



Escola Tècnica
Superior d'Enginyeria
Informàtica

Projecte Final de Carrera

Desenvolupament d'un web per controlar la llar digital

AUTOR

Francisco Faulí Tarazona

(frfaut1@ei.upv.es)

DIRECTORS

Joan Fons i Cors

(jjfons@dsic.upv.es)

Ignacio Mansanet Benavent

(imansanet@dsic.upv.es)

Curs 2010-2011

València

5

Agraïments

Voldria donar les gràcies a tota la meua família per haver-me donat el seu suport incondicional durant tot aquest temps i en especial a mare per haver tingut tanta paciència amb mi.

També voldria agrair a tots els meus companys de classe que han estat fent-me companyia durant totes eixes llargues sessions de biblioteca. La veritat és que sense ells açò hagués segut molt dur.

Tampoc voldria oblidar-me dels meus amics i amigues que han estat al meu costat sempre que els he necessitat i han sabut animar-me quan les coses no anaven molt be.

Finalment, vull agrair a Nacho Mansanet tota la atenció que ha dipositat sobre mi i el meu projecte. Perquè m'ha donat molta llibertat a l'hora de la seua creació i perquè sempre m'ha ajudat en qualsevol problema que he tingut, sobretot amb la creació d'aquesta memòria.

GRÀCIES!

Índex

Agraïments	3
La llar digital	9
1. Introducció	10
1.1 Història de la domòtica i la llar digital	11
1.2 Objectiu del Projecte	12
2. Tecnologia JSF	13
2.1 Per què utilitzar JSF	14
2.2 Com funciona JSF	15
2.3 Llibreries	15
PrimeFaces (Versió 2.2 RC2)	15
Jdom (Versió 1.1)	16
Antecedents	18
3. Altres programes	19
3.1 Domotiga	19
3.2 OpenDomo	20
Domestica	22
4. Primera Versió	23
5. Segona Versió	27
6. Interfície d'usuari	31
7. Evolució de les versions	38
8. Estructura dels fitxers	39
9. Llicència	41
Conclusions	43
10. Treballs Futurs	44
Apèndixs	48
A. Estructura XML	50
B. Primers passos amb JSF 2.0	53
B.1 Prerequisits	53
B.2 Crear un nou projecte	54
B.3 Importar les llibreries	57

C. Exemple JSF: Hola món	58
D. Importar Primefaces	63
E. Importar <i>Domestica</i>.....	65
G. Modificar Fitxer XML.....	68
H. Connexió amb el Servlet	70
I. Glossari De Termes	72
J. Bibliografia	75

Índex de Figures

Figura 2. 1: Esquema Funcionament JSF	15
Figura 3. 1: Captura de Pantalla Domotiga	20
Figura 3. 2: Captura de Pantalla Domotiga	21
Figura 4. 1: Domotica 1.0	23
Figura 4. 2: Estructura mapa de les localitzacions	24
Figura 4. 3: Domestica 1.1	25
Figura 4. 4: Domestica 1.2	26
Figura 5. 1: Exemple component p:layout	28
Figura 5. 2: Informació sobre un dispositiu amb Domestica 1	29
Figura 5. 3: Informació sobre un dispositiu amb Domestica 2	29
Figura 5. 4: Domestica 2.0	30
Figura 6. 1: Pàgina d'inici	31
Figura 6. 2: Finestra Canvi IP	31
Figura 6. 3: Menu	32
Figura 6. 4: Mapa de FirstFloor	33
Figura 6. 5: Mapa de Kitchen	34
Figura 6. 6: Informació sobre KitchenLighting1	35
Figura 6. 7: Informació sobre GradualLighting1	35
Figura 6. 8: Botons Device_type_switch	36
Figura 6. 9: Botons Device_type_percentual_dimmer	36
Figura 6. 10: Botons Device_type_pulse	37
Figura 6. 11: Botons Device_type_percentual_dimmer	37
Figura 6. 12: Botons Device_type_vertical_movement	37
Figura 6. 13: Botons Device_type_angular_dimmer	38
Figura 6. 14: Botons Device_type_hex_dimmer	38
Figura 8. 2: Directori /WebContent	40
Figura 10. 1: Exemple p:columnChart	45
Figura 10. 2: Exemple p:lineChart	46
Figura B. 1: Instal·lació Tomcat	54
Figura B. 2: Creació Web Dinàmica	55
Figura B. 3: Crear Web Dinàmica 2	56
Figura B. 4: Estructura fitxer per defecte	57
Figura B. 5: Importar Lliberies	57
Figura C. 1: Provar Exemple	62
Figura C. 2: Resultat	62
Figura D. 1: Resultat Primefaces	64

Figura E. 1: Importar Projecte	65
Figura E. 2: Importar Projecte 2	66
Figura E. 3: Executar Domestica.....	67
Figura G. 1: Modificar XML	68
Figura G. 2: Importar XML.....	69

Part I

La llar digital

1. Introducció

La Llar Digital o l'Habitatge Intel·ligent és una casa on les necessitats dels habitants referent a la seguretat, confort, gestió i control, oci i entreteniment, telecomunicacions, estalvi d'energia, temps i recursos són ateses mitjançant la integració de sistemes, productes, noves tecnologies i serveis d'àrees com la Domòtica, Seguretat, Multimèdia, Informàtica i Telecomunicació.

La llar avui en dia disposa d'un gran nombre d'equips i sistemes autònoms i xarxes digitals no connectats entre ells com la telefonia, els sistemes d'accés, la televisió, les xarxes de dades (cablejats i sense fil), electrodomèstics, equipament d'àudio i vídeo, calefacció, aire-condicionat, seguretat, enllumenat, etc.

La llar digital és el resultat del procés d'integració d'aquests equips i sistemes autònoms. Això ens porta a la següent definició del concepte de la Llar Digital:

"La Llar Digital és un habitatge que a través d'equips i sistemes, i la integració tecnològica entre ells, ofereix als seus habitants funcions i serveis que faciliten la gestió i el manteniment de la llar, augmenten la seguretat, incrementen el confort, milloren les telecomunicacions; estalvien energia, costos i temps, i ofereixen noves formes d'entreteniment, oci i altres serveis dins de la mateixa i el seu entorn. "

Els productes i sistemes relacionats amb la Llar Digital poden ser agrupats en les següents àrees:

- **La Domòtica** és l'automatització i control local i remota de la llar (apagar / encendre, obrir / tancar i regular) d'aplicacions i dispositius domèstics, amb instal·lacions, sistemes i funcions per a il·luminació, climatització, persianes, portes i finestres, panys, electrodomèstics, control de subministrament d'aigua, gas, i electricitat, etc.

La paraula domòtica és la unió de les paraules Domus (casa en llatí) i Tica (automàtica en grec).

- **La Multimèdia** son els continguts de informació i entreteniment, relacionats amb la captura, tractament i distribució d'imatges i sons dins i fora de l'habitatge, amb instal·lacions, sistemes i funcions com ràdio, televisió, àudio/vídeo "multi-room", cinema a casa, pantalles planes, videojocs, porters i vídeo porters.

- **La Seguretat i Alarmes** són sistemes i funcions per a alarmes d'intrusió, càmeres de vigilància, alarmes personals, alarmes tècniques (incendi, fum, aigua, gas, fallada de subministrament elèctric, fallada de línia telefònica etc.).

- **Les Telecomunicacions** son la distribució de fitxers de text, imatges i sons,

compartint recursos entre dispositius, l'accés a Internet i nous serveis, amb instal·lacions, sistemes i funcions com la xarxa de telefonia, telefonia sobre IP, xarxa local de dades, routers, accés a Internet Banda Ampla, etc.

1.1 Història de la domòtica i la llar digital

La Història de la Llar Digital és molt breu. Realment no es va començar a considerar la integració de sistemes al nivell comercial fins als anys 80. I llavors es tractava principalment d'edificis terciaris i van ser denominats edificis intel·ligents. En el sector domèstic la integració de sistemes a escala comercial s'ha desenvolupat més tard coincidint amb l'evolució i desplegament d'Internet. Va començar als anys 90 al Japó, Estats Units i alguns països en el nord d'Europa. Els diferents sistemes autònoms com la Domòtica, la Seguretat, el Multimèdia i les Comunicacions, però, tenen cadascuna, una història més llarga. Durant molt de temps, la inclusió de tecnologia a la llar, però, s'ha realitzat a través d'un augment de les prestacions o funcions pròpies dels equips domèstics. Aquesta ha estat habitualment conseqüència de la voluntat d'augmentar el valor afegit en si mateix dels equips domèstics, però de forma aïllada, és a dir, sense considerar altres possibilitats de millora relacionades amb el control i la comunicació. Per exemple, en la capacitat de comunicació amb altres dispositius de l'habitatge.

Aquesta situació va suposar el desenvolupament d'un mercat purament vertical, on els equips domèstics que es desenvolupaven eren totalment independents, és a dir, que funcionen de forma autònoma, sense necessitat de comunicar-se amb altres dispositius de la llar. Aquesta forma de concebre els productes ha dificultat la definició i el desenvolupament de serveis susceptibles de ser prestats a la llar i al propi usuari. La introducció de la tecnologia domòtica en el mercat tampoc va trencar amb aquesta realitat. L'automatització d'equips domèstics es realitzava mitjançant un control de la seua alimentació elèctrica, sent una manera molt senzilla de gestió, i de poc atractiu tecnològic. Els equips domèstics no tenien cap tipus de comunicació eficient amb el sistema domòtic. Per això, la Domòtica estava relegada a un mercat molt reduït, comparat amb la totalitat del mercat de productes domèstics, i limitant-se, per tant, a donar resposta a necessitats de control en l'habitatge. Per exemple, les possibilitats de comunicació amb l'exterior es reduïen a senzilles transmissions de senyals o avisos d'alarma o al control remot d'un nombre reduït de sistemes o equips.

Recentment, amb la plena irrupció d'Internet a la llar i, en general, les anomenades TIC (Tecnologies de la Informació i les Comunicacions), s'ha forjat una nova forma d'entendre l'aplicació de tecnologia en l'habitatge, molt més positivista i realista, on l'únic important és el propi usuari i no aquesta. És a dir, de la tecnologia per la tecnologia s'ha passat a assegurar la consecució de les necessitats o desitjos dels usuaris a través de serveis, on evidentment la

tecnologia adquireix un paper de suport molt important a aquests serveis. Amb això, la tecnologia és una cosa transparent per a l'usuari, el qual no té un interès tècnic sinó simplement d'utilitat. L'usuari no està interessat en la tecnologia sinó en resoldre el seu problema, necessitat o desig. Per aquest motiu, cal avançar en el concepte de tecnologia al servei de l'usuari, i que permetrà aportar solucions fàcils, útils i econòmiques, amb les finalitats clares d'assegurar el benestar i la seguretat.

1.2 Objectiu del Projecte

L'objectiu d'aquest projecte serà, doncs, la realització de una aplicació per satisfer les definicions esmentades anteriorment al mateix temps que les necessitats actuals d'una aplicació d'aquest tipus.

Realitzar una interfície senzilla, clara i intuïtiva serà fonamental per poder arribar a tot tipus d'usuari. Plantejar-ho des del punt de vista d'una persona que no està familiaritzada amb els ordinadors ens pot ajudar a solucionar aquest problema. També convindrà la utilització d'elements gràfics (botons, missatges, etc) amb un mida més gran del normal per poder arribar a un client d'una edat més avançada.

Facilitar la gestió i manteniment de la llar també serà un altre aspecte a tindre en compte. Com que es tracta d'una pàgina web, la casa podrà ser controlada des de qualsevol punt del planeta amb connexió a Internet (la mateixa casa, l'oficina, lloc de vacances, etc).

El fet de gestionar la llar des d'una sola web i amb pocs clics de ratolí també suposarà una gran facilitat.

2. Tecnologia JSF

JavaServer Faces (JSF) és un marc de treball per aplicacions web basades en Java que simplifica el desenvolupament d'interfícies d'usuari per a aplicacions Java EE.

JSF està basat en el patró MVC (Model-Vista-Controlador) i per tant el seu disseny està separant en: interfície d'usuari, model de dades i lògica de control. Per la representació de la interfície d'usuari JSF utilitza JavaServer Pages (JSP) com a tecnologia per fer el desplegament de les pàgines, però també pot utilitzar altres tecnologies, com per exemple XUL.

A la interfície d'usuari JSF compta amb dues llibreries d'etiquetes, HTML i CORE, que estan formades per components reutilitzables i extensibles (es poden modificar o es poden crear nous) que faciliten i agilicen el desenvolupament de la interfície d'usuari.

La llibreria HTML representa els components HTML com els camps de text, botons, formularis, etc. Es poden dividir en els següents grups:

- 4 **Entrants** (inputText, inputTextarea)
- 5 **Sortints** (outputText, outputLabel)
- 6 **Comandaments** (commandButton)
- 7 **Selectors** (selectOneRadio, selectOneListbox, Listboxes, menú etc)
- 8 **Disseny** (panelGrid)
- 9 **Taula de dades** (dataTable)
- 10 **Errors i missatges**

Exemple:

```
<h:inputText id="ID1" value="value"/>
```

La llibreria Core ens facilita el registrament esdeveniments, validadors i altres accions dels components de la interfície d'usuari. També compta la etiqueta view necessària per poder crear la vista i així mostrar tots els components en la pàgina. Podríem dir que aquesta llibreria suporta la llibreria HTML.

Algunes etiquetes importants:

- 11 **f: view** Crea la vista general
- 12 **f: subview** Crea una subvista
- 13 **f: validator** afegeix un validador al component.
- 14 **f: converter** Afegeix un convertidor arbitrari a un component
- 15 **f: actionListener** afegeix un actionListener a un component
- 16 **f:valueChangeListener** Afegeix un valuechange listener a un component

Exemple:

```
<f:view>
  <h:outputText value="label" />
</f:view>
```

JSF pot associar a cada component de la interfície d'usuari un objecte Java que li dona suport en el costat del servidor (BackingBean) els quals son manejats pel controlador (Managed Beans) que facilita la recollida, manipulació, i visualització dels diferents valors mostrats en els diferents components de les vistes. El controlador determina quins BackingBeans utilitza la pàgina i on ha de emmagatzemar-los.

La comunicació entre els components de la interfície d'usuari i el BackingBean es realitza mitjançant EL (Expression Language).

Exemple:

```
#{BackingBean.Nom}
```

El fitxer de configuració del ManagedBeans s'anomena *faces-config.xml* i es troba en el directori *WEB-INF* (Servlet) quan es compila l'aplicació.

