

Manual de usuario - Planificador de CPU

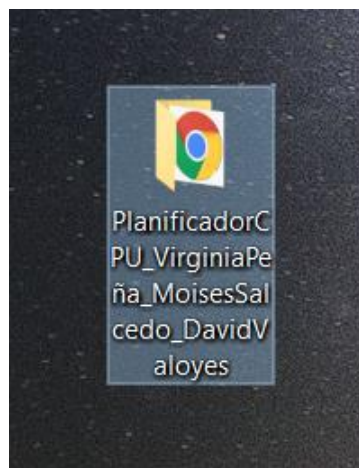
Requisitos previos

- Tener instalado Java 8 como mínimo.
- Tener instalado el IDE Eclipse.

Configuración del proyecto

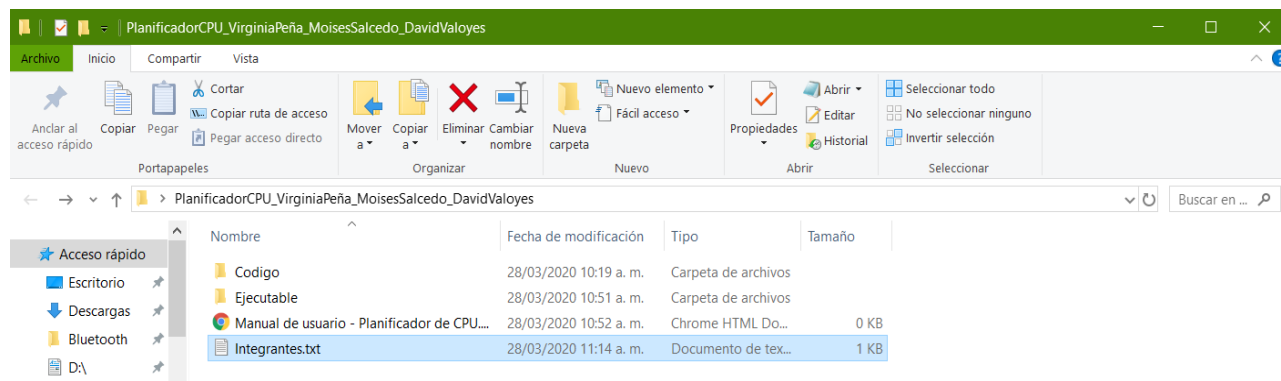
1.

El entregable subido al aula consiste en un archivo con extensión “.zip” la cual se titula “PlanificadorCPU_VirginiaPeña_MoisésSalcedo_DavidValoyes.zip” se encuentran 2 carpetas, un archivo de texto y un archivo PDF. El primer paso consiste en ubicar dicho archivo en el escritorio y descomprimirlo, de tal manera que queda una carpeta de la siguiente manera:



2.

Dentro de esta carpeta el contenido se vera de la siguiente manera:



3.

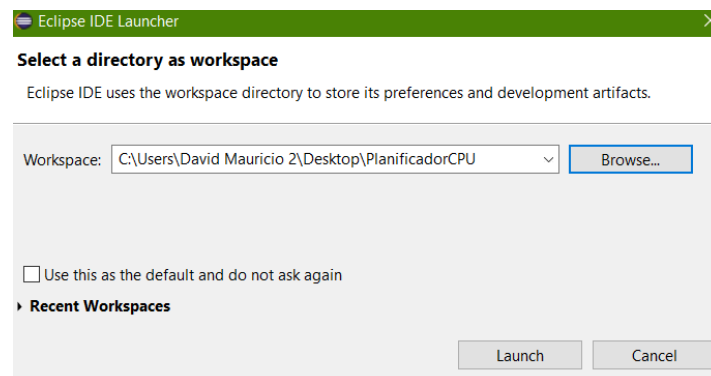
En el escritorio se debe de crear una carpeta titulada “PlanificadorCPU”, la cual va a funcionar como workspace del proyecto.



4.

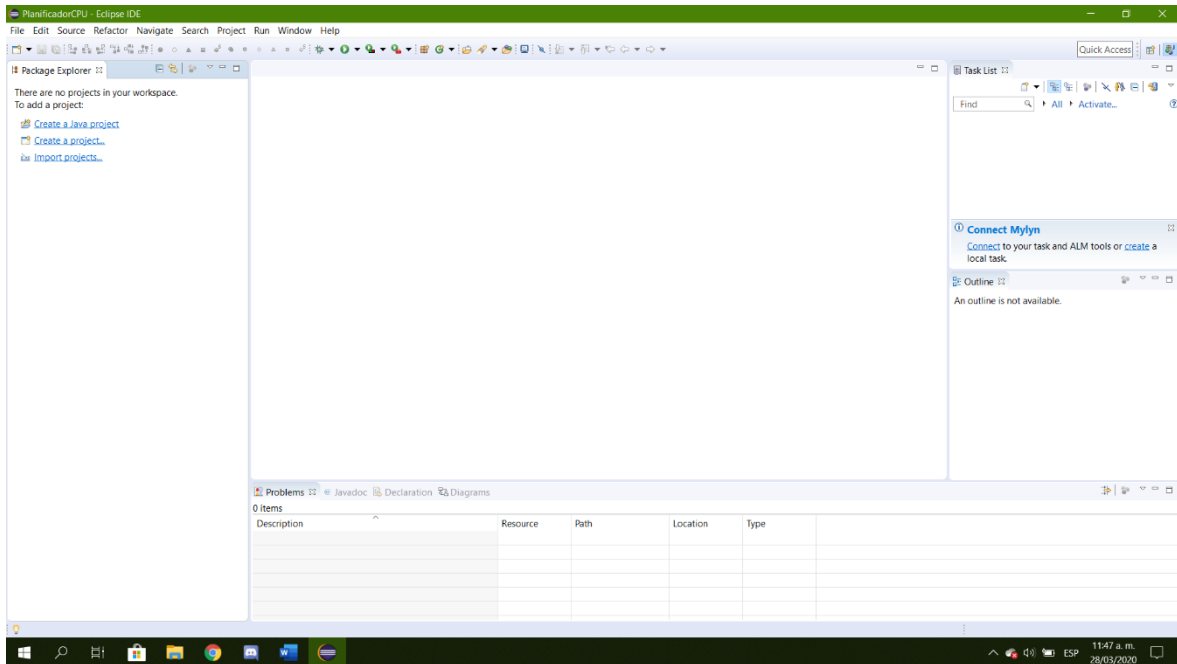
A continuación, se debe de ejecutar el IDE Eclipse de la siguiente manera:

