

# **Anàlisi, diseny i implementació d'un aplicació de gestió d'espais fent servir l'arquitectura J2EE**

**Xavier Vega Cobo**  
ETIS

**Albert Grau Perisé**

11/05/2004

# Dedicatòria i agraïments

Per a Hèctor.

M'agradaria aprofitar aquestes línies per a expressar el meu més sincer agraïment a totes les persones que directa o indirectament han fet possible la dificultosa elaboració d'aquest treball de fi de carrera, principalment al meu consultor, Albert Grau Perisé, pels seus ànims als moments difícils i el seu suport incondicional i, finalment, a la meva dona Angels per la seva infinita comprensió.

# Resum

L'objectiu principal d'aquest treball és exposar tots els passos seguits pel desenvolupament d'un projecte (anàlisi, disseny, implementació i proves) utilitzant una tecnologia determinada: l'arquitectura J2EE, una arquitectura escalable que permet un desenvolupament simplificat dels projectes d'aplicacions

El projecte escollit és el de una aplicació de Gestió d'Espais, la finalitat del qual és gestionar els espais que pertanyen a una entitat multiespaial, com per exemple una Universitat o un hospital, que pugui ser delimitat físicament per un àrea concreta i la explotació d'aquesta informació, tot això tenint el nostre usuari com a eina de treball el navegador.

El document comença explicant els objectius de la aplicació i els seus clients potencials. Posteriorment exposo els requisits del projecte, i els detalls de major importància del seu anàlisi, de la feina de disseny i d'implementació que he realitzat.

S'inclueix també una petita introducció als conceptes i arquitectura J2EE, al model MVC (Model, View, Controller) i a la tecnologia d'Struts. Finalment s'exposan les conclusions obtingudes al llarg de tot aquest procés de desenvolupament així com les possibles línies de treball futures

# Index de continguts

<b>DEDICATÒRIA I AGRAÏMENTS .....</b>	<b>2</b>
<b>RESUM .....</b>	<b>3</b>
<b>INDEX DE CONTINGUTS.....</b>	<b>4</b>
<b>INDEX DE FIGURES.....</b>	<b>6</b>
<b>EL PROJECTE DE GESTIO D'ESPAIS .....</b>	<b>7</b>
<b>INTRODUCCIÓ. ....</b>	<b>7</b>
<i>Justificació i context del TFC.....</i>	<i>7</i>
<i>Objectius del TFC.....</i>	<i>7</i>
<i>Enfocament i mètode seguit.....</i>	<i>7</i>
<i>Planificació del projecte.....</i>	<i>8</i>
<i>Productes obtinguts.....</i>	<i>8</i>
<i>Estructura i contingut dels següents capítols de la memòria.....</i>	<i>8</i>
<b>REQUISITS .....</b>	<b>9</b>
<i>Requisits Funcionals .....</i>	<i>9</i>
Gestió i consulta dels Edificis de la Organització.....	9
Gestió i consulta dels subespais.....	9
Generació d'informes i llistats.....	9
<i>Requisits No Funcionals.....</i>	<i>11</i>
Requisits d' Interfície .....	11
Comportament .....	11
Estil.....	11
Facilitat d'ús.....	11
<i>Requisits de Producció.....</i>	<i>12</i>
Rendiment.....	12
Precissió.....	12
Compromisos de temps i horaris.....	12
Volums d'informació.....	12
Entorn físic i tecnològic .....	12
Nombre d'usuaris finals i concurrents .....	12
Distribució del software .....	13
Consumibles .....	13
Impacte .....	13
<i>Requisits de Formació.....</i>	<i>14</i>
Seguretat .....	14
Manteniment .....	14
Reutilització.....	14
<i>Requisits de les Proves.....</i>	<i>15</i>
Necessitats i Mitjans .....	15
Eines .....	15
Entorn Hardware/Software .....	15
<b>ANÀLISI.....</b>	<b>16</b>
<i>Diagrama de Casos d'ús principal.....</i>	<i>16</i>
<i>Descripció textual dels casos d'us.....</i>	<i>17</i>
Llistar espais .....	17
Sel.leccionar espai .....	17
Crear espai .....	18
Modificar espai .....	19
Eliminar espai .....	20
Crear subespai.....	21
Sel.leccionar subespai.....	22
Modificar subespai.....	23
Eliminar subespai.....	23
<i>Diagrama de Classes del Sistema .....</i>	<i>25</i>

<b>DISSENY</b> .....	27
<i>Programari</i> .....	27
<i>Disseny de la base de dades</i> .....	28
<i>Disseny de les interfícies gràfiques</i> .....	29
<i>Disseny arquitectònic</i> .....	33
Us de J2EE .....	33
Model MVC (Model Vista Controlador) .....	33
Us d' Struts .....	34
<b>IMPLEMENTACIÓ</b> .....	36
<i>Implementació de la base de dades</i> .....	36
Model : package ge.model .....	38
View : jsp's .....	38
Controller: package ge.controller .....	39
<b>CONCLUSIONS</b> .....	41
<b>GLOSSARI</b> .....	42
<b>BIBLIOGRAFIA</b> .....	43

## Index de figures.

Il·lustració 1: Diagrama de casos d'ús principal .....	16
Il·lustració 2 : Diagrama de classes del sistema .....	25
Il·lustració 3 : Disseny de la base de dades.....	28
Il·lustració 4 : Pantalla de selecció de l'espai.....	30
Il·lustració 5 : Pantalla de visualització d'un edifici .....	31
Il·lustració 6 : Pantalla de modificació d'un local.....	32
Il·lustració 7 : Esquema del model MVC .....	34
Il·lustració 8 : Esquema d'Struts.....	35

# El projecte de Gestio d'espais

## ***Introducció.***

### **Justificació i context del TFC**

A l'hora de plantejar-me el TFC, tenia clar que ho faria relacionat amb la arquitectura J2EE i això és així perquè actualment en qualsevol revista especialitzada, congrés d'informàtica o simplement xarradeta entre programadors i analistes n'hi ha una idea que sembla molt extesa: que l'arquitectura J2EE representa un pas endavant en la evolució de la tecnologia Java, ja que és una arquitectura escalable que permet un desenvolupament simplificat de les complexes aplicacions empresarials on es separa les tres capes de funcionalitat del model MVC en tipus específics de components.

A més de que el fet de l'aprenentatge d'aquesta arquitectura era molt important en el TFC, volia que l'aplicació que l'exemplificava fos també interessant per un possible mercat, i per aquesta raó vaig escollir una aplicació de Gestio d'Espais perquè penso que és una problemàtica molt extesa, tant dins de l'administració com en el món empresarial, i que actualment no existeixen una gran quantitat de productes que la puguin solucionar.

### **Objectius del TFC.**

En totes les administracions es necessari tenir un coneixement detallat del espais físics amb que es compta i quines activitats es fan i es poden fer en ells, sobre tot en grans entorns com poden ser universitats u hospitals, que poden estar composades de més d'un edifici i les tasques poden ser de molts tipus (docència, recerca, administració, etc...). Aquests espais, de gran heterogeneïtat i amb constants modificacions internes dels edifici, construccions de nova planta, canvis en les activitats, etc... haurien de ser gestionats per una secció genèrica, com pot ser un Servei d'Obres de la institució en la feines d'alta, baixa i modificacions d'edificis pero podrien ser també gestionats per una utilització concreta d'alguna de les seves parts per coordinadors de cada edifici o zona i finalment es podria explotar aquesta informació oferint en internet les dades principals: adreces, parts components de l'edifici, etc...

Així doncs l'objectiu d'aquest treball és la gestió de qualsevol espai físic que formi part d'una entitat multiespacial, com per exemple una Universitat o un hospital, que pugui ser delimitat físicament per un àrea concreta i la explotació d'aquesta informació.

### **Enfocament i mètode seguit.**

L'enfocament inicial del projecte va ser seguir una metodologia de desenvolupament clàssic e incremental on cada etapa de desenvolupament d'un projecte (requisits, anàlisi, disseny, implementació i prova) estigués tutoritzada pel consultor, que provocaria un replantejament i retoc de cadascuna de les etapes abans de passar a la següent.

