

UNIVERSIDAD RAFAEL LANDÍVAR
FACULTAD DE INGENIERÍA
SISTEMAS OPERATIVOS



PROYECTO SISTEMAS OPERATIVOS

CARLOS RODOLFO SANTISTEBAN GRAMAJO 1559419

FERNANDO ENRIQUE QUIÑONEZ GARCIA 1518619

MARCO JAVIER DE LEÓN VASQUEZ 1521719

LUIS MARIANO GUTIERREZ DIVAS 1535719

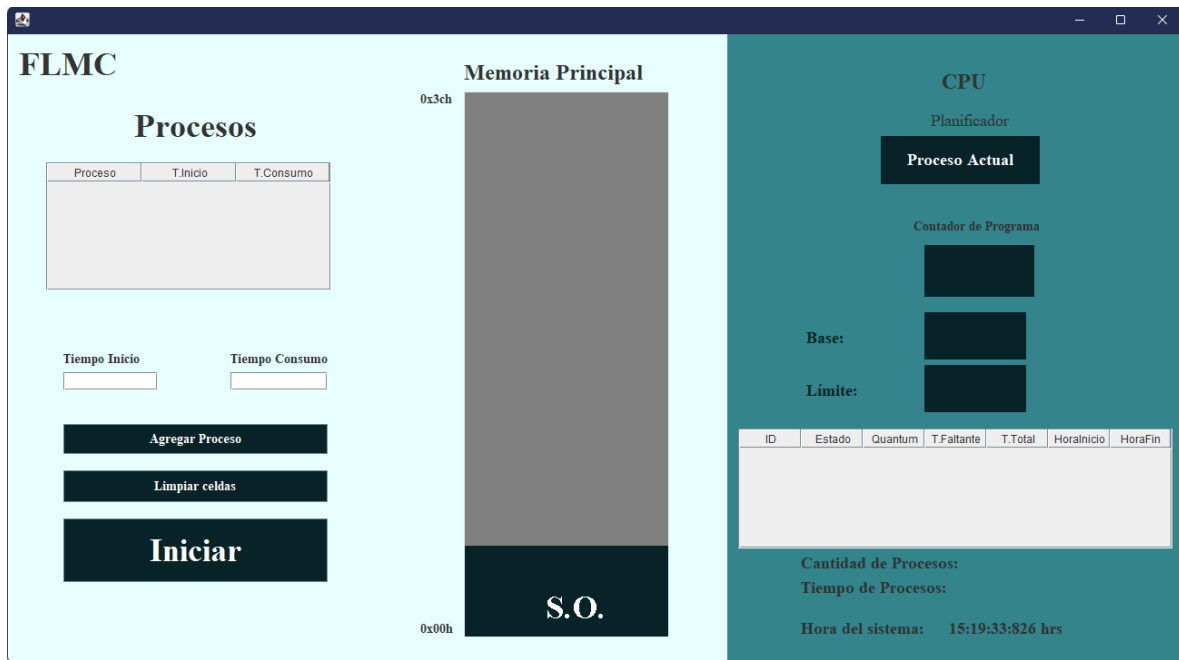
MARÍA LUISA COS ALVAREZ 15675-19

QUETZALTENANGO, ENERO DE 2023

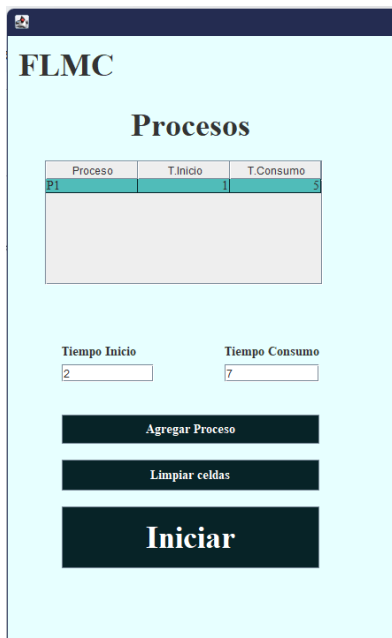
Capturas del funcionamiento

Interfaz principal del programa

Esta se divide en 3 bloques que son: Procesos, Memoria Principal y CPU

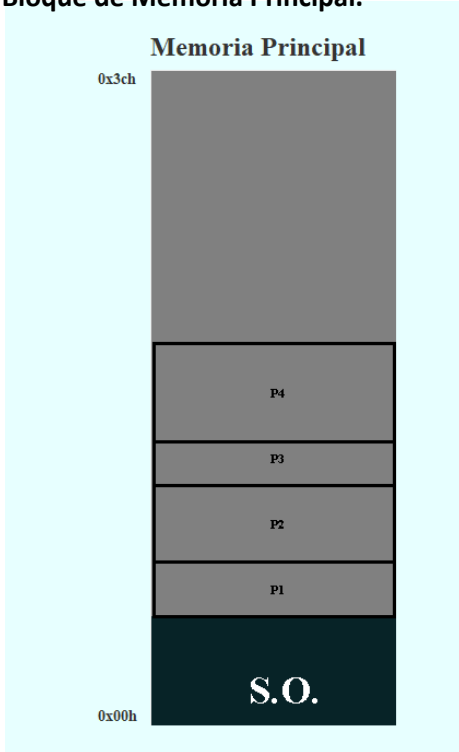


Bloque de procesos:



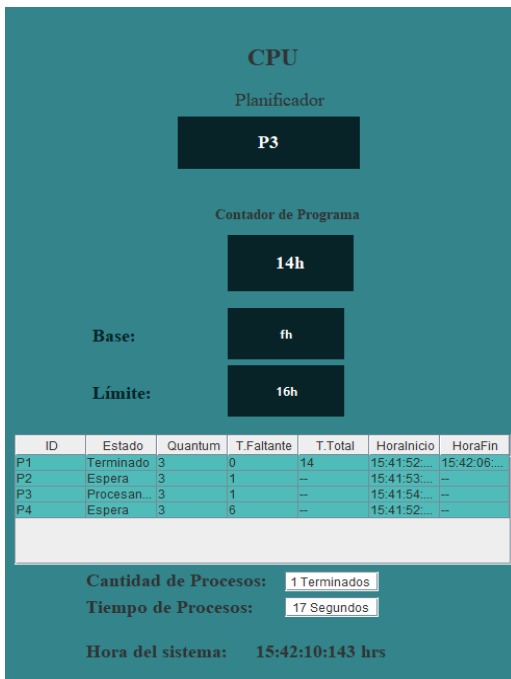
- En este bloque es en el que podemos agregar el tiempo de inicio y el tiempo de consumo de cada proceso que se va a ingresar, llenando los campos indicados y pulsando el botón “Agregar Proceso”.
