Memòria TFC

Aplicació Web per a l'obtenció d'SNPs de rellevància

Jesús Maria López Martí

Índex

Introducció....3
Objectius....5
Planificació....6
Especificacions....7
Cas d'ús creació d'usuari:....8
Cas d'ús connexió d'usuari:....9
Cas d'ús creació de projecte:....10
Cas d'ús Selecció de Seqüències:....11
Cas d'ús Creació D'Alineament:....12
Cas d'ús Anotació D'Alineament:....13
Cas d'ús Baixar dades....14
Implementació....15
Persistència....16
Interfície gràfica – JSFs (myfaces)....22
'State of Art'....31
Conclusions....33
Apèndixs....35
Bibliografia....36

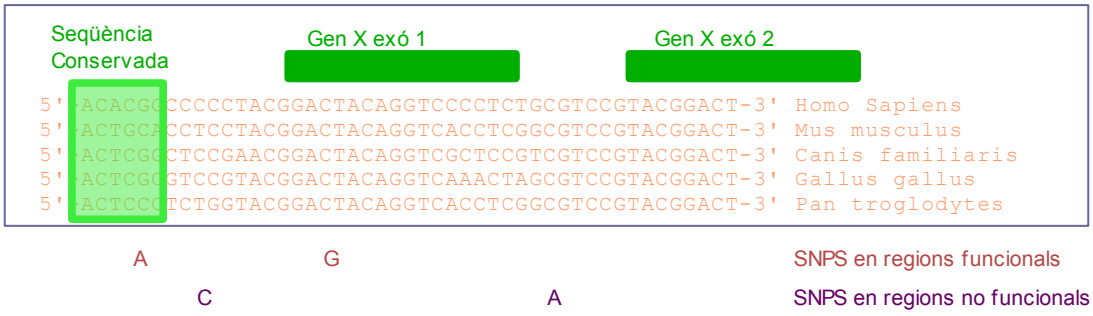
Introducció

El principal objectiu del treball de fi de carrera es familiaritzar-se amb les tecnologies J2EE i alhora desenvolupar una aplicació que en faci ús. L'aplicació en que he centrat els meus esforços te a veure amb el meu àmbit de treball: la genètica i el càncer. Més especificament es centra en l'estudi dels canvis que es produeixen a petita escala en el genoma i que poden estar associat a l'aparició i progressió d'una determinada enfermetat. Un tipus de canvis són els SNPs (Single Nucletide Polymorphism) i en el sentit més restrictiu del terme fan referència a la presència en el genoma de llocs polimòrfics, és a dir, que poden tenir dues bases nucleotídiques (A, C, T o G) en una posició determinada. Es diferencien de les mutacions en la freqüència en que tenen lloc a les poblacions (major de 5%).

Els SNPs poden trobar-se en qualsevol lloc al llarg del genoma, però principalment ens interessen únicament aquells SNPs que s'ubiquen en les regions funcionals. Aquestes són els promotors dels gens, els gens pròpiament dits i altres regions reguladores. Un canvi que es trobi present en aquestes regions pot afectar a la expressió d'una proteína i per tant a afectar-ne la seva funció.

Moltes de les regions funcionals están caracteritzades ja que es coneixen la major part dels gens. De totes maneres, determinar la presència de sequències reguladores no és una tasca tan fàcil ja que solen ser molt més curte i degenerades: admente més variabilitat. En aquest sentit, l'alineament de la seqüència problema amb la d'altres organisme pot aportar-nos molta informació i posar-nos de relleu aquelles regions o zones que estàn més conservades. La conservació de les zones ens indica que al

llarg de l'evolució la pressió selectiva perquè aquests romanguin constants ha estat important ja que juguen un paper important en l'expressió d'una proteïna o en la regulació del genoma.



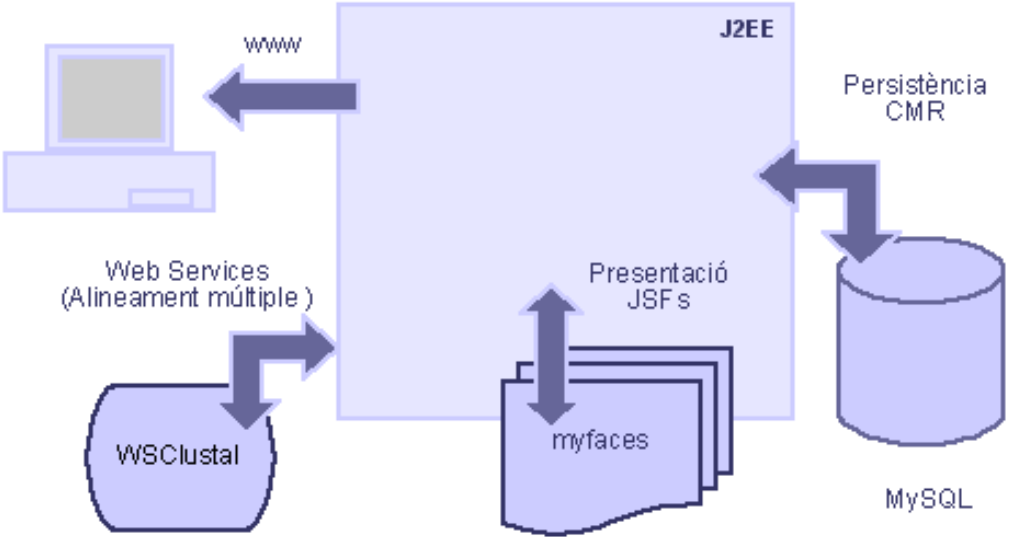
Per tant, per poder detectar en una determinada regió del genoma aquells SNPs que tinguin un possible impacte en el desenvolupament d'una enfermetat cal emprar seqüències d'altres espècies allunyades evolutivament de l'Homo sapiens. Existein molts mètodes per realitzar alineaments múltiples però en tot cas és un procés que té un alt cost computacional. En alguns casos a més, cal assegurar-se manulament que l'alineament obtingut és correcte.

Objectius

L'objectiu principal del projecte és doncs obtenir una eina amb una interfície web que ens permeti fàcilment obtenir la màxima quantitat d'informació d'una alineament múltiple. Per realitzar l'alineament múltiple i obtenir les seqüències es farà ús de web services de l'European Bioinformatics Institute (EBI) el qual aporta alguns exemples que es poden adaptar a les nostres necessitats.

Per altra banda, cal obtenir la informació de caire biològic de les seqüències: a quin gen pertany, cromosoma, quins SNPs té,... i per això es necessari fer ús o bé de bases de dades locals o bé accedir a les bases de dades que trobem disponibles via web.

Per últim la metodologia que es vol emprar per crear la interfície web és els JSFs ja que permeten de forma fàcil diseñar pantalles reutilitzant component que ja estiguin creats i a més permeten de forma molt còmoda internacionalitzar la nostra aplicació. En aquest projecte es fa ús de les llibreries myfaces per implementar alguns dels objectes més vistosos.



Planificació

En la primera PAC ens vam centrar en la creació d'un pla de treball que consideràvem adequat per dur a terme el projecte. El principal problema de realitzar una planificació d'aquest tipus radica en el desconeixement total de les tecnologies que s'utilitzen i per tant es cau en el parany d'estimar a la curta el nombre d'hores que seran necessàries per familiaritzar-se amb les eines. En el meu cas aquest ha estat un factor clau juntament amb els problemes de configuració i m'ha obligat a dedicar molt més temps del planificat per al desenvolupament de l'aplicació. A pesar d'això, m'hagués fet falta encara molt més temps per finalitzar les principals funcionalitats del projecte.

Per tant, una de les fases que m'ha faltat afegir en la planificació és la familiarització amb exemples de la problemàtica que pot sorgir al desenvolupar un projecte com el meu. De fet aquesta fase considero que és molt important i que només amb la lectura i uns pocs exemples bàsics no és suficient per tal de poder configurar les eines adequadament.

Tot i que la corba d'aprenentatge inicialment ha estat molt lenta m'he donat compte que els darrers dies de desenvolupament les hores que he invertit han estat molt més profitoses. De totes maneres a pesar que ho vaig considerar en la preparació de la planificació hagués estat necessari fer una ulladada més detallada a les eines per tal de valorar el temps adequat per utilitzar-les. Per tant, la planificació és del tot errònea i en cap cas durant la implementació he pogut guiar-me d'ella.

Especificacions

Al contrari que la planificació les especificacions que vaig establir en les PACs anteriors m'han servit d'orientació per no perdren en cap moment els objectius principas de cadascuna de les funcionalitats a implementar.

L'única modificació en els casos d'ús necessària és la connexió al sistema. La principal raó de la introducció d'aquest nou cas d'ús radica en el fet que el procés de creació d'un alineament múltiple és molt lent. Per tant, per tal de poder mantenir la informació entre sessions diferents cal permetre l'opció a l'usuari d'identificar-se al sistema per poder recuperar dades d'altres sessions.

L'organització de les dades generades en projectes facilita per tant el tractament i la recuperació d'aquestes. Un usuari determinat pot disposar de varis projecte i en cada projecte de varis alineaments en funció de les seves necessitats. D'aquesta manera facilitem l'ún de l'eina a l'usuari.

Si l'usuari anteriorment no s'havia connectat mai es permet crear un usuari indicant a la vegada la clau d'accés. Posteriorment a la creació l'usuari haurà de crear el primer projecte per tal d'anar guardant en aquests les dades. Si es necessari crear altres projectes la funcionalitat està disponible desde el menú principal ('Projects') visible en tot moment.

Amb les modificacions introduïdes els casos d'ús a implementar són els següents:

- Creació d'Usuari
- Connexió Usuari
- Creació d'un Projecte
- Selecció de Seqüències
- Creació d'Alineament
- Anotació d'Alineament
- Baixar dades

Cas d'ús creació d'usuari:

Mitjançant aquest cas d'ús es permet crear un usuari introduïnt un nom d'usuari i una clau d'accés. Mitjançant aquesta clau d'accés es podrà conectar posteriorment al sistema. El sistema validarà que no existeix cap usuari amb l'identificador introduït i en cas contrari es notificarà de tal incidència a l'usuari. Per altra banada també es validarà que la clau d'accés no estigui buida i compleixi certs requisits de seguretat. Si no es produeix cap error en la creació de l'usuari es mostrarà la pantalla de creació de projectes necessària per a començar a tractar les dades.

Precondició: no existeix l'usuari en la base de dades

Postcondició: s'ha creat un usuari amb un identificador i clau d'accés correctes però sense cap projecte assignat.