## Planificació del projecte.

La meua planificació inicial era:

Feina	Entregable	Data
Pla de treball i instal·lació del programari	Pla de treball Temporització	30/09
Anàlisi prèvia i anàlisi de requisits	Diagrama de classes; Diagrama de casos d'ús	29/03
Disseny	Disseny Físic BD; Disseny arquitectònic; Disseny de les pantalles	29/03 29/03 29/03
Programació	Programació Clases	26/04
Prova	Plà de proves	01/06
Entrega del projecte	Memoria del projecte; Aplicació	18/06

## Productes obtinguts

El producte obtingut és una aplicació que gestiona espais, la seva alta, baixa i modificació i la seva inclusió d'uns dintre d'altres, juntament aamb la informació associada a aquests, com pot ser el tipus (si és un edifici, un local, etc...) la activitat a la qual es dedica, l'adreça de l'espai, etc...

## Estructura i contingut dels següents capítols de la memòria.

Als següents capítols explico detalladament les principals característiques de les diferents etapes de desenvolupament del projecte: requisits, anàlisi, disseny, implementació i proves.

També explico breument l'arquitectura J2EE, el model MVC i la tecnologia d'Struts, acabant amb unes conclusions i la bibliografia utilitzada.



## **Requisits**

### **Requisits Funcionals**

#### **Gestió i consulta dels Edificis de la Organització**

La gestió dels espais de l'organització implicaria no només l'alta i baixa dels edificis amb la seva informació associada, que inclou el fet de poder aportar documents fotogràfics d'aquests. Haurà d'incloure, a més, la possibilitat d'agrupar varis edificis en àrees ( ex. Àrea nord, Àrea sud, etc...). A més, haurà d'informar de cada edifici: el seu nom, la seva descripció, un codi d'ús intern, els metres quadrats aproximats i la seva adreça, formada per carrer, num, codi postal, poblacio, provincia, apart d'un camp d'observacions. Aquesta gestió haurà de realitzar-se de la forma més senzilla possible, facilitant l'entrada de dades i la modificació de les mateixes per a maximitzar la quantitat d'informació que un número limitat de recursos puguin posar a disponibilitat dels usuaris consultants.

#### **Gestió i consulta dels subespais**

La gestió dels subespais es farà indicant en cada Edifici, els subespais dels qual forma part. Aquest subespais, podran ser bàsicament:

- plantes
- locals, que hauran de ser obligatoriament creats dins d'una planta

Les plantes tindran la informació següent: el seu nom, la seva descripció, un codi d'ús intern, i els metres quadrats aproximats

Mentre que els locals tindran la següent informació: el seu nom, la seva descripció, un codi d'ús intern, els metres quadrats aproximats, el tipus de superfície (Administració, Docència, Hospitalaria, etc...), l'activitat assignada, la capacitat de persones i observacions

#### **Generació d'informes i llistats**

Amb totes les dades es podran fer informes i llistats

##### **□ Subespais per espai**

**Parametres d'entrada:** Nom d'espai

**Sortida:** Subespais que pertanyen a aquest espai, ordenats per plantes i dintre de cada planta els diferents locals

##### **□ Adreces**

**Parametres d'entrada:** Cap

**Sortida:** Edificis dels que es compona l'organització amb la informació expressa de la seva adreça

□ **Capacitat dels locals**

**Parametres d'entrada:** Activitat, num. de persones

**Sortida:** Locals on es fa una determinada activitat amb una capacitat mínima del nombre de persones demanat

## **Requisits No Funcionals**

### **Requisits d' Interfície**

La interfície haurà de ser intuïtiva, ràpida i fàcil de fer servir. A més a més, haurà d'oferir a l'usuari, en tot moment, un sistema d'ajuda on-line on es resolguin tots els dubtes que se li puguin plantejar a l'usuari sobre el funcionament de l'aplicació a més a més d'informació complementària sobre el servei que aquesta proporciona.

La interfície també haurà de notificar, quan calgui, a l'usuari d'una operació en curs per mitjà d'un gràfic animat o altre mitjà que es consideri adequat per a aquesta finalitat, de manera que les operacions que requereixin un llarg temps de processament reflexin activitat de fons.

### **Comportament**

L'aplicació haurà de mostrar un comportament estable, de manera que si existeixen problemes durant l'execució, s'informarà adequadament a l'usuari, si és pertinent, per a que aquest pugui dirigir la gestió de l'error per les vies tradicionals (trucada a HelpDesk, etc.)

### **Estil**

La part visual del servei haurà de complir amb el llibre d'estils de la Intranet Corporativa.

### **Facilitat d'ús**

L'aplicació haurà de complir amb els requisits estàndard d'intuïtivitat i facilitat d'ús d'una aplicació Web.

## **Requisits de Producció**

L'aplicació haurà d'estar localitzada com a part del Portal Corporatiu de la Intranet de l'empresa

L'aplicació haurà de servir les dades d'una manera ràpida, amb un funcionament estable, accessible les 24 h.

S'haurà de tenir en compte la possible concurrència en accessos a dades i l'escalabilitat en quant als requeriments del sistema.

## **Rendiment**

L'aplicació haurà de servir les dades a l'usuari en el mínim temps possible.

## **Precisió**

Les dades que l'aplicació serveixi hauràn de ser actuals.

## **Compromisos de temps i horaris.**

La disponibilitat del servei serà de 24 h..

## **Volums d'informació**

Degut a què el servei no existeix a l'actualitat, no es pot fer una estimació vàlida sobre els volums d'informació. És per això que s'haurà de dissenyar l'aplicació de manera que un alt volum de dades no signifiqui una pèrdua d'estabilitat ni de rendiment del servei.

## **Entorn físic i tecnològic**

L'entorn físic es basarà en un servidor d'aplicacions ORACLE i una base de dades del mateix tipus

Tecnològicament s'utilitzarà l'entorn J2EE.

## **Nombre d'usuaris finals i concurrents**

S'ha de distingir els 3 àmbits del projecte, amb el seu respectiu rang d'usuaris. Així doncs, distingirem:

- Gestors: Els usuaris seràn, principalment el departament d'Obres i Patrimoni (entre 10 i 15 usuaris)
- Coordinadors: Els usuaris seràn 1 o 2 coordinadors per cada edifici de la base de dades
- Usuaris: Tots aquells usuaris de la Intranet i de Internet sensibles de poder visualitzar les dades dels espais

### **Distribució del software**

La distribució del software serà la ja estandarditzada per a aplicacions d'entorn Web Intranet.

### **Consumibles**

N.A.

### **Impacte**

Degut a què no es substitueix cap aplicació existent, no s'ha de contemplar un impacte sobre cap aplicació existent tret del moment de l'implantació de l'aplicació a l'entorn de Producció que, sota cap concepte haurà d'afectar al servei de cap aplicació existent.

## **Requisits de Formació**

Els requisits de formació per als dos perfils d'administració (Gestori Coordinador) seran semblants però no es necessitarà una formació específica per a l'usuari.

L'informació sobre l'aplicació i com fer-la servir ha d'estar accessible on-line.

## **Seguretat**

Es seguiràn els estàndards de l'entorn Intranet de Seguretat.

## **Manteniment**

El manteniment evolutiu del servei, un cop posat a l'entorn de Producció, estarà a càrrec del departament de Manteniment d'aplicacions Web Intranet/Internet.

## **Reutilització**

Tota la construcció haurà de fer-se amb la base del desenvolupament de components reutilitzables complint les normes definides a l' Organització.

## **Requisits de les Proves**

### **Necessitats i Mitjans**

Es requeriràn llocs amb l'instal·lació estàndard de la Intranet (navegador, ...) amb la configuració estàndard, però amb accés a la URL de pre-producció (entorn on es duràn a terme les proves d'usuari).

Els usuaris escollits per a realitzar les proves hauràn de ser donats d'alta a l'entorn de pre-producció.

S'haurà de disposar, per tant, dels diferents entorns de pre-producció que es designin.