Exemple:

```
<managed-bean>
  <managed-bean-name>BackingBean</managed-bean-name>
  <managed-bean-class>src.Bean</managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
```

2.1 Per què utilitzar JSF

JSF és un Standard de Sun i per tant es utilitzat i recolzat per un gran nombre d'empreses. A més a més, al estar creat amb Java pot ser utilitzat en diferents dispositius (Web, mòbils, etc) i traduït a un gran nombre d'idiomes(internacionalització - i18n).

La utilització del patró MVC facilita el treball en equip a l'hora de treballar en la mateixa aplicació. En un equip format per dissenyadors i programadors, els primers podrien dedicar-se a treballar en l'aspecte visual i els segons a treballar en el model de dades.

Finalment tenim que ressaltar que JSF també permet la utilització de Javascript i Ajax, augmentant la velocitat, usabilitat i usabilitat de l'aplicació.

2.2 Com funciona JSF

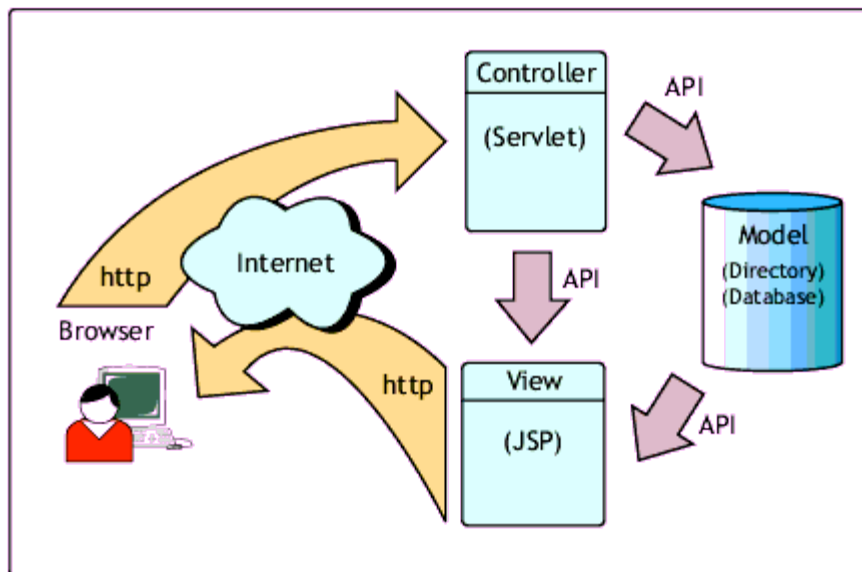


Figura 2. 1: Esquema Funcionament JSF

La petició del client és enviada al servidor. El controlador Servlet examina la petició i extrau el view-ID de la pàgina des d'on s'ha fet la petició. Si no existeix, el controlador crearà un nou view-ID. Seguidament s'actualitzen els components amb els nous valors que fa la petició. També és validen per comprovar que els nous valors son vàlids i a més a més, son diferents dels anteriors valors. Si la validació és correcta, es crearà un esdeveniment ValueChangeEvent que invocarà el mètode valueChangeInput del model (BackingBean). Aleshores els valors del model son actualitzat. Finalment, si la petició tenia un esdeveniment, aquest serà executat i la pagina serà mostrada al client.

2.3 Lliberies

PrimeFaces (Versió 2.2 RC2)

Primefaces és una llibreria de components visuals per a JSF desenvolupada per Prime Technology. Les principals característiques son:

- Format per més de 100 components

- Suport nadiu de Ajax
- Kit per crear aplicacions web per a mòbils.
- Compatible amb altres llibreries.
- Ben documentat

Pàgina web: www.primefaces.org

Demo: <http://www.primefaces.org/showcase/ui/home.jsf>

Jdom (Versió 1.1)

Jdom és una biblioteca de codi obert per a manipulació de Dades XML optimitzats per a Java.

Pàgina web: <http://www.jdom.org/>

Part II

Antecedents

Avui en dia podem trobar una gran varietat de programari dedicat al control de la llar digital. La majoria han estat desenvolupat per companyies especialitzades en domòtica però també podem trobar un gran nombre d'aplicacions desenvolupades per persones amb la finalitat de donar suport a la seues instal·lacions demòtiques personals que finalment han anat creixent de tal forma que donen suport a un gran nombre de dispositius domòtics. el gran avantatge d'aquestes aplicacions desenvolupades anònimament és que la majoria han estat creades sense ànim de lucre i per tant poden ser descarregades lliurement.

D'aquestes aplicacions podem destacar-se'n dos, Domotiga i OpenDomo, que encara que ara per ara encara es troben en fase de desenvolupament, la seues seues comunitats de desenvolupadors i usuaris son molt grans i per tant el seu desenvolupament és molt actiu.

3. Altres programes

3.1 Domotiga

ha estat desenvolupat amb el llenguatge de programació Gambas Basic i corre sota el sistema operatiu Linux. Domotiga està pensat per poder suportar el major nombre de protocols per tal de poder comunicar-se amb el dispositius, entre els que destaquem X10 i Z-wave.

La seua interfície gràfica es clara i senzilla i s'hi pot accedir des de qualsevol plataforma. El seu esquema gràfic utilitzat per mostrar la informació es el mateix que utilitzarem en la nostra aplicació. Conta amb un menú lateral on es mostren les diferents localitzacions i dispositius i des del qual es pot mostrar, en la part central, la seua informació.

Web: <http://www.domotiga.nl/>

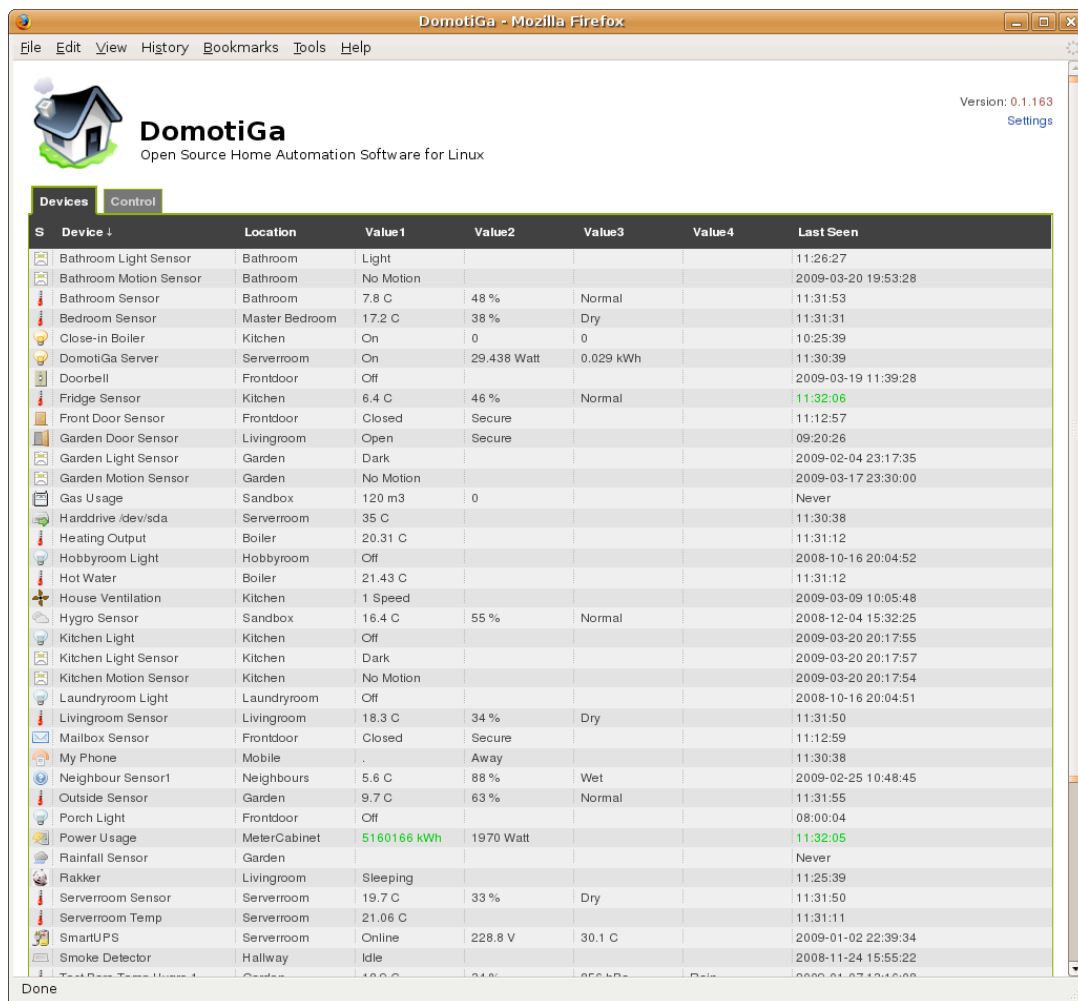


Figura 3. 1: Captura de Pantalla Domotiga

3.2 OpenDomo

¡Error! Marcador no definido. ha estat desenvolupat sota el llenguatge de programació C i també s'executa amb el sistema operatiu Linux. OpenDomo sorgeix de la necessitat d'unificar les diferents tecnologies existents en el món de la domòtica, com UPnP, X10, EIB, etc, amb el protocol de comunicacions més utilitzat en l'actualitat TCP / IP.

La seua interfície gràfica està dissenyada per ser mostrada en la majoria de dispositius web encara que no és tant intuïtiva com pot ser la de domotiga. Segons diuen al seu web, aquesta serà una de les millores que realitzaran pròximament.

Web: www.opendomo.org

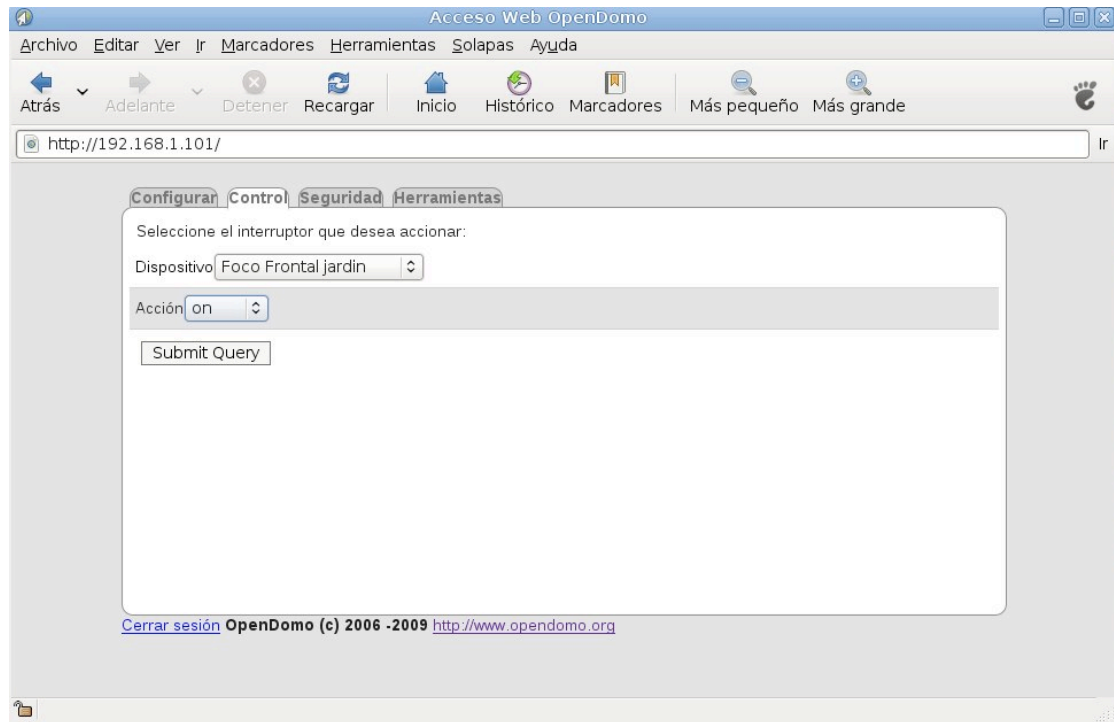


Figura 3. 2: Captura de Pantalla Domotiga

El gran inconvenient d'aquestes aplicacions es que els dispositius i les localitzacions tenen que ser afegides una per una. Això suposa un gran esforç quan estem treballant sobre un edifici on podem trobar un gran nombre d'elements. Una bona forma de solucionar aquest problema serà l'elaboració d'un XML on emmagatzemar tota la informació d'aquests elements per a que pugui ser llegit per l'aplicació.

En aquest projecte també s'intentarà millorar l'experiència de l'usuari afegint una mapa de l'edifici on es mostren totes les localitzacions i dispositius tal i com es troben en la realitat i que al clicar sobre ells es pugui accedir a la seua informació. Aquesta característica la trobem a faltar en les aplicacions esmentades anteriorment.

