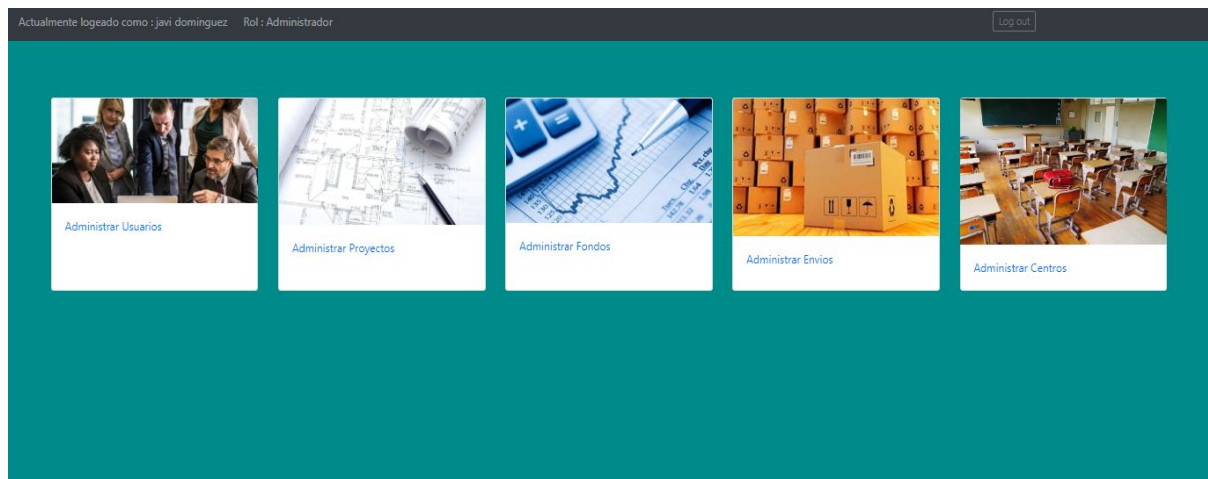




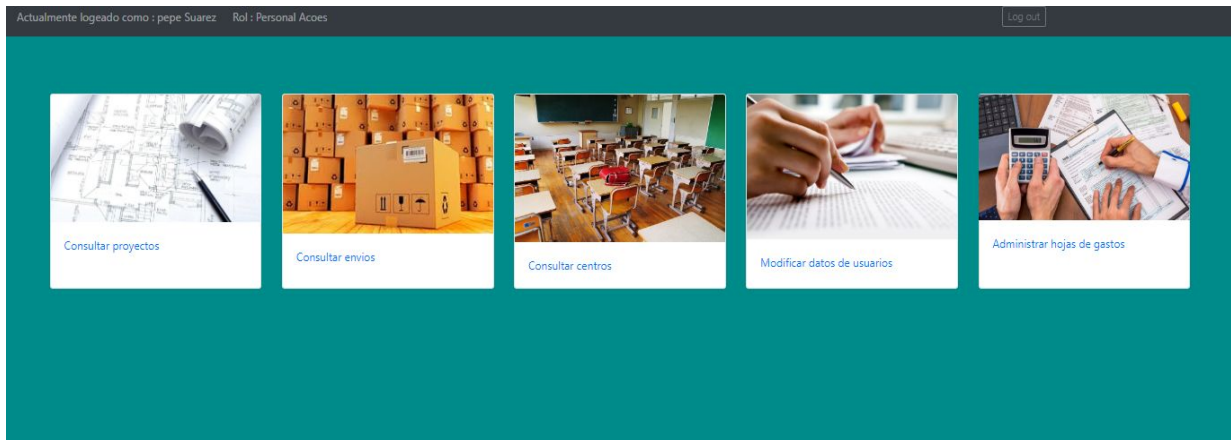
Integrantes del grupo 13:

Carlos Cañero Fernández
Darío Jesús Flores Sevilla
Francisco Javier Dominguez Suarez
Jesús Marques Galán
Marina Muñoz Paradas