1. Se ejecuta el IDE Eclipse.
2. Se selecciona como workspace la carpeta creada en el paso 3 de este manual.



3. Se da click en “Launch”.

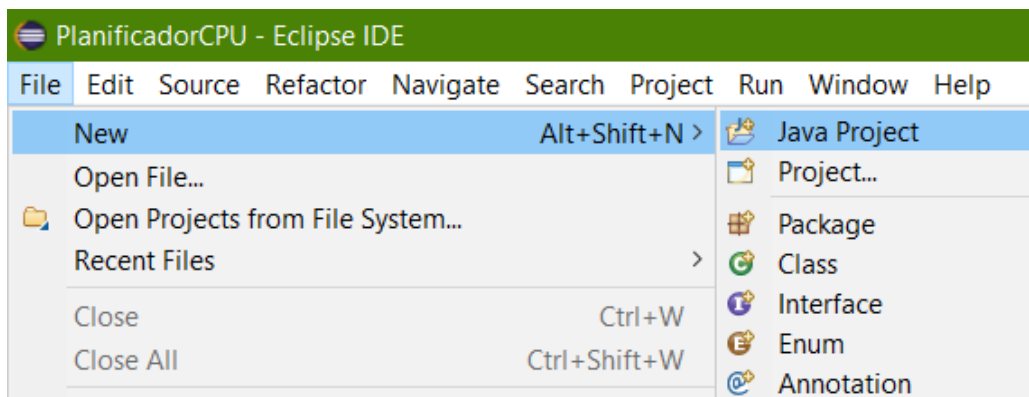
Posterior a estos pasos se va a abrir la interfaz grafica de eclipse la cual ha de verse de la siguiente manera dependiendo de su versión de IDE:



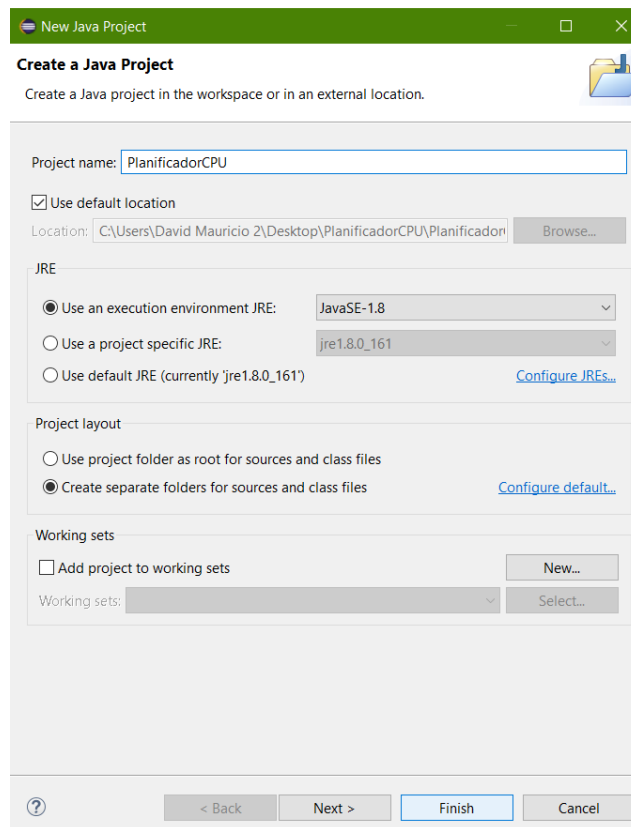
- 5.

A continuación, se debe de crear un nuevo proyecto Java el cual se debe titular “PlanificadorCPU”, para lo cual se debe de realizar los siguientes pasos:

1. Seleccionar “File”.
2. Seleccionar “New”.
3. Dar click en “Java Project”.

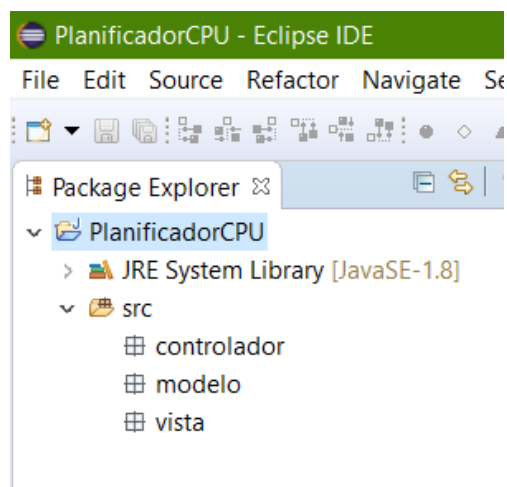


Posterior a estos pasos se va a abrir una ventana para configurar el proyecto, donde lo único que se debe hacer es ingresar el nombre del proyecto (PlanificadorCPU) y dar click en “Finish”.



