INFORME DE LA PRACTICA

Contenido

[Memorias: 3](#_Toc536123996)

[Manual para la construcción y lanzamiento de la aplicación 3](#_Toc536123997)

[Descargar y carga proyecto: 3](#_Toc536123998)

[Ejecutar el programa: 6](#_Toc536123999)

# Memorias:

En una primera instancia, hemos creado un menú bastante intuitivo y visual basado en la inserción numérica para elegir una opción, esto da bastante versatilidad al usuario a la hora de elegir que opción desea hacer, pues permite en caso de error hasta volver una tarea hacia atrás o avanzar dos estados de golpe en caso de que se olvidase, da bastante libertad al usuario en general.

Nos decantamos por crear dos Singlenton: AdministradosDeTareas, AdministradorDeMiembros y AdministradosDeRequisitos, esto se debe a que estas clases van a manejar la lista de tareas y de miembros, por lo cual necesitamos que haya una única instancia accesible globalmente, para que solo haya una lista y tener una información verídica sin “bifurcaciones”. Estas dos clases se encargar, como el nombre indica, de administrar las listas de miembros y de tareas, hasta para la persistencia.

También decidimos que era buena idea, hacer del backlog una especie de observador, product backlog y sprint backlog contendrán dos listas independientes, la primera con las tareas pendientes, y la segunda con el resto de las tareas, decidimos que sería observador, debido a que cuando una tarea modifica su estado, o directamente se crea una nueva tarea, el backlog se actualiza automáticamente, e intentamos que no haya actualizaciones imprevistas

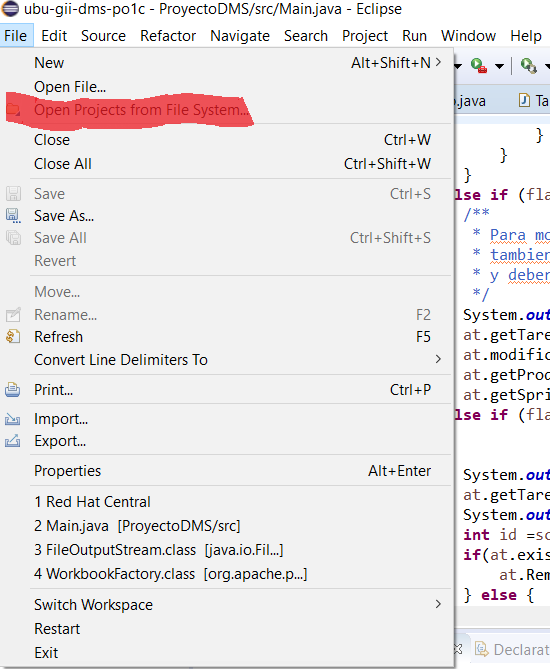
La persistencia la logramos a través de la librería de Apache poi, para leer y escribir en un fichero Excel. Abstraemos Modelo, Vista y Controlador separándolo en distintas clases, en esta última versión quitamos del main, todo lo posible para que este lo básico y creamos una clase llamada VistaPrincipal que hace de estados, para la vista, y sacamos las funcionalidades del modelo y controlador que había en el main.

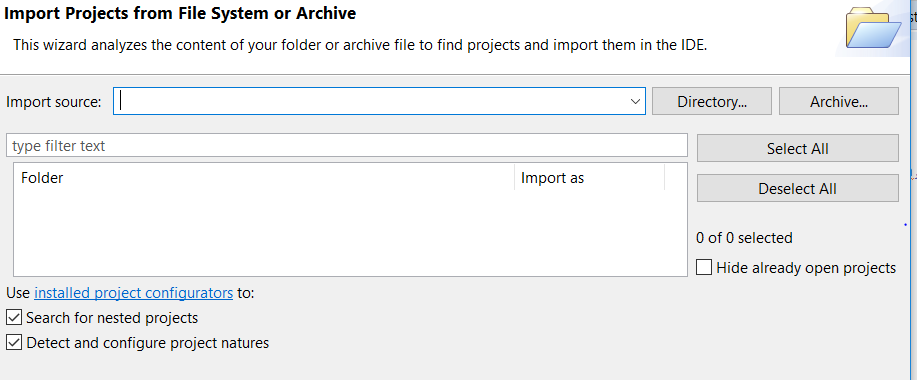
# Manual para la construcción y lanzamiento de la aplicación

## Descargar y carga proyecto:

Descargar los archivos necesarios del enlace: <https://github.com/jct0024/ubu-gii-dms-po1c> , una vez lo tengamos descargado lo guardamos en el lugar que más cómodo nos resulte del equipo. Más tarde deberíamos tener descargado eclipse en nuestro escritorio.