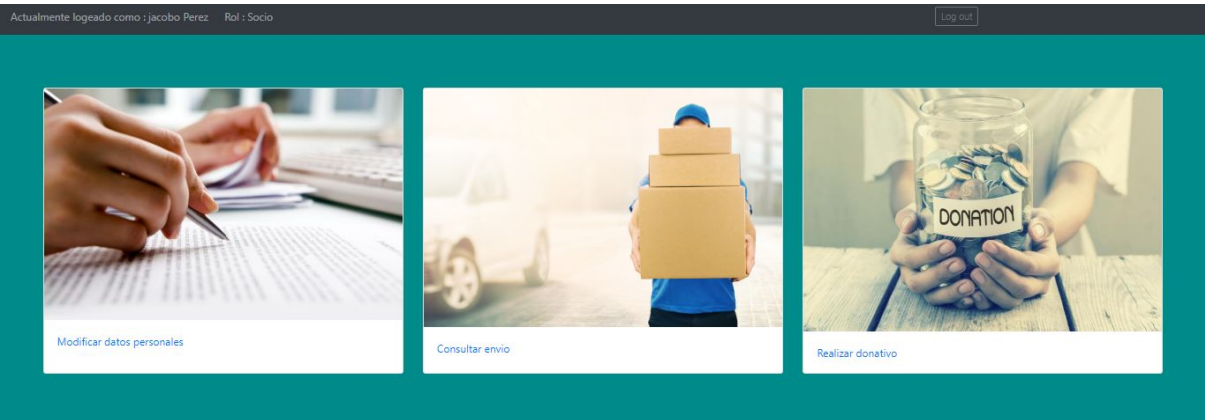
Requisitos funcionales del producto:



- RF1 El administrador de la plataforma podrá gestionar los usuarios de la aplicación : beneficiarios, personal de la asociación y socios
- RF2 El administrador de la plataforma podrá gestionar los proyectos de la ONG
- RF3 El administrador de la plataforma podrá gestionar los fondos de la ONG, en Ingresos/Egresos.
- RF4 El administrador de la plataforma podrá gestionar los envíos a realizar.
- RF5 El administrador de la plataforma podrá consultar y modificar la información de los centros de la asociación



- RF6 El personal de la organización podrá modificar sus datos personales
- RF8 El personal de la organización podrá gestionar a los socios de la aplicación
- RF9 El personal de la organización podrá realizar las funcionalidades anteriormente descritas también con los beneficiarios de la ONG
- RF10 El personal de la organización podrá consultar la información de los proyectos
- RF11 El personal de la organización no podrá modificar o eliminar la información de los proyectos
- RF12 El personal de la organización podrá consultar y crear hojas de gastos e ingresos.
- RF13 El personal de la organización no podrá eliminar o modificar las hojas de gastos sin autorización
- RF14 El personal de la organización podrá consultar la información de los envíos
- RF15 El personal de la organización podrá consultar la información de los centros de la ONG



-RF16 Un socio podrá acceder a su cuenta.

-RF17 Un socio podrá modificar sus datos personales

-RF18 Un socio podrá eliminar su cuenta.

-RF19 Un socio podrá realizar ingresos (donativos) en la cuenta de la ONG.

-RF20 Un socio no tendrá acceso a la información financiera de la ONG.

-RF21 Un socio podrá consultar el estado de un envío realizado por el niño apadrinado.

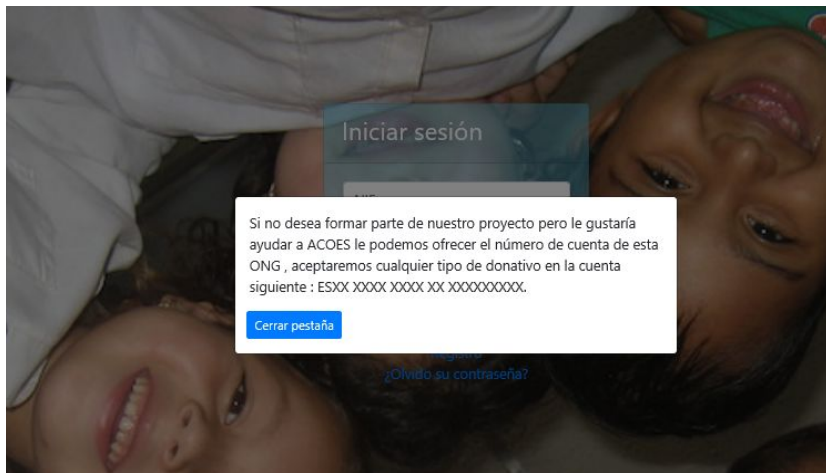
Operaciones que podrá realizar un usuario sin registrar :



-RF26 Un usuario sin registrar podrá registrarse.

A screenshot of a web application's registration form. The background is the same photograph of smiling children. Overlaid on this is a semi-transparent white box containing the registration form. The form has a title 'Formulario de registro' at the top. It contains several input fields: 'NIF/NIE', 'Contraseña', 'Repita su contraseña', 'Nombre', 'Apellidos', 'Fecha de nacimiento' (with a date format 'DD/MM/AAAA'), 'Provincia', 'Población', 'Codigo Postal', 'Calle/numero', 'Teléfono fijo', 'Teléfono móvil', and 'Correo electrónico' (with the example 'ejemplo@gmail.com'). At the bottom of the form, there are two buttons: 'Cancelar' (Cancel) and 'Continuar' (Continue).

-RF27 Un usuario sin registrar podrá realizar ingresos (donativos) de forma anónima en la cuenta de la ONG.



-RF28 Un usuario sin registrar podrá ver el contenido público de la página web.

