manu.md

ESTE PDF SE HA GENERADO DESDE UN ARCHIVO .md. PARA VISUALIZARLO CORRECTAMENTE VISITE EL SIGUIENTE REPOSITORIO:
https://github.com/JorgeBarcena3/Task-Manager

Task Manager - Jorge Bárcena Lumbreras

Este proyecto consistía en realizar una herramienta en forma de librería dinámica en C++. También se debería crear una interfaz gráfica para el uso de dicha herramienta.

Task Manager, es una herramienta que busca facilitar la organización de tareas a sus usuarios. Task Manager está orientado para usuarios que quieren mantener de manera organiza y mantener un seguimiento de las tareas que tiene que llevar a cabo en el trabajo, vida diaria.

En el siguiente documento se especificarán algunos detalles del proyecto, sobre cómo funcionan ciertos objetos y notas sobre su arquitectura. A demás podremos encontrar una guía de usuario básica de la herramienta.

Todo el código está documentado para que doxygen genere los documentos pertinentes (Se debe tener activada la opción de JAVA_DOC_BRIEF).

Índice

Librerías externas

Para la realización de este proyecto he utilizado las siguientes versiones de librerías externas:

- Lua: Maquina virtual de LUA.
- LuaState: Permite adaptar las funciones de C++ con el lenguaje LUA.
- RapidXML: Lectura y escritura de archivos XML.

Organización del proyecto

En la carpeta de este proyecto encontraremos dos carpetas, cada una de esta carpeta tendrá una solución para visual studio 2019. Las carpetas son las siguientes:

- **TaskManagerTool:** Esta carpeta contiene el proyecto de la herramienta. Este proyecto generara una biblioteca de enlace dinámico (.dll), tras la compilación del mismo, los archivos se copiaran (Comando de VS 'xcopy') a los lugares necesarios para la ejecución del proyecto "TaskManagerEditor".
- **TaskManagerEditor:** Esta carpeta contiene el proyecto de la UI de la herramienta. La UI se ha realizado con el framework QT en Visual Studio, también se ha utilizado la herramienta "QT Designer", para diseñar la interfaz de usuario. Este proyecto genera un ejecutable de la herramienta, y hace uso de la librería de enlace dinámico de "TaskManagerTool". Este proyecto hace uso de el comando de VS 'xcopy', para que el proyecto de QT funcione correctamente.

Además de estas carpetas encontraremos tres carpetas más:

- **VS-Solution:** Contiene una solución configurada para su ejecución de visual studio 2019, con los dos proyectos anteriores.
- **binaries:** Contiene una versión ejecutable de la aplicación para su uso final.
- **documents:** Contiene una serie de documentos para

el uso de la herramienta.

Estructura de la herramienta

La herramienta presenta el siguiente modelo de datos para almacenar las tareas:

- **Tareas:** Guarda el título, descripción, fecha de creación y a quien esta asignada la tarea.
- **Estados:** Guarda la una serie de tareas.
- **Paneles:** Contiene los estados.