### **Eines**

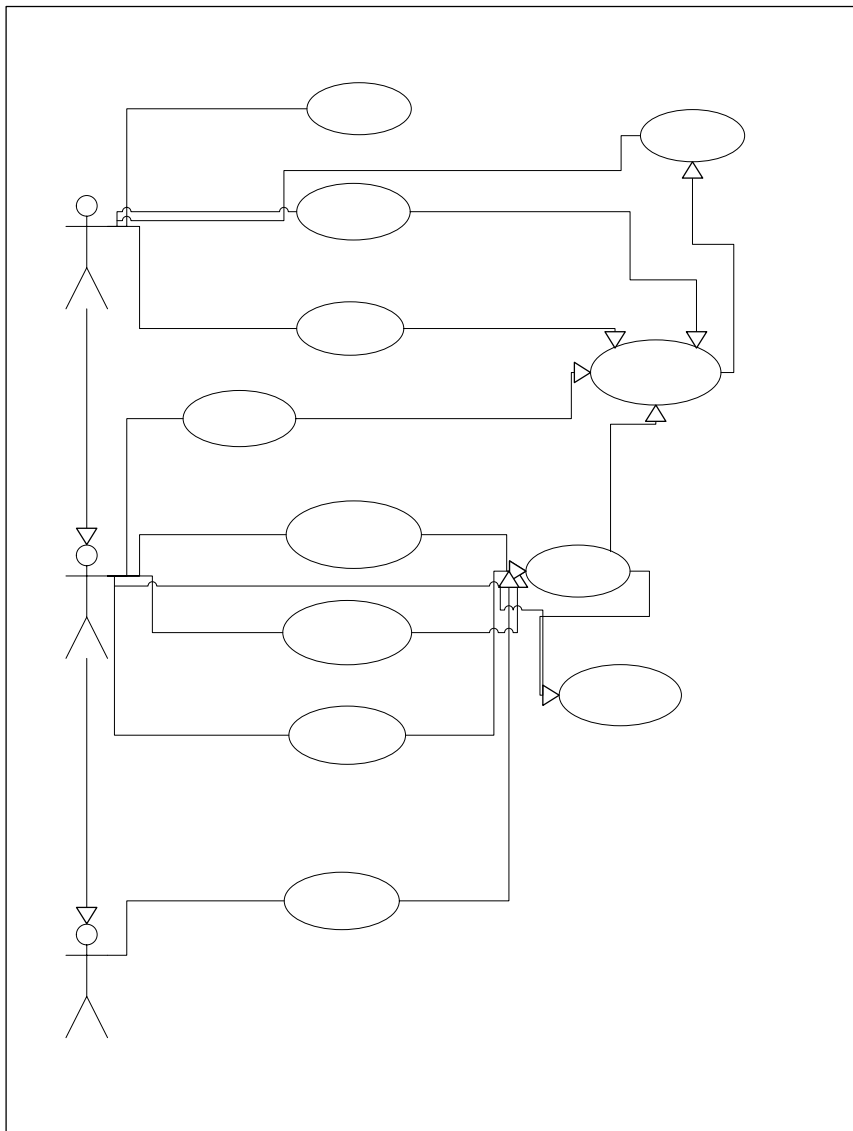
Les estàndard per a les aplicacions Web de la Intranet

### **Entorn Hardware/Software**

L'estàndard per a les aplicacions Web de la Intranet.

## Anàlisi

### Diagrama de Casos d'ús principal



Il·lustració 1: Diagrama de casos d'ús principal



## Descripció textual dels casos d'ús

### Llistar espais

#### Descripció

Aquest cas d'ús encapsula la funcionalitat encarregada de llistar tots els espais. Els espais es correspondrien als edificis principals, que poden estar organitzats en campus o àrees

#### Pre-Condicions

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

#### Post-Condicions

N/A

#### Fluxos d'events

##### *Flux Principal – Llistat deEspais*

1. El sistema llistarà a l'entrada de l'aplicació els edificis principals ja introduïts a l'aplicació, organitzats, si es vol, en àrees, que s'obtenen de l'entitat EDIFICIS
2. L'usuari pot desplegar una àrea determinada per veure els edificis dels quals formen part
3. L'usuari pot replegar una àrea determinada per netejar la pantalla
4. L'usuari pot seleccionar una àrea o un edifici concret (cas d'ús Seleccionar espai)
5. L'usuari pot crear un nou espai (cas d'ús Crear espai)

### Seleccionar espai

#### Descripció

Aquest cas d'ús encapsula la funcionalitat encarregada de seleccionar un espai determinat. Els espais es correspondrien als edificis principals, que poden estar organitzats en campus o àrees

### **Pre-Condicions**

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

### **Post-Condicions**

N/A

### **Fluxos d'events**

#### ***Flux Principal – Sel.leccionar Espai***

1. Al menú d'espais l'usuari sel.lecciona un espai determinat
2. El sistema mostra una pantalla amb la següent informació:
  - Dades principals de l'espai
    - Codi
    - Tipus Espai (Edifici, Modul, Planta, etc..)
    - Nom
    - Descripció
    - Mesures
    - Relació : Propietat / Lloguer / Cessió / Col·laboració
    - Adreça
  - Llistat de subespais, que formen part d'ell, amb agrupació expressa en plantes i una indicació de la seva activitat genèrica. (cas d'us Llistar subespais)
3. El sistema permetrà:
  - a. Modificar l'espai (cas d'us Modificar espai)
  - b. Eliminar l'espai (cas d'us Eliminar espai)
  - c. Crear un espai depenent d'aquest (cas d'us Crear espai). Unicament en els casos d'àrees
  - d. Crear un subespai depenent d'aquest (cas d'us Crear subespai)

## **Crear espai**

### **Descripció**

Aquest cas d'ús encapsula la funcionalitat encarregada de crear un espai determinat. Els espais es correspondrien als edificis principals, que poden estar organitzats en campus o àrees i aquests campus o àrees determinats

### **Pre-Condicions**

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

### **Post-Condicions**

N/A

### **Fluxos d'events**

#### ***Flux Principal – Crear Espai***

1. Del menú principal l'usuari sel.lecciona l'opció Crear espai
2. L'usuari introdueix les dades principals de l'espai
  - Nom
  - Codi
  - Tipus Espai (Edifici, Modul, Planta, etc..)
  - Descripció
  - Mesures
  - Relació : Propietat / Lloguer / Cessió / Col·laboració
  - Adreça

#### ***Flux Alternatiu – Crear Espai dins d'un altre espai***

1. Del menú principal l'usuari sel.lecciona un espai determinat que sigui un àrea (Cas d'ús seleccionar espai)
2. Del menú principal l'usuari sel.lecciona l'opció Crear espai
3. L'usuari introdueix les dades principals de l'espai
  - Nom
  - Codi
  - Tipus Espai (Edifici, Modul, Planta, etc..)
  - Descripció
  - Mesures
  - Relació : Propietat / Lloguer / Cessió / Col·laboració
  - Adreça
4. El sistema guarda el nou espai a l'entitat d'ESPAIS

## **Modificar espai**

### **Descripció**

Aquest cas d'ús encapsula la funcionalitat encarregada de modificar un espai determinat. Els espais es correspondrien als edificis principals, que poden estar organitzats en campus o àrees i aquests campus o àrees determinats.

### **Pre-Condicions**

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

#### **Post-Condicions**

N/A

#### **Fluxos d'events**

##### ***Flux Principal – Modificar Espai***

1. L'usuari selecciona l'espai que vol modificar (Cas d'us seleccionar espai)
2. Del menú l'usuari selecciona l'opció Modificar espai
3. L'usuari modifica les dades principals de l'espai
  - Nom
  - Codi
  - Tipus Espai (Edifici, Modul, Planta, etc..)
  - Descripció
  - Mesures
  - Relació : Propietat / Lloguer / Cessió / Col·laboració
  - Adreça

## **Eliminar espai**

#### **Descripció**

Aquest cas d'ús encapsula la funcionalitat encarregada d'eliminar un espai determinat. Els espais es correspondrien als edificis principals, que poden estar organitzats en campus o àrees i aquests campus o àrees determinats. Eliminat un espai s'eliminarien tots els espais i subespai que formessin part d'aquest

#### **Pre-Condicions**

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

#### **Post-Condicions**

N/A

#### **Fluxos d'events**

##### ***Flux Principal – Eliminar Espai***

1. L'usuari selecciona l'espai que vol eliminar (Cas d'us seleccionar espai)

2. Del menú, l'usuari sel.lecciona l'opció Eliminar espai
3. El sistema demana a l'usuari si desitja confirmar l'eliminació
4. El sistema elimina l'espai de l'entitat ESPAIS i tots els subespais depenents

## Crear subespai

### Descripció

Aquest cas d'ús encapsula la funcionalitat encarregada de crear un subespai determinat. Els subespais es correspondrien a espais dintre d'edificis, com poden ser plantes o locals.