Part III

~~~~~

## *Domestica*

## 4. Primera Versió

L'elaboració d'aquesta primera versió va començar en el mes de febrer de 2010 encara que no seria fins a dos mesos més tard quan la versió 1.0 veuria la llum.

Durant aquesta primera versió l'aplicació s'anomenava WebHomeControl i no va ser fins a la segona versió quan es va canviar el nom a *Domestica*.

Els dos primers mesos va suposar, bàsicament, el primer contacte amb Eclipse, la tecnologia JSF, les seues llibreries i la llibreria JDOM. Entre totes les llibreries que utilitzen JSF com a tecnologia vaig decantar-me per IcesFaces i Tomahawk, encara que finalment vaig escollir aquesta ultima perquè IcesFaces no sempre funcionava correctament.

El que feia la versió 1.0 de *Domestica* (aleshores WebHomeControl) era llegir el fitxer "EjemploDevicesV0.2.xml", mostrar la informació amb un component tree2 de tomahawk, i si un element de l'arbre era clicat, es mostrava la seua imatge.

Exemple visual de la versió 1.0:



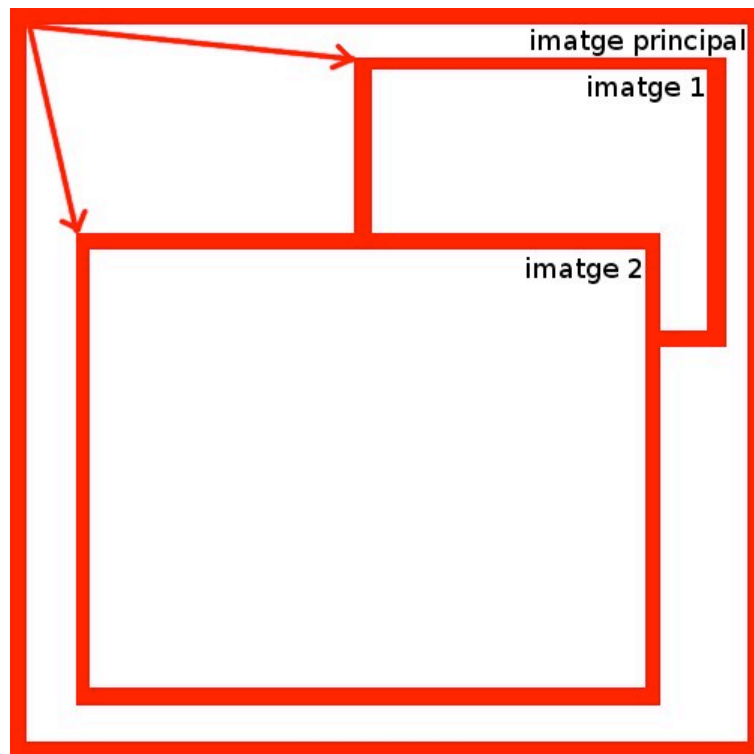
Figura 4. 1: Domotica 1.0

Posteriorment, el fitxer XML es va canviar i es va utilitzar el fitxer "ScenarioSpecificationV0.3" (el mateix que s'utilitza actualment) així que es va actualitzar l'aplicació perquè la informació pugues ser llegida i mostrada correctament.

Una de les coses que més em va costar implementar en aquest projecte va ser

el mapa de les localitzacions.

Principalment, el mapa de les localitzacions és una imatge que al mateix temps compta altres subimatges clicables a sobre d'ella, en una posició determinada.



*Figura 4. 2: Estructura mapa de les localitzacions*

Com podem veure en la imatge de dalt, tenim una imatge principal que al mateix temps compta altres subimatges ( imatge 1, imatge 2 ) a sobre d'ella. Aquestes subimatges son representades en la posició X,Y en funció de la posició 0,0 de la imatge principal.

El que vaig fer va ser, bàsicament, crear una llista amb totes les subimatges i recórrer-la amb l'atribut `ui:repeat` i per cada iteració crear un imatge clicable ( `commandLink` i `graphicImages` ) utilitzant els elements CSS `position:absolute`, `left` i `top` per establir la seua posició.

Exemple bàsic de codi:



```

<p:graphicImage value="#{image.url}"/> #Imatge Principal
<ui:repeat value="#{image.map}" var="item">
  <h:commandLink actionListener="#{image.getImage}">
    <h:graphicImage value="#{item.url}" #SubImatges
      style="position:absolute;left           :#{item.posX};top:#{item.posY};"
      height="#{item.height}" width="#{item.width}">
    <f:attribute name="docNum" value="#{item.locationIDRef}"/>
  </h:graphicImage>
</h:commandLink>
</ui:repeat>

```

Una vegada el mapa de les localitzacions va ser implementat va sorgir la versió 1.1 ( abril de 2010 ).

A hores d'ara aquesta versió encara tenia moltes manques, sobretot amb aspectes relacionats amb els dispositius, però ja anava agafant la forma, que en un futur, tindria la versió final.

Exemple visual de la versió 1.1 :

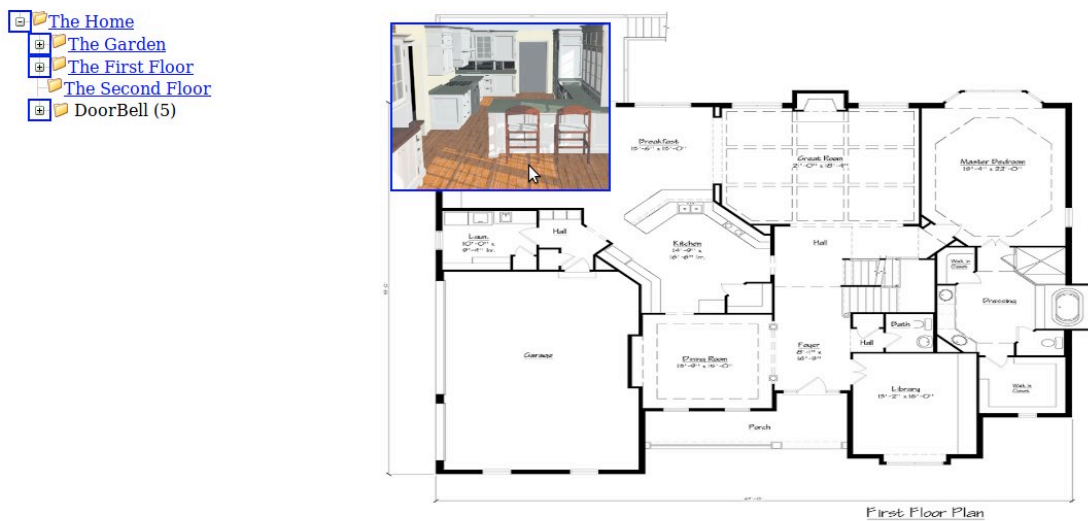


Figura 4. 3: Domestica 1.1

Finalment, a finals de maig de 2010, va sorgir l'última branca d'aquesta primera versió, la versió 1.2.

Aquesta versió ja implementava totes les manques de les anteriors versions, seguint la més important la relacionada amb els dispositius.

Exemple visual de la versió 1.2 :

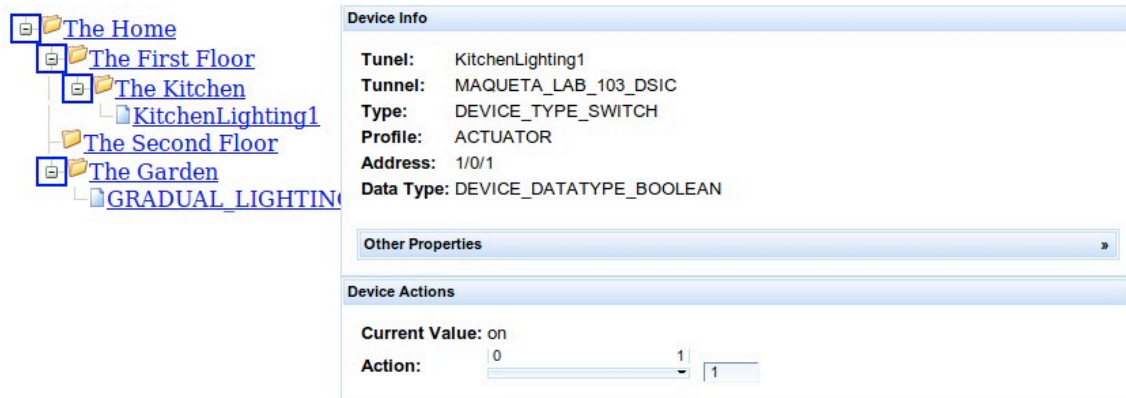


Figura 4. 4: Domestica 1.2

Com podem veure a l'exemple, a l'arbre existien dos tipus de node, les localitzacions i els dispositius.

Les localitzacions es representaven amb la icona d'una carpeta mentre que els dispositius es representaven amb la d'un fitxer.

Quan es clicava sobre un d'aquests últims, es mostrava una taula com la de l'exemple, on es mostrava la informació del dispositiu i les accions sobre aquest.

Totes les accions es representaven amb un component slider fent de la seua utilització, una acció poc intuïtiva

No serà fins a la segona versió de l'aplicació quan s'utilitzen botons per interactuar amb els dispositius.

La de Tomahawk i una mala estructuració del codi faran que la primera versió de *Domestica* sigues una versió poc atractiva visualment i difícil d'entendre per a qualsevol persona que intentés llegir el codi.

Per tant, encara que l'aplicació, funcionalment, ja feia tot allò que tenia fer, una reescriptura del codi serà necessària. Serà aleshores, quan va començar l'elaboració de la segona versió de *Domestica*.

## 5. Segona versió

Aquesta segona i última versió de *Domestica* suposarà la reescriptura total l'aplicació així com el reemplaç de la llibreria Tomahawk per la llibreria Primefaces. Aquest canvi ens facilitaria la d'Ajax en el projecte, ja que Primefaces està construït sobre l'API de JSF 2.0 Ajax, i així poder augmentar la velocitat, interactivitat i usabilitat.

Com que les noves versions de Primefaces utilitza JSF 2.0, l'aplicació es va actualitzar de JSF 1.2 a JSF 2.0.

La primera tasca a realitzar va ser la de reestructurar el codi. En la primera versió de *Domestica*, el BackingBean i els objectes es trobaven en el mateix directori així que van ser dividits a dintre dels directors *beans* i *objects* respectivament. A més a més, es van crear diferents BackingBeans per a cadascun del components de l'aplicació. En la primera versió hi havia un BackingBean *TreeLocation* per donar suport a l'arbre i un altre anomenat *WebActions* per donar suport a la resta de components. Això dificultava enormement la tasca de lectura i modificació del codi així que es varen crear els següent BackingBeans: *DevicesPage*, *LocationPage*, *LoadData* i *Tree* que donaven suport als dispositius, localitzacions, carrega de dades i a l'arbre respectivament.

Respecte a l'aspecte gràfic, el canvi més considerable va ser el de la creació d'una estructura general per poder mostrar els diferents components en els seus respectius llocs independentment de la resolució de la pantalla o del navegador. Açò es va aconseguir mitjançant la del component Primefaces *p:layout* que divideix la pantalla en diferents sectors left, top, center, bottom and right com podem veure en la imatge.

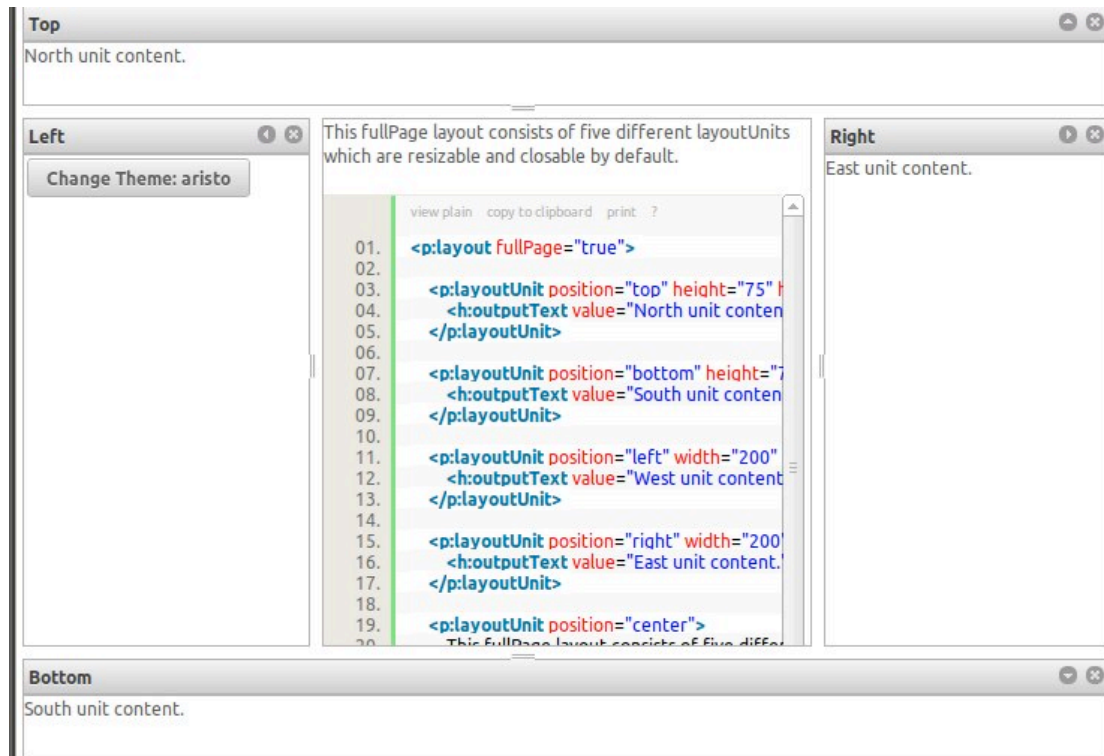


Figura 5. 1: Exemple component *p:layout*

Així doncs, es va col·locar en la part de dalt el logo de l'aplicació, en la part esquerra el menú i en el centre el mapa de les localitzacions i la informació sobre els dispositius. La part de baix i la de la dreta no s'utilitzaven així que es varen amagar per donar més espai a la resta de la web.

També es va redissenyar la pàgina encarregada de mostrar la informació dels dispositius.

**Device Info**

**Tunnel:** KitchenLighting1  
**Tunnel:** MAQUETA\_LAB\_103\_DSIC  
**Type:** DEVICE\_TYPE\_SWITCH  
**Profile:** ACTUATOR  
**Address:** 1/0/1  
**Data Type:** DEVICE\_DATATYPE\_BOOLEAN

**Other Properties** »

**Device Actions**

**Current Value:** on  
**Action:**

Abans

*Figura 5. 2: Informació sobre un dispositiu amb Domestica 1*



**Tunnel:** KitchenLighting1  
**Tunnel:** MAQUETA\_LAB\_103\_DSIC  
**Type:** DEVICE\_TYPE\_SWITCH  
**Profile:** ACTUATOR  
**Address:** 1/0/1  
**Data Type:** DEVICE\_DATATYPE\_BOOLEAN

**Properties**

- LOCATION : The Kitchen
- NAME : Kitchen Lighting
- MODEL : R23
- SERIAL\_NUMBER : xxxx-3454-cv
- DESCRIPTION : The Kitchen Lighting
- MANUFACTURER : ABB

Després

*Figura 5. 3: Informació sobre un dispositiu amb Domestica 2*

Mostrant la informació i les propietats a la banda dreta i afegint, a la banda esquerra, una imatge del dispositiu i els botons per interactuar amb aquest. A partir d'aquesta versió, el motor Ajax ha seguit l'encarregat de l'actualització parcial de la pàgina evitant la recarrega total d'aquesta. Així doncs, quan es clica un node de l'arbre o una subimatge del mapa de les localitzacions, sols s'actualitza la part central de la pàgina o quan es clica un dels botons que tenim a l'exemple de dalt, sols s'actualitza la part esquerra de la part central,

actualitzant l'estat de la imatge i els botons.

D'aquesta manera augmentarem la velocitat, interactivitat i usabilitat de l'aplicació.

A principis d'octubre de 2010 va sorgir la versió 2.0 de *Domestica*.

Exemple visual de la versió 2.0 :

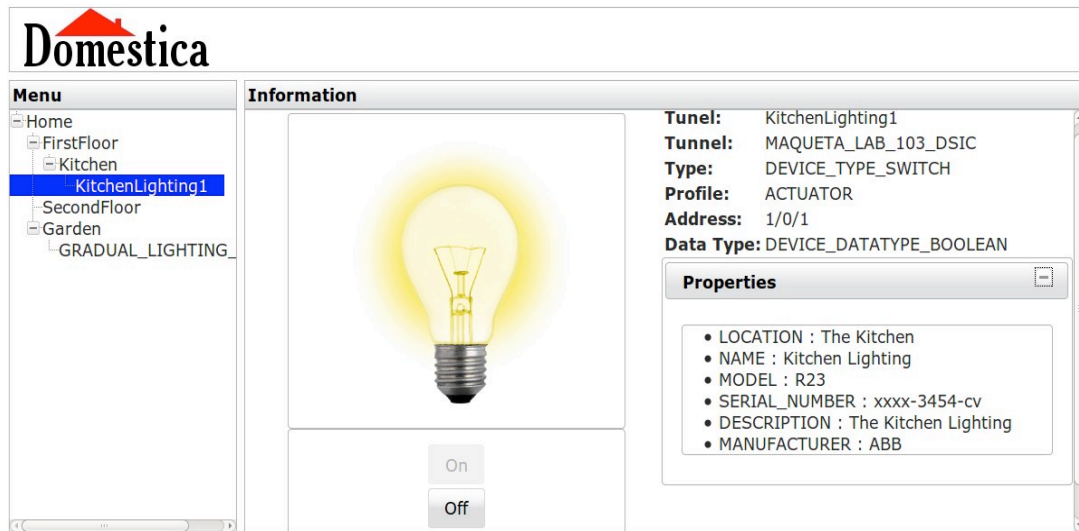


Figura 5. 4: Domestica 2.0

Finalment la versió 2.1 ( versió final ) seria alliberada al desembre de 2010 i entre les seues novetats destaquem l'afegiment d'una pàgina de benvinguda amb l'opció de poder canviar l'adreça IP, un botó per poder expandir / amagar l'arbre, diferents colors segons els nodes de l'arbre, canvi de les imatges per unes altres més realistes, capçaleres als fitxers amb la informació de la llicència GPL i divisió del fitxer *home.xhtml* en diferents fitxer.

El problema era que *home.xhtml* contenia tots els elements gràfics de l'aplicació i s'havia fet massa gran per a la tasca de llegir-lo o modificar-lo així que es varen crear els fitxer *devices.xhtml*, *locations.xhtml* i *tree.xhtml* per als elements gràfics dels dispositius, de les localitzacions i de l'arbre respectivament.

Així doncs, ara per ara *home.xhtml* compta l'estructura general de la web i amb l'atribut *iu:include* es fa referència als altres fitxers.

Una vegada la versió 2.1 va ser alliberada va ser el moment de fer la prova definitiva que verifiqués que tot funcionava correctament. La interacció entre l'aplicació i una maqueta domòtica.

Per realitzar aquesta tasca es va crear el mètode *\_requestHTTP(String DeviceType, String Action)* en el fitxer */src/beans/DevicesPage.java* el qual envia una petició HTML al servlet.

Com a paràmetres d'entrada tenim el tipus de dispositiu i l'acció a realitzar sobre eixe dispositiu.

## 6. Interfície d'usuari

el primer que ens trobem quan accedim a l'aplicació es el següent:

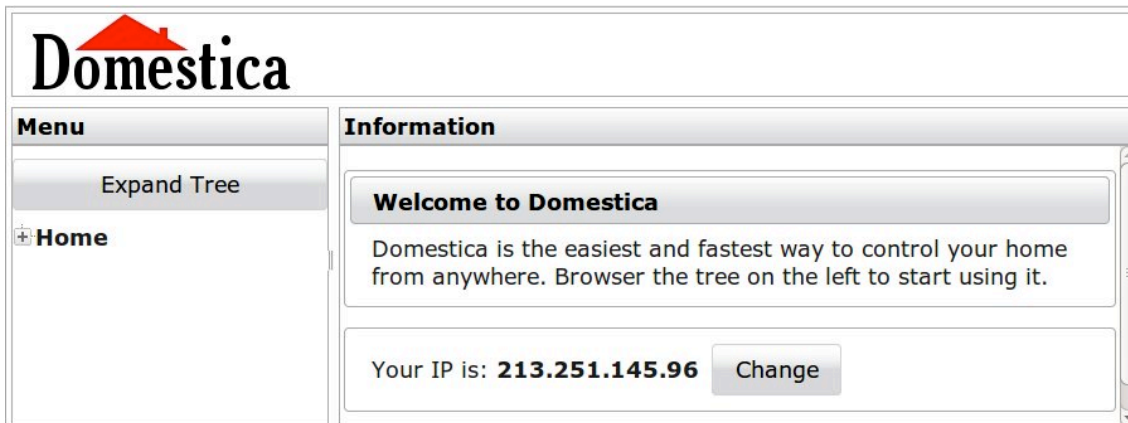


Figura 6. 1: Pàgina d'inici

Es tracta de la pàgina de benvinguda i principalment compta un missatge on s'explica breument com començar a utilitzar-la així com també l'IP del servlet i un botó per poder canviar-la. Quan el polsem es mostra la següent finestra:

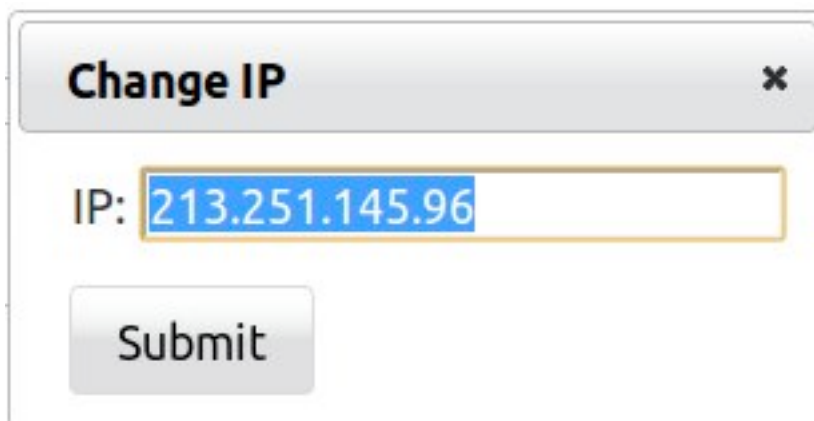
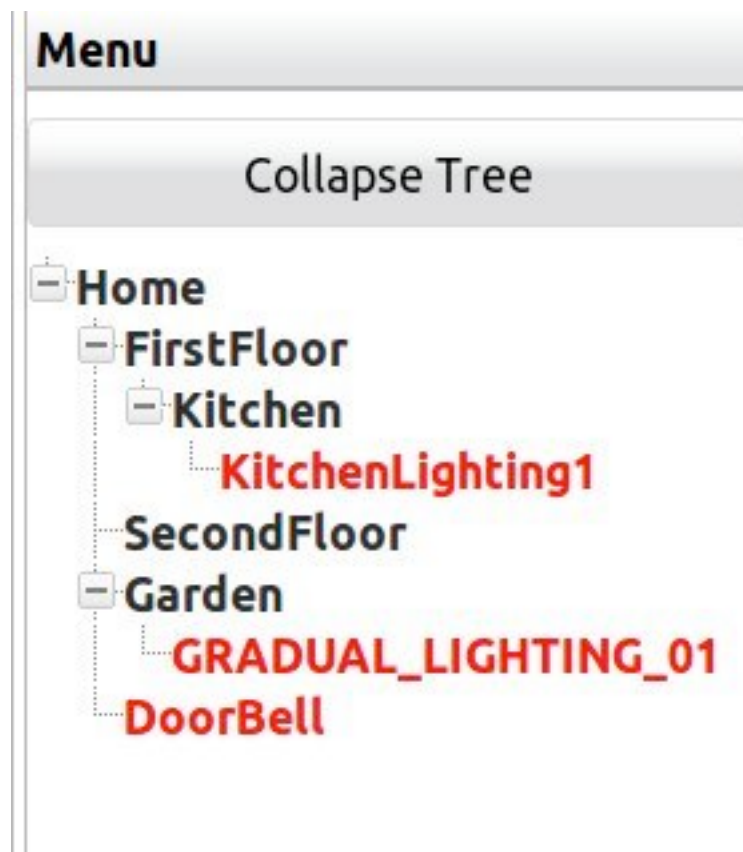


Figura 6. 2: Finestra Canvi IP

La primera vegada que l'usuari accedeix a l'aplicació no hi ha cap IP establerta i aleshores és mostra el missatge NO IP. Una vegada es establida, aquesta es guardarà en el servidor per a les pròximes vegades.

En la banda esquerra de la pàgina d'inici trobem l'arbre o menú on l'usuari pot interactuar amb els diferents elements de la casa.



*Figura 6. 3: Menu*

L'arbre es tracta d'un menú desplegable que mostra totes les localitzacions i dispositius de la casa, els quals es representen amb un color negre o roig respectivament. Des d'aquest menú es pot accedir tant a la informació de les localitzacions com la dels dispositius. La representació en forma d'arbre té l'objectiu de mostrar els elements segons la seua posició en la casa. Per exemple, si mirem en la imatge, KitchenLighting1 és un dispositiu que es troba a dintre de la localització Kitchen ( la cuina ) que al mateix temps es part de FirstFloor ( Planta primera ) i que al mateix temps es part de Home ( Llar )

El menú també compta amb un botó per poder expandir o amagar tot l'arbre amb un sol clic.

Si per exemple cliquem sobre l'element "FirstFloor" en el centre de la pàgina se'ns mostrarà una imatge com aquesta:





*Figura 6. 4: Mapa de FirstFloor*

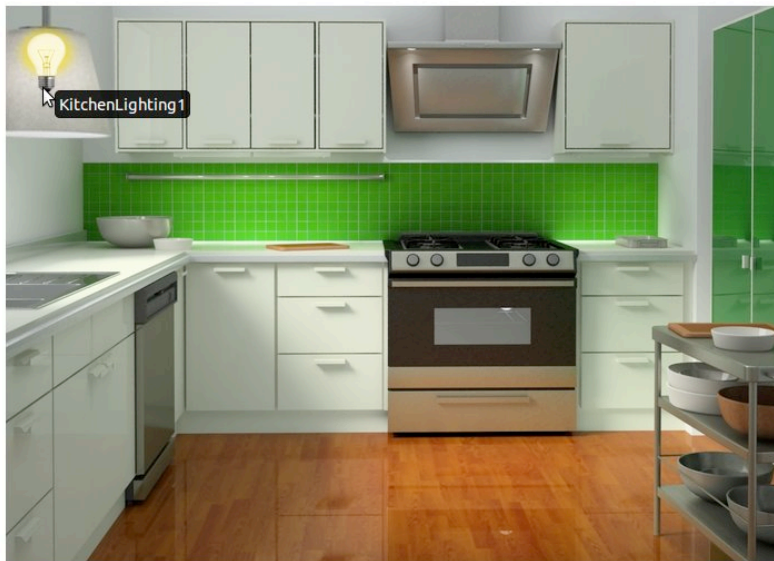
Aquesta imatge es tracta del mapa de les localitzacions de l'element "FirstFloor". Com podem veure en la imatge el número 1 es correspon amb el plànol de l'element "FirstFloor".

Si la localització té subelements ( com en aquest cas) es mostraran les subimatges d'aquestos. La posició d'aquestes subimatges serà en respecte al plànol i clicant sobre elles es pot accedir a la seua pàgina. Si es tracta d'una localització s'accedirà al seu plànol i si es tracta d'un dispositiu s'accedirà a la seua informació. En la imatge, el número 2 fa referència a la subimatge de la localització "Kitchen".

El número 3 és tracta del dispositiu "KitchenLighting1". Si una localització ( En aquest cas "FirstFloor ) té sublocalitzacions ( en aquest cas "Kitchen" ) que tenen dispositius a dintre d'ella ( en aquest cas "KitchenLighting1 ) aquestos també seran mostrats sobre la imatge, però clicant sobre ells no es podrà accedir a la seua informació. Aquests elements es representaran en relació al seua pare, o siga, a la posició 0,0 de la imatge de "Kitchen".

Si cliquem sobre la subimatge "Kitchen" se'ns mostrarà el següent:


## Information



*Figura 6. 5: Mapa de Kitchen*

Ara el que veiem és el plànol de “Kitchen” i la subimatge del dispositiu “KitchenLighting1” que en aquest cas sí que es pot clicar. Com podem veure a la imatge, aquest dispositiu es troba encès així que si cliquem sobre ell podrem accedir a la seua informació i des d’ací podrem apagar-lo o realitzar altre tipus d’accions.

**Information**



State: on


**Id:** KitchenLighting1  
**Tunnel:** MAQUETA\_LAB\_103\_DSIC  
**Type:** DEVICE\_TYPE\_SWITCH  
**Profile:** ACTUATOR  
**Address:** 1/0/1  
**Data Type:** DEVICE\_DATATYPE\_BOOLEAN

**Properties**

LOCATION : The Kitchen  
NAME : Kitchen Lighting  
MODEL : R23  
SERIAL\_NUMBER : xxxx-3454-cv  
DESCRIPTION : The Kitchen Lighting  
MANUFACTURER : ABB

*Figura 6. 6: Informació sobre KitchenLighting1*

**Information**



State: 50-74

**Id:** GRADUAL\_LIGHTING\_01  
**Tunnel:** MAQUETA\_LAB\_103\_DSIC  
**Type:** DEVICE\_TYPE\_PERCENTUAL\_DIMMER  
**Profile:** ACTUATOR  
**Address:** 1/0/5  
**Data Type:** DEVICE\_DATATYPE\_INTEGER

**Properties**

LOCATION : The Garden  
NAME : Garden Gradual Lighting  
MODEL : R26-9-xz  
SERIAL\_NUMBER : xyzt-3278-cv  
DESCRIPTION : The Garden Gradual Lighting  
MANUFACTURER : ABB

*Figura 6. 7: Informació sobre GradualLighting1*

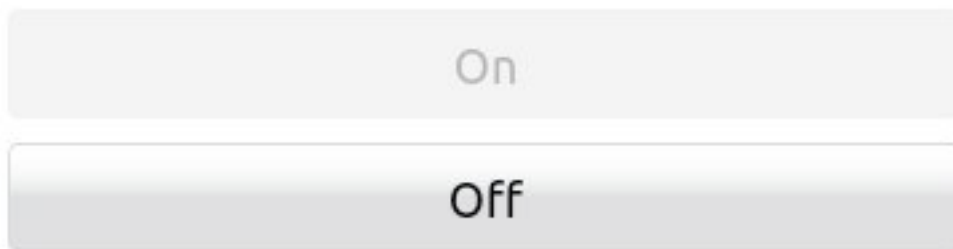