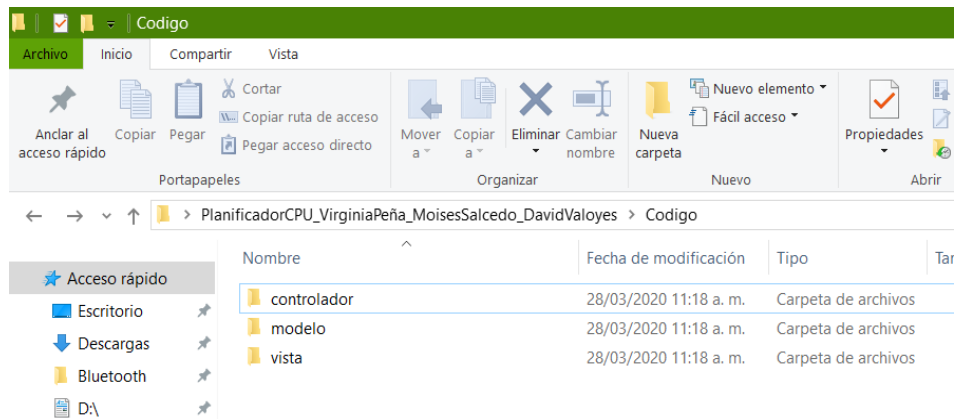
6.

A continuación, dentro de la carpeta “src” del proyecto se deben de crear 3 paquetes titulados: modelo, vista y controlador respectivamente, de tal forma que luego de crearlos el proyecto se ha de ver así:

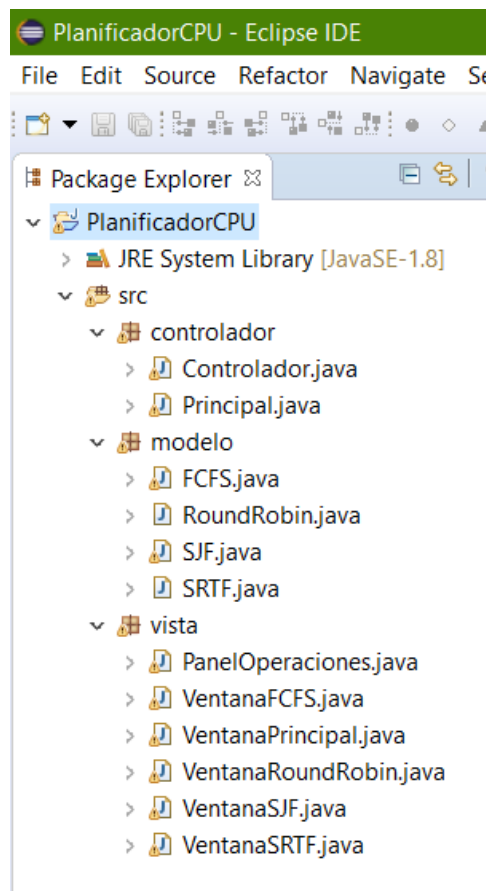


7.

Dentro de la carpeta descomprimida en el primer paso se encuentra una carpeta titulada código, dentro de la cual se encuentran a su vez tres carpetas tituladas: modelo, vista y controlador, donde a su vez tienen cada una sus respectivos archivos con extensión “.java”.



Se deben de copiar cada uno de los archivos de estas carpetas en su respectivo paquete dentro del proyecto (Es decir, el paquete con el mismo nombre de la carpeta donde se encontraban dichos archivos), de tal forma que posterior a esto el proyecto debe de verse así:

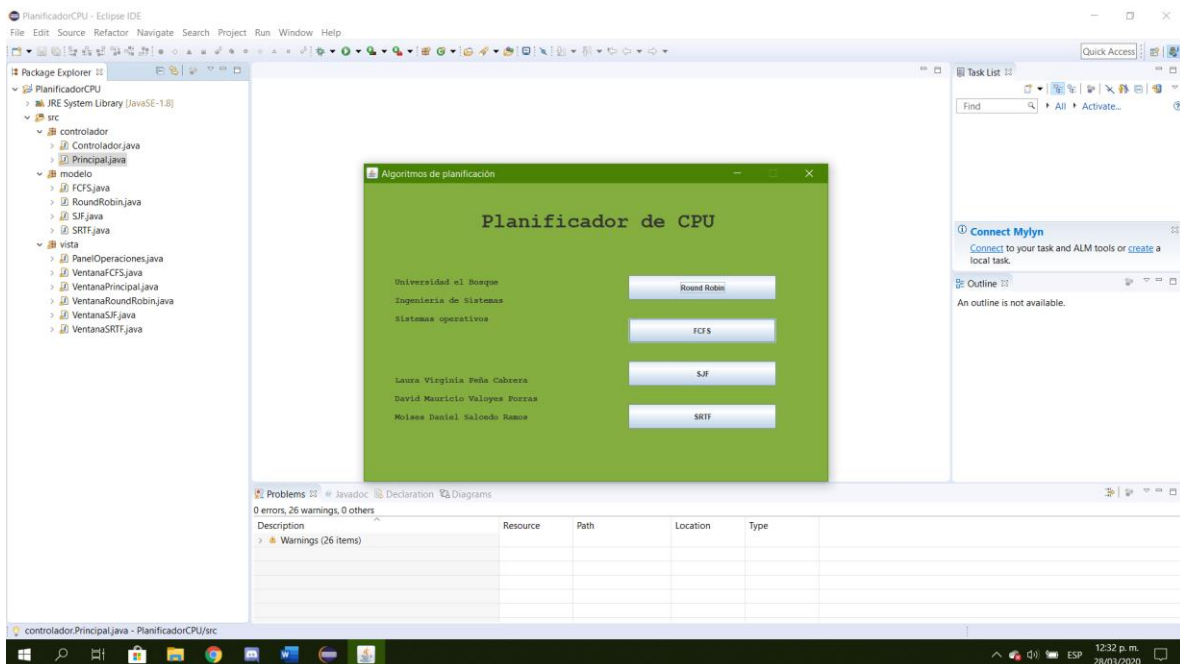


Ejecución del programa

El proyecto se puede ejecutar de dos maneras, la primera de estas es mediante el IDE Eclipse, o la segunda es mediante el archivo ejecutable de extensión “.jar”.