- Al pulsar el botón “Limpiar celdas” la tabla de procesos se vaciará, limpiando el espacio de memoria que estaba ocupado.
- Al pulsar el botón “Iniciar” todos los procesos que estén en la tabla se empezará a simular el planificador por turno circular (Round-robin RR).

Bloque de Memoria Principal.



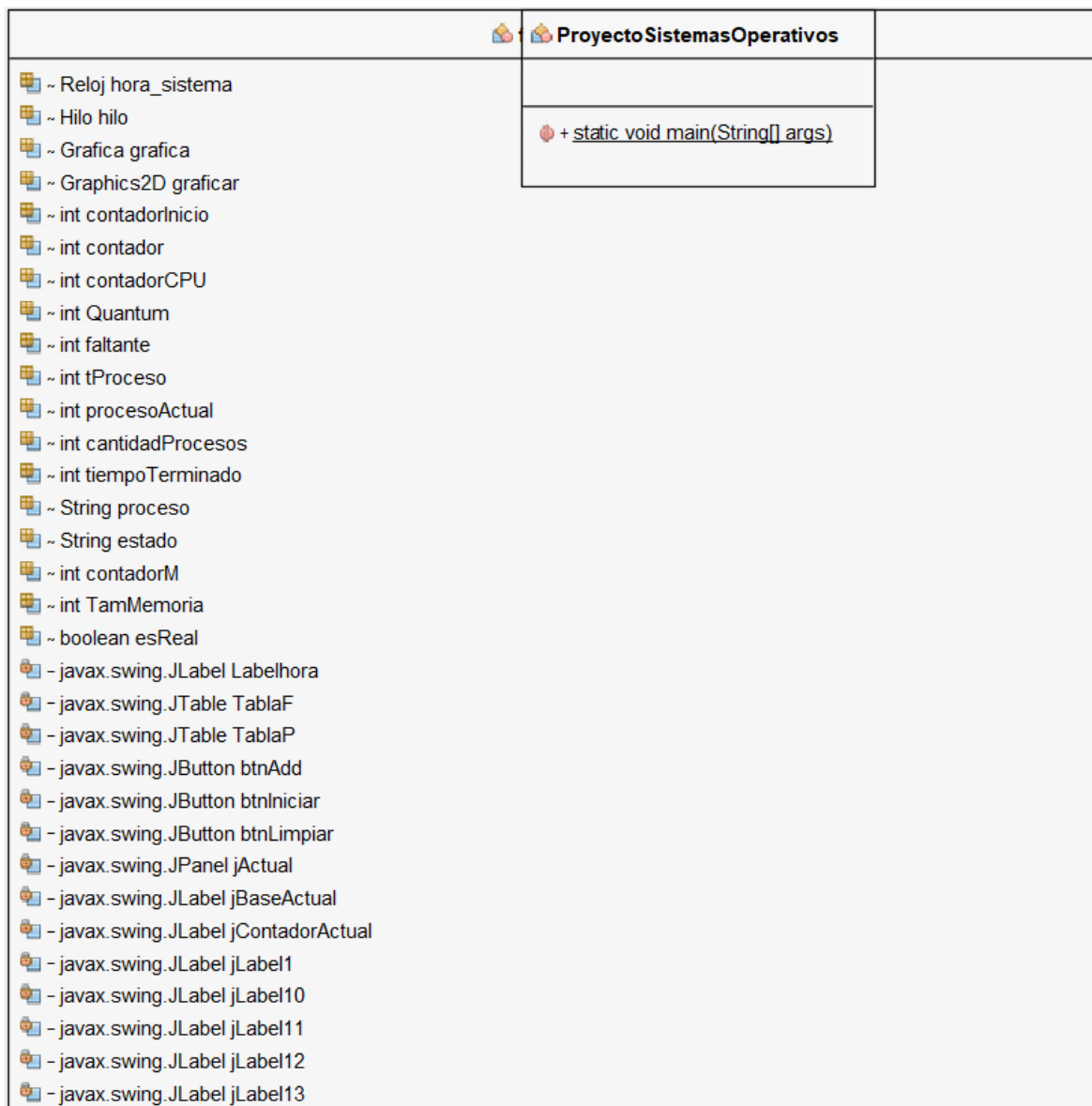
- En este bloque se irán apilando los procesos en el orden en que se van agregando a la tabla de procesos.
- Cada proceso tendrá un bloque con su ID respectivo y el tamaño del bloque dependerá del espacio de memoria que consume cada proceso.
- Nuestra memoria principal tiene la capacidad máxima de 60 direcciones de memoria, de las cuales el Sistema Operativo ocupa 10 de ellas, estas están representadas en base hexadecimal.