Diagrama de seqüència:

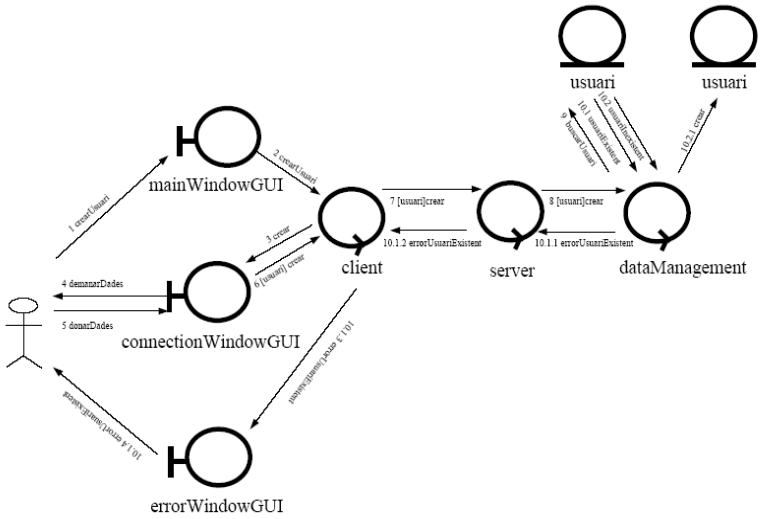
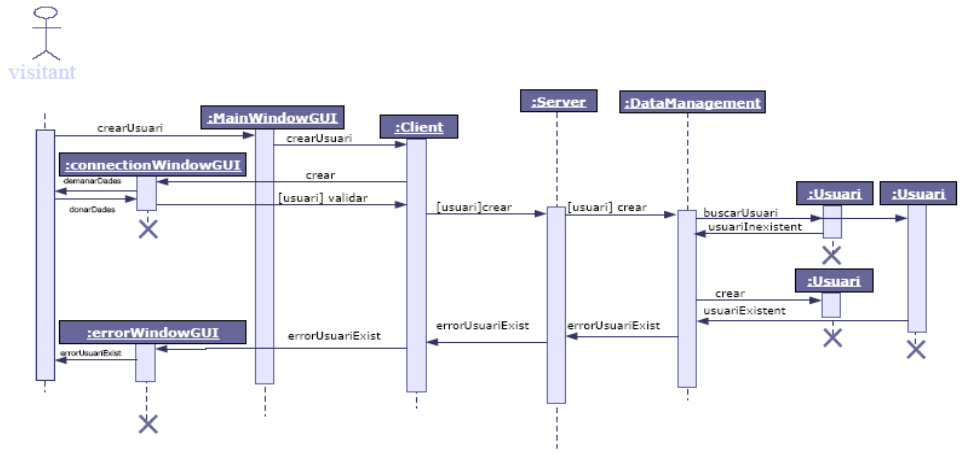


Diagrama de col.laboració:



Cas d'ús connexió d'usuari:

Mitjançant aquest cas d'ús l'usuari es pot connectar al sistema indicant el seu nom identificador i la clau d'accés. Si l'usuari existeix i a més la clau d'accés és correcta es mostra les dades dels projectes. En cas contrari s'informa mitjançant un error que la validació ha estat incorrecta mostrant de nou la pantalla de connexió.

Precondició: l'usuari ja ha estat creat a la base de dades.
Postcondició: les dades han estat validades i es mostra els projectes que l'usuari ha creat anteriorment.

Diagrama de seqüència:

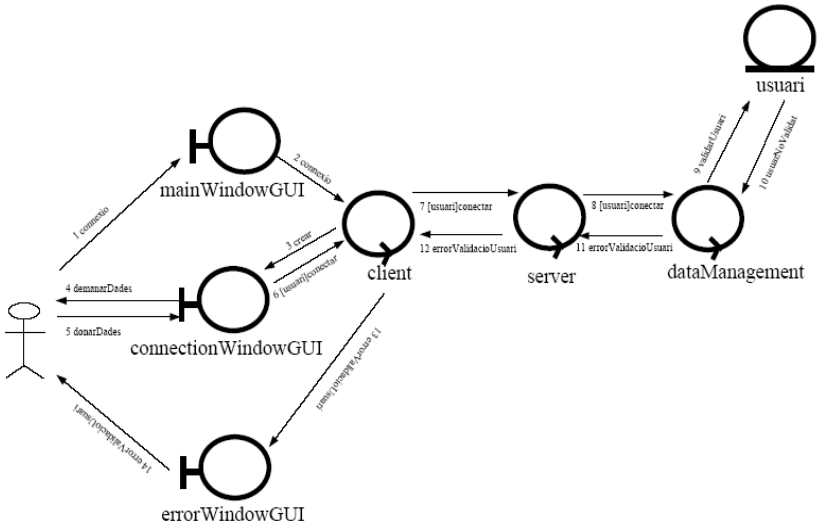
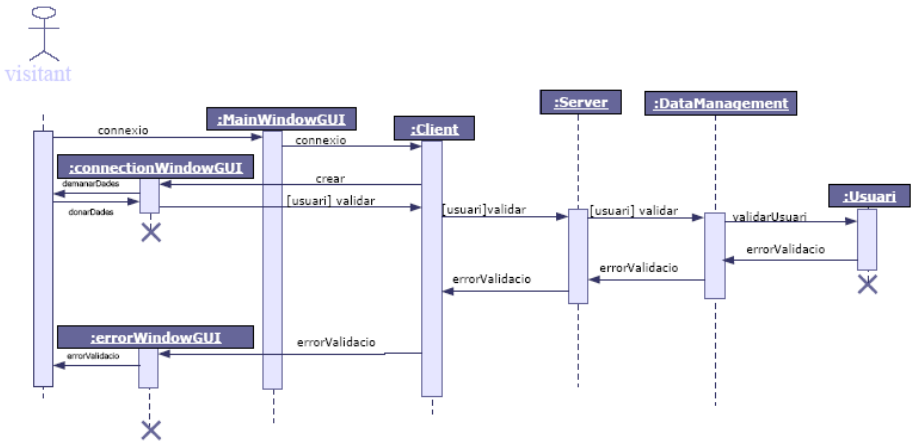


Diagrama de col.laboració:



Cas d'ús creació de projecte:

Mitjançant aquest cas d'ús l'usuari crearà un projecte nou seleccionant els organismes o espècies en els que estarà centrat. Cal indicar un nom al projecte o en cas contrari el sistema n' assignarà un per defecte. Si el nom que s'ha assignat al projecte coincideix amb el nom d'un projecte anterior s'advertirà a l'usuari de tal circunstància. Aquest projecte creat passarà a ser l'objecte de treball de l'usuari.

Precondició: no existeix un projecte amb el nom indicat.

Postcondició: s'ha creat en la base de dades un projecte amb el nom i les condicions especificades.

Diagrama de seqüència:

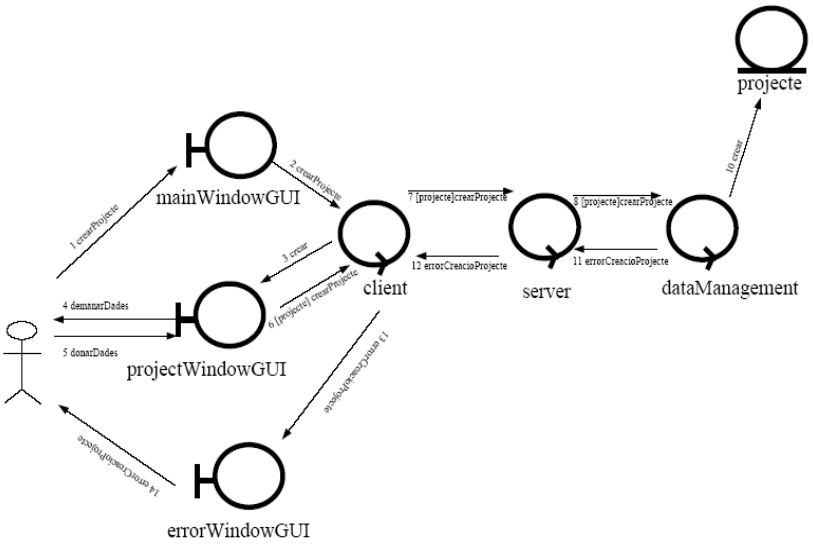
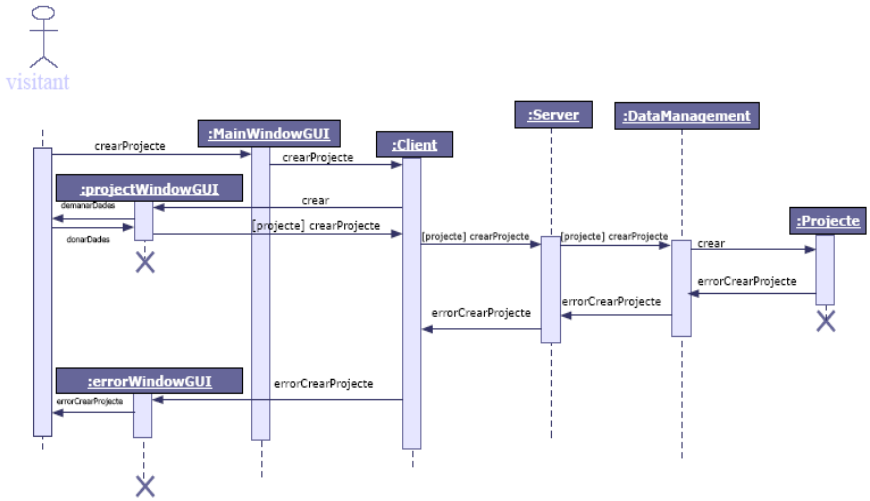


Diagrama de col.laboració:



Cas d'ús Selecció de Seqüències:

Mitjançant aquest cas d'ús es permetrà a l'usuari obtenir les seqüències de partida amb les que realitzaran els alineaments múltiples. L'usuari pot seleccionar-les mitjançant la seva función indicant un terme GO, mitjançant semblança pegant una seqüència en format FASTA o bé mitjançant ortologia u orígen comú (Homolgene). El resultat final és un llistat de les seqüències que compleixen els requisits per a què l'usuari pugui seleccionar aquelles que l'interessi.

Precondició: existeix un projecte de treball.

Postcondició: s'ha afegit en el project de treball les seqüències seleccionades per l'usuari.

Els tres subcasos d'ús son molt semblant i tan sols s'indiquen el diagrama de seqüència i de col.laboració per un d'ells.