Abrimos eclipse y nos vamos a archivo (File) una vez hay pulsamos <<Open projects from file system>>



Mas tarde nos saldrá una pantalla similar a esta: 

Damos en “directory” y buscamos el fichero del proyecto halla donde le hallamos guardado.

Una vez llegados a este punto, Instalar las librerias necesarias desde el siguiente enlace:

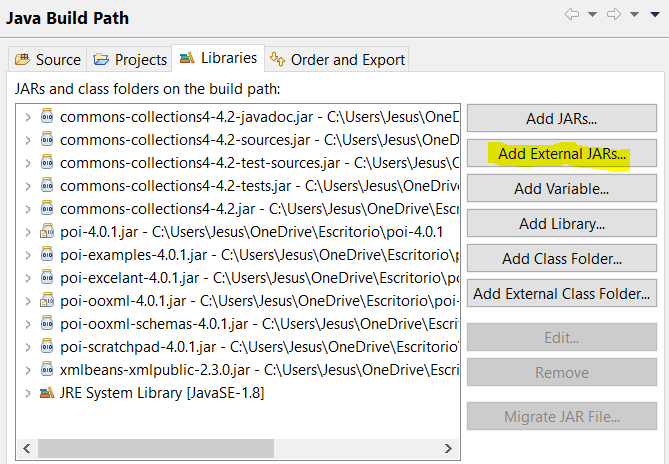
<https://www.dropbox.com/s/z59i4ayg7a2uc7j/poi-4.0.1.rar?dl=0>

Pero antes, eliminar toda librería que encontremos en las propiedades del build path, que aparezcan con una x roja de error y ELIMINARLAS, una vez eliminadas, Instalar las librerias de esta carpeta, con esto deberia ser suficiente, en caso de fallo, pruebe a descargar todas las librerias del siguiente enlace e instalarlo aunque no deberia ser necesario:

<https://www.dropbox.com/s/10stplzcgv43n99/librerias.rar?dl=0>

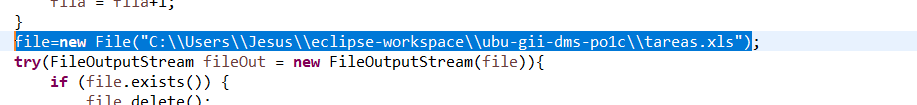
Estas bibliotecas deberián solucionar todo error posible.

Para instalarlas nos vamos al proyecto, pulsamos el click derecho del ratón, y nos vamos a Build Path -> Configure Build Path. Nos saldra esta pantalla:



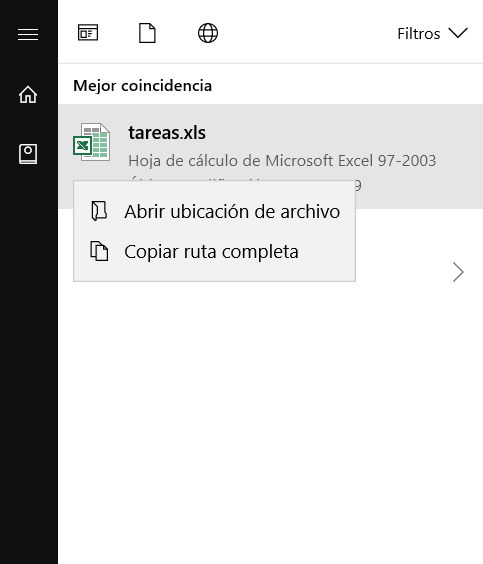
Pulsamos: Add External JARs, y buscamos el directorio donde hemos guardado las librerias y las selecionamos.