### Pre-Condicions

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

### Post-Condicions

N/A

### Fluxos d'events

#### ***Flux Principal – Crear Subespai (Ex: crear una planta)***

1. Del menú principal l'usuari sel.lecciona un espai determinat (Cas d'us seleccionar espai)
2. Del menú principal l'usuari sel.lecciona l'opció Crear subespai
3. L'usuari introdueix les dades principals del subespai
  - Identificador
  - Tipus local
  - Tipus de funcionalitat
  - Capacitat : nombre de persones
  - Descripció
4. El sistema guarda el nou subespai a l'entitat d'ESPAIS

#### ***Flux Alternatiu – Crear subespai dins d'un altre subespai (Ex: crear un local dins d'una planta)***

1. Del menú principal l'usuari sel.lecciona un espai determinat que sigui un edifici (Cas d'us seleccionar espai)

2. Del menú l'usuari sel.lecciona un subespai determinat que sigui una planta (Cas d'us seleccionar subespai)
3. Del menú principal l'usuari sel.lecciona l'opció Crear subespai
4. L'usuari introdueix les dades principals del subespai
  - o Identificador
  - o Tipus local
  - o Tipus de funcionalitat
  - o Capacitat : nombre de persones
  - o Descripció
5. El sistema guarda el nou subespai a l'entitat d'ESPAIS

## **Sel.leccionar subespai**

### **Descripció**

Aquest cas d'ús encapsula la funcionalitat encarregada de sel.leccionar un subespai determinat. Els subespais es correspondrien a espais dintre d'edificis, com poden ser plantes o locals.

### **Pre-Condicions**

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

### **Post-Condicions**

N/A

### **Fluxos d'events**

#### ***Flux Principal – Sel.leccionar subespai***

1. Al menú d'espais l'usuari sel.lecciona un espai determinat (Cas d'us seleccionar espai)
2. Del llistat de subespais, que formen part d'ell, l'usuari sel.lecciona un subespai determinat
3. El sistema permetrà:
  - a. Modificar el subespai (cas d'us Modificar subespai)
  - b. Eliminar l'espai (cas d'us Eliminar subespai)

- c. Crear un subespai dependent d'aquest (cas d'us Crear subespai).  
Unicament en els casos de plantes
- d. Assignar una activitat a un subespai (cas d'us Assignar activitat).

## **Modificar subespai**

### **Descripció**

Aquest cas d'ús encapsula la funcionalitat encarregada de modificar un subespai determinat. Els subespais es correspondrien a espais dintre d'edificis, com poden ser plantes o locals.

### **Pre-Condicions**

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

### **Post-Condicions**

N/A

### **Fluxos d'events**

#### ***Flux Principal – Modificar subespai***

1. L'usuari selecciona l'espai on està el subespai que vol modificar (Cas d'us seleccionar espai)
2. L'usuari selecciona el subespai que vol modificar (Cas d'us seleccionar subespai)
3. Del menú l'usuari selecciona l'opció Modificar subespai
4. L'usuari modifica les dades principals del subespai
  - Identificador
  - Tipus local
  - Tipus de funcionalitat
  - Capacitat : nombre de persones
5. L'usuari accepta la modificació

## **Eliminar subespai**

### **Descripció**

Aquest cas d'ús encapsula la funcionalitat encarregada d'eliminar un subespai determinat. Els subespais es correspondrien a espais dintre d'edificis, com poden ser plantes o locals. Si es eliminada una planta s'eliminen tots els locals dependents

### **Pre-Condicions**

L'usuari haurà de tenir permisos per a accedir a aquesta funcionalitat i ha d'haver-se validat a l'entrada de l'aplicació.

### **Post-Condicions**

N/A

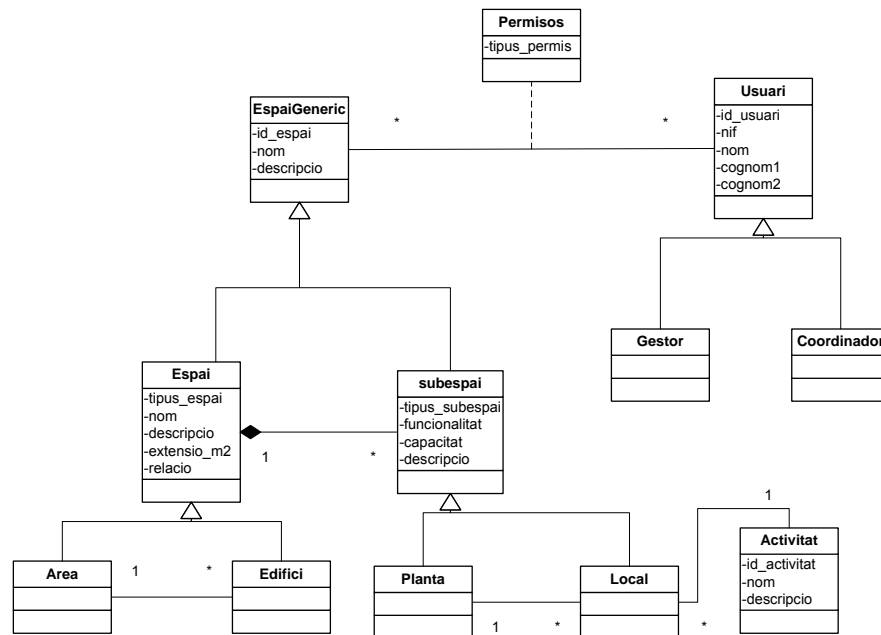
### **Fluxos d'events**

#### ***Flux Principal – Eliminar subespai***

1. L'usuari selecciona l'espai que vol eliminar (Cas d'us seleccionar espai)
2. L'usuari selecciona el subespai que vol eliminar (Cas d'us seleccionar subespai)
3. Del menú, l'usuari selecciona l'opció Eliminar
4. El sistema demana a l'usuari si desitja confirmar l'eliminació
5. El sistema elimina l'espai de l'entitat ESPAIS i tots els subespais dependents



## Diagrama de Classes del Sistema



**Il·lustració 2 : Diagrama de classes del sistema**

A l'hora de buscar les classes del sistema ens trobem amb les següents característiques:

- de la classe usuari genèric podem especificar dos tipus d'usuaris, els gestors i els coordinadors, que encara que ara no tinguin atributs específics, més endavant podem pensar altres característiques

- he creat una classe associativa entre usuari i espai genèric per indicar els tipus de permisos (lectura, modificació) que un usuari que un usuari pot tenir
- he creat una superclasse Espai genèric, de la que heretan Espai i subEspai com a generalització dels atributs comuns
- Per relexar que un subespai (locals o plantes) tan sols poden estar dins d'un espai determinat he creat una relació de composició entre espai i subespai
- He creat quatre classes: Area, Edifici, Planta i Local, las dues primeres especialitzacions d'Espai i les dues segones de Subespai, amb relacions d'associació per indicar que un edifici pot formar part d'una area i un Local d'una planta
- Finalment, com tan sols els locals poden tenir activitats he creat una relació entre subespai i activitat