En la primera imatge podem veure el que es mostra quan cliquem sobre la subimatge "KitchenLighting1".

La segona imatge és un altre exemple de la informació d'un altre tipus de

dispositiu.

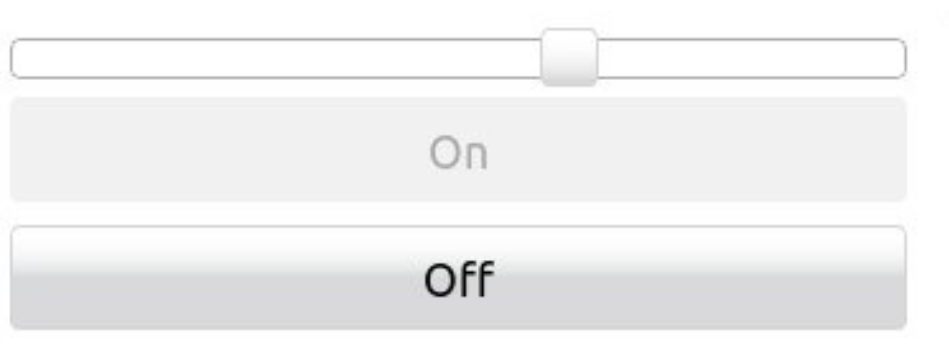
La informació que es mostra es la següent:

- Imatge del dispositiu ( **Opcional** )
- Informació general ( **obligatori** )
  - ID ( Identificador )
  - Tunnel ( Túnel )
  - Type ( Tipus )
  - Profile ( Perfil )
  - Address ( Adreça )
  - Data Type ( Tipus de data )
- Propietats ( **Opcional** )
- Botons d'Acció ( **Depenen del tipus de dispositiu** )
  - DEVICE\_TYPE\_SWITCH



*Figura 6. 8: Botons Device\_type\_switch*

- DEVICE\_TYPE\_PERCENTUAL\_DIMMER



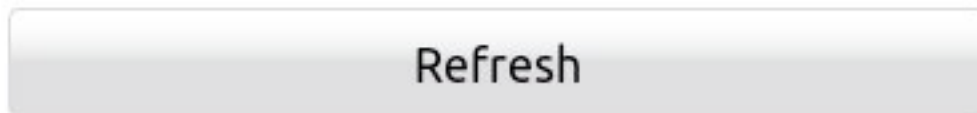
*Figura 6. 9: Botons Device\_type\_percentual\_dimmer*

- DEVICE\_TYPE\_PULSE



*Figura 6. 10: Botons Device\_type\_pulse*

- DEVICE\_TYPE\_SENSOR



*Figura 6. 11: Botons Device\_type\_percentual\_dimmer*

- DEVICE\_TYPE\_VERTICAL\_MOVEMENT



*Figura 6. 12: Botons Device\_type\_vertical\_movement*

- DEVICE\_TYPE\_ANGULAR\_DIMMER

---

*Figura 6. 13: Botons Device\_type\_angular\_dimmer*

- `DEVICE_TYPE_HEX_DIMMER`

---

*Figura 6. 14: Botons Device\_type\_hex\_dimmer*

## 7. Evolució de les versions

### Primera Versió

- *Domestica* 1.0 : 24-03-10
- *Domestica* 1.1 : 28-04-10
- *Domestica* 1.2 : 19-05-10

### Segona Versió

- *Domestica* 2.0 : 07-10-10
- *Domestica* 2.1 (Final) : 07-10-10

# 8. Estructura dels fitxers

El fitxers estan estructurats de la següent forma:

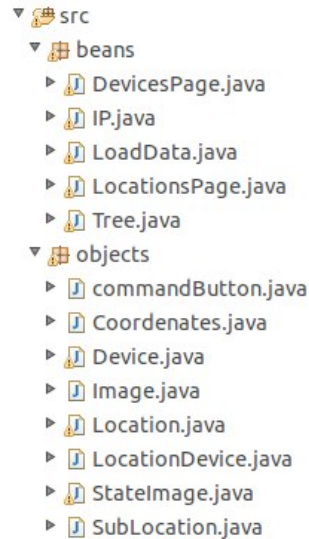


Figura 8. 1: Directori /SRC

- **src/beans:** Directori on s'emmagatzemen totes les classes Java que componen la part Servidor de l'aplicació. Fitxers:
  - DevicePage.java: Classe encarregada de totes les accions relacionades amb els dispositius.
  - IP.java: Classe encarregada del canvi o establiment de una adreça IP.
  - LoadData.java: Classe encarregada de la lectura del fitxer XML amb la informació relacionada amb els elements de la casa. S'executa soles una vegada ( quan l'usuari carrega la pàgina)
  - LocationPage.java: Classe encarregada de totes les accions relacionades amb les localitzacions.
  - Tree.java: Classe encarregada de la creació del menú que es mostra a la part esquerra de la web.
- **src/objects:** Directori on s'emmagatzemen tots els objectes utilitzats per les classes citades anteriorment.

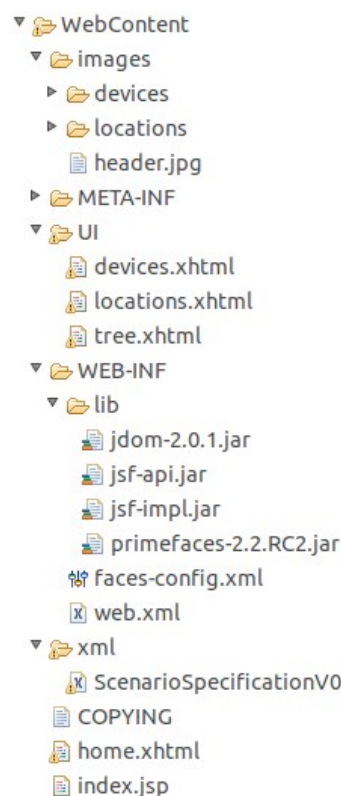


Figura 8. 1: Directori /WebContent

- **WebContent:** Directori on s'emmagatzemen totes les aplicacions web utilitzades per l'aplicació. Fitxers:
  - index.jsp: És el primer fitxer que s'executa quan es carrega l'aplicació. L'únic que fa és cridar al fitxer home.xhtml. Podríem definir-lo com un fitxer pont.
    - home.xhtml: Crea els components JSF per poder mostrar el disseny general de la web. A més a més crida a LoadData.java (si és necessari).
  - COPYING: llicència del projecte.
- **WebContent/images:** Directori on s'emmagatzemen totes les imatges utilitzades en la web.
- **WebContent/UI:** Directori on s'emmagatzemen tots els fitxers relacionats amb l'entorn gràfic de la web. Fitxers:
  - devices.xhtml: Crea els components JSF per poder mostrar els diferents elements relacionats amb els dispositius.
  - locations.xhtml: Crea els components JSF per poder mostrar els diferents elements relacionats amb les localitzacions.
  - tree.xhtml: Crea els components JSF per poder mostrar el menú.
- **WebContent/xml:** Directori on s'emmagatzema el fitxer XML amb la informació sobre els dispositius i les localitzacions.



- **WebContent/WEB-INF**: Directori creat automàticament durant la creació de l'aplicació que emmagatzema tota la metadata de l'aplicació. Fitxers:
  - web.xml: Proveeix la informació sobre configuració i utilització per als components web que componen l'aplicació. Descriu, entre altres, el servlet.
  - faces-config.xml: Fitxer de configuració del JSF. Llista les classes Java i defineix les regles de navegació.
- **WebContent/WEB-INF/lib**: Directori on s'emmagatzemen les diferents llibreries utilitzades per l'aplicació.

## 9. Llicència

Aquest projecte està registrat sota llicència GPL ( General Public Licence ) i per tant pot ser copiat, distribuït i modificat lliurement sempre que qualsevol modificació es continue distribuint amb la mateixa llicència GPL. Podeu trobar més informació sobre la llicència a les capçaleres dels fitxers del projecte així com també al fitxer WebContent/COPYING



# **Part IV**

# **Conclusions**

*Domestica* és el resultat d'un llarg període de desenvolupament on s'han utilitzat les últimes tecnologies, com per exemple JSF 2, i s'ha pogut, com dèiem al començament d'aquest document, donar solució als objectius principals del projecte: La creació d'una aplicació amb una interfície d'usuari fàcil i senzilla i el poder gestionar la llar digital des d'una sola web i amb pocs clics de ratolí.

Respecte al meus objectius i aspiracions personals, la sensació de satisfacció és més que present per haver pogut adquirir un gran coneixement en la tecnologia JSF. El fet de desenvolupar una aplicació d'aquestes característiques requereix uns coneixements avançats que han creat en mi certa familiaritat amb aquesta tecnologia.

## 10. Treballs Futurs

Bàsicament hi han 3 coses que podrien ser implementar i millorarien la interactivitat i seguretat de l'aplicació.

### Utilització de registres

La primera seria la creació de registres per a totes les accions realitzades sobre els dispositius. Els registres emmagatzemarien el dia, l'hora, el dispositiu i l'acció realitzada sobre aquest i podrien ser mostrats tant en la pàgina d'inici com en cadascuna de les diferents pàgines dels dispositius.

El registre mostrat en la pàgina d'inici contindria la informació sobre tots els dispositius mentre que el de cada dispositiu sols mostraria la informació d'aquest.

Un exemple del registre de la pàgina d'inici seria:

| Day          | Time     | Device              | Action |
|--------------|----------|---------------------|--------|
| Today        | 06:54:43 | KitchenLighting1    | On     |
| Today        | 01:01:10 | GRADUAL_LIGHTING_01 | 34%    |
| Today        | 12:34:56 | GRADUAL_LIGHTING_01 | 50%    |
| Yesterday    | 12:10:22 | KitchenLighting1    | Off    |
| 2 days ago   | 23:45:12 | TwilightSensor1     | --     |
| 2 days ago   | 23:10:24 | GRADUAL_LIGHTING_01 | 0%     |
| 2 months ago | 23:10:22 | TwilightSensor1     | --     |

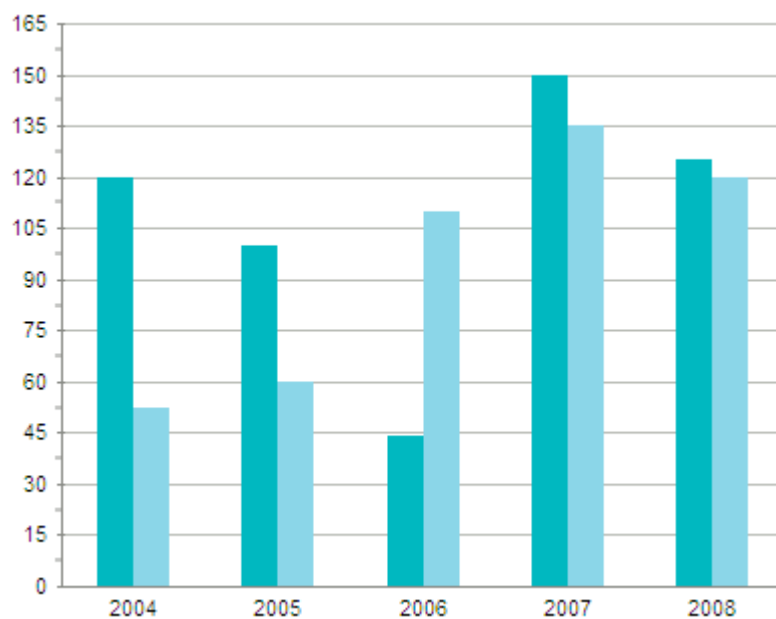
I per a un dispositiu concret (KitchenLighting1):

| Day   | Time     | Action |
|-------|----------|--------|
| Today | 06:54:43 | On     |
| Today | 01:01:10 | Off    |

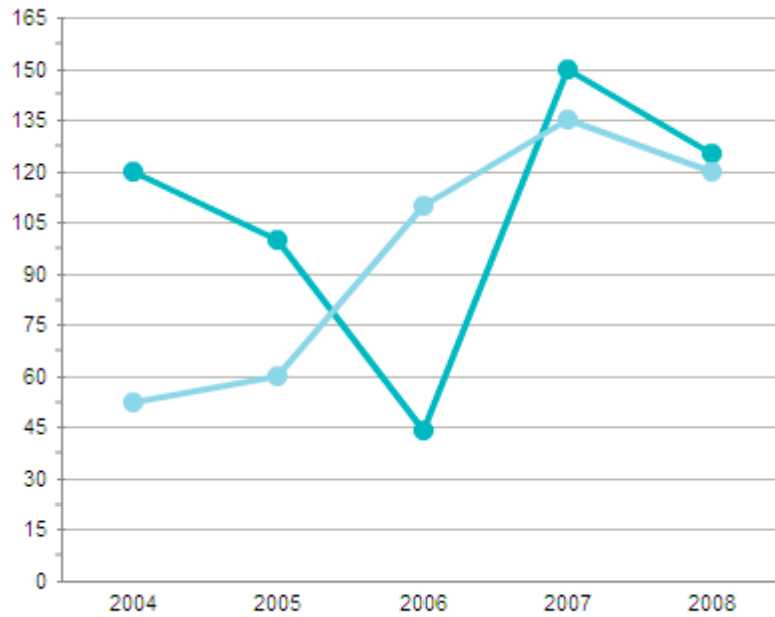
### Utilització de Gràfics

La segona cosa que podria portar-se a terme seria la creació de gràfics de forma dinàmica per poder mostrar les despeses de gas, aigua i electricitat.

Primefaces compta amb un gran nombre de components per realitzar gràfics entre els que podem destacar `p:columnChart` i `p:lineChart`:



*Figura 10. 1: Exemple p:columnChart*



*Figura 10. 2: Exemple p:lineChart*

### **Gestió d'usuaris i contrasenyes**

Finalment convindria crear un nou apartat en la web per poder gestionar els diferents usuaris que accedeixen a la web. Això podria realitzar-se mitjançant la creació d'un administrador que pugui controlar un base de dades on s'emmagatzemen els usuaris.

L'autenticació d'usuaris al iniciar l'aplicació també seria una cosa important per garantir la seguretat i evitar que terceres persones tinguessin control sobre la llar digital.



# **Part V**

# **Apèndixs**





# A. Estructura XML

- Locations
  - Location
    - ID
      - ID
    - Name
      - name
    - SuperLocationIDRef
      - superLocationID
    - Image
      - URL
      - PosX
      - PosY
      - Width
      - Height
  - Sublocations
    - Sublocation
      - LocationIDRef
        - locationID
      - ClickableArea
        - PosX
        - PosY
        - Width
        - Height
      - Thumbnail
        - URL
        - PosX
        - PosY
        - Width
        - Height
  - LocationDevices
    - LocationDevice

- DeviceIDRef
  - deviceID
- ImagePosition
  - PosX
  - PosY
  - Width
  - Height
- ThumbnailPosition
  - PosX
  - PosY
  - Width
  - Height
- Devices
  - Device
    - ID
      - ID
    - Tunnel
      - tunnelName
    - Type
      - deviceType
    - Profile
      - profile
    - Address
      - address
    - DataType
    - Platform
      - PlatformName
        - name
      - StateImage
        - StateImage
          - state
      - ImageFile
        - URL

- ThumbnailFile
  - URL
- Properties
  - Property
    - Literal
    - Value

# B. Primers passos amb JSF 2.0

## B.1 Prerequisits

- **JRE ( Java Runtime Environment )** és el conjunt de programes i llibreries de Java necessaris per a executar aplicacions desenvolupades en aquest llenguatge de programació.

Web:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- **JDK ( Java Development Kit )** és un programari que proveu ferramentes de desenvolupament per a la creació de programes en Java.

Web:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

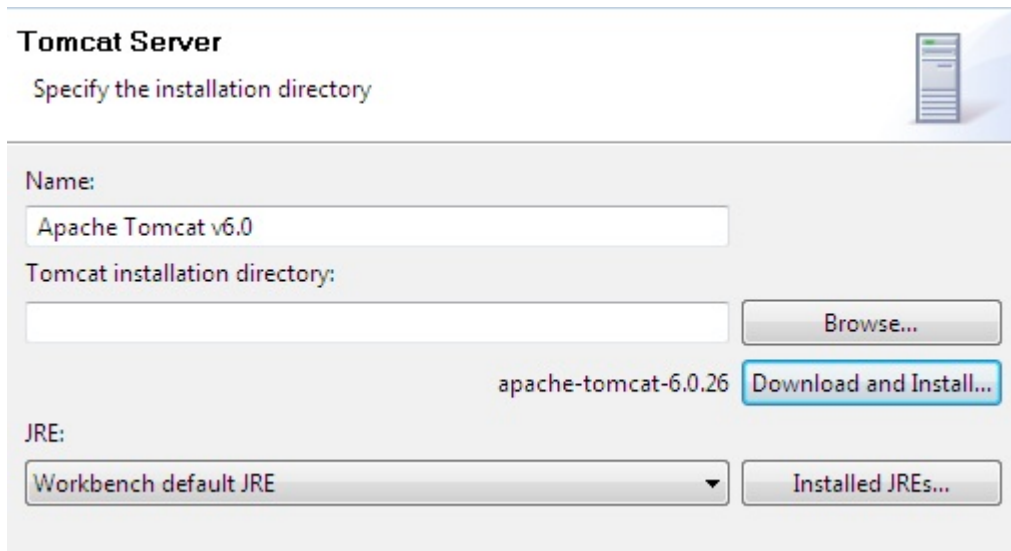
- **Eclipse ( Java EE Developers )** és un entorn integrat de desenvolupament de codi obert programat principalment amb Java per desenvolupar projectes amb gairebe qualsevol llenguatge de programació.

Web: <http://www.eclipse.org/>

- **Apache Tomcat ( Core )** funciona com un contenidor de Servlets de codi obert desenvolupat per la Apache Software Foundation ( ASF ). Tomcat implementa les especificacions dels Servlets i de JSP.

Web: <http://tomcat.apache.org/index.html>

Instal·lem JRE, JDK i l'Eclipse i descomprimim el fitxer de Tomcat en qualsevol directori. Seguidament iniciem l'Eclipse i ens dirigim a Window -> Preferencies -> Server -> Runtime Environment i cliquem el botó Add. Seleccionem la versió de Tomcat que hem descarregat ( en aquest projecte s'ha utilitzat Tomcat 6 ) i cliquem el botó Next. Ens apareixerà una finestra com aquesta:

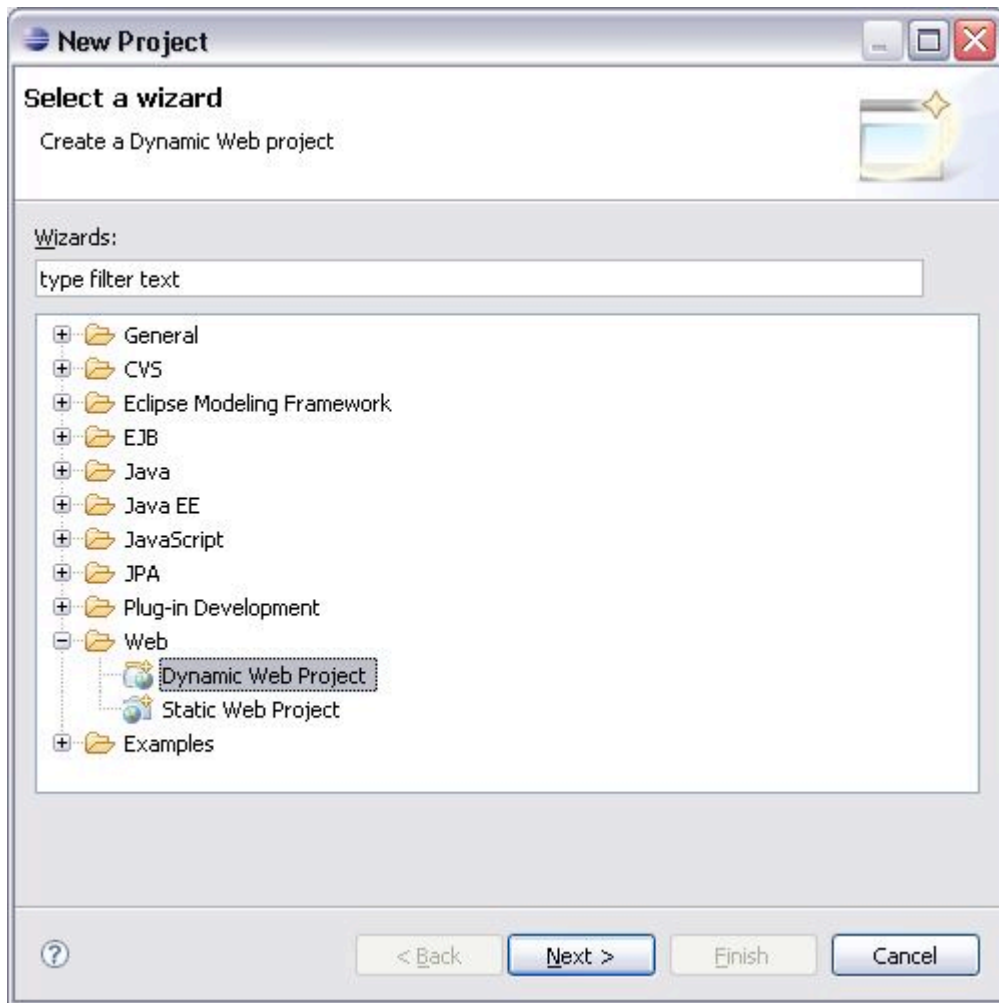


*Figura B. 1: Instal·lació Tomcat*

Aleshores cliquem en “Browser” i seleccionem el directori on hem descomprimit Tomcat. Finalment cliquem en “Download” and Install” i seguidament en “Finish” i ja tenim Tomcat instal·lat en Eclipse.

## **B.2 Crear un nou projecte**

Per poder crear un nou projecte JSF crearem un “Dinamic web project” clicant en File -> New -> Project -> Web -> Dynamic Web Project.



*Figura B. 2: Creació Web Dinàmica*

Cliquem en Next i en la següent finestra assignem el nom al projecte i seleccionem Apache Tomcat v6.0 ( o la versió que em instal·lat ) com a servidor predeterminat. Seguidament cliquem en Finish.



*Figura B. 3: Crear Web Dinàmica 2*

L'estructura per defecte dels fitxers del projecte és la següent:



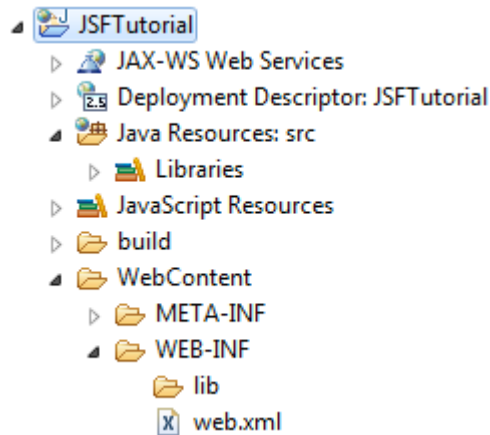


Figura B. 4: Estructura fitxer per defecte

## B.3 Importar les llibreries

Ara anem a <http://javaserverfaces.java.net/> , descarreguem l'última versió de JSF 2 i descomprimim el fitxer.

Per carregar les llibreries en Eclipse cliquem amb el botó secundari en el directori `lib` del projecte ( `WebContent/WEB-INF/lib` ) i seleccionem Import. En la nova finestra anem a General -> File System, cliquem en següent i després en Browser. Ara ens apareixerà una finestra on tenim que seleccionar el directori `lib` del fitxer descomprimit anteriorment. Finalment seleccionem els fitxers `jsf-api.jar` i `jsf-impl.jar` i cliquem en "Finish".

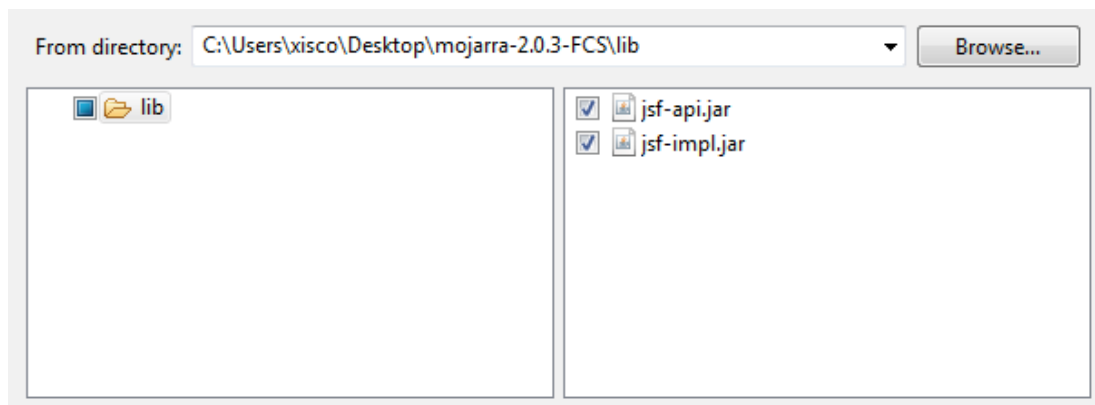


Figura B. 5: Importar Llibreries

# C. Exemple JSF: Hola món

## Crear el JSP i els XHTML

Cliquem amb el botó secundari sobre Webcontent, New -> File i creem el fitxer *index.jsp*.

/WebContent/index.jsp