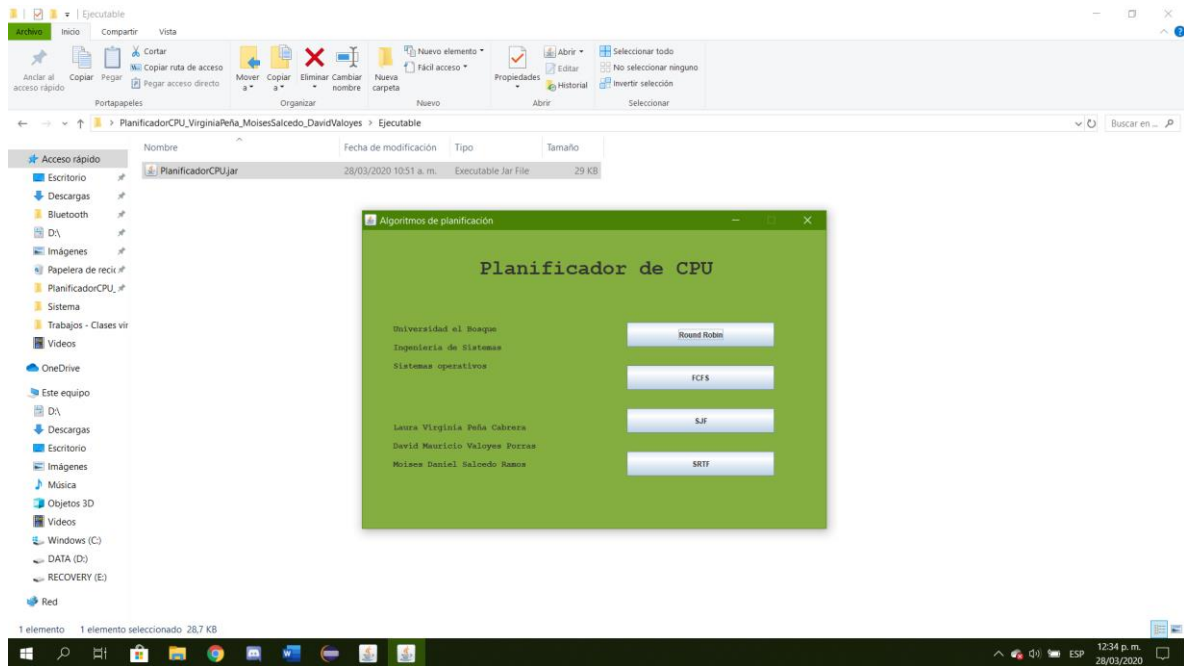
Ejecución mediante el IDE Eclipse:

Se debe de ejecutar la clase llamada “Principal” la cual pertenece al paquete controlador. En esta clase se encuentra el método “Main” del proyecto. Al ser ejecutada se debe ver algo así:



Ejecución mediante el archivo ejecutable:

Dentro de la carpeta descomprimida en el primer punto de este manual se encuentra una carpeta titulada “Ejecutable”, la cual a su vez tiene el archivo ejecutable llamada “PlanificadorCPU-Ejecutable.jar”. Para ejecutarlo solo es necesario hacer doble click sobre este archivo. Al ser ejecutado se debe ver algo así:



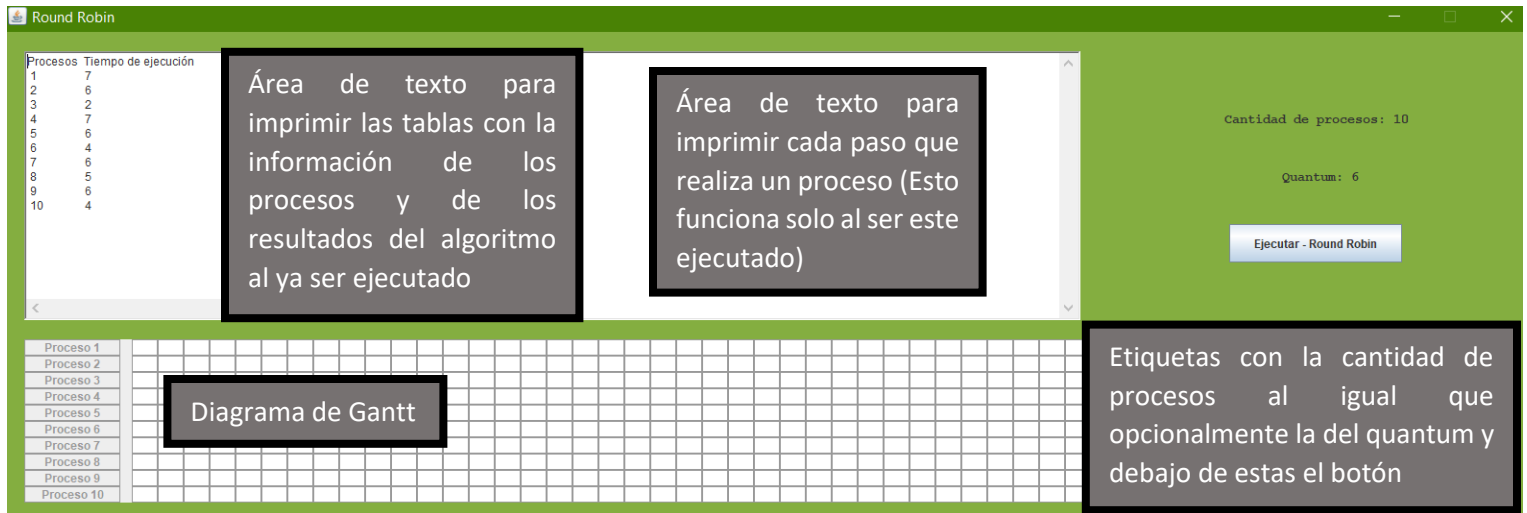
Explicación del funcionamiento general del programa

La interfaz gráfica principal del programa consiste en una ventana que posee el nombre de la universidad, el nombre del programa, el nombre de la materia y el nombre de los 3 integrantes del grupo, al igual que 4 botones donde se ejecuta la interfaz de cada uno de los ejercicios en el siguiente orden: Round Robin, FCFS, SJF y SRTF.