## ***Disseny***

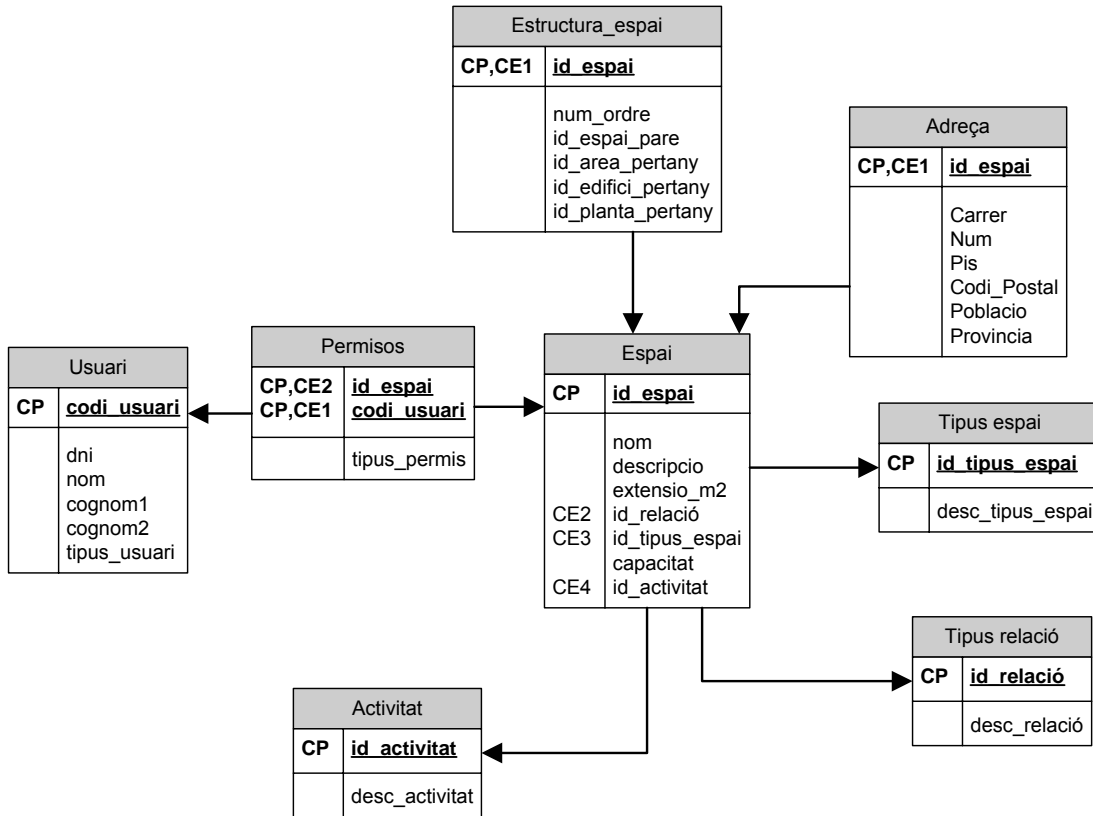
### **Programari**

Per fer aquest treball, en un principi se m'ha proposat fer ús del programari lliure, Jboss, MySQL i Eclipse, però degut al fet de que podia accedir amb facilitat, gràcies a la meva feina personal, a tota la tecnologia d'ORACLE per desenvolupament d'aplicacions amb l'arquitectura J2EE i sent aquesta una de les plataformes més importants dins de l'entorn empresarial, he decidit utilitzar aquesta. Així com a IDE, faré servir el Jdeveloper 9.0.3.1 i com a servidor d'aplicacions el mateix entorn J2EE incrustat del Jdeveloper, el OC4J.

OC4J (Oracle9iAS Containers for J2EE) es un servidor d'aplicacions amb contenidors que ofereix un entorn J2EE i que està integrat al JDeveloper i en tots els Oracle9i Application Servers (Oracle9iAS). L'OC4J inclueix un contenidor d'Enterprise JavaBeans (EJB's), un contenidor de Servlet's i de JSP's.

Finalment com a base de dades penso utilitzar Access de Microsoft, ja que penso que és un entorn de base de dades molt distribuït dins l'àmbit de l'administració, lloc bàsic on es podrà utilitzar l'aplicació.

## Disseny de la base de dades



**Il·lustració 3 : Disseny de la base de dades**

A l'hora d'obtenir la definició de l'estructura de la base de dades del diagrama UML de l'anàlisi de classes he tingut en compte el següent:

- he suprimit l'herència de Gestor i Cordinador amb Usuari, definint una sola taula per a tota la jerarquia de classes, la taula Usuari, que tindria tots els atributs propis de cada subclasse més els de la superclasse, ja que l'espai ocupat pels valors dels atributs de les subclasses es quasi nul.
- respecte a les classes heretades d'EspaiGenèric i a la seva vegada d'Espai i de Subespai he decidit, després de moltes discussions interiors reduir tota la jerarquia de l'herència a una única classe Espai, que englobaria tots els camps de les superclasses i subclasses i a més afegir una taula Estructura\_Espai que ens donaria la relació d'un espai dintre del complex donant l'àrea, edifici i planta a la que pertany. He escollit aquesta sol.lució en front d'altres en vistes a la eficiència temporal i a que la majoria dels espais seran Locals.

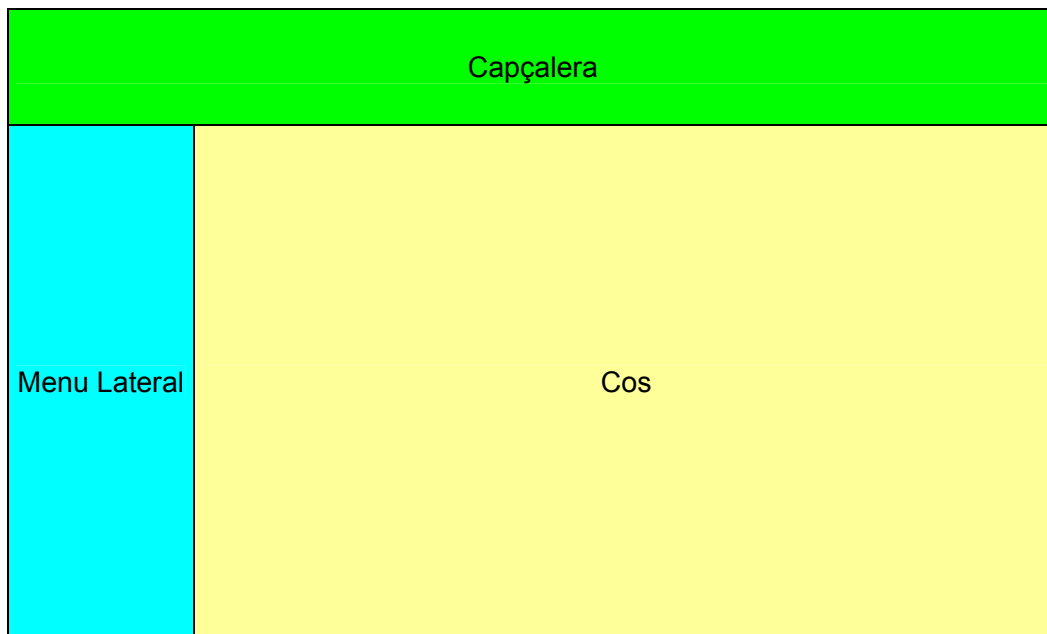
## Disseny de les interfícies gràfiques

Les possibles pantalles serien:

- Pantalla de connexió on es demanaria el codi i el password per determinar el rol
- Afegir espai: pantalla on es demanaria la següent informació del espai:
  - Codi
  - Tipus Espai (Edifici, Modul, Planta, etc..)
  - Nom
  - Descripció
  - Mesures
  - Relació : Propietat / Lloguer / Cessió / Col·laboració
  - Adreça
- Seleccionar espai, per poder després modificar-ho, esborrar-ho o afegir un subespai
- Afegir subespai, on es demanaria la següent informació:
  - Identificador
  - Tipus local
  - Tipus de funcionalitat
    - Docència : Funcions docents
    - Sanitària: Funcions sanitàries
    - Administració : Funcions administratives
    - Serveis :
    - Trànsit : vestíbuls, passadissos, etc.
    - Altres :
  - Capacitat : nombre de persones
  - Descripció
- Seleccionar subespai per poder després modificar-ho, esborrar-ho o assignar-li una activitat
- Consultar espai