Diagrama de seqüència:

Mitjançant introducció terme GO:

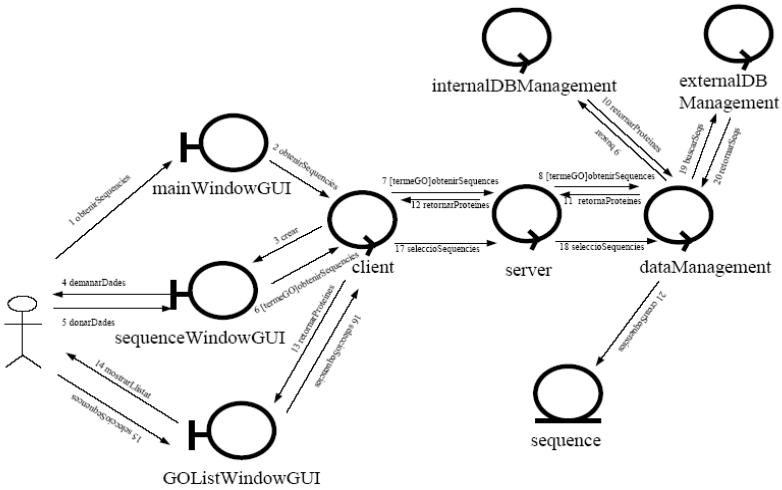
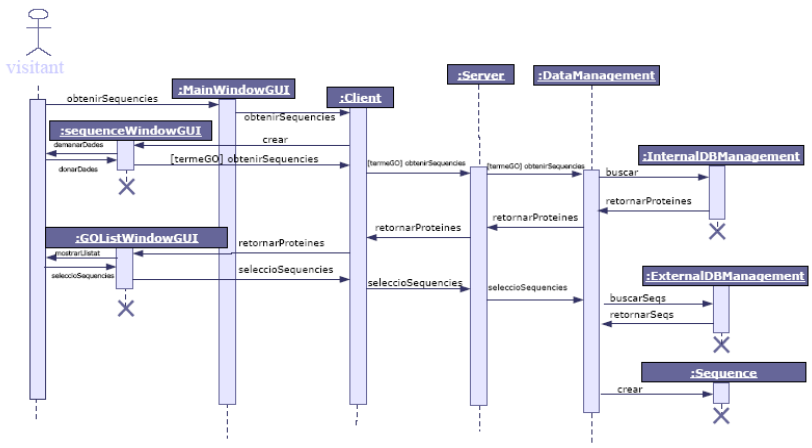


Diagrama de col.laboració:

Mitjançant codi GO:



Cas d'ús Creació D'Alineament:

Mitjançant aquest cas d'ús es permetrà a l'usuari crear a partir de les següències seleccionades en el cas d'ús anterior crear un alineament múltiple. Cal evidentment haber seleccionat més de dues seqüències ja que en cas contrari no es pot dur a terme l'alineament. El procés d'alineament es duu a terme mitjanánt web service i en cas de produir-se algun tipus d'error es notificarà a l'usuari mitjançant un missatge d'error de tal circunstància.

Precondició: existeix un projecte de treball en el qual hi ha més de dues seqüències seleccionades.

Postcondició: s'ha creat un alineament múltiple i s'ha afegit al projecte de treball de l'usuari.

Diagrama de seqüència:

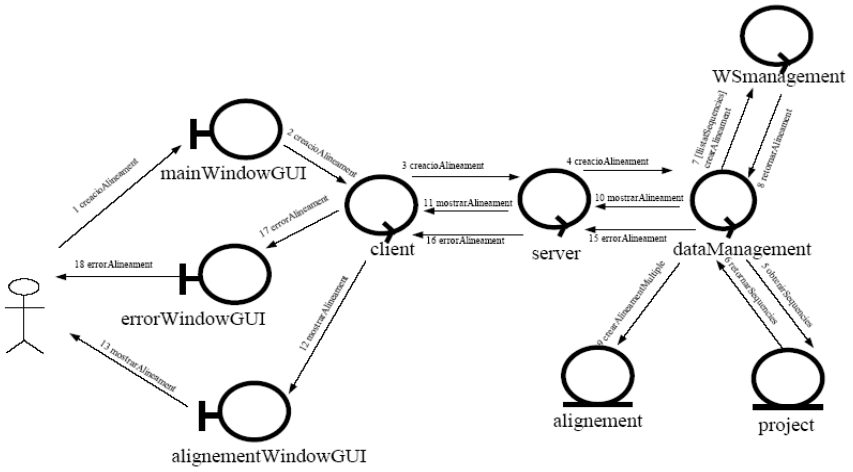
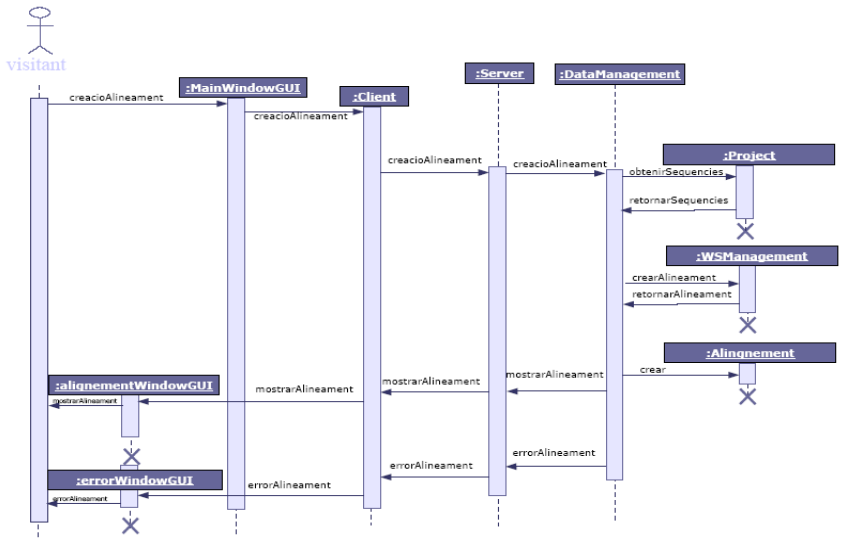


Diagrama de col.laboració:



Cas d'ús Anotació D'Alineament:

Mitjançant aquest cas d'ús es permetrà a l'usuari seleccionar el tipus d'SNP del qual es vol obtenir informació. Cal haver creat un alineament múltiple en el qual hi ha d'haver una seqüència d'Homo sapiens. Es pot filtrar els SNPs per tipus: en zona no codificador, sinònim o no-sinònim. També es pot filtra pel tipus de validació de l'SNP i per la seva freqüencia en la població europea. En el cas de no obtenir cap SNP que satisfaci les condicions s'informarà a l'usuari de tal circunstància.

Precondició: existeix un projecte de treball en el qual hi ha un alineament múltiple amb una seqüència d'Homo sapiens.

Postcondició: s'ha creat una alineament anotant tots els SNPs que compleixen els requisits indicats per l'usuari.

Diagrama de seqüència:

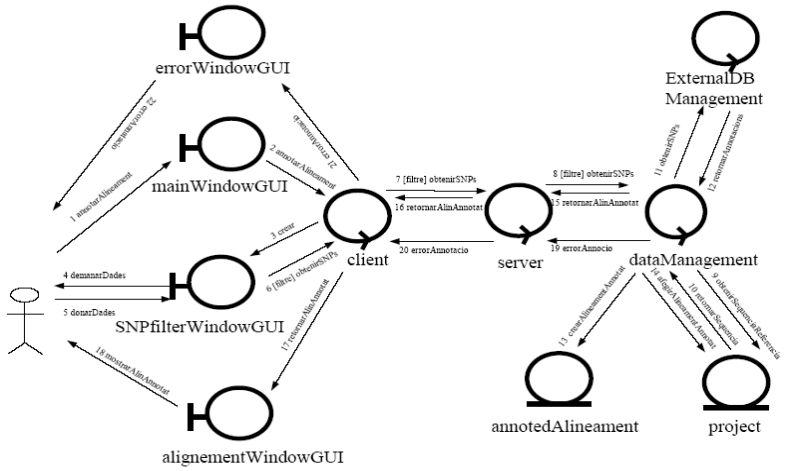
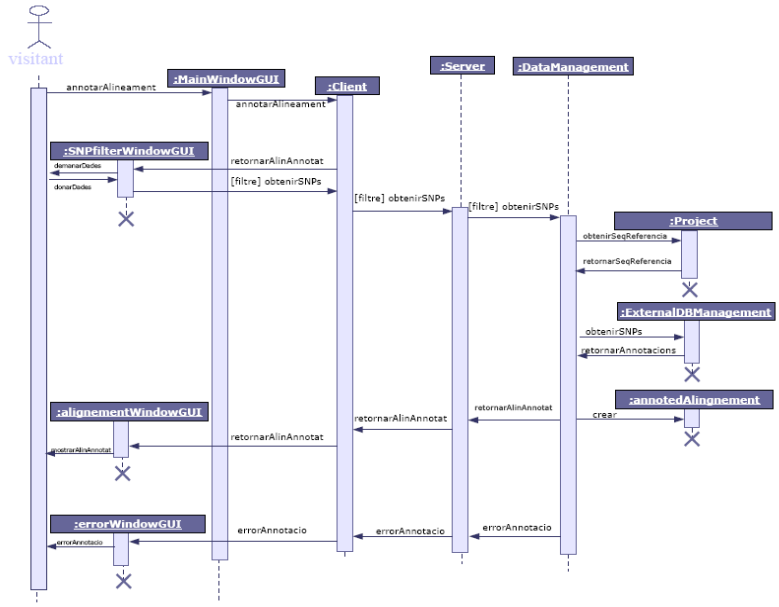


Diagrama de seqüència:



Cas d'ús Baixar dades

Mitjançant aquest cas d'ús es permetrà a l'usuari baixar-se les dades que fins al momen ha processat en el projecte de treball. Es crearà un fitxer per cadascún dels tipus d'elements distintius: seqüències, alineament i alineaments anotats. Si el projecte està buit l'opció estarà deshabilitada. Es notificarà a l'usuari de qualsevol error que es produixi en le procés.

Precondició: existeix un projecte en el qual hi ha almenys unes seqüències seleccionades.

Postcondició: s'ha creat uns fitxers amb les dades del projecte que l'usuari podrà seleccionar i baixar-se desde la web.