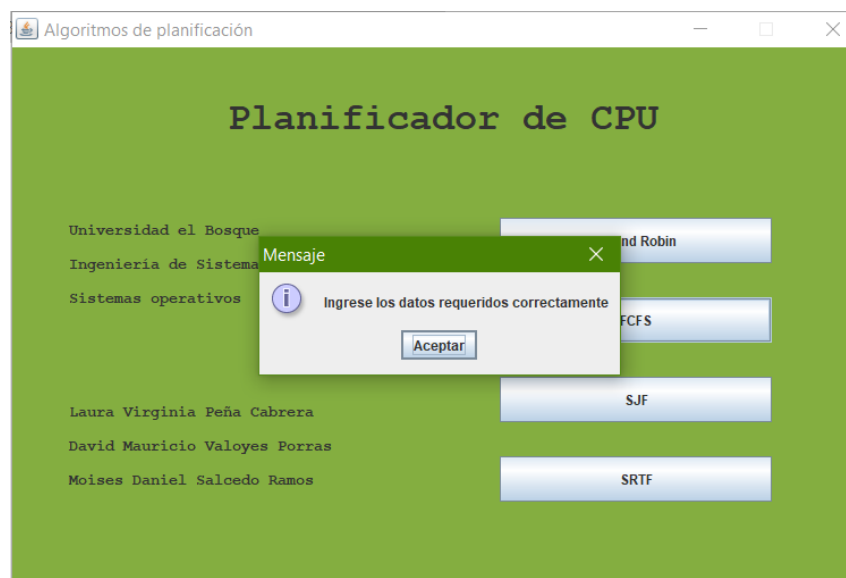


La interfaz gráfica de cada uno de los ejercicios es prácticamente la misma puesto que consiste en dos áreas de texto donde en la que se ubica al lado se imprime la tabla de resultados y al lado

derecho se imprimen los procesos (O pasa) ocurridos durante la ejecución del algoritmo, al lado de estas se encuentra una etiqueta con la cantidad de procesos a ejecutar (Y en el caso de Round Robin también hay una etiqueta con el quantum utilizado), debajo de la etiqueta se encuentra el botón para ejecutar el algoritmo y en la parte inferior de la interfaz gráfica se encuentra el diagrama de Gantt que representa visualmente la ejecución del algoritmo. En términos generales la interfaz gráfica de los algoritmos está constituida de la siguiente manera:



Para ejecutar un algoritmo debe de seleccionarse en la interfaz grafica principal el algoritmo a ejecutar, donde al momento de seleccionarlo va a emerger una ventana en la cual se debe ingresar la cantidad de procesos a ejecutar, los cuales por efectos prácticos del ejercicio y de la organización de la interfaz gráfica solamente se puede ejecutar mínimo 1 proceso y máximo 10. La información ingresada en esta ventana se valida de tal forma que no se pueden ingresar menos de un proceso o mas de 10, al igual que no se pueda ingresar letras, números decimales, caracteres especiales o simplemente dar click en aceptar sin que se haya ingresado información alguna.



Al momento de ingresar correctamente la información sobre la cantidad de procesos que desea ejecutar, se carga la interfaz grafica del algoritmo deseado, en donde se puede visualizar que:

1. En el área de texto destina a las tablas de los algoritmos, se imprimirá la información referente a el estado inicial de cada proceso, es decir, los parámetros iniciales de estos, los cuales dependen de cada algoritmo.
2. El diagrama de Gantt se va a dividir en filas y columnas en función de la cantidad de procesos y de sus respectivos tiempos de ejecución, donde cada fila representa un proceso diferente y cada columna representa un ms de tiempo.
3. En la etiqueta de “Cantidad de procesos” se visualiza la cantidad de procesos que ingresó el usuario. Para el caso de Round Robin también se actualiza la información sobre el quantum de tiempo.

Posteriormente, al momento de ya haber ejecutado un algoritmo, en el cuadro de texto destinado al proceso que ejecuta cada algoritmo, se puede visualizar de manera textual los pasos de la ejecución de todos los procesos, de tal forma que se hace más entendible que fue lo que paso con cada proceso durante la ejecución del algoritmo, por su parte el diagrama de Gantt colorea el o los intervalos de ejecución de cada uno de los procesos (Los colores se configuran automáticamente) y el botón de ejecución se bloquea automáticamente.

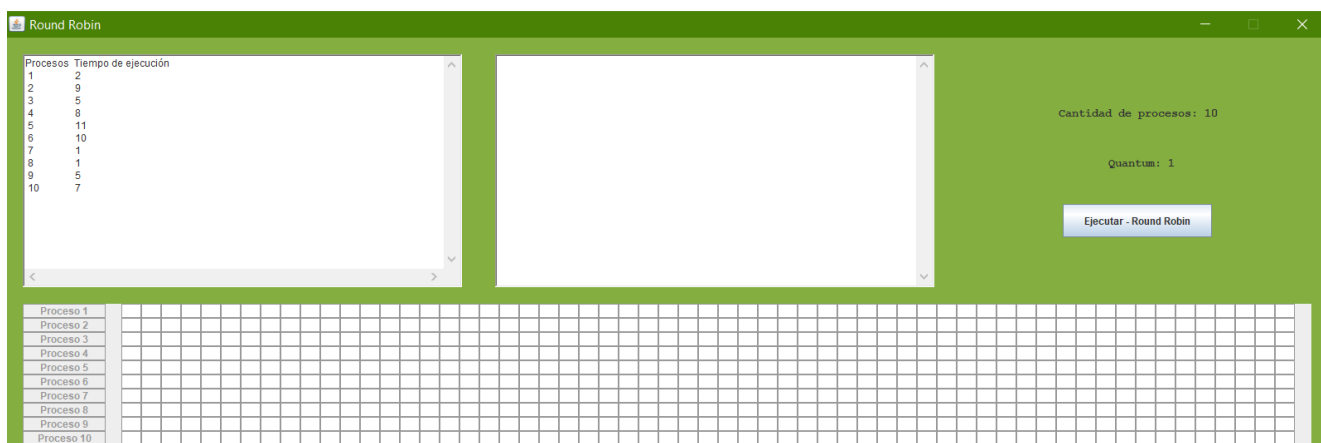
Explicación del funcionamiento de cada algoritmo

