**Cosas necesarias 3º entrega.**

**Backing beans**

login.java

entrar()

decide que usuario es el que ha entredo y le redirecciona a la vista correspondiente

solicitudcontrol.java

inyectar sesión y negocio

aprobarsolicitud(solicitud)

entra en la capa de negocio

refreshuser(usuario) del usuario de la solicitud, nos devuelve el usuario de la persistencia actualizado

Llama a la capa de negocio getnj()

Llama a la capa de negocio crearap(usuario, nj)

Eliminar la solicitud de la persistencia llamando a eliminarsolicitud(solicitud)

return “solicitudes.xhtml”

rechazarsolicitud(solicitud)

eliminarsolicitud(solicitud)

return “solicitud.xhtml”

gestionarsolicitud()

si el checbox está clicado crea una solicitud y la manda a la persistencia

con crearsolicitud() de la capa de negocio

return “socio.xhtml”

si no esta clicado return “socio.xhtml”

sociocontrol.java (Para modificar socio en botón modificar)

modificar()

public list<usuario> getsocios()

return neg. getusuarios(“socio”)

infosesion.java

getempleados()

empleadocontrol.java

Inyecta la sesemp

getsolicitudes()

accede a la negocio y llama a getsolicituddes()

public list<usuario> getempleados()

return neg.getusuarios(“emp”)

getempleado()

return neg.refreshuser(sesión.getusuario())

admincontrol.java

Inyecta la sesión y la negocio

@requestscope

public list<usuario> getadmins()

return neg. getusuarios(“admin”)

njcontrol.java

insertarnj(“Pasar todos los datos de un nj”)

Se llama al constructor con todos los datos y luego a la función insertarnj() de la capa de negocio.

Public list<jovenes> getnjs()

return

eliminarnj(Jovenes nj)

eliminarnj(Jovenes nj)

modificarnj(Jovenes nj)

consultar modificar contacto de la agenda (Suerte)

apcontrol.java

eliminarap(Apadrinamiento ap)

modificarap(Apadrinamiento ap)

consultar modificar contacto de la agenda (Suerte)

getaps(Usuario u)

coger el socio de la sesión

comprueba si el parámetro es igual al socio de la sesión y si lo es…

llamar a refreshuser(usuario) de la persistencia y asignar a socio

get apadrinamientos del refresh socio

public list<Apadrinamientos> getaps()

return neg.getaps()

**Capa de negocio**

comprobarlogin()

crearsolicitud(Usurario)

modificarsocio()

public usuario refreshuser(usuario)

Devuelve el usuario de la persistencia, al pasarle el usuario de la sesion

getsolicitudes()

accede a la persistencia y nos devuelve todas las solicitudes

getnj()

Nos devuelve el candidato más apropiado de nuestra lista de jóvenes para ser apadrinado.

crearap(usuario, nj)

Apadrinamiento = new apadrinamiento(user, nj, fecha)

em.persist(ap)

public boolean eliminarsolicitud(Solicitud sol)

Solicitud S = em.find(Solicitud.class, sol.getid())

If(S ¡= null){

Em.Remove(S) ;

}else (lanzar excepcion)

public boolean eliminarusuario(Usuario u)

public list<usuario> getusuarios(rol)

Devuelve todos los usuarios según el rol

Crea una lista de usuarios vacia L

Query q = new Query(“Select e from usuario e where e.rol = ‘”+rol”’ “)

L = q.getResultList() ;

Return L

public void Insertarnj(Jovenes nj)

Em.persist(nj)

Public list<Apadrinamientos> getaps()

Consulta SQL (Buscate la vida)

Public list<jovenes> getnjs()

Consulta SQL (Buscate la vida) (Apuntes jpa)

Eliminarap(Apadrinamiento AP)

Em.remove(em.merge(AP))

Eliminarnj(jovenes nj)

Em.remove(em.merge(nj))

Modificarap(Apadrinamientos ap)

Modificarnj(Jovenes nj)

**Tareas varisa**

- Eliminar socio y sustituir empleados por usuario.

- Revisar si los botones de cada vista estan bien hechos

**Cambios en las vistas**

Solicitudes.xhtml

Imprimimos llamando al baking beans empleadocontrol.getsolicitudes las solicitudes, datos:

Fecha

Usuario

Botones: (Commandbutton)

Aprobar

Llama a un método de solicitucontrol.aprobarsolicitud

Rechazar

Llama a un método de solicitucontrol.rechazarsolicitud

Admin.xhtml

Crear una vista para que el Admin pueda modificar sus datos

**Funcionalidades**

1. Login A
2. Registro S
3. Listar apadrinamientos del socio (socio) F
4. Modificar datos del socio (Socio) M
5. Solicitar apadrinamiento (Socio) J
6. Listar, aprobar y rechazar solicitudes (empleado y admin (So la misma funcionalidad)) A
7. Listar empleados (Empleado) S
8. Mostrar y modifical datos del empleado (Empleado) F
9. Listar administradores (Empleado) M
10. Listar modificar y eliminar socios (Empleado y administrador (Son la misma funcionalidad)) J
11. Dar de alta un joven (Empleado y administrador) A
12. Listar modificar y eliminar apadrinamientos (Empleado y administrador) S
13. Listar modificar y administrador (Empleado y administrador) F
14. Listar, modificar y eliminar empleados (Administrador solo) M
15. Listar mensajes (Administrador) J
16. Listar, modificar y eliminar administradores (Administrador) A
17. Dar de alta a un administrador (Administrador) S
18. Dar de alta un empleado (Administrador) F
19. Dar de alta mensajes (Administrador y Empleados) M