```
<% response.sendRedirect("pagina1.jsf"); %>
```

Creem un altre fitxer anomenat *pagina1.xhtml*

/WebContent/pagina1.jsp

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html">

  <h:head>
    <title>JSF 2.0 Hola món</title>
  </h:head>
  <h:body>
    <h3>Exemple JSF 2 - Hola món</h3>
    <h:form>
      <h:inputText value="#{helloBean.name}"></h:inputText>
      <h:commandButton value="Benvingut"
action="pagina2"></h:commandButton>
    </h:form>
  </h:body>
</html>
```

I per últim creem el fitxer *pagina2.xhtml*

/WebContent/pagina2.jsp

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">

  <h:head>
    <title>JSF 2.0 Hola món</title>
  </h:head>
  <h:body bgcolor="white">
    <h3>Exemple JSF 2 - Hola món</h3>
    <h4>Benvingut #{helloBean.name}</h4>
  </h:body>
</html>

```

## Crear el BackingBean

Creem una directori en “src: Java Resources “ ( Botó secundari, New -> Package ) amb el nom bean, i creem el fitxer myBean.java a dintre seu ( Botó secundari, New -> Class ).

/src/bean/myBean

```

package bean;

import java.io.Serializable;

public class myBean implements Serializable {

    private static final long serialVersionUID = 1L;

    private String name;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}

```

## Crear la configuració de JSF

Creem el fitxer faces-config.xml ( Botó secundari, New -> File ) en el directori WEB-INF i li afegim el següent codi.

/WebContent/WEB-INF/faces-config.xml

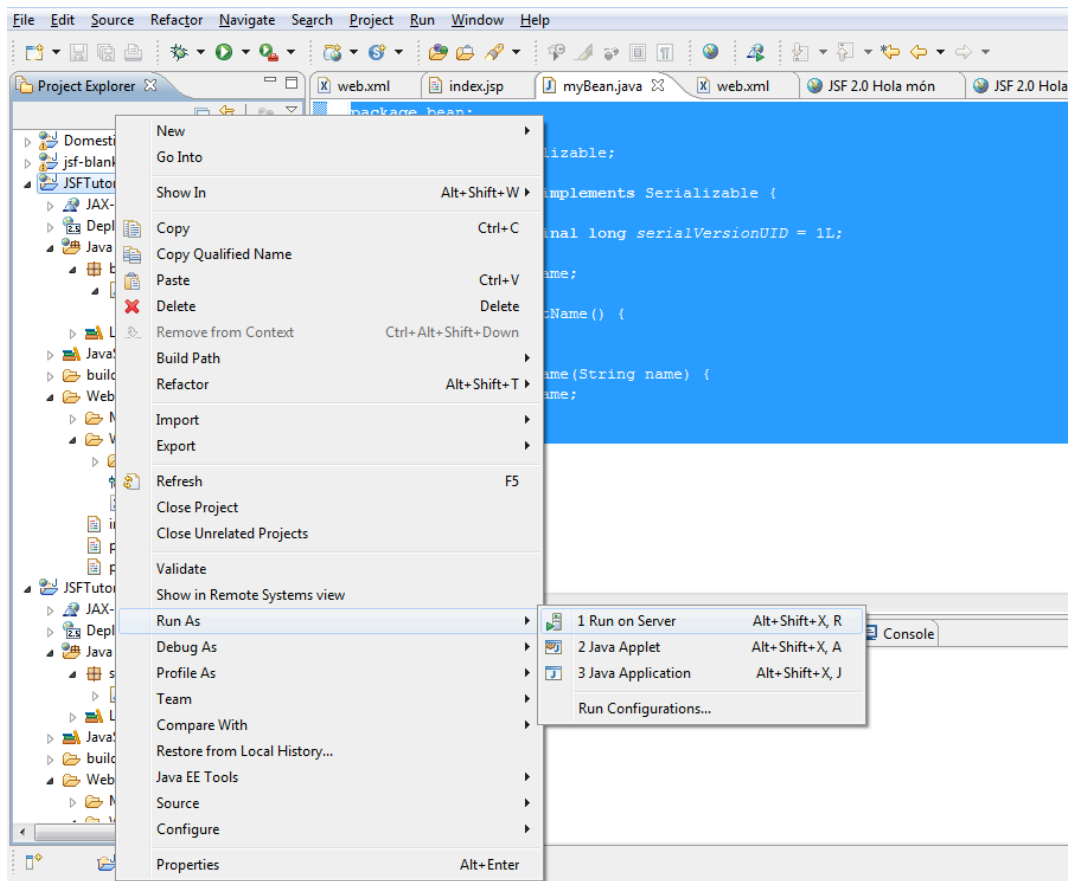
```
<?xml version="1.0"?>
<faces-config xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
  version="2.0">
  <managed-bean>
    <managed-bean-name>holaBean</managed-bean-name>
    <managed-bean-class>bean.myBean</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean>
</faces-config>
```

## Editar el fitxer web.xml

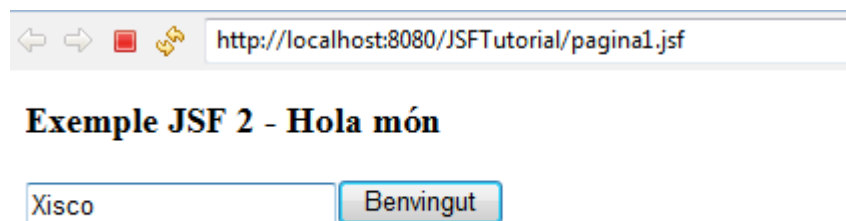
/WebContent/WEB-INF/web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.jsf</url-pattern>
  </servlet-mapping>
  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Finalment provem l'aplicació. Botó secundari sobre el nom del projecte, Run As  
-> Run on Server



*Figura C. 1: Provar Exemple*



*Figura C. 2: Resultat*

## D. Importar Primefaces

Ens descarreguem el fitxer binari de l'última versió de Primefaces de <http://www.primefaces.org/downloads.html> i l'importem a la carpeta /WebContent/WEB-INF/lib ( botó secundari, Import -> General -> File System ).

Finalment agregem a les capçaleres del fitxer XHTML la següent línia i ja tenim la llibreria Primefaces carregada.