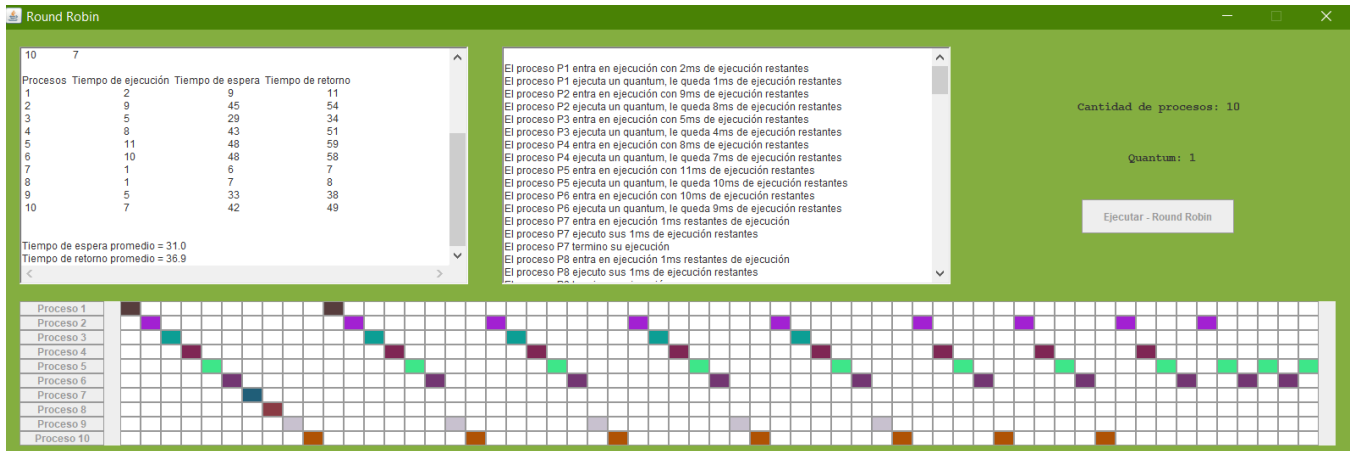
Para efectos demostrativos, en este manual cada algoritmo se ejecutará con la cantidad máxima de procesos permitidos por el programa (10 procesos).

1. Round Robin:

Los parámetros iniciales de cada proceso se configuran automáticamente mediante el uso de números enteros aleatorios de la siguiente manera:

- El tiempo de ejecución de cada proceso es un aleatorio entre 1 y 12.
- El quantum es un aleatorio entre 1 y 7 puesto que un quantum más grande podría hacer que en varios casos la ejecución del algoritmo se asemejara más a un FCFS.
- El tiempo de llegada para cada proceso es 0.

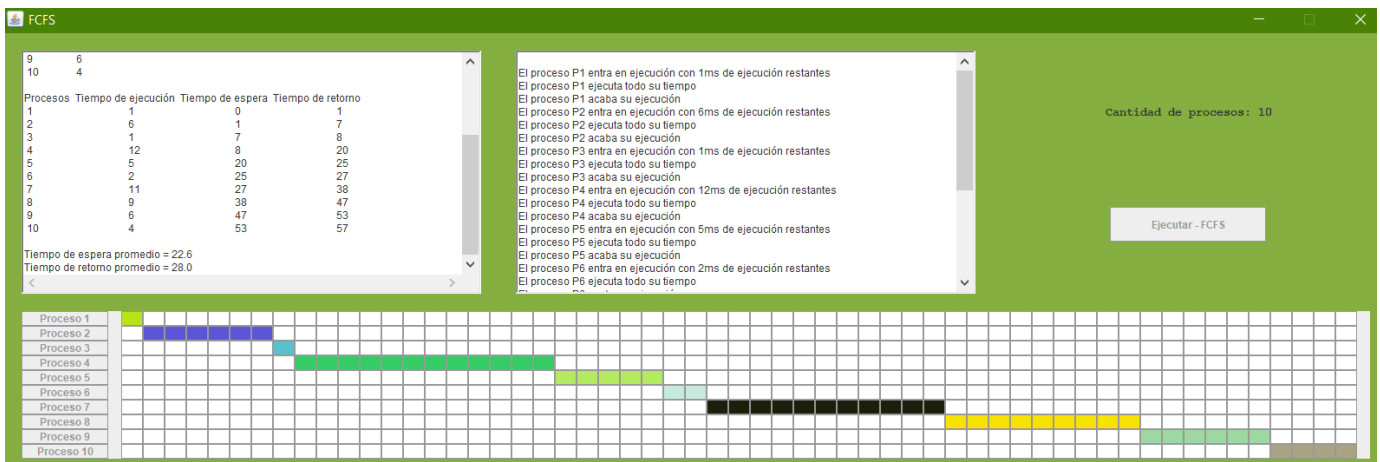
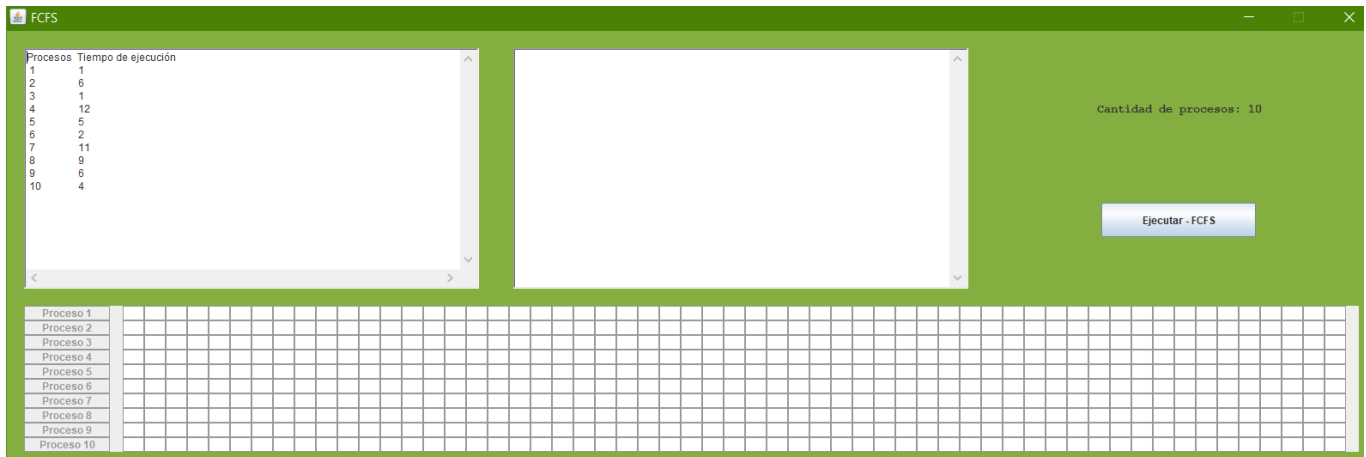