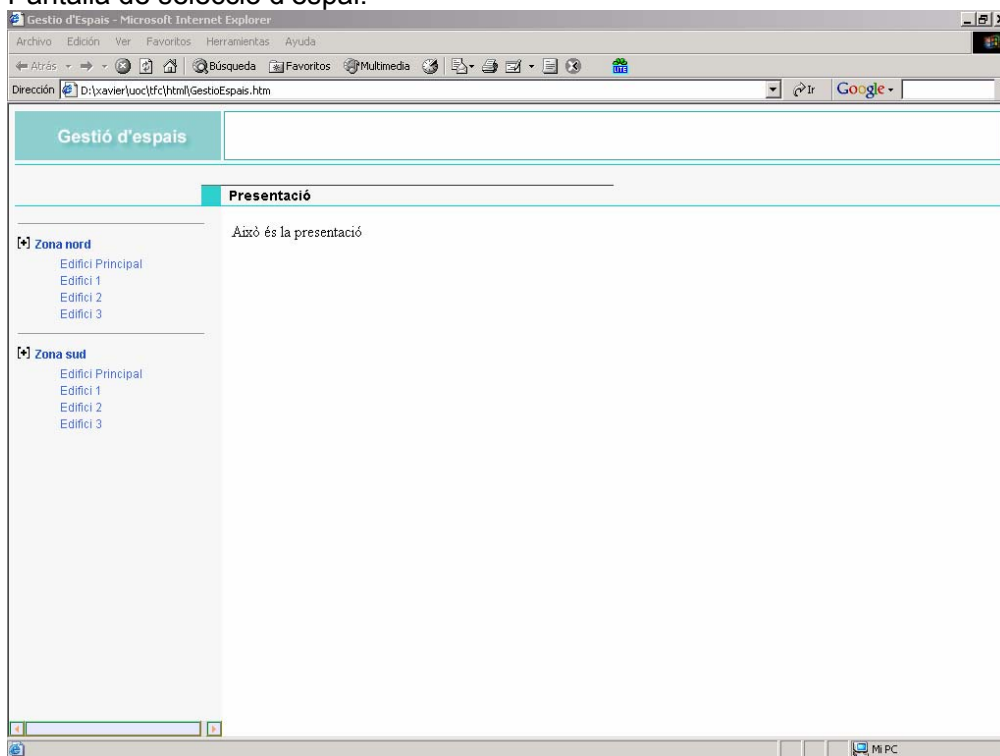
L'estructura de la interfície gràfica serà el d'una pàgina amb marcs amb tres zones diferenciades:

- capçalera, on apareixerà el logotip, i dues línies de navegació dins del nostre mapa del web
- menú lateral, on apareixerà el menú de les diverses, àrees, edificis, plantes i locals que l'usuari podrà escollir
- cos, on apareixerà la informació relativa a allò escollit per l'usuari: descripció de l'espai, formularis d'introducció, etc...



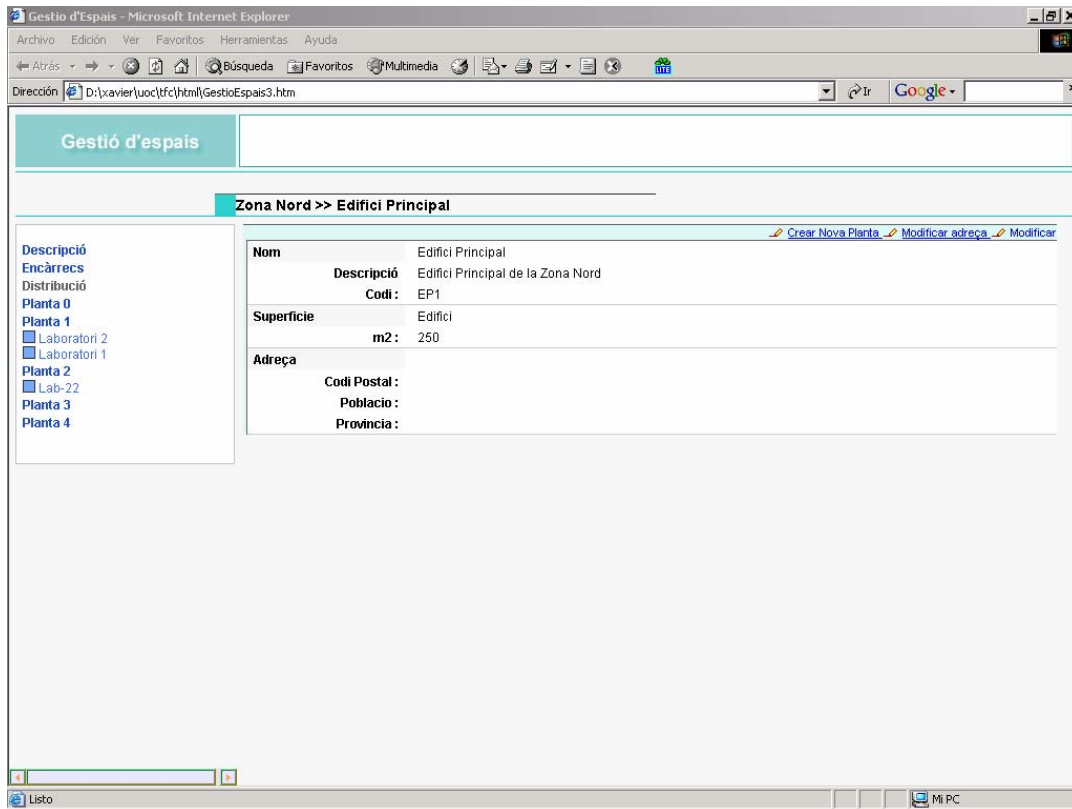
Un exemple gràfic d'alguna d'aquestes pantalles seria

Pantalla de selecció d'espai:



**Il·lustració 4 : Pantalla de selecció de l'espai**

Pantalla de visualització d'un edifici:



Il·lustració 5 : Pantalla de visualització d'un edifici

Pantalla de modificació d'un local:

The screenshot shows a web browser window titled "Gestió d'Espais - Microsoft Internet Explorer". The address bar shows the URL "D:\xavier\uoc\lfc\html\GestioEspais3.htm". The page content includes a navigation menu on the left with options like "Descripció", "Encàrrecs", "Distribució", and "Planta 1" through "Planta 4". The main area displays a form for editing "Laboratori 2" under the path "Zona Nord >> Edifici Principal >> Planta 1 >> Laboratori 2". The form fields are: "Nom" (Laboratori 2), "Descripció" (Laboratori d'ús general), "Codi" (Lab 01), "Superfície" (Docència i investigació), "m2" (30), "Activitat" (Laboratori d'investigació/docència), and "Capacitat" (entre 0 i 15 persones). There are "Confirmar" and "Cancel·lar" buttons at the top right of the form.

<b>Nom</b>	Laboratori 2
<b>Descripció</b>	Laboratori d'ús general
<b>Codi</b>	Lab 01
<b>Superfície</b>	Docència i investigació
<b>m2</b>	30
<b>Activitat</b>	Laboratori d'investigació/docència
<b>Capacitat</b>	entre 0 i 15 persones

Il·lustració 6 : Pantalla de modificació d'un local



## Disseny arquitectònic

### Us de J2EE

J2EE és un conjunt d'especificacions d'APIs Java per a la construcció d'aplicacions empresarials que estandaritzen les principals característiques d'aquestes aplicacions empresarials, com son:

- Accés a bases de dades normalment relacionals
- Accions transaccionals
- Escalables: han de poder soportar més càrrega de treball sense necessitat de modificar el software
- Disponibilitat: Idealment no han de deixar de prestar servei
- Segures: No tots els usuaris poden accedir a la mateixa funcionalitat
- Integració: Es necessari integrar aplicacions construïdes amb distintes tecnologies
- Separació clara entre la interfaz gràfica i el model que encapsula la lògica de negoci

J2EE proporciona les següents tecnologies:

- APIs que permeten implementar la interfaz gràfica d'una aplicació web: Servlets, pàgines JSP
- Componentes EJB (J2EE):
  - Entity Beans, que permeten implementar els objectes persistents del model que no depengui d'un tipus particular de BD
  - Session Beans, que permeten implementar fachadas del modelo

### Model MVC (Model Vista Controlador)

Entre les característiques que tindrà la meua aplicació, intentaré fer us del Model MVC i de l'entorn d'Struts

El model MVC separa 3 capes: Model, Vista i Controlador, independents una de l'altre amb l'intenció de que les aplicacions es puguin mantenir fàcilment i amb parts reusables

Struts és un entorn normalitzat de presentació, que normalitza el desenvolupament de la capa Vista i que per aquesta finalitat proporciona mecanismes per a treballar amb la capa Controlador, però que mai es mesclarà amb la capa Model, separant, d'aquesta manera la lògica de presentació de la lògica de negoci