```
xmlns:p="http://primefaces.prime.com.tr/ui"
```

Exemple:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.prime.com.tr/ui">

<h:head>

</h:head>

<h:body>

    <p:breadcrumb>
        <p:menuItem value="Categories" url="#" />
        <p:menuItem value="Sports" url="#" />
        <p:menuItem value="Football" url="#" />
        <p:menuItem value="Countries" url="#" />
        <p:menuItem value="Spain" url="#" />
        <p:menuItem value="F.C. Barcelona" url="#" />
        <p:menuItem value="Squad" url="#" />
        <p:menuItem value="Lionel Messi" url="#" />
    </p:breadcrumb>

</h:body>
</html>

```

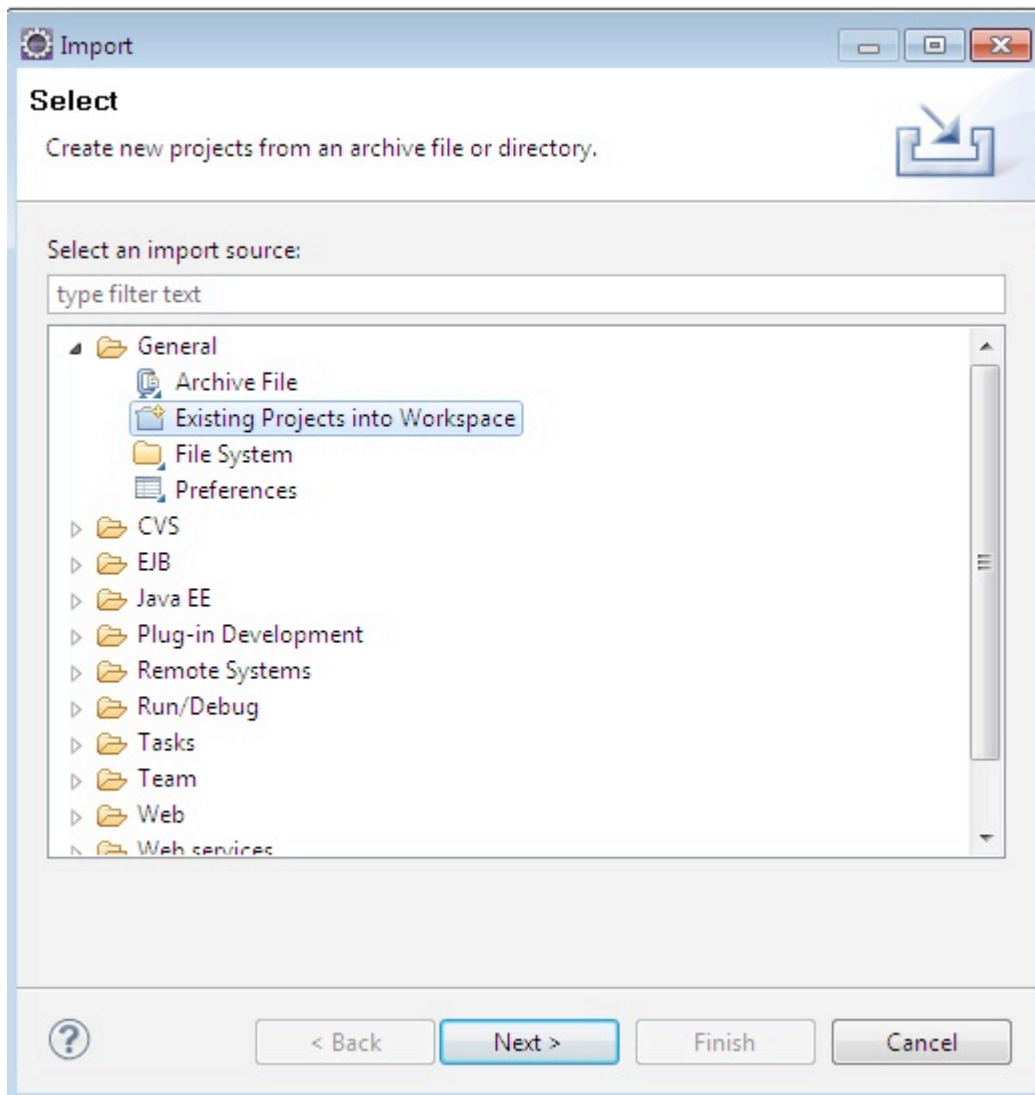


*Figura D. 1: Resultat Primefaces*



## E. Importar *Domestica*

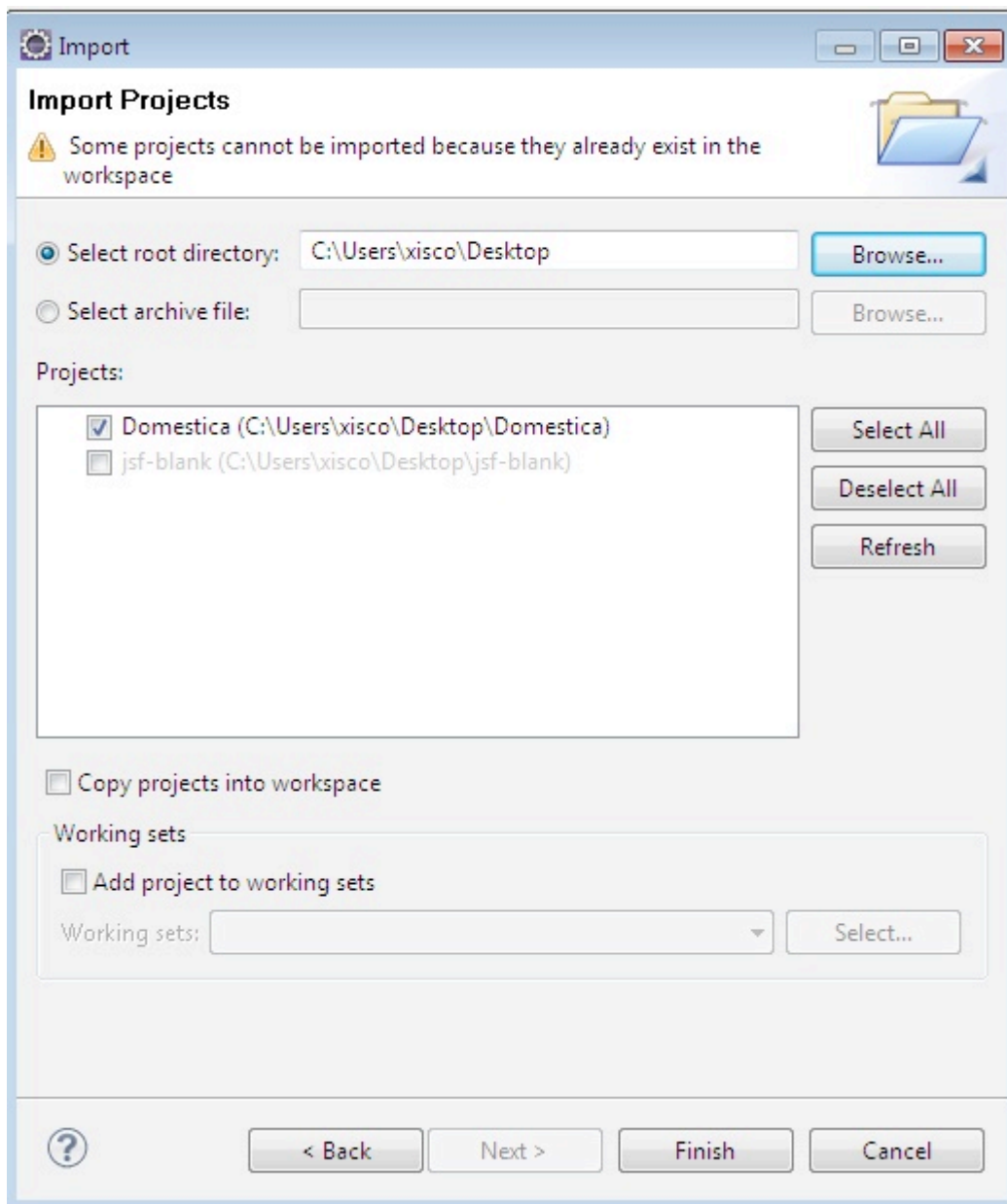
Anem a File -> Import i en eixirà una finestra com la següent:



*Figura E. 1: Importar Projecte*

Cliquem sobre “Existing Projects into Workspace” i seguidament en el botó “Browser” de la nova finestra. Naveguem fins a la carpeta on es troba la carpeta “Domotica” i cliquem en Acceptar.

**Es recomanable** copiar la carpeta del CD al disc dur abans d'importar-la, següent l'opció més recomanable la carpeta de treball d'Eclipse, anomenada “workspace”. Quan cliquem en Acceptar es mostra la següent finestra:



*Figura E. 2: Importar Projecte 2*

En la part central de la finestra es mostren tots el projectes existents a la carpeta seleccionada com arrel. Tenim que seleccionar “Domestica” i clicar en “Finish”.

Finalment, per poder executar el projecte, soles tenim que clicar amb el botó secundari sobre el seu nom en el “Project Explorer” -> Run As -> Run On Server.

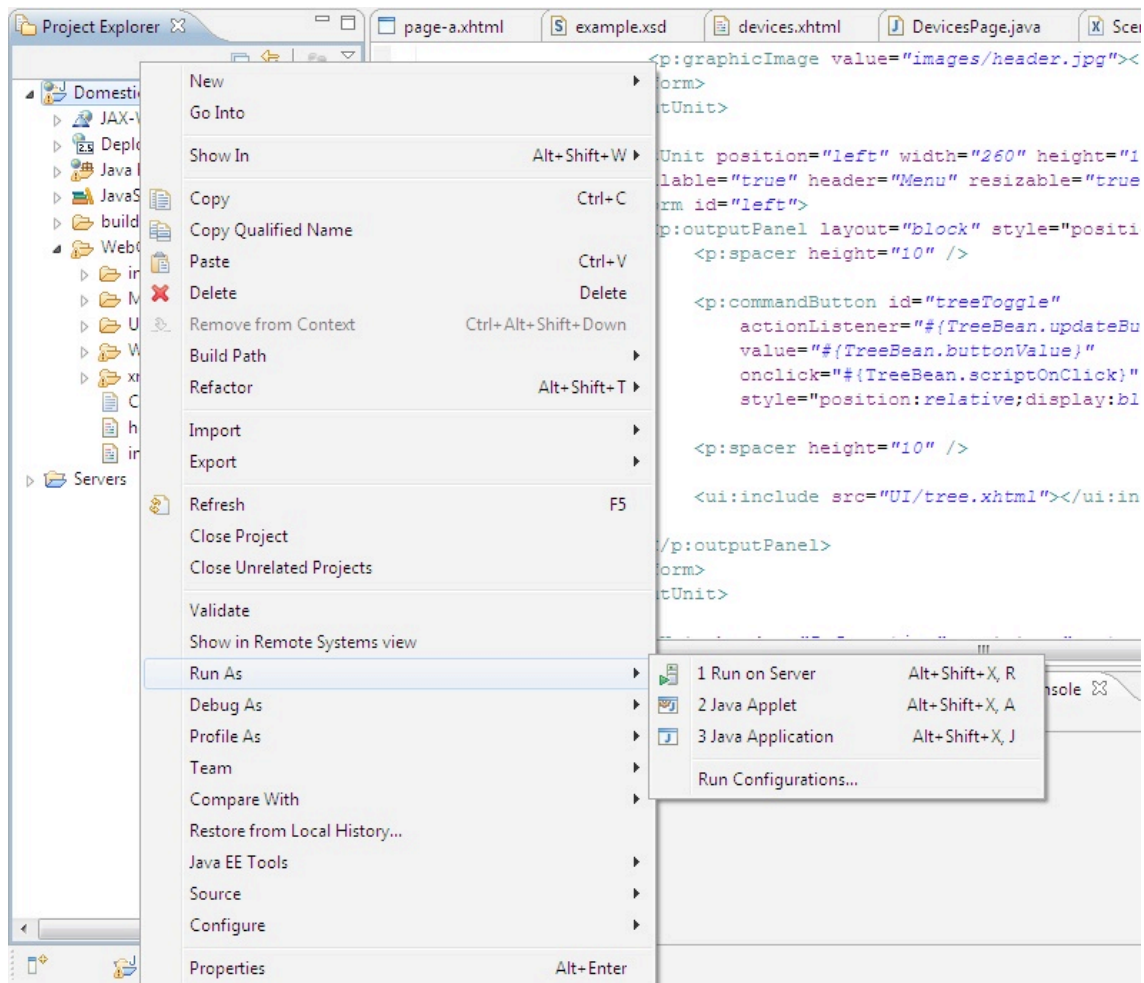


Figura E. 3: Executar Domestica

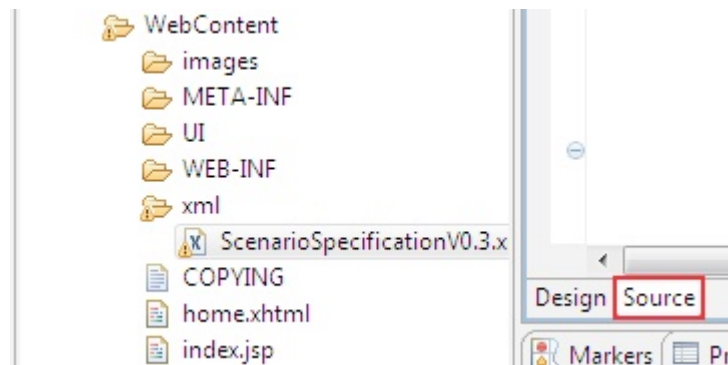
## F. Provar *Domestica* amb dispositius reals

Durant el desenvolupament de *Domestica*, l'aplicació no ha pogut realitzar, en algunes ocasions, la connexió amb el servidor Servlet i per tant, algunes línies de codi han segut comentades per tal de facilitar el seu desenvolupament. Si es vol provar l'aplicació en una situació real, sols tindrem que descomentar les línies 175 i 256 del fitxer `/src/beans/DevicesPage.java` que criden a la funció `requestHTTP` que és l'encarregada de realitzar les peticions HTTP.

```
_requestHTTP(DeviceType, Action);
```

## G. Modificar Fitxer XML

El fitxer XML es troba al directori WEBContent/xml. Podrem modificar-lo fent doble clic sobre el seu nom i seleccionant la pestanya "Source".