2. FCFS (First Come First Served):

Los parámetros iniciales de cada proceso se configuran automáticamente mediante el uso de números enteros aleatorios de la siguiente manera:

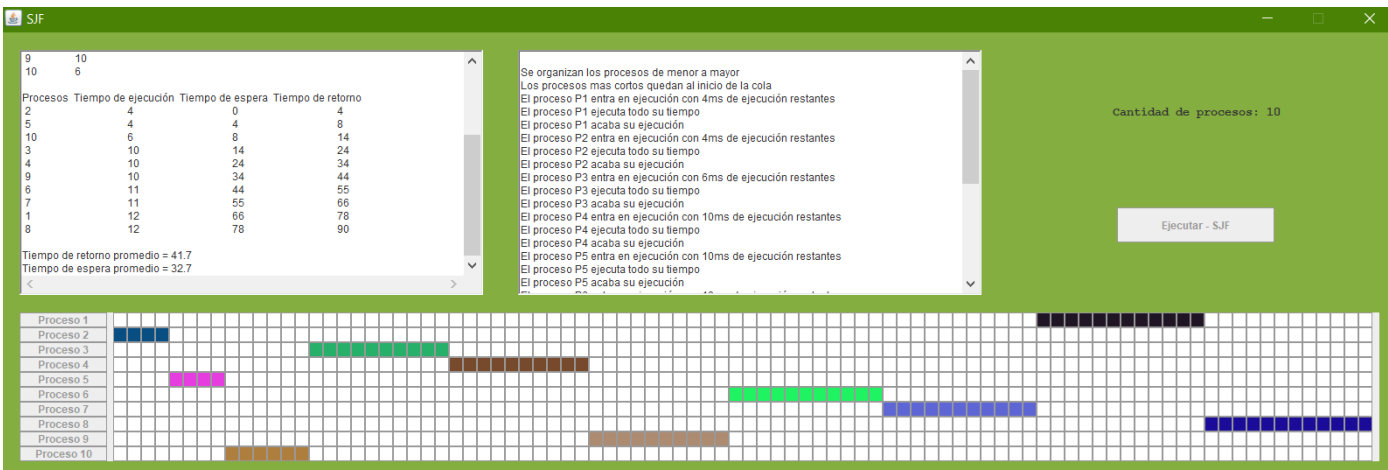
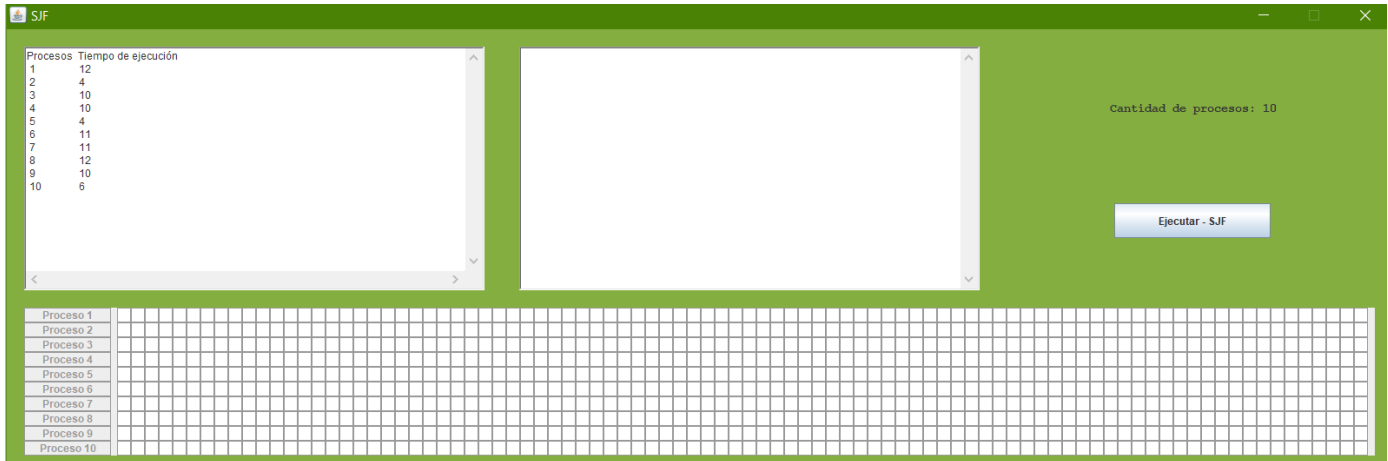
- El tiempo de ejecución de cada proceso es un aleatorio entre 1 y 12.
- El tiempo de llegada para cada proceso es 0.