Bloque de CPU.




















- En este bloque tenemos una tabla en donde podemos ver el estado de cada proceso, el quantum del planificador (el quantum por defecto es de 3), el tiempo que le hace falta para terminar el proceso, el tiempo total que necesitó el proceso para terminar desde que se inició el simulador, la hora de inicio y final del proceso.
- Tenemos el planificador que nos enseñará el proceso que se está ejecutando, el contador del programa que nos dirá en qué dirección de memoria está apuntando y la base y límite que nos dirá los rangos de las direcciones de memoria que tiene el proceso que se está ejecutando

Diagrama de clases



-  - javax.swing.JLabel jLabel4
-  - javax.swing.JLabel jLabel5
-  - javax.swing.JLabel jLabel6
-  - javax.swing.JLabel jLabel7
-  - javax.swing.JLabel jLabel8
-  - javax.swing.JLabel jLabel9
-  - javax.swing.JLabel jLimiteActual
-  - javax.swing.JPanel jPGrafica
-  - javax.swing.JPanel jPanel1
-  - javax.swing.JPanel jPanel2
-  - javax.swing.JPanel jPanel3
-  - javax.swing.JPanel jPanel4
-  - javax.swing.JPanel jPanel6
-  - javax.swing.JPanel jPanel7
-  - javax.swing.JLabel jProcesoActual
-  - javax.swing.JScrollPane jScrollPane1
-  - javax.swing.JScrollPane jScrollPane2
-  - javax.swing.JTextField jtTerminados
-  - javax.swing.JTextField jtTiempoProcesos
-  - javax.swing.JLabel lblhorasistema
-  - javax.swing.JTextField txtInicio
-  - javax.swing.JTextField txtTiempo

-  + frmMain()
-  ~ void Organizar()
-  ~ void agregar()
-  ~ void limpiar()
-  ~ void revisarBase(int i)
-  - // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()
-  - void btnAddActionPerformed(java.awt.event.ActionEvent evt)
-  - void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt)
-  - void btnIniciarActionPerformed(java.awt.event.ActionEvent evt)
-  - void TablaFComponentShown(java.awt.event.ComponentEvent evt)

-  + static void main(String args)
-  + void HoralInicio(int i)
-  + void Informar(int i)
-  + void RevisarListo()
-  + void Revisar(int i)
-  + void Dormir()
-  + void paint(Graphics g)

Enlace del Repositorio de Git

<https://github.com/carlosrsg11/ProyectoSistemasOperativos>