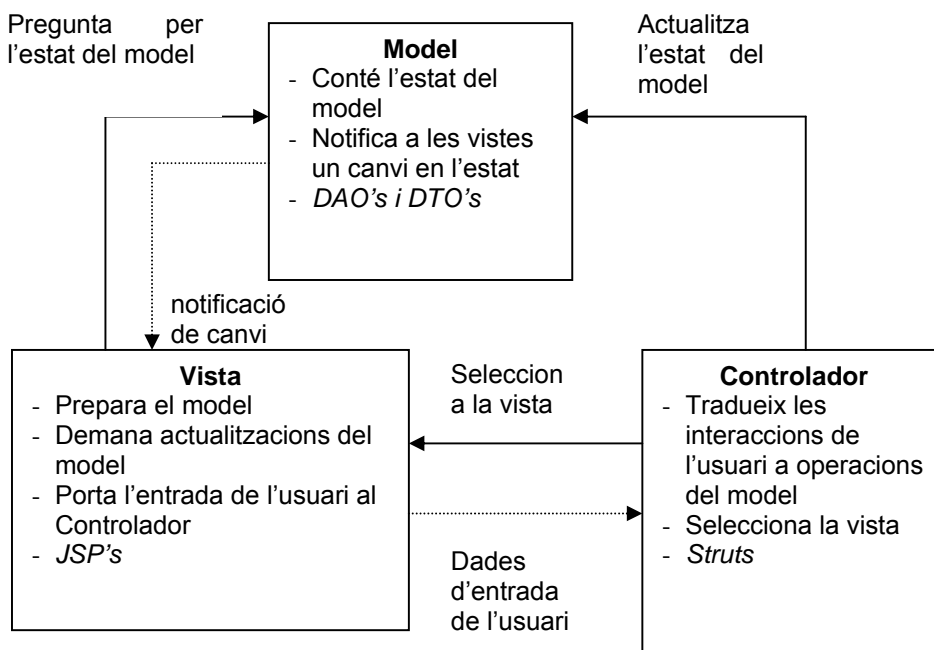
Utilitzant aquest model i aquest entorn dins la tecnologia J2EE desenvoluparem per diferents costats:

- la interfaz del client de Web (la Vista): que estarà formada bàsicament per JSP's (Java Server Pages) i ActionForms

- la part de la lògica de negoci (el Model). Aquests es pot implementar utilitzant EJB's però també es pot utilitzar, com al final així he escullit, una sol.lució més sencilla basada en dos patrons de disseny:
  - o DTO's (Data Transfer Objects)
  - o DAO's (Data Acces Object) per a la connexió amb la BD mitjançant ODBC
- la part de control de l'aplicació, la part que rep les peticions i les deriva cap a la capa de model o de vista (el Controlador), que jo intentaré implementar utilitzant Struts, principalment Actions

A més aquest model es desenvolupa dins d'un contenidor per a guardar tota la informació necessària de sessió

Un esquema d'aquest model seria el següent:

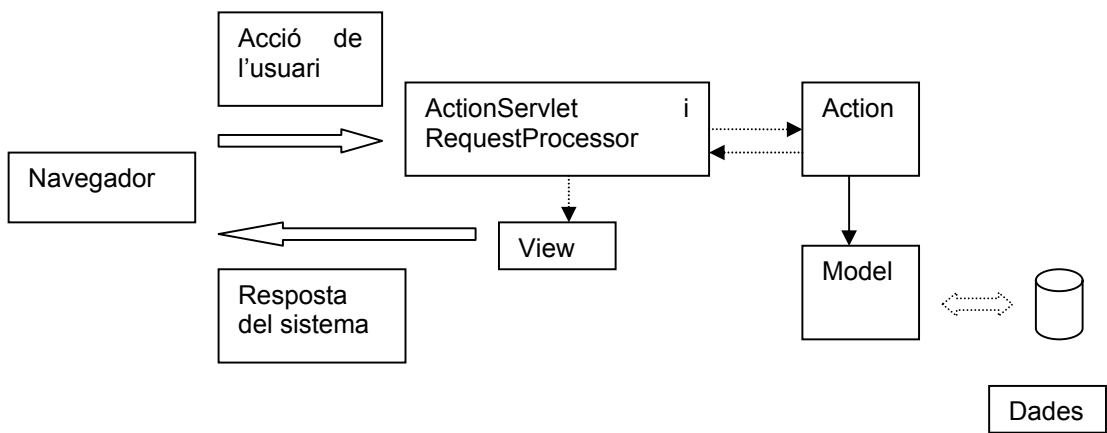


**Il·lustració 7 : Esquema del model MVC**

## Us d'Struts

Per a la implementació de la capa Controller s'utilitza el FrameWork Struts del projecte JAKARTA, que segueix el patró MVC, es a dir que una petició http realitzada per un usuari es rebuda per un controlador que delega la seva execució en una Action que invoca la lògica de negoci necessària i emmagatzema els resultats en algún scope de servlets, seguidament el controlador escull la vista a mostrar a la que passa el control i mostra les dades.

Un esquema dels struts seria:



**Il·lustració 8 : Esquema d'Struts**

## ***Implementació***

Tal com he explicat anteriorment he fet servir com a IDE el Jdeveloper 9.0.3.1 i com a servidor d'aplicacions el mateix entorn J2EE incrustat del Jdeveloper, el OC4J.

Després com a base de dades he utilitzat l'Access de Microsoft.

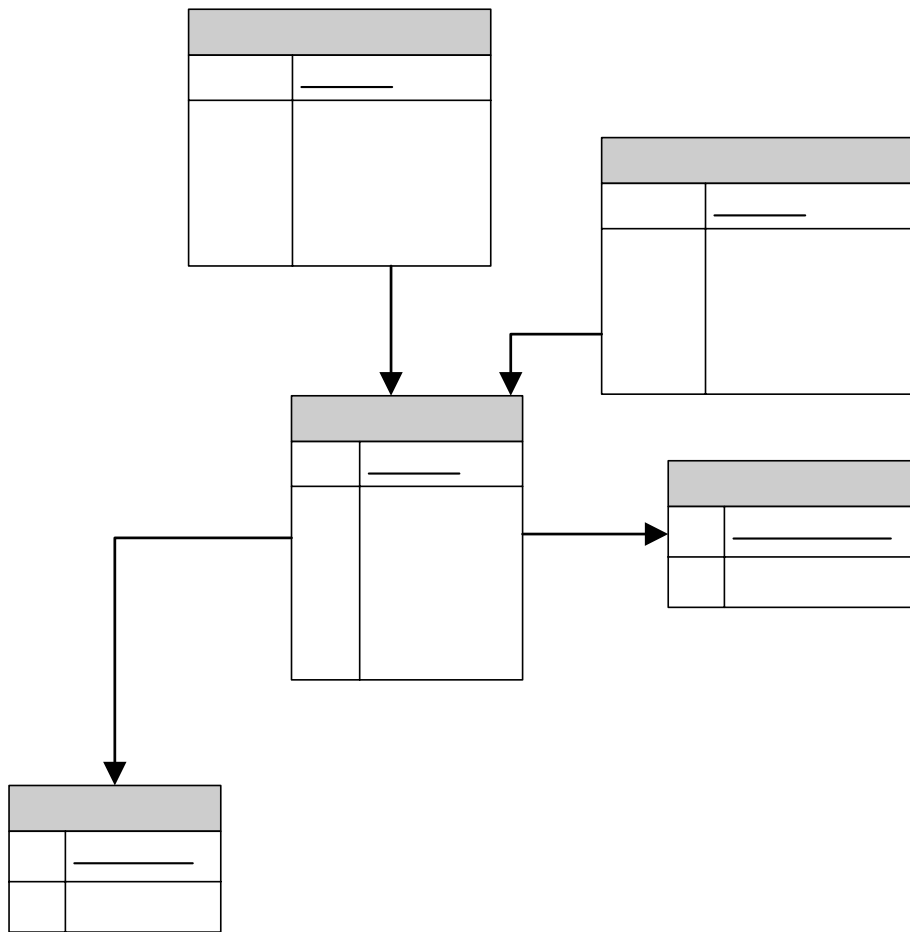
Així el resultat de la meva implementació és

- el fitxer GestioEspais.mdb, la base de dades creada en MS Access amb les taules exposades a la fase de diseny
- El fitxer .jar amb tots els fitxers java necessaris per ser implementats al servidor d'aplicacions.

En relació a l'anàlisi i al disseny de l'aplicació hi ha hagut una serie de modificacions que explico a continuació

## **Implementació de la base de dades**

En relació al diseny de la base de dades finalment i per motius en algún cas de falta de temps (d'aquesta manera no he implementat tot allò relacionat amb els usuaris i els seus permisos) i en altres casos al veure que l'aplicació feia innecessaris algun dels camps previament analitzats, el diseny ha quedat de la següent manera:



Estruc

CP,CE1 ic

Heu de tenir en compte que per que es pugui accedir a la base de dades s'ha de crear una connexió ODBC amb nom GestioEspais que apunti al fitxer d'Access.