MUY IMPORTANTE: Ir a la parte de codigo de ReadExcellFile y GuardarEnExcell y nos vamos a las partes del código similares a esta: (Donde haya una ruta de directorio)



En esa zona debemos poner la ruta del directorio donde tengamos el proyecto, junto al nombre de las bases de datos, sino dará error por no encontrar la base de datos.

Para encontrar esta ruta buscamos: tarea.xls o miembros.xls o requisito.xls, en la barra de tareas de nuestro otrdenador.



Pulsamos click derecho, y damos a abrir ubicación del archivo.

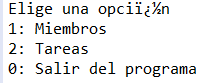


Copiamos esta dirección y le añadimos “\\tareas.xls” y otra vez con: “\\miembros.xls” y por ultimo: “\\requisitos.xls”

RECORDATORIO: Tanto en la carga del sistema operativo, como donde se guarda. La ruta de carga la encontramos en el main y la de guardar los datos en la persistencia.

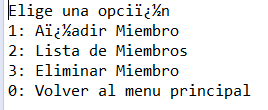
## Ejecutar el programa:

Una vez cargado el programa, vamos al Main, que es la clase que se organizara de todo, y nos saldra un menu en la consola como el siguiente:



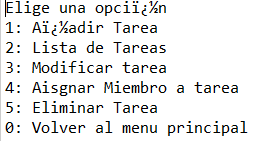
Desde aquí podemos elegir Miebro, Tareas, o salir del programa, este ultimo cierra el programa guardando todo dato posible. (Las opciones siempre se eligen a traves de los números, para su facil uso)

* Si elegimos “Miembros”, pasaremos a un menu desde el que podemos elegir cualquier opción de miembro.



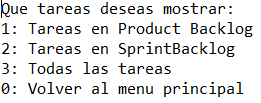
Podemos añadir miembro, listar todos los miembros que existen, o eliminar un miembro en base al ID.

* Si elegimos “Tareas”, pasaremos al menú desde el cual podemos modificar y listar las tareas necesarias:

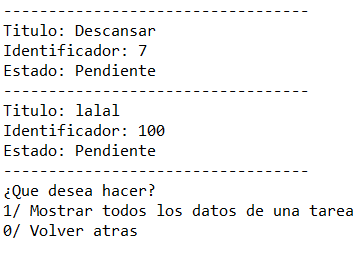


Aquí tenemos las opciones básicas: Añadir tarea, listar tarea, asignar miembro a tarea y eliminar la tarea en base al ID.

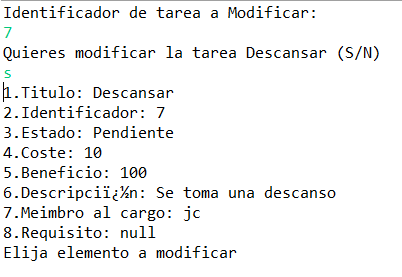
A la hora de **listar la tarea**, se muestra tres opciones:



Puedes elegir si te muestra todas las tareas, o las que se encuentran en el product o en el sprint backlog, una vez elijas una de esas opciones te mostrará las tareas pertinentes, y la opción de mostrarte toda la información de una tarea concreta, o salir, pues en un principio esta opción te muestra la información básica:



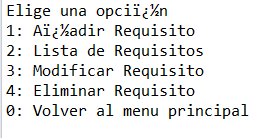
En **Modificar tarea** pasa algo similar, te muestra la información básica de las tareas, y te pide que insertes el id de la tarea que queires modificar y ya te muestra todos las posibles opciones:



Se elegiría la opción a cambiar y se cambiaría por lo nuevo.

Importante: A la hora de añadir una descripción o nombre compuesto, no añadir espacios, sino hacerlo en este formato: Esto\_Es\_un\_ejemplo\_de\_texto

Si escogemos requisito funciona de manera similar, hemos decidido no variar mucho el diseño, para hacer una interfaz mas sencilla al usuario, y así simplificar su comprensión.



Tendríamos estas 4 opciones, cada una de ella te va a pedir los datos básicos para añadir o modificar un requisito o eliminarlo. Para listarlo no hace falta.