Diagrama de seqüència:

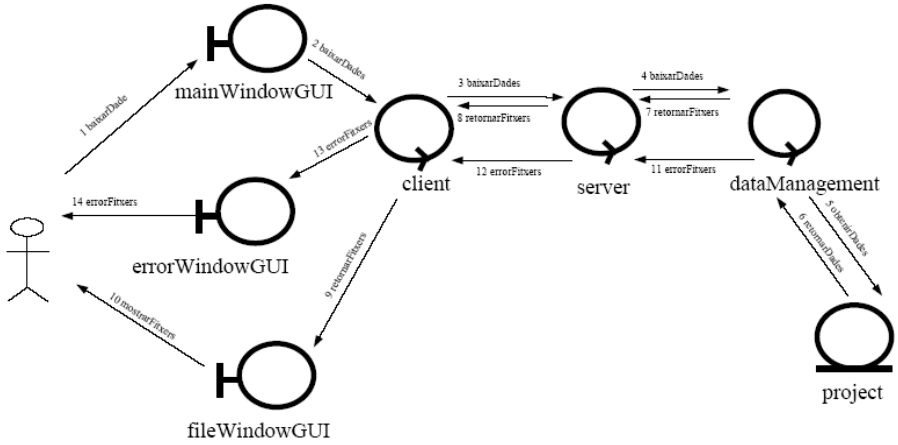
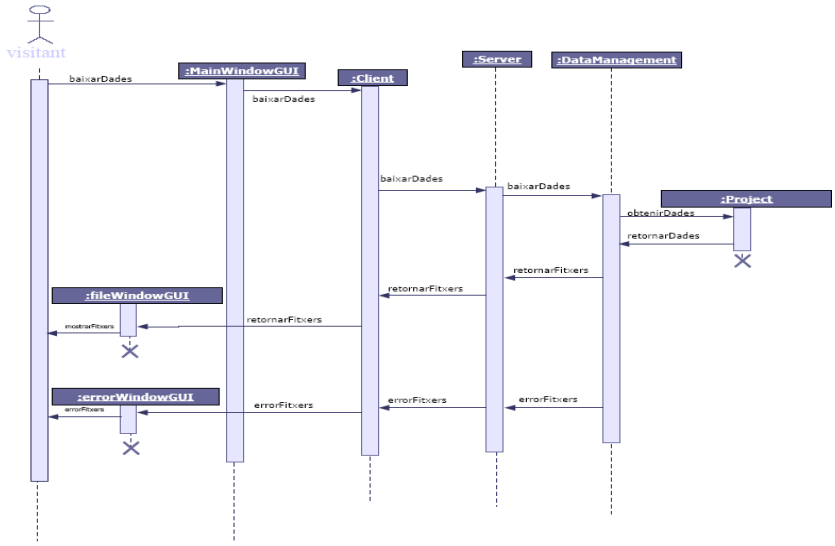


Diagrama de col.laboració:



Implementació

Abans de poder començar a implementar el projecte cal configurar les eines indespensables con l'entorn de desenvolupament (eclipse), el gestor de base de dades (MySQL) y el servidor d'aplicacions (Jboss). A més per comoditat y claredat s'ha instal.lat el plugin que permet gestionar desde eclipse el Jboss tal com descriu el tutorial de pràctiques lliurat en el forum del laboratori de pràctiques. Aquest tutorial ha resultat indispensable no només per configurar eclipse sino per modificar els fitxers de configuració adients en el Jboss. De totes maneres, per alguns problemes en concret la web s'ha convertit en un recurs molt valuós del qual obtenir informació. A tall d'exemple problemes que han anat sorgint entre el Jboss i l'eclipse són la falta de sincronisme o l'incapacitat de poder desplegar aplicacions degut a una falta de memòria.

Un cop configurades les eines s'ha partit de la següent estructura de fitxers per tal de tenir en tot moment organitzades les dades.

/vo
/inc
/interfaces
/resources
/images
/ejb
/css
/client
/WEB-INF
/META-INF

Els directoris més importants per a la gestió de la persistència són el **/ejb** i **/interfaces**. En el primer d'ell s'hi troben els Beans necessaris per gestionar les dades que s'han d'emmagatzemar en la base de dades i en el segon s'hi ubica les interfaces.

El directori **/vo** conté els 'value objects' necessaris per tal de poder gestionar en remot les dades d'una forma entenedora. També s'utilitzen en local quan no es necessari que una

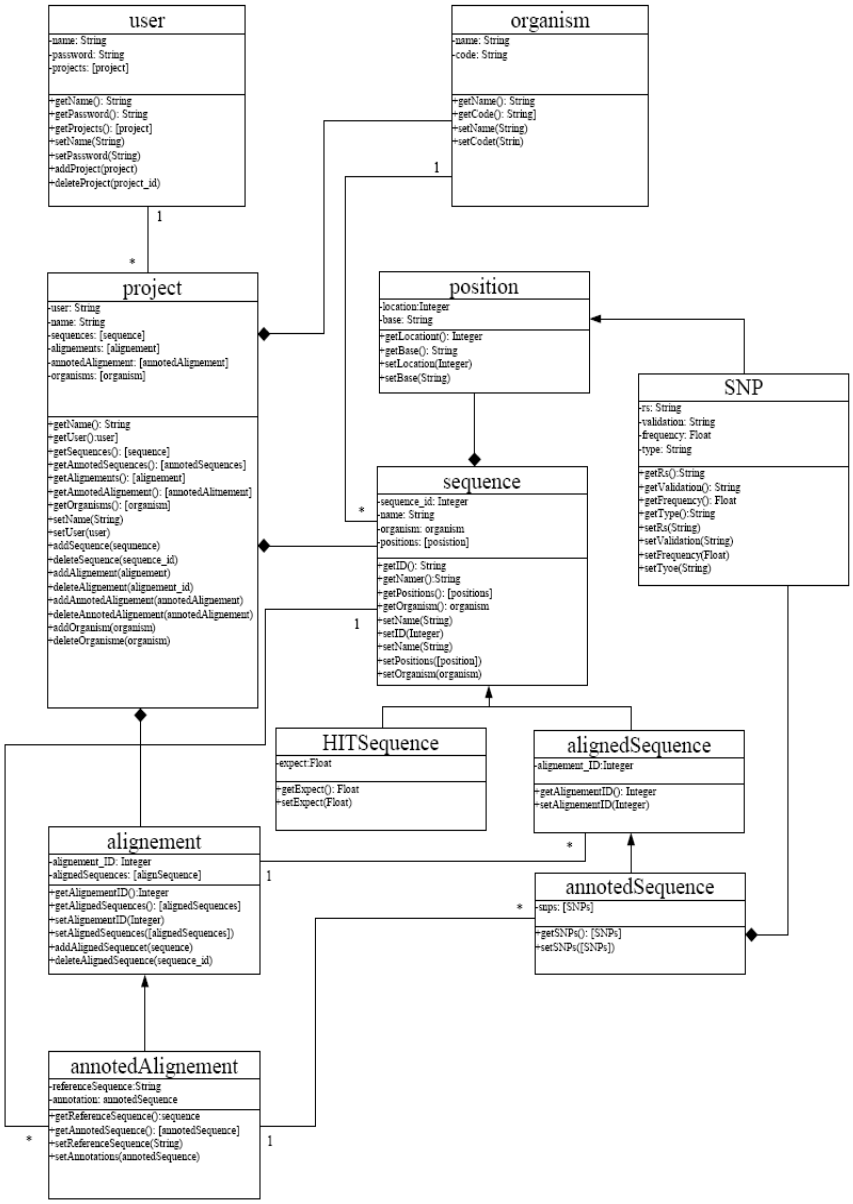
determinada entita hagi de ser persistent.

Per altra banda els directoris **/inc**, **/resources**, **/images** i **/css** contenen tota la informació necessària per tal de poder crear la interfície web mitjançant JSFs.

Per últim en el directori **/client** s'hi ubica un classe que fa ús dels diferents Beans i que té com a funció principal poder testejar tota la persistència.

Persistència

El diagrama de classe definitiu que s'ha utilitzat per a poder gestionar la persistència és el següent:



En el diagrama anterior cal destacar les dues jerarquies d'herència existents: Alineaments i Seqüències. La primera d'elles únicament conté dues classes mentre que la segona és més complexa i està formada per quatre classes. A nivell d'interfaces els Beans no presenten cap problema a excepció dels 'home' que han d'heredar forçosament de la *interface* **EJBLocalHome** o **EJBHome**. A continuació s'il.lustra el cas dels alineaments:

```
package interfaces;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBLocalObject;
import javax.ejb.RemoveException;

public interface AlignementLocal extends EJBLocalObject {
        public Integer getAlignementID();
        public void setAlignementID(Integer alignementID);
        public String getName();
        public void setName(String name);
        public void addAlignedSequence(AlignedSequenceLocal alignedSequence) throws  CreateException;
        public void deleteAlignedSequence(AlignedSequenceLocal alignedSequence) throws RemoveException;
        public Collection getAlignedSequences();
        public void setProject (ProjectLocal project);
        public ProjectLocal getProject();
}
```

```
package interfaces;

public interface AnnotedAlignementLocal extends AlignementLocal {
        public void setReferenceSequence(SequenceLocal referenceSequence);
        public SequenceLocal getReferenceSequence();
              public void setAnnotedSequence(AnnotedSequenceLocal referenceSequence);
        public AnnotedSequenceLocal getAnnotedSequence();
}
```

Per als Beans (**/ejb**) trobem la mateixa situació que per a les interfaces i la gestió de l'herència és l'habitual:

```
package ejb;

import interfaces.*;

import java.rmi.RemoteException;
import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.ejb.EntityBean;
import javax.ejb.EntityContext;
import javax.ejb.RemoveException;

public abstract class AlignementBean implements EntityBean {
        protected EntityContext ctx;
        private static int NUM_ALIGNEMENTS;

        public Integer ejbCreate(String name) throws CreateException {
                NUM_ALIGNEMENTS++;
                setAlignementID(new Integer(NUM_ALIGNEMENTS));
                setName(name);
                return null;
        }

        public void ejbPostCreate(String name) {

        }

        // Persistent fields
        public abstract void setName(String name);
        public abstract String getName();
        public abstract void setAlignementID(Integer ID);
        public abstract Integer getAlignementID();
        public abstract void setProject (ProjectLocal project);
        public abstract ProjectLocal getProject();

        // relationships methods
        public abstract void setAlignedSequences(Collection alignedSequences);
        public abstract Collection getAlignedSequences();

        // business methods
        public void addAlignedSequence(AlignedSequenceLocal sequence) throws  CreateException {
                try {
                        Collection sequences = getAlignedSequences();
                        sequences.add(sequence);
                } catch(Exception ex) {
                        throw new EJBException(ex.getMessage());
                }
        }

        public void deleteAlignedSequence(AlignedSequenceLocal sequence) throws RemoveException {
                try {
                        Collection sequences = getAlignedSequences();
                        sequences.remove(sequence);
                } catch(Exception ex) {
                        throw new EJBException(ex.getMessage());
                }
        }

        public void setEntityContext(EntityContext ctx) throws EJBException,
                        RemoteEx
                // TODO Auto-generat
                this.ctx = ctx;
        }

        ...

}
```

```
package ejb;

import interfaces.*;
import javax.ejb.CreateException;

public abstract class AnnotedAlignementBean extends AlignementBean {

        public Integer ejbCreate(String name) throws CreateException {
                super.ejbCreate(name);
                return null;
        }

        public void ejbPostCreate(String name) {

        }

        // Persistent fields
        public abstract void setReferenceSequence(SequenceLocal referenceSequence);
        public abstract SequenceLocal getReferenceSequence();
        public abstract void setAnnotedSequence(AnnotedSequenceLocal annotedSequence);
        public abstract AnnotedSequenceLocal getAnnotedSequence();

}
```

Alguns dels objectes que es necessari guardar mantener una relació molts a molts o bé 1 a molts. Per tal de poder guardar les relacions que s'estableixen entre els diferents objectes cal fer ús del CMR (Container-Managed Relationships).