*Figura G. 1: Modificar XML*

En canvi, si el que volem es utilitzar un altre fitxer XML, cliquem amb el botó secundari sobre la carpeta "xml" -> Import -> File System i seleccionem l'XML.

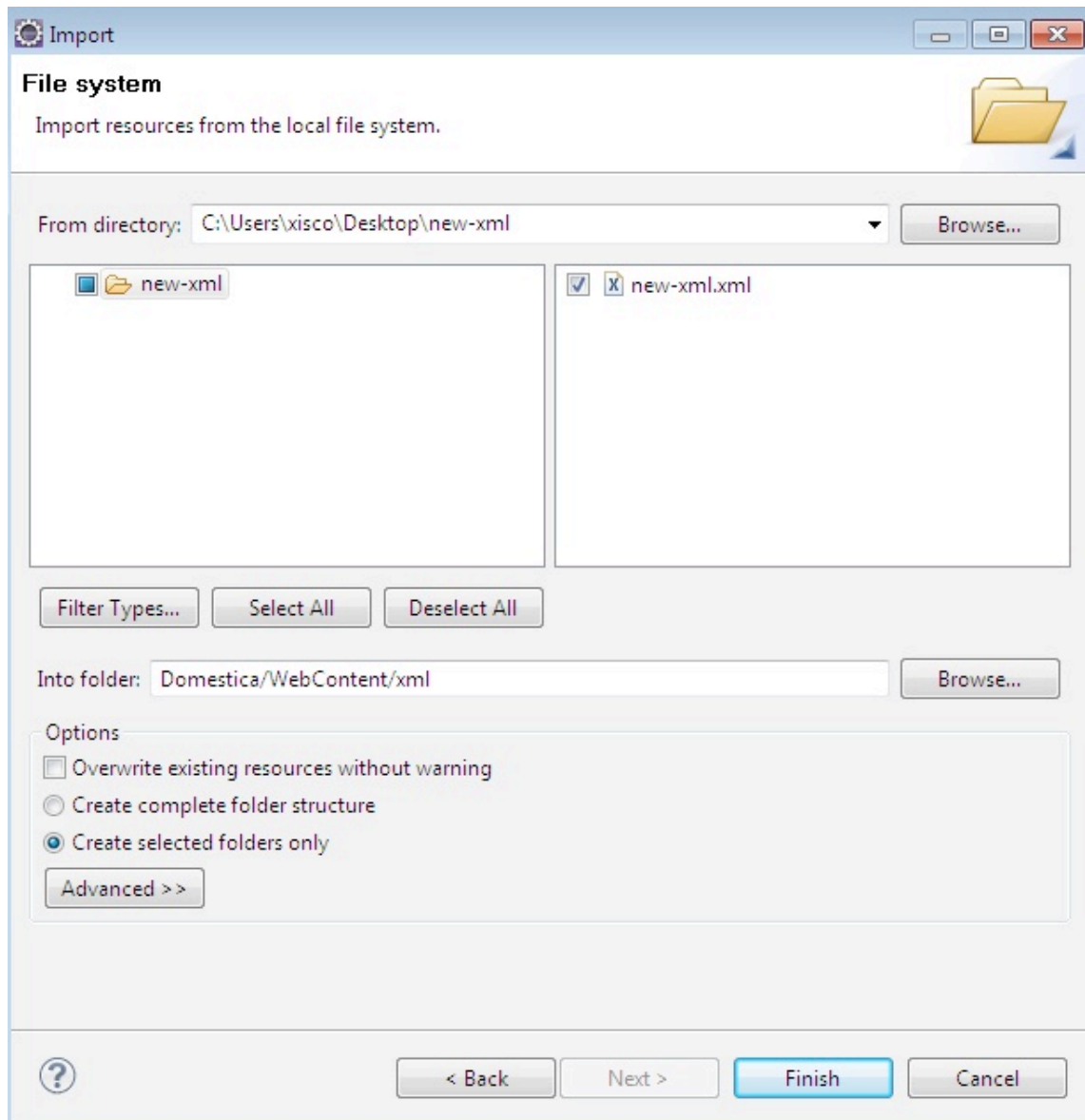


Figura G. 2: Importar XML

Per últim, sols tindrem que modificar la línia 64 del fitxer  
/src/beans/LoadData.java

```
String filepath =
context.getRealPath("xml/ScenarioSpecificationV0.3.xml");
```

**Exemple:**

Si el nou Xml s'anomena *new-xml.xml* quedaria així:

```
String filepath = context.getRealPath("xml/new-xml.xml");
```

# H. Connexió amb el Servlet

```
public void _requestHTTP(String DeviceType, String Action) {
    // Method that interact with the devices
    Map session = FacesContext.getCurrentInstance().getExternalContext()
        .getSessionMap();

    String server = (String) session.get("server");
    String requestUrl = "http://" + server + "/muteServlet/" +
        "?DeviceID=" + currentDevice.getId() + "&DeviceType=" + DeviceType
        + "&Action=" + Action;

    try {
        URL url = new URL(requestUrl.toString());
        BufferedReader in = new BufferedReader(new
InputStreamReader(url
        .openStream()));

        String state = in.readLine();
        in.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Aquest mètode es troba dins de la classe DevicePage i es l'encarregat de connectar amb el servlet per poder interactuar amb els dispositius. El mètode rep dos valors. El primer és el tipus de dispositiu:

- es.upv.pros.tunnel.interfaces.IActuatorSwitch
- es.upv.pros.tunnel.interfaces.IActuatorPulse
- es.upv.pros.tunnel.interfaces.IActuatorVerticalMovement
- es.upv.pros.tunnel.interfaces.IActuatorPercentualDimmer
- es.upv.pros.tunnel.interfaces.IActuatorAngularDimmer
- es.upv.pros.tunnel.interfaces.IActuatorHexDimmer
- es.upv.pros.tunnel.interfaces.ISensor

i el segon l'acció a realitzar sobre eixe dispositiu:

- es.upv.pros.tunnel.interfaces.IActuatorSwitch
  - switchOn
  - switchOff
- es.upv.pros.tunnel.interfaces.IActuatorPulse
  - pulse
- es.upv.pros.tunnel.interfaces.IActuatorVerticalMovement
  - moveUp
  - moveDown
  - stop
- es.upv.pros.tunnel.interfaces.IActuatorPercentualDimmer
  - setPercentualValue

- on
- off
- es.upv.pros.tunnel.interfaces.IActuatorAngularDimmer
  - setAngularValue
- es.upv.pros.tunnel.interfaces.IActuatorHexDimmer
  - setHexValue
- es.upv.pros.tunnel.interfaces.ISensor
  - getState

# I. Glossari De Termes

## JavaEE

JavaEE ( de l'anglès Java Platform, Enterprise ) és una plataforma de programació per desenvolupar i executar programari escrit amb el llenguatge Java amb una arquitectura distribuïda amb nivells, basada en components de programari, tot plegat executant-se en un servidor d'aplicacions.

## XUL

XUL ( de l'anglès XML-based User-interface Language ) és un llenguatge basat en XML per la interfície d'usuari.

XUL només proporciona la part gràfica, per tant, s'ha de fer servir altres llenguatges de programació per al funcionament d'un programa.

## JavaBean

Els JavaBeans són un model de components creats per al desenvolupament d'aplicacions en Java.

L'especificació de JavaBeans de Sun Microsystems els defineix com "components de programari reutilitzables que es puguin manipular visualment en una eina de desenvolupament".

Per a funcionar com una classe JavaBean, una objecte classe ha d'obeir certes convencions sobre nomenclatura de mètodes, construcció, i comportament. Aquestes convencions permeten tenir eines que puguin utilitzar, reutilitzar, substituir, i connectar JavaBeans.

Les convencions requerides són:

- La classe ha de ser serializable (capaç de salvar persistentment i de restablir el seu estat).
- Ha de tenir un constructor sense arguments.
- Les seves propietats han de ser accessibles mitjançant mètodes get i set que segueixen una convenció de nomenclatura estàndard.
- Ha de contenir determinats mètodes de maneig d'esdeveniments.



## Servlet

Un Servlet és un programa que s'executa en un servidor o contenidor JEE especialment dissenyat per generar pàgines web de forma dinàmica a partir dels paràmetres de petició que envia el navegador web.

La seua informació s'emmagatzema en el fitxer web.xml.

Un exemple pot ser Tomcat.

## XML

El llenguatge de marques extensible (de l'anglès eXtensible Markup Language) és un llenguatge extensible d'etiquetes desenvolupat per el "World Wide Web Consortium" (W3C) que permet definir la gramàtica de llenguatges específics.

Per tant, XML no és realment un llenguatge en particular, sinó una manera de definir llenguatges per a diferents necessitats.

XML és una tecnologia relativament senzilla que té al seu voltant altres que la complementen i la fan notablement més extensa, a més de proporcionar-li unes possibilitats molt més grans. A l'actualitat té un paper molt important, ja que permet la compatibilitat entre sistemes, permetent de compartir informació d'una manera segura, fiable i fàcil.

Es pot utilitzar en una gran varietat d'aplicacions com poden ser les bases de dades, editors de text, fulles de càlcul, etc.

## API

Una Interfície de Programació d'Aplicacions (de l'anglès Application Programming Interface) és un conjunt de declaracions que defineix el contracte d'un component informàtic amb qui farà ús dels seus serveis.

Al moment de construir un sistema informàtic o llibreria de programació, per donar suport a les invocacions a serveis fetes per un altre programa, cal oferir una API, tant als programes externs (que podran usar els serveis oferts), com al programador (que disposa del manual indispensable per poder treure el màxim suc del component que ha adquirit).

Sovint, una API és una part del Kit de Desenvolupament d'una Aplicació (SDK). L'API en si mateixa és abstracta en tant que especifica una interfície i no està lligada als detalls de la implementació. Quan un programari proveeix la funcionalitat descrita per una API, llavors diem que és una implementació de l'API.

En aquesta abstracció apareixen la llista de variables públiques, funcions i

procediments (o mètodes, si parlem de programació orientada a objectes), que el component informàtic ofereix.

## JSP

JSP (JavaServer Pages) és una tecnologia que permet als desenvolupadors de pàgines web, generar respostes dinàmicament a peticions HTTP. La tecnologia permet que codi Java i certes accions predefinides siguin incrustades en un context estàtic.

La sintaxi de JSP incorpora etiquetes XML addicionals, anomenats accions de JSP, per ser usats per invocar altres funcions. Addicionalment, la tecnologia permet la creació de llibreries d'etiquetes que actuen com extensions de l'estàndard d'etiquetes HTML o XML. Les llibreries d'etiquetes aporten una forma multiplataforma d'ampliar les capacitats d'un servidor web.

## MVC

MVC ( de l'anglès Model-Vista-Controlador) és un patró de disseny per al desenvolupament de programari que separa el model de dades, la interfície usuari i la lògica de control. El patró MVC es veu freqüentment en aplicacions web, on es pot visualitzar una pàgina HTML i el codi que proveeix de dades dinàmiques a la pàgina, el controlador és el sistema de gestió de bases de dades i el model és el model de dades.

## Ajax

Ajax ( de l'anglès Asynchronous Javascript And Xml ), és un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina.

Ajax és asíncron en tant que les dades addicionals són demanades i carregades en un segon pla, sense interferir en la presentació i el comportament de la pàgina. Habitualment les funcions d'Ajax es criden des del llenguatge JavaScript. Les dades són adquirides usant l'objecte XMLHttpRequest, que està disponible als llenguatges de Script que corren en navegadors moderns, o també usant Remote Scripting en navegador que no suportin XMLHttpRequest. En qualsevol cas, no és necessari que el contingut asíncron sigui formatat en XML.

# J. Bibliografia

- **Beginning JSP, JSF, and Tomcat web development : from novice to professional**

Zambon, Giulio

Berkeley : Apress , 2007

ISBN: 1590599047

9781590599044

- **Primefaces User's Guide**

Prime Technology, 2010

<http://www.primefaces.org/documentation.html>

- **Apache Tomcat 6**

Chopra, Vivek

Madrid : Anaya Multimèdia , 2007

ISBN: 9788441523777

- **Professional Eclipse 3 for Java developers**

Daum, Berthold

Chichester : John Wiley & Sons , 2005

ISBN: 0470020059

- **Informacion y Historia de la Domòtica y el Hogar Digital**

Casa Domo

<http://www.casadomo.com/>

- **JSF 2.0 + Ajax hello world example**

Mkyong

<http://www.mkyong.com/jsf2/jsf-2-0-ajax-hello-world-example/>

- **JSF Hello World Example on Eclipse and Tomcat**

Myhomepageindia.com

<http://www.myhomepageindia.com/index.php/2009/05/04/jsf-hello-world-example-on-eclipse-and-tomcat.html>

- **JSF Tags**

Roseindia.net

<http://www.roseindia.net/jsf/jsftags.shtml>

- **Introducción a JSF Java Server Faces**

Cristóbal Gonzalez Almirón

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=IntroduccionJSFJava>