A aquesta base de dades on he afegit unes vistes:

- la consulta C\_menu\_principal que em permet crear el menu lateral de l'aplicació.
- la consulta C\_espais, que visualitza totes les dades d'un espai
- la consulta C\_menu\_edifici, que visualitza els espais que pertanyen a un edifici determinat

n  
ic  
ic  
ic  
ic

## Implementació de les classes del sistema

Seguint el model MVC he creat les següents classes :

### Model : package ge.model

Implementa les classes de la lògica de negoci. S'utilitzen bàsicament dos patrons de disseny:

- DTO's (Data Transfer Objects) :
  - ActivitatDTO.java, que representa un objecte que correspon als registres de la taula Activitat
  - EspaisDTO.java, que representa un objecte que correspon als registres de les taules Espai i EstructuraEspai enllaçats en una view de la BBDD anomenada C\_espais
  - GeAdrecaDTO.java, que representa un objecte que correspon als registres de la taula Adreca
  - GeNavegacioDTO.java, que representa un objecte que ens servirà per les línies d'ubicació dins la nostra web
  - MenuDTO.java, que representa un objecte que correspon a la informació del menú lateral de l'aplicació
- DAO's (Data Access Object) per a la connexió amb la BD mitjançant ODBC
  - EspaisDAO.java, que s'encarrega de l'accés a les dades de les taules d'Espais
  - MenuDAO.java, que s'encarrega de l'accés a les dades del menú lateral

A més apareixen altres classes com:

- Espais.java: Clase principal que representa l'objecte Espai, dintre del qual té el DAO que li permet la connexió a la BBDD, el DTO que representa les taules i la distribució on està tota la seva distribució espacial completa
- PrepararPrincipal.java: Servlet cridat pel jsp espais.jsp i que prepara el marc principal de la pàgina espais.jsp
- Utilitats.java: Clase d'utilitats on es guarda aquelles utilitats que s'utilitzen a l'aplicació d'Espais

### View : jsp's

Implementa la interfàcia del client de Web. S'utilitzen els fitxer següents:

- ClientMenu.jsp: Jsp que presenta el menú lateral de la aplicació
- Descripcio.jsp: Jsp que presenta el menú lateral de la aplicació
- Errors.jsp: Jsp que presenta l'error produït
- Espais.jsp: Jsp que prepara els marcs de l'aplicació
- Ge\_crear\_espai.jsp: Jsp que presenta el formulari per crear un espai
- Ge\_descripcio\_ed.jsp: Jsp que presenta el formulari per modificar les dades d'un espai
- Ge\_frame\_cap.jsp: Jsp que presenta la capçalera de l'aplicació

- Ge\_modificar\_adreca.jsp: Jsp que presenta el formulari per modificar l'adreça
- Inici.jsp: Jsp que inicia l'aplicació de Gestió d'Espais
- Presentacio.jsp: Jsp que presenta el marc principal de Presentació

En aquesta implemtació també he utilitzat tècniques pròpies del disseny de pàgines HTML, com poden ser l'ús de Javascript, css's i frames.

## Controller: package ge.controller

Utilitzant la tecnologia d'Struts, principalment Actions, els fitxers del controlador son:

- CrearEspaiAction.java
- MenuAction.java
- MenuEdDescripcionAction.java
- MenuGeneralAction.java
- ModificarAdrecaAction.java

De tota la tecnologia d'Struts, per manca de temps he utilitzat únicament una de les seves característiques, com son els Action, clase que respresenta accions úniques e independitza les peticions del client del model de negoci. El seu mètode més important és:

```
public ActionForward execute(ActionMapping mapping,
    HttpServletRequest request, HttpServletResponse response) throws
    Exception;
```

Com podem veure aquest mètode retorna un objecte ActionForward. Aquest objecte és exclusiu d'Struts i permet definir en un fitxer extern (struts-config.xml) mitjançant mapejats les característiques de les accions: nom, ruta, què fer si el mètode té un resultat existòs y què fer en cas contrari, és a dir cap a on adreçar la petició. D'aquesta forma, en el nostre codi podem indicar mitjançant return mapping.findForward("Success") que continui el fluxe de l'aplicació per una altre acció o JSP o reenviar-li a un altre.

Així per exemple quan el fitxer CrearEspai.jsp té l'ordre d'activació del seu formulari d'HTML

```
<form action="CrearEspai.do" method="post" name="form_descripcion" target="_top">
```

aquest formulari fa un submit envia el control al struts-config.xml que té mapejat CrearEspai com:

```
<action path="/CrearEspai" type="ge.controller.CrearEspaiAction">
  <forward name="Confirmar" path="/MenuGeneral.do"/>
  <forward name="Cancelar" path="espais.jsp?op1=G"/>
</action>
```

el que vol dir que envia el seu control a CrearEspaiAction.java. Depenent del seu retorn

```
    return mapping.findForward("Confirmar");  
o  
    return mapping.findForward("Cancelar");
```

Struts enviarà al control a un altre mapejat (en el cas de confirmar a MenuGeneral) o retornara a l'usuari un fitxer jsp (espais.jsp)



## ***Conclusions.***

Aquest projecte és el resultat de quatre mesos d'intens desenvolupament des del moment en el que es van conèixer els requisits fins el moment de la implantació i proves, aturat aquests desenvolupament temporalment per l'arribada d'un nou vingut a aquest mon, el que va fer encara més complicat la meva dedicació a ells. Com es veurà al projecte està molt més desenvolupada la part View del model MVC, ja que com va ser la primera que vaig començar és a on vaig poder dedicar més hores abans de l'atur parcial que vaig patir benhauradament el 17 d'abril d'aquest any.

Durant aquest temp s'ha intentat aplicar els diferents coneixements de l'enginyeria adquirits durant els tres cursos de la carrera (però molts més anys) principalment els provinents de les assignatures de programació, juntament amb altres coneixements adquirits de manera personal.

Com opinió personal dir que s'han trobat grans avantatges en l'us de la tecnologia JSP i la tecnologia d'.struts. Això permet per una part de la creació de la part View amb pàgines web dinàmiques el diseny de les quals podrà ser actualitzat i millorat fàcilment amb qualsevol eina de diseny web i per una altra part la creació de la capa de Control, que he intentat implementar utilitzant Struts, principalment Actions

## Glossari.

<b>Gestor</b>	Personal del Departament de'Obres o de Patrimoni que alimentarà el sistema de dades amb els espais principals i realitzarà l'explotació de les dades.
<b>Coordinador</b>	Personal amb capacitat de crear subespais dins del sistema de dades y de assignar activitats a aquests
<b>Usuaris</b>	Personal de la resta de l'organització i de fora de l'organització. Consultarà les dades sobre els espais a través del portal corporatiu de la Intranet o inclusiu des d'Internet
<b>Espai</b>	Qualsevol edifici o agrupació d'edificis (àrea)
<b>Subespai</b>	Qualsevol espai contingut dins d'un altre
<b>Area</b>	Agrupació d'edificis (EX: campus nord, àrea clínic, etc...)
<b>JSP (Java Server Pages)</b>	És una extensió dels Servlets que permet de forma fàcil la mescla de codi amb pàgines HTML estàtiques.
<b>J2EE</b>	Es una arquitectura complerta d'aplicacions de n-capes basat en la plataforma Java, a la vegada que un model de programació i una plataforma
<b>Servlets</b>	Un servlet és una classe Java que pot carregar-se dinàmicament per estendre la funcionalitat del servidor web.
<b>Struts</b>	Variació del paradigma de disseny MVC (Model View Controler) on s'intenta separar al màxim allò que fa referència a la visualització del que fa referència a la lògica del sistema.

## Bibliografía.

García de Jalón, J.; Rodríguez, José Ignacio; Imaz, Aitor *Aprenda Servlets de Java como si estuviera en segundo*. Universidad de Granada 1999.

Marinescu, F. *EJB Design Patterns* John Wiley & Sons Inc: 2002

Bergman, Noel J. *Introduction to JSP technology*.

Murray, Jeanne. *Building Java HTTP servlets*

Dick, Andrew J.. *Introduction to Struts*. Colorado Software Summit

Geary, D. M. *JavaServerPages avanzado*. Prentice Hall: Madrid 2003