Arribats en aquest punt un fitxer que és clau i on indiquem les taules, els atributs i les relacions dels diferents objectes que formen part del nostre sistema és el *ejb-jar.xml*. Part del fitxer es mostra a continuació:

```
<?xml version="1.0"?>

<!DOCTYPE ejb-jar PUBLIC
"-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN"
"http://java.sun.com/dtd/ejb-jar_2_0.dtd">

<ejb-jar>
 <enterprise-beans>
         <entity>
           <ejb-name>Alignement</ejb-name>
           <local-home>interfaces.AlignementHomeLocal</local-home>
           <local>interfaces.AlignementLocal</local>

           <ejb-class>ejb.AlignementBean</ejb-class>
           <persistence-type>Container</persistence-type>
           <prim-key-class>java.lang.Integer</prim-key-class>
           <reentrant>False</reentrant>
           <cmp-version>2.x</cmp-version>
           <abstract-schema-name>Alignement</abstract-schema-name>
           <cmp-field><field-name>alignementID</field-name></cmp-field>
           <cmp-field>
                 <field-name>name</field-name>
           </cmp-field>
           <primkey-field>alignementID</primkey-field>
           <security-identity><use-caller-identity/></security-identity>
           <ejb-local-ref>
                 <ejb-ref-name>ejb/local/AlignedSequence</ejb-ref-name>
                 <ejb-ref-type>Entity</ejb-ref-type>
                 <local-home>interfaces.AlignedSequenceHomeLocal</local-home>
                 <local>interfaces.AlignedSequenceLocal</local>
                 <ejb-link>AlignedSequence</ejb-link>
           </ejb-local-ref>
           <query>
             <query-method>
                 <method-name>findByUserProject</method-name>
                           <method-params>
                                     <method-param>java.lang.String</method-param>
                           <method-param>java.lang.String</method-param>
                 </method-params>
             </query-method>


             <ejb-ql>
                 <![CDATA[SELECT OBJECT(a) FROM Alignement AS a WHERE a.project.name = ?2 AND a.project.user.name =?1 ]]>
             </ejb-ql>
           </query>

         ...
</enterprise-beans>

<relationships>
             <ejb-relation>
             <ejb-relation-name>User-Project</ejb-relation-name>
             <ejb-relationship-role>
                       <ejb-relationship-role-name>User-has-a-Project</ejb-relationship-role-name>
                       <multiplicity>One</multiplicity>
                       <relationship-role-source>
                                 <ejb-name>User</ejb-name>
                       </relationship-role-source>
                       <cmr-field>
                             <cmr-field-name>projects</cmr-field-name>
                       </cmr-field>
             </ejb-relationship-role>
             <ejb-relationship-role>
                       <ejb-relationship-role-name>Project-belongs-to-User</ejb-relationship-role-name>
                       <multiplicity>Many</multiplicity>
                       <cascade-delete></cascade-delete>
                       <relationship-role-source>
                                 <ejb-name>Project</ejb-name>
                       </relationship-role-source>
                       <cmr-field>
                             <cmr-field-name>user</cmr-field-name>
                       </cmr-field>
             </ejb-relationship-role>
             </ejb-relation>
</relationships>
</ejb-jar>
```

Atributs del Bean

Camp clau

Mètodes de búsqueda definits en el 'home'

paràmetres

SQL

Relacions

En el fitxer *ejb-jar.xml* indiquem el nom dels atributs així com el del camp clau de la taula que s'ha de crear per tal de fer persistent l'objecte:

```
<cmp-field>
    <field-name>name</field-name>
</cmp-field>
<primkey-field>alignementID</primkey-field>
```

En aquest cas estem definin un Alineament el qual estarà format per objectes del tipus seqüències i per tant cal indicar-ho de la següent forma:

```
<ejb-local-ref>
    <ejb-ref-name>ejb/local/AlignedSequence</ejb-ref-name>
    <ejb-ref-type>Entity</ejb-ref-type>
    <local-home>interfaces.AlignedSequenceHomeLocal</local-home>
    <local>interfaces.AlignedSequenceLocal</local>
    <ejb-link>AlignedSequence</ejb-link>
</ejb-local-ref>
```

A més, hem de afegir els detalls de la relació que s'estableix entre ambdós objectes dins de l'apartat <relationships>:

```
<ejb-relation-name>User-Project</ejb-relation-name>
      <ejb-relationship-role>
            <ejb-relationship-role-name>
                User-has-a-Project
            </ejb-relationship-role-name>
            <multiplicity>One</multiplicity>
            <relationship-role-source>
                  <ejb-name>User</ejb-name>
            </relationship-role-source>
            <cmr-field>
              <cmr-field-name>projects</cmr-field-name>
            </cmr-field>
      </ejb-relationship-role>
      <ejb-relationship-role>
            <ejb-relationship-role-name>
                    Project-belongs-to-User
            </ejb-relationship-role-name>
            <multiplicity>Many</multiplicity>
            <cascade-delete></cascade-delete>
            <relationship-role-source>
                  <ejb-name>Project</ejb-name>
            </relationship-role-source>
            <cmr-field>
              <cmr-field-name>user</cmr-field-name>
            </cmr-field>
      </ejb-relationship-role>
</ejb-relation>
```

Cal destacar a més que totes les funcions de búsqueda que hagin estat definides en els 'home' dels Bean cal definir-se en el fitxer de la següent forma:

```
<query>
        <query-method>
                <method-name>findByUserProject</method-name>
                        <method-params>
                                <method-param>
                                        java.lang.String
                                </method-param>
                        <method-param>java.lang.String</method-param>
                </method-params>
         </query-method>

        <ejb-ql>
                <![CDATA[SELECT OBJECT(a) FROM Alignement AS a WHERE a.project.name
        = ?2 AND a.project.user.name =?1 ]]>
         </ejb-ql>
</query>
```

```
package interfaces;

import java.util.Collection;

import javax.ejb.CreateException;
import javax.ejb.EJBLocalHome;
import javax.ejb.FinderException;

public interface AlignementHomeLocal extends EJBLocalHome {
        public AlignementLocal create(String name) throws CreateException;
        public AlignementLocal findByPrimaryKey(Integer alignementID) throws FinderException;
        public Collection findByUserProject(String user, String Project) throws FinderException;
        public AlignementLocal findByUserProjectName(String user, String project, String name) throws FinderException;
}
```

En funció del tipus de búsqueda que descrivim mitjançant SQL cal tenir molt present que en tot moment esten fent referència a objectes. Aquest fet marca de forma distintiva el tipus de seqüència quan volem accedir als atributs d'un objecte referenciat. Un exemple d'aquest tipus el trobem en la sentència SQL anterior:

```
SELECT OBJECT(a) FROM Alignement AS a WHERE a.project.name = ?2 AND
     a.project.user.name =?1
```

L'anterior sentència busca tots aquells projectes que tenen per nom ?2 (el segon paràmetre passat) i que a més pertanyen a l'usuari que té de nom ?1 (primer paràmetre).

Sino tenim en compte la mecànica de funcionament descrita

no es podrà produir el desplegament de l'aplicació.

Finalment comentar que el gestor de base de dades elegit no pot encara gestionar l'herència vertical i per tant per cadascún dels objectes persistents indistintament de l'herència es guardarà en una taula diferent. Així doncs tot i que **AlineamentAnnotat** hereti de Alineament cal crear una taula nova on es guardarà tan la informació de la superclasse com els nous atributs de la classe filla. Aquest fet pot afegir molta complicació alhora de recuperar tots els objectes **Alineament** siguin **Alineament** o **AlineamentAnnotat** que compleixin certes característiques. En aquest punt cal utilitzar l'ingeni i sobrescriure mètodes de buscar per tal que es realitzin búsquedes independents en la taula Alineament i la taula Alineament múltiple.

Tal com s'ha comentat anteriorment es va crear una classe per tal de comprobar el correcte funcionament de la persistència. A més, si és el cas es pot utilitzar per a inicialitzar la base de dades per realitzar proves de la interfície gràfica.

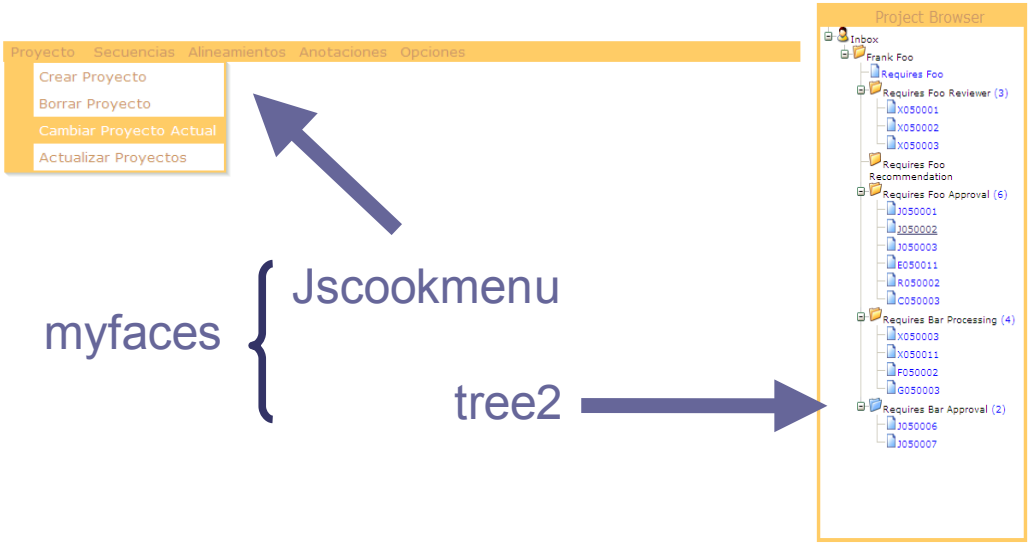
Interfície gràfica – JSFs (myfaces)

La interfície gràfica s'ha creat aprofitant la potència que aporten els Java Server Faces. De totes maneres, la corba d'aprenentatge és molt llarga i el procés de familiarització amb la tecnologia és lent. Si a més tenim en compte que la major part dels objecte, en le meu cas de les llibreries myfaces, no estàn estandaritzat l'esforç es multiplica ja que cal realitzar un aprenentatge diferent per a cadascún dels objectes. Aquest fet es producte de que moltes de les implementacion no són conseqüència d'un equip productiu sino que són més aviat un recull d'objectes que han implementat gent arreu del món.