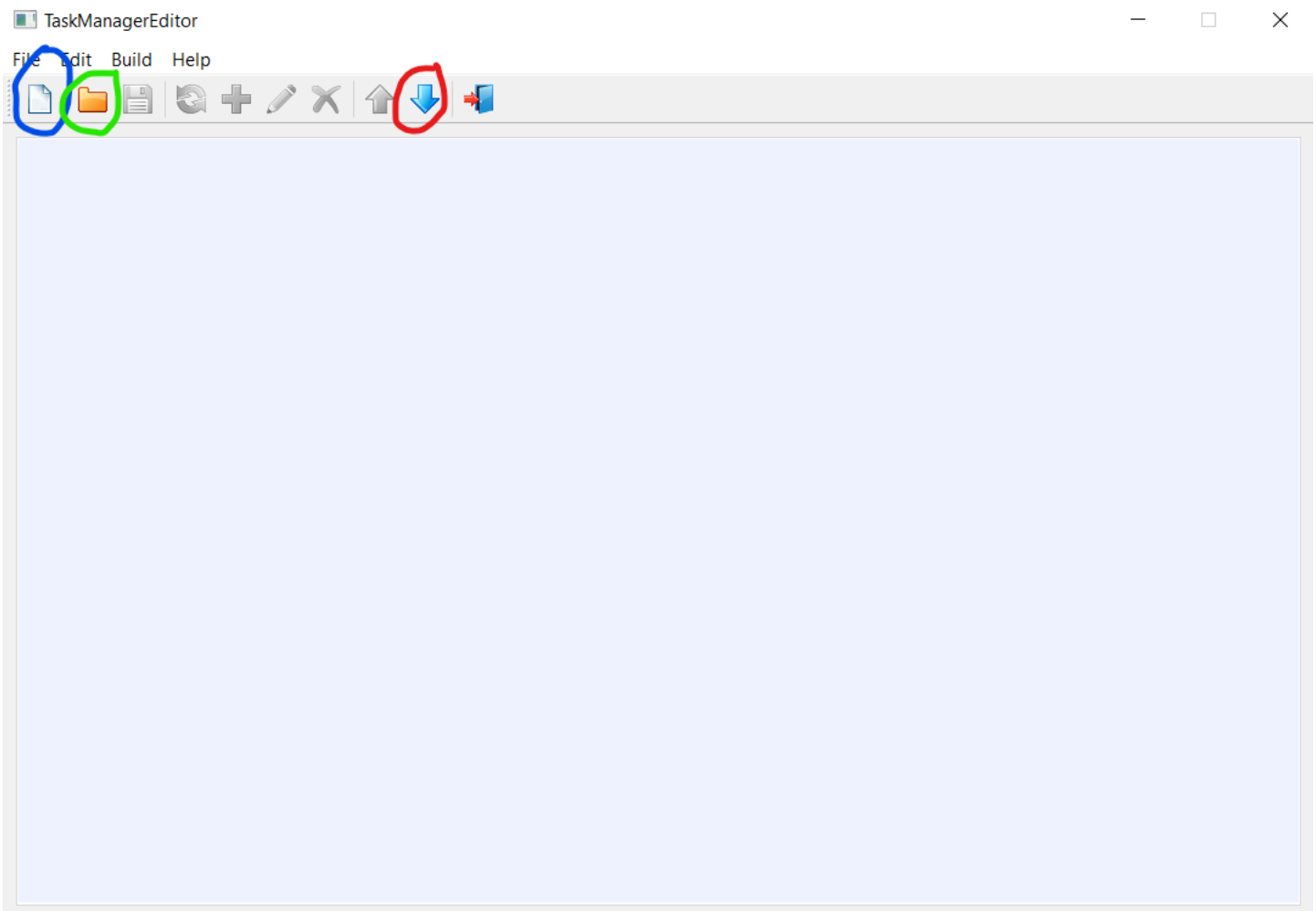
Conocer este modelo de datos es indispensable para el correcto uso de la misma, ya que lo tendremos que tener presente a utilizar la herramienta.

Manual de usuario

En primer para ejecutar la aplicación debemos ir a la siguiente ruta (de los archivos descargados) y ejecutar el archivo "**TaskManagerEditor.exe**".

binaries\bin\TaskManagerEditor.exe

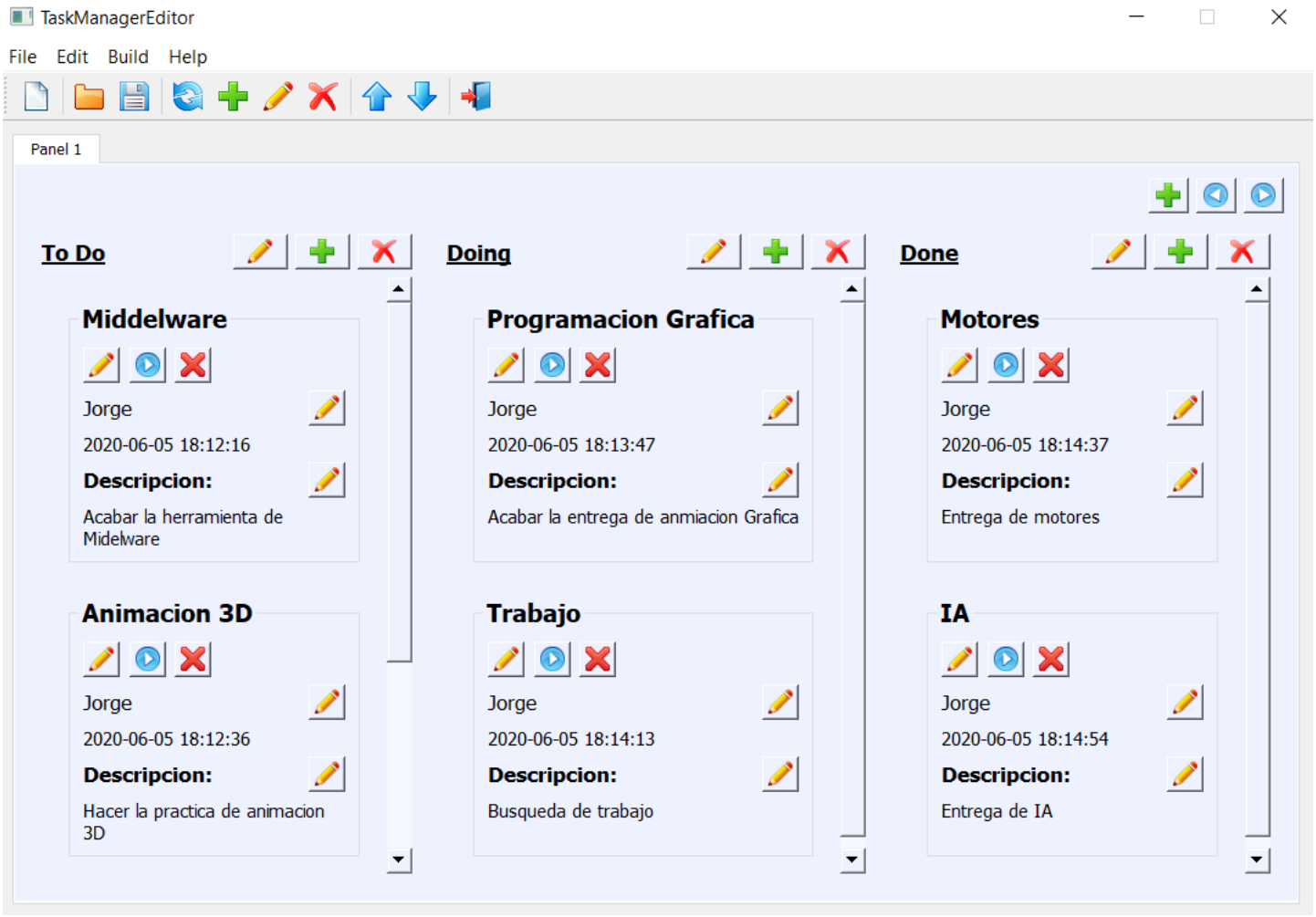
Una vez ejecutado el siguiente archivo se nos presentará la siguiente pantalla:



Si hacemos click sobre el botón redondeado de azul, se nos creara un archivo nuevo. También podemos cargar un archivo XML previamente exportado por la aplicación. Nosotros elegiremos el botón redondeado verde, para cargar un archivo ya existente. El archivo que queremos cargar se encontrara en la siguiente ruta:

binaries\assets\ExampleTab.sav

Una vez escogido el siguiente archivo, el panel debería mostrarse de la siguiente forma



A continuación, se detallan las funciones de cada botón de la barra de tareas

1. Nuevo archivo
2. Abrir archivo
3. Guarda el archivo
4. Refresca todo el panel (en caso de que el archivo se haya modificado desde fuera)
5. Añade un nuevo panel
6. Edita el nombre del panel activo
7. Elimina un panel
8. Exporta el proyecto a un archivo XML
9. Importa los datos de un archivo XML
10. Salir de la aplicación

A continuación, se detallan las funciones de cada botón de la barra paneles

1. Añadir un nuevo estado
2. Mover un estado hacia delante (En caso de que haya más estados)
3. Mover un estado hacia detrás(En caso de que haya más estados)

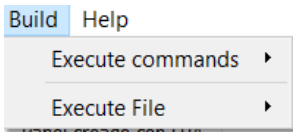
A continuación, se detallan las funciones de cada botón de la barra estados

1. Editar título de un estado
2. Añadir una tarea al estado
3. Eliminar el estado

A continuación, se detallan las funciones de cada botón del panel de tareas

1. Editar el título de la tarea
2. Cambiar la tarea de estado
3. Eliminar la tarea
4. Editar la persona a la que esta asignada
5. Editar la descripción de la tarea

También hay una serie de funciones que se pueden utilizar para la automatización de tareas mediante el lenguaje de script LUA. Estas funciones se encuentran en el menú de "build".



1. Ejecutar una línea de comando con LUA
2. Ejecutar un archivo LUA

Para probar esta característica podemos ejecutar el script de LUA situado en la ruta:

binaries\assets\scriptLua.LUA

Este archivo nos creará un panel mediante LUA.

Scripting con LUA

La herramienta permite ejecutar alguna serie de tareas con el lenguaje de scripting LUA. Las tareas que se pueden ejecutar desde LUA son las siguiente:

```
importPanelAsXML(XMLpath)
exportDataAsXML(XMLPath)
importPanel(SAVPath)
exportData(SAVPath)
addState(NewStateName)
addTaskToState(StateName, TaskTitle, TaskDescription, TaskAssigned)
removeTaskOfState(StateName, TaskTitle)
changeTaskToState(StateName, TaskTitle)
removeState(StateName)
createPanel(StateName)
changeToPanel(StateName)
```

```
getTaskFromState(StateName)
getStatesFromPanel(StateName)
getAllPanels()
```