informat, el manager ja podrà “acceptar” o “rebutjar” el sinistre. Si aquest és acceptat, es podrà anar informant de tots els passos seguits, tant per el client com per el corredor, de cara a resolució del mateix.

Possibles estats d'un sinistre:

- Obert (per defecte)
- Acceptat
- Rebutjat
- Tancat



# Resultats obtinguts

Totes les tasques planificades s'han assolit al 100%, a falta de fer la posada en marxa dins de l'empresa... fet que trobo un èxit.

Com a següent punt, l'empresa plantejarà el desenvolupament de noves funcionalitats com: gestió de notícies, gestió d'incidències, descarrega de nomines, irpf i contractes laborals... i moltes d'altres que ja tenen pensat.

Al tractar-se d'una aplicació totalment escalable no hi haurà cap problema per satisfer les seves necessitats.



# Conclusions

D'aquest projecte n'he tret una experiència molt positiva. He profunditzat molt més en el funcionament de la tecnologia J2EE, i que de ben segur podré aprofitar en futurs projectes.

En quan a SegurGest, ha sigut dels primers projectes que he portat a terme amb una planificació dia a dia, documentació, etc... I la veritat és que ajuda a preparar molt millor tot el procés de implementació.

Considero que seria molt interessant poder dedicar sempre un temps a analitzar i planificar un projecte, tal i com he fet amb aquest. Però la veritat és que quan un és petit, normalment el client no deixa marge de maniobra per poder dedicar-hi aquest temps.