Un altre aspecte que dificulta l'ús dels JSFs és que són poc flexibles, almenys els que he tingut oportunitat d'usar i la documentació que s'adjunta és poc clara i extensa. Els exemples a més són poc pràctics i en la majoria de casos cal tenir experiència en l'ús d'altres classes com **TreeNodeBase**.

A pesar de tot això, un cop s'ha assolit el nivell necessari el rendiment és prou alt i el resultat molt vistós i pràctic.

Dos dels components de myfaces que he utilitzat en la creació de l'interfície gràfica han estat els **Jscookmenu** i el **Tree2**:



Aquest dos components ja que s'utilitzen arreu s'han modificat per mostrar la informació adequada i s'han afegit en el directori **/inc** de l'aplicació. Per tal de poder mostrar les imatges de l'arbre **Tree2** cal afegir en el directori **/images** les icones a les quals es fa referència en el *include*.

Per altra banda els JSFs permeten d'una manera molt fàcil internacionalitzar l'aplicació, és a dir, traduir-la a qualsevol idioma. Per tal de fer això, s'ha creat un directori **/resources** en el qual s'ha afegit dos fitxers: *messages.properties* i *messages_es.properties*. Aquests fitxers tenen el següent format:

```
/resources/messages_es.properties
nav_Project              = Proyecto
nav_Sequences            = Secuencias
nav_Alignement                        =
Alineamientos
nav_Annotations            = Anotaciones
nav_Options              = Opciones
```

```
/resources/messages.properties
nav_Project              = Project
nav_Sequences            = Sequences
nav_Alignement                        =
Alignements
nav_Annotations            = Annotations
nav_Options              = Options
```

Per valer-nos d'aquest facilitat cal, en cadascuna de les pàgines jsf a crear, afegir mitjançant etiquetes inputLabel o outputLabel el missatge de la següent manera:

```
...
<td align='right'>
      <br>       
      <font color="#CC9966" bgcolor="#FFCC66" size="4">
            <h:outputText value="#{messages['label_User']}" />
      </font>
</td>
...
```

Llavors en funció de l'idioma del navegador sen's mostrarà un missatge en anglès o bé en castellà. Si es necessari afegir altres idiomes únicament cal crear el fitxer *messages_idioma.properties* i traduir les variables.

Per a poder utilitzar adequadament les llibreries myfaces amb el Jboss cal realitzar les següents modificacions:

- Borrar de qualservol directori del Jboss totes les llibreries myfaces. Per defecte aquestes venen instalades i poden provocar problemes d'incompatibilitat.
- Per altra banda cal afegir en la propia estructura del projecte i dins del directori /WEB-INF/lib totes les llibreries myfaces que el nostre projecte necessita.

Per altra banda cal crear una Session Bean que s'encarregui de totes les funcionalitats que la nostra aplicació dugui a terme. Aquest s'anomena UserManagementBean.

```
import java.util.Collection;
import javax.ejb.EJBObject;
import org.apache.myfaces.custom.tree2.TreeNode;
import org.apache.myfaces.custom.tree2.TreeNodeBase;
import java.rmi.RemoteException;
import vo.*;

public interface UserManagement extends EJBObject {
	public void addProject(ProjectDO project, UserDO user) throws  RemoteException;
	public void deleteProject(ProjectDO project, UserDO user) throws RemoteException;
	public Collection getProjects(UserDO user) throws RemoteException;
	public void addSequence(UserDO user, SequenceDO sequence, ProjectDO project) throws  RemoteException;
	public void deleteSequence(UserDO user, SequenceDO sequence, ProjectDO project) throws RemoteException;
	public Collection getSequences(UserDO user, ProjectDO project) throws RemoteException;
	public void addHITSequence(UserDO user, HITSequenceDO sequence, ProjectDO project) throws  RemoteException;
	public void deleteHITSequence(UserDO user, HITSequenceDO sequence, ProjectDO project) throws RemoteException;
	public Collection getHITSequences(UserDO user, ProjectDO project) throws RemoteException;
	public void addOrganism(UserDO user, OrganismDO organism, ProjectDO project) throws  RemoteException;
	,,,
	public String obtainUser() throws RemoteException;
	public String createUser() throws RemoteException;
	public String getMessage() throws RemoteException;
	public void setMessage(String message) throws RemoteException;
	public UserDO getUser() throws RemoteException;
	public void setUser(UserDO user) throws RemoteException;
}
```

L'anterior és la interface que els JSFs utilitzen per tal de realitzar alguna de les funcions: crear un usuari nou al sistema, crear un alineament, ...

La classe UserManagementBean implementa totes aquestes funcions i altres de privades per tal de facilitar el treballs amb els Beans persistents. Les funcions més importants són aquelles que són cridades per les pàgines JSF i que han de retornar un String que és el resultat de l'execució de la funció. Si s'ha pogut realitzar per exemple la creació de l'usuari correctament llavors retornarà 'usuari creat' i en cas contrari un missatge amb l'explicació de l'error com 'usuari existent'.

Els missatges que retornen totes les funcions cridades dels dels JSFs no són visibles per l'usuari ja que són simplement un mecanisme de guiar la lògica del procés i per tant, de saber en tot moment quina és la següent pantalla a mostrar. Si l'usuari s'ha creat correctament llavors la següent pantalla que s'hauria de mostrar és la de creació de projectes sino s'hauria de tornar a mostrar la pantalla de conexió al sistema.

Tot aquest procés està guiat pel fitxer *examples-config.xml* que es descriu a continuació.

```
<faces-config>
	<application>
		<locale-config>
			<default-locale>en</default-locale>
			<supported-locale>es</supported-locale>
		</locale-config>
	</application>

	<managed-bean>
		<managed-bean-name>userManagement</managed-bean-name>
		<managed-bean-class>ejb.UserManagementSessionBean</managed-bean-class>
		<managed-bean-scope>session</managed-bean-scope>
	</managed-bean>

	<navigation-rule>
		<from-view-id>/user.jsp</from-view-id>

		<navigation-case>
			<from-outcome>user created</from-outcome>
			<to-view-id>/project.jsp</to-view-id>
			<redirect />
		</navigation-case>

		<navigation-case>
			<from-outcome>user not created</from-outcome>
			<to-view-id>/user.jsp</to-view-id>
		</navigation-case>
	</navigation-rule>
</faces-config>
```

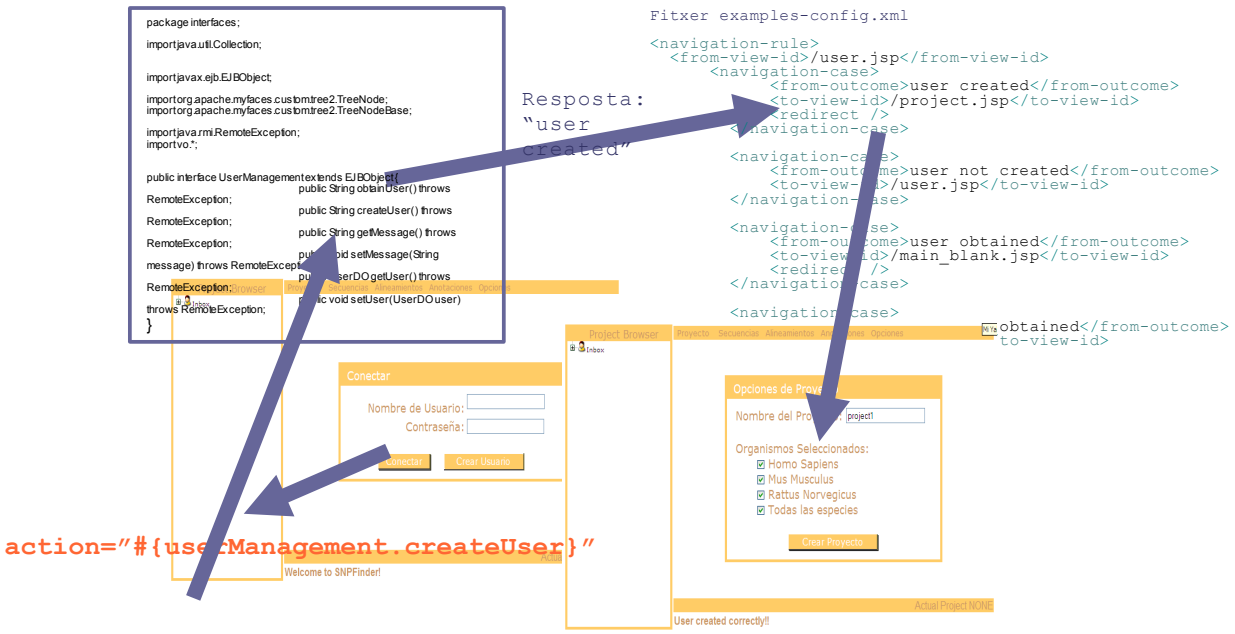
Cadascuna dels Beans necessaris per crear les pàgines JSFs s'han d'indidicar dins l'etiqueta `<managed-bean>` i perquè aquest puguin ser instanciats perl servlet `javax.faces.webapp.FacesServlet` cal afegir un mètode de creació sense paràmetres, d'altra banda, es produirà un error en l'execució de l'interfície.

En el cas anterior sempre que partim de la pàgina jsf **user.jsp** (`<from-view-id>`) i es retorni un missatge 'user created' la pantalla següent a mostrar és **project.jsp**. D'altra banda si el missatge retornat alhora de crear l'usuari en la pàgina **user.jsp** és 'user not created' és mostrarà la mateixa pàgina **user.jsp**.

Per a cadascuna de les pàgines jsf a crear cal indicar les regles de navegació en cas que es produixi una o una altra situació. A més es pot indicar regles que s'apliquin de forma global sigui quina sigui la pàgina que generi el missatge.

Tan el servlet que s'utilitza en el sistema com el fitxer de configuració de myfaces s'indiquen en el fitxer *web.xml*. A més en aquest fitxer també es configuren altrea paràmetres importants com la primera pàgina a mostrar, els filtres a aplicar, ...

Per tant, tots dos fitxers són indispensables per al correcte funcionament de l'aplicatiu i han d'estar ubicats en el directori **/WEB-INF**. El següent esquema mostra el flux principal d'informació que s'estableix per tal de dur a terme la lògica de procés adequada.



Alhora d'implementar i prova les pàgines jsf creades cal tenir molt clar com es gestionen els errors. Així doncs, si existeix algun error en la pàgina la major part de les vegades en pitjar qualsevol botó refresca la mateixa pàgina encara que no hagués de ser aquest el seu comportament habitual.

Per obtenir més informació sobre l'error produit cal afegir en cada pàgina creada `<h:messages />` per tal de mostrar en format llista els missatges generats.