3. JSF (Shortest Job First):

Los parámetros iniciales de cada proceso se configuran automáticamente mediante el uso de números enteros aleatorios de la siguiente manera:

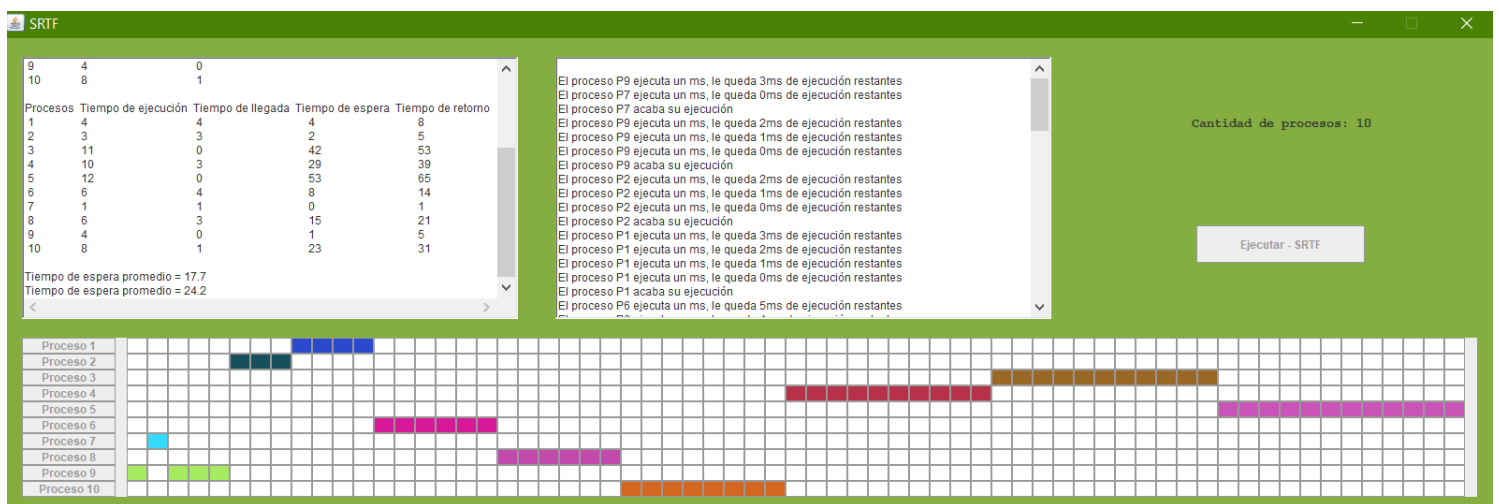
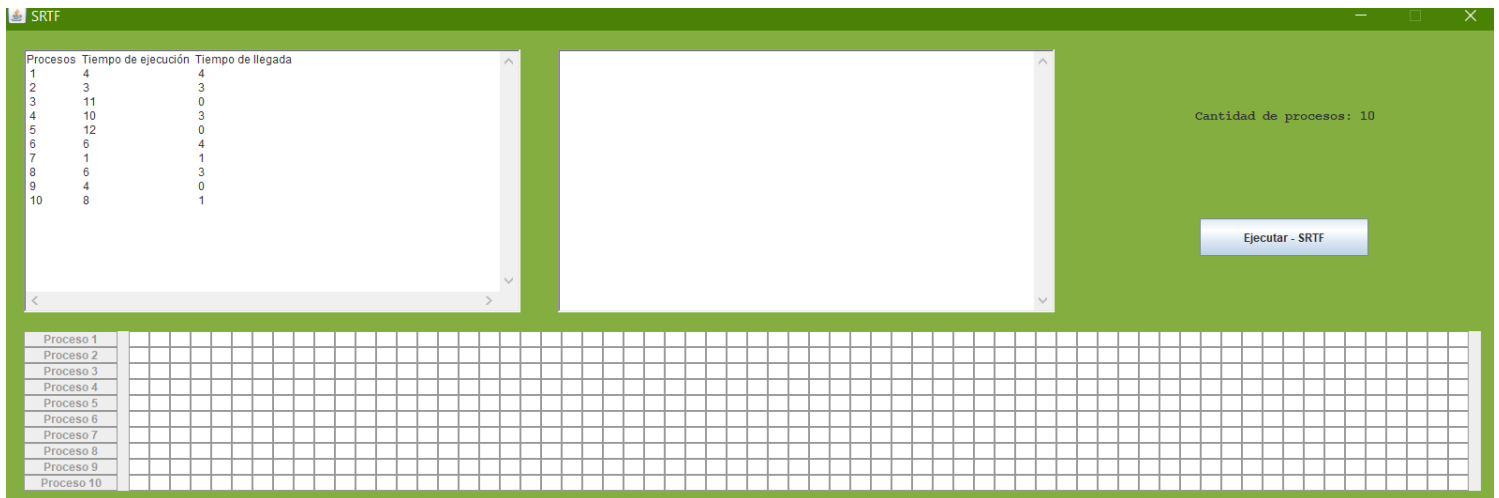
- El tiempo de ejecución de cada proceso es un aleatorio entre 1 y 12.
- El tiempo de llegada para cada proceso es 0.



4. SRTF (Shortest Remaining Time First):

Los parámetros iniciales de cada proceso se configuran automáticamente mediante el uso de números enteros aleatorios de la siguiente manera:

- El tiempo de ejecución de cada proceso es un aleatorio entre 1 y 12.
- El tiempo de llegada para cada proceso es un aleatorio entre 1 y 5.



Como se puede evidenciar, cada algoritmo al finalizar su ejecución presenta toda la información relevante de dicha ejecución:

1. En el área de texto del lado izquierdo se visualiza la tabla con los respectivos resultados, es decir el tiempo de espera y el