Descripción de los EJBs usados:

En primer lugar tenemos nuestras Excepciones , `AplicaciónException` y `CuentaExistenteException`, por otra parte hemos utilizado una interfaz donde se recogen los métodos que implementará nuestra clase `InterfazImplementada.java` , los métodos que hemos utilizado son los siguientes:

`buscarSocio` -> Busca un socio proporcionando su dni (RF1,RF8)

`buscarPersonal` -> Busca a miembros del personal al introducir id (RF1)

`add` -> Añade cualquier usuario o proyecto la Base de Datos (RF2)

`update` -> Actualiza la Entidad de la BD que recibe

`delete` -> Elimina la Entidad de la BD que reciba

`registrar` -> Sirve para registrar Socios (RF26)

`registrarUsuario` -> Registra beneficiarios y personal

refrescarUsuario -> Actualiza los datos de alguien del personal o de un socio (RF17,RF6)

refrescarProyecto -> Actualiza los proyectos (RF2)

listar_personal -> Devuelve una lista con el personal (RF1)

listar_beneficiarios -> Devuelve una lista con los beneficiarios (RF9,RF1)

listar_socios -> Devuelve una lista con los socios (RF8,RF1)

getProyectos -> Devuelve una lista con los proyectos (RF10)

listar_centros -> Devuelve una lista con los centros (RF5)

getEnvios -> Devuelve una lista con los envíos(RF4,RF21,RF14)

listar_ingresos -> Devuelve una lista con los ingresos (RF3)

listar_proveedores -> Devuelve una lista con los proveedores (RF14)

añadirProyecto -> Crea un nuevo proyecto (RF2)

actualizarProyecto -> Modifica un Proyecto(RF2)

eliminarProyecto -> Elimina un proyecto al proporcionar su id (RF2)

añadirPersonalAProyecto -> Añade una persona a un proyecto (RF2)

eliminarPersonalDeProyecto -> Elimina a alguien de personal de un proyecto (RF2)

añadirEnvio -> Crea un nuevo envío (RF14)

actualizarEnvio -> Modifica la información de un envío (RF4)

eliminarEnvio -> Borra un envío (RF4)

añadirCentro -> Añade un nuevo centro a la BD (RF15)

Cambios en los JPA :

- Gestión de proveedores: hemos tomado los proveedores como los socios que realizan ingresos cuyo atributo relación es "proveedores".
- Gestión de hoja de gastos: hemos tomado la hoja de gastos como la relación entre los ingresos y los proyectos a los que son destinados.
- Modificado JPA por error encontrado en atributo fecha de pago.
- Arreglado error al definir el atributo Foto .
- Hemos borrado la tabla municipios.
- Agregado campo contraseña a Socio y Personal.
- Personal tiene ahora un campo fecha de nacimiento.

Cambios en los backing bean:

Para esta entrega principalmente nos hemos dedicado a crear nuestros Beans, ya que para la otra aun no teníamos muchos de estos, para cumplir con nuestros requisitos , usaremos los siguientes :

Añadir.java : nos permite añadir, actualizar y eliminar un proyecto, como también nos permite añadir personal o quitarlo de un proyecto, nos da lugar también a añadir, actualizar o eliminar envíos y añadir centros que se sumen a la causa. (RF2,RF4,RF5)

GestionIngresos.java : podemos crear o editar un ingreso (RF3,RF12)

GestionUsuarios.java : tenemos métodos para editar cualquier tipo de usuario, centro y guardar así la información editada (RF6,RF17)

Login.java : permite a los distintos tipos de usuario loguearse en la app (RF16)

Registro.java : se pueden registrar tanto un socio , como el personal como los beneficiarios.

Sesion.java : controla el mantenimiento de las sesiones , aparte de dar comprobar ciertos permisos por motivos de seguridad. (RF26)

Conclusión :

Hemos trabajado para cumplir con los requisitos especificados, aun así no nos ha dado tiempo de implementar y cumplir los requisitos de búsqueda por nombre en gestión de usuarios , aparte nos ha faltado tiempo añadir una hoja de gastos en gestion_fondos, por otra parte analizando el proyecto por partes y los requisitos de este, vemos la utilidad de usar esta estructura de trabajo y como de una forma relativamente sencilla podemos unir distintos módulos , tanto JPA y EJB como los JSF y los Beans nos permite ir cumpliendo y dando las funcionalidades que el proyecto requería , aparte también cabe resaltar la utilidad que cobran herramientas como Git Hub , a la hora de trabajar en grupo sobre cualquier plataforma o proyecto en grupo, dándonos también posibilidades para corregir errores y organizar tareas.

También proporcionamos aquí la url del repositorio online:

<https://github.com/Sonny064/Proyecto-Sistemas-Inf-Internet>

Los primeros 10 caracteres del hash del último commit) :

a2da2b670c