Finalment, la visualització dels missatges d'error o d'avís que es produexin durant el tractament de les dades es localitza en el 'message board' a peu de pàgina web:

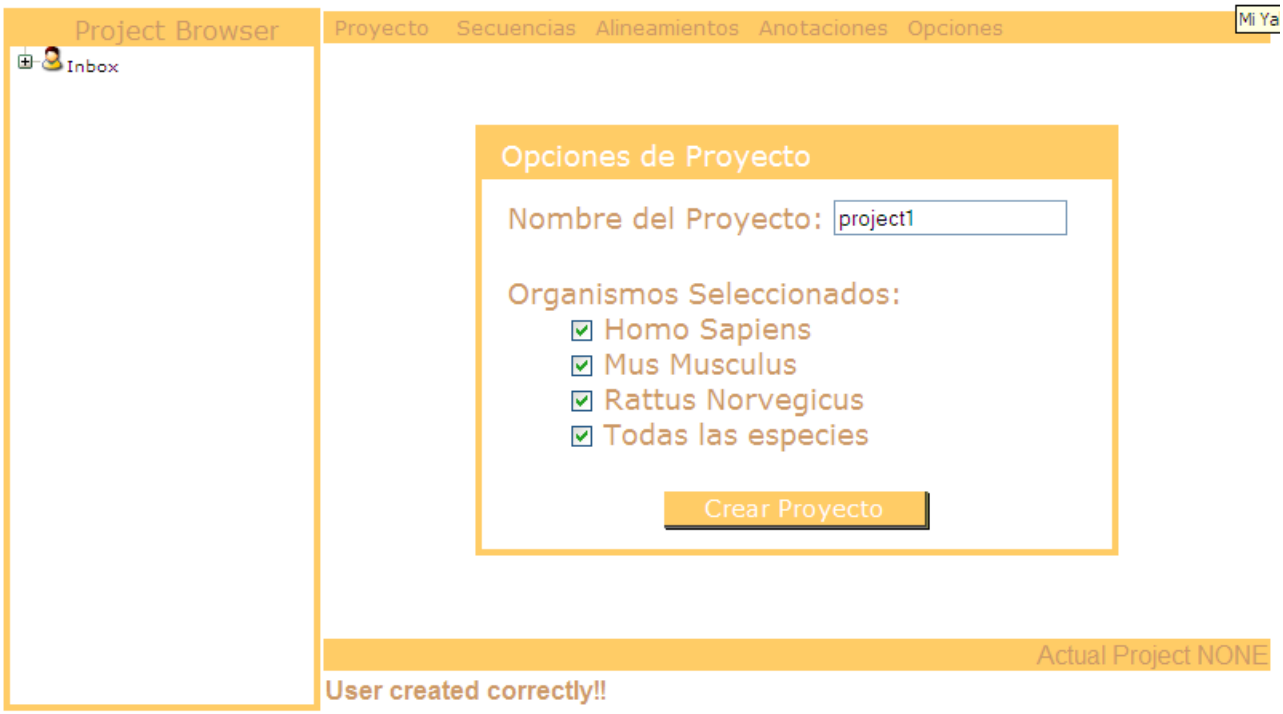
La manera de centralitzar la gestió d'error en tot l'aplicatiu en facilita la visualització. En primer lloc s'ha creat un atribut *message* de tipus cadena en el Bean **UserManagementBean** i després s'ha lligat la visualització del seu contingut al lloc de la pàgina web adequat ('message board'). Així doncs, cada cop que que una funció modifiqui l'atribut *message* automàticament passarà a ser visible a l'usuari. La forma d'accedir a l'atribut *message* de **UserManagementBean** des d'una pàgina jsf és la següent:

```
<td height='10px' valign='middle' align='left'>
      <b>
      <font color="#CC9966">
            <h:outputText value="#{userManagement.message}" />
      </font>
      </b>
</td>
```

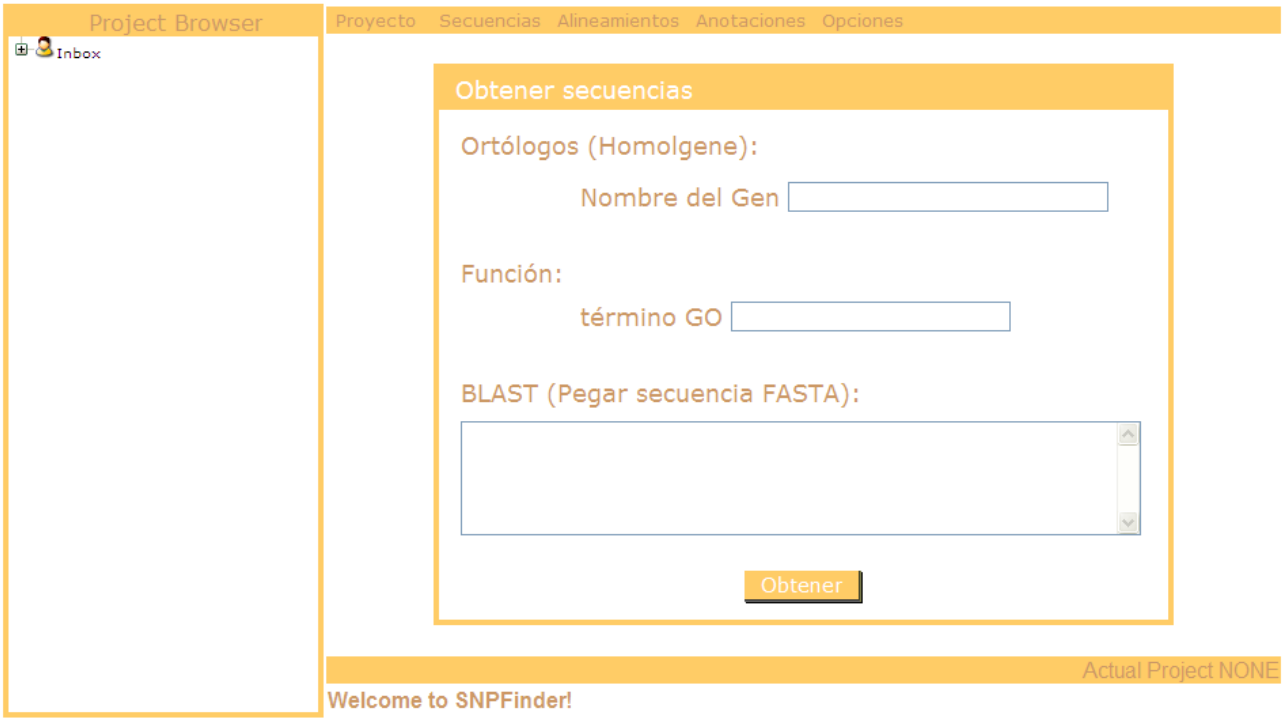
Algunes de les pantalles dissenyades en la PAC anterior ja finalitzades són les següents:

pantalla de conexió al sistema:

pantalla de creació de projectes:

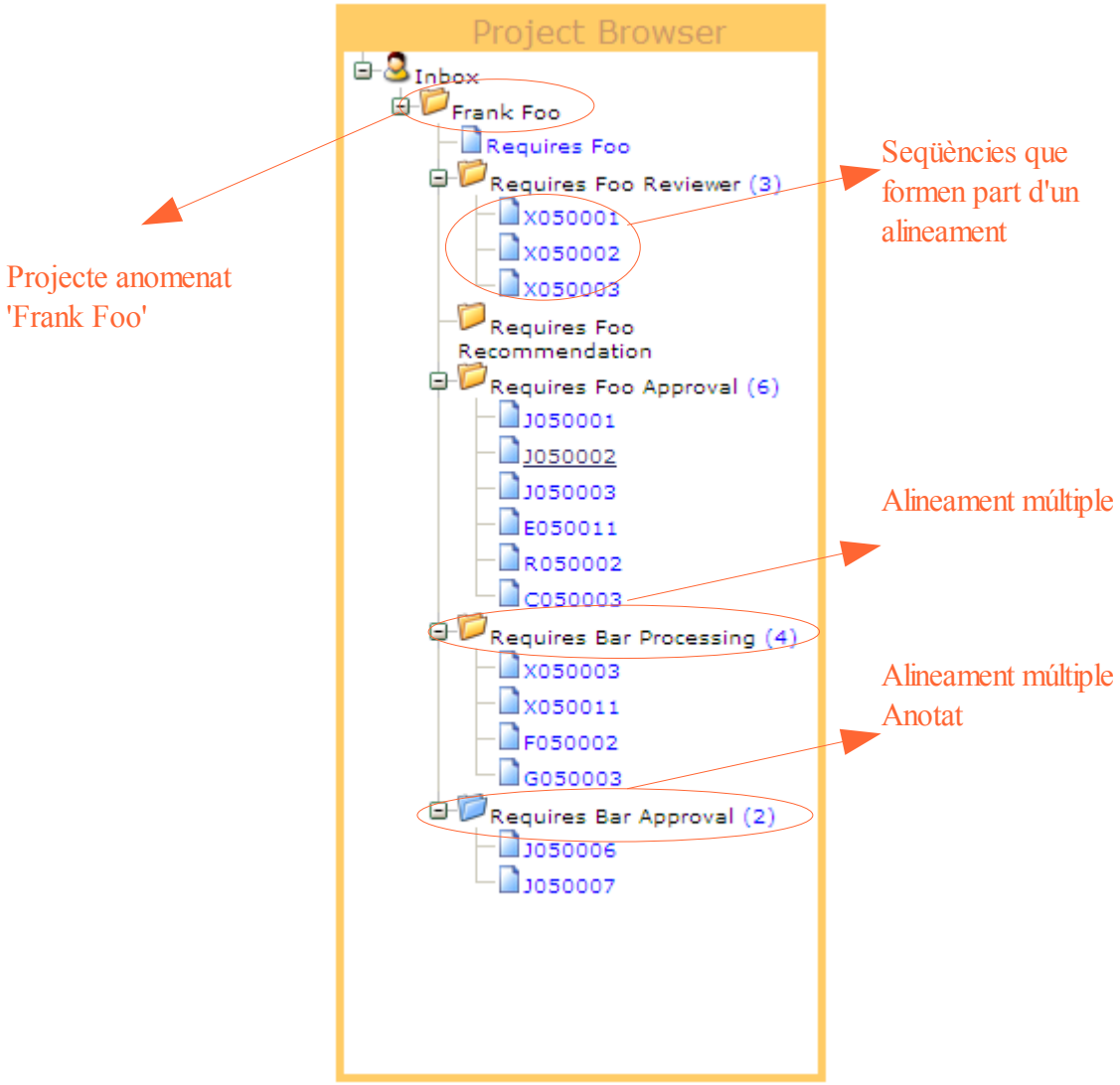


pantalla d'obtenció de seqüències:

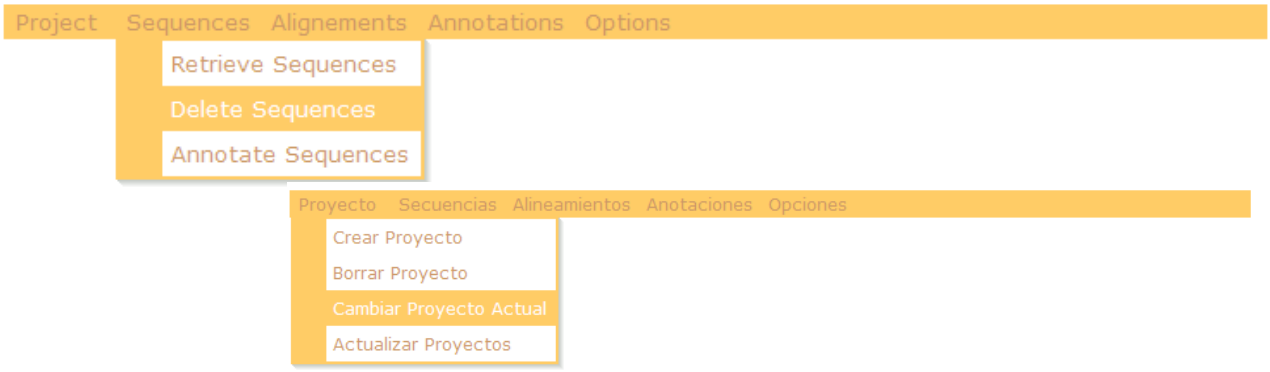


pantalla d'anotació d'alineaments:

detall del 'project browser':



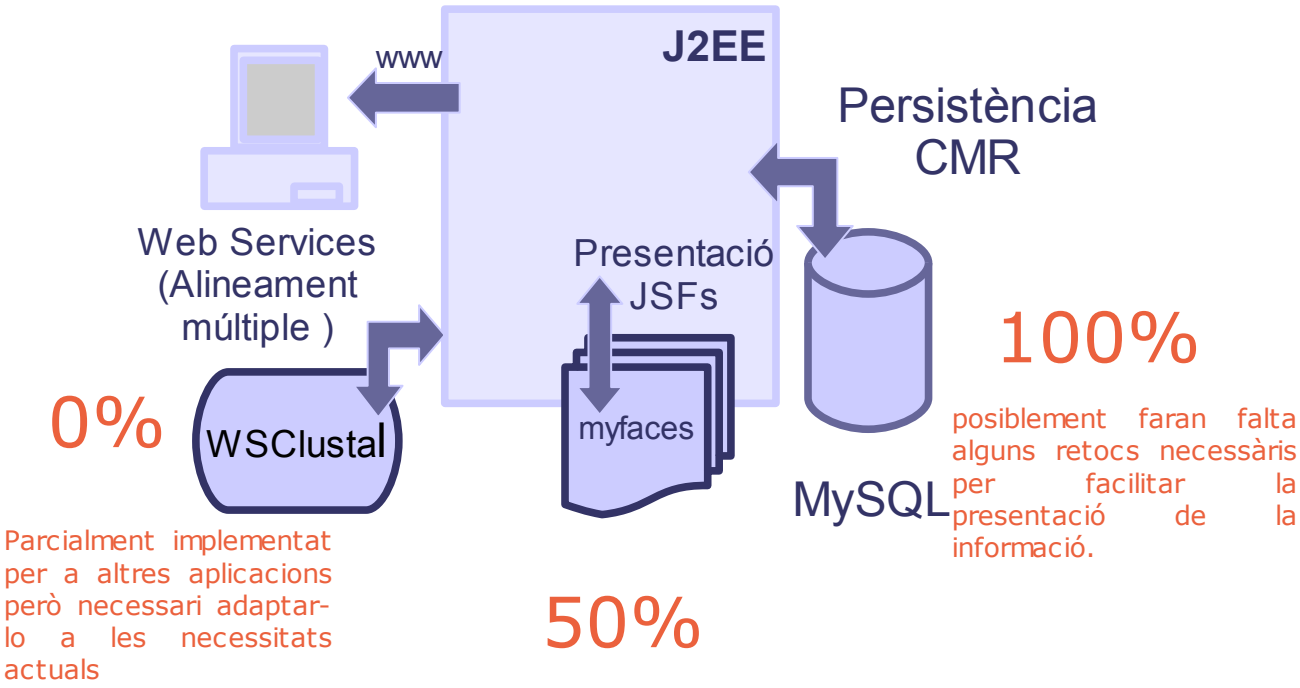
detall dels idiomes soportats:



'State of Art'

Tot i que inicialment el projecte s'havia concebit per a poder finalitzar-lo durant el treball de fi de carrera, tal i com vaig plantejar en la planificació, finalment únicament una part d'aquest s'ha pogut concloure.

El següent diagrama preten il.lustrar el percentatge implementat en cada subsistema:



En el moment de l'entrega de la PAC la implementació de la persistència està finalitzada tot i que es molt possible que s'hagin d'introduir petits canvis en algunes de les taules per tal de facilitar el tractament de les dades per part de l'usuari.

En quan a la interfície gràfica a pesar que no totes les pantalla estàn creades si s'ha realitzar la part més costosa. Un cop familiaritzat amb els JSFs la creació de la lògica no hauria de resulta difícil.

L'única part que resta quasi per complet per implementar és

el procés mitjançant es creen els alineaments múltiples (web services). De totes maneres, la web de l'European Institut of Bioinformatics (EBI) aporta alguns exemples prou bons de com realitzar els clients que facin ús dels web services. Per tant, únicament caldria modificar-los mínimament per adaptar-los a les nostre necessitat i en major grau simplificar-los de codi no important per al projecte. Durant el projecte s'ha realitzar algunes proves amb èxit dels web services d'exemple generats.

Conclusions

El projecte de fi de carrera m'ha permés coneixer la tecnologia J2EE i totes les ventatges que aquesta aporta. He pogut endinsar-me en la creació de pàgines web mitjançant JSFs, eina que encara no considero prou madura però que pot arribar a ser molt i molt útil. A més he pogut gestionar la persistència dels Beans mitjantçant CMR i apropar-me una mica a la problemàtica de com els diferents gestors de bases de dades poden implementar-la totalment o parcialment. En el meu cas, he hagut de realitzar algunes modificaciones amb les que en un principi no contava.

Degut a que no tenia cap tipus de coneixement sobre J2EE he hagut de realitzar molt esforç, sobrepassant el nombre d'hores que inicialment en la planificació havia cregut adients. De fet, el nombre d'hores que he dedicat diàriament excedeix de les 6 hores els dies feines i de més de 12 els dies festius (cap de setmana). Això sens dubte es fruit d'una molt mala planificació.

Un segon punt que ha enrarerit encara més la implementació del projecte ha estat la configuració de totes les eines necessàries per a poder desnvolupar l'aplicació. En molts casos he hagut de canviar de versió del Jboss i l'eclipse. Tot i això, els problemes van continuar apareixent i internet ha resultat una base de dades d'informació molt valuosa.

El factor principal al que atribueixo la manca de planificació ha estat el desconeixement total de la plataforma. Les lectures preparatories m'han ajudat a tenir una visió globa, és a dir, una perspectiva de J2EE, però en cap cas m'hagués imaginat el temps necessari per a poder resoldre determinats errors. En molts casos he esta apunt de tirar la tovallola ja que considerava que la informació que trobava a internet o bé era massa específica o bé massa general.

De tots els exemples pràctics realitzats, sens dubte el que més a contribuit a agilitzar el meu aprenentage és el tutorial del laboratori de pràctiques.

En quan a la corba d'aprenentatge, ja he comentat que és

molt lenta i que en poc més d'un mes es difícil assolir i digerir tots els coneixements. M'he donat compte que els darrers dies d'implementació la productivitat havia augmentat molt. Tot i això, els problemes de configuració han estat constants fins al final.

A pesar que el projecte no s'ha finalitzar durant el treball de fi de carrera és probable que finalitzi el projecte ja que suposaria tenir una eina que facilita molt el tractament i la interpretació de les dades.

Apèndixs

Glossari:

Concepte	Definició
SNP	Canvi de bases en un seqüència d'ADN que es troba en una freqüència major al 5%. En altre cas es considera mutació. També es considera SNP a la inserció o deleció de una o vàries bases.
bases	Cadascun dels elements que formen l'ADN.
Seqüència reguladora	Conjunt de Bases en una seqüència que es reconegut per una proteïna per tal de regular l'expressió d'una altra proteïna.
Format FASTA	Format mitjançant el qual s'identifiquen les seqüències d'ADN. Consta d'una breu descripció i de totes les bases que formen la pròpia seqüència
Alineament Múltiple	Alineació de vàries seqüències que posa en relleu les zones o regions d'aquestes que són semblants.
Alineament Múltiple Anotat	Alineament sobre el qual s'ha afegit la informació relativa als SNPs que conté: tipus, validació i freqüència
Homòleg	Un Gen o proteïna homòleg és aquell gen o proteïna d'una altra espècie amb un origen evolutiu comú, és a dir, provenen d'una mateixa seqüència que es duplica i posteriorment es diferencia mitjançant un procés de mutació.
exó	Un gen en el genoma es pot trobar framentat en petits troços. Entre aquests troços existeix material genètic que jugar un paper estructural. Aquest troços s'anomenen exons.

Abreviacions:

Abreviació	Definició
SNP	Single Nucleotide Polymorphism
GO	Gene Ontology
BLAST	Basic Local Search Alignement Tool
ADN	Àcid Desoxiribonucleic
EBI	European Bioinformatics institute
JSF	Java Server Faces
WS	Web Services
TFC	Treball de fi de carrera

Bibliografia

Llibres:

- *Beginning Java EE5, from Novice to Professional*. Kevin Mukhar and Chris Zelenak. Editorial Apress 2006.

Tutorials de www.coreservlets.com:

- JSF: The Myfaces Custom Component (Tomahawk)
- JSF: Data Tables
- JSF: Managed Beans
- Using the JSP Standard Library (JSTL) with JSF
- JSF: Controling Page Navigation
- Accesing Nested Bean Properties

Altres tutorials:

- Tutorial de Laboratori de 'Enginyeria del programari de components i de sistemes destribuïts'. Versió 1.09
- Pràctica tutoritzada. 'Enginyeria del programari de components i de sistemes destribuïts'. 2006

webs:

- http://wiki.jboss.org
- http://wiki.apache.org/myfaces
- http://www.irian.at/myfaces (JSF examples)
- http://myfaces.apache.org
- http://www.onjava.com (Tutorial EJB inheritance)
- http://www.developer.com/java/ejb/
- http://www.biojava.org

Fòrums:

- http://www.nabble.com/Myfaces-f180.html
- http://forum.java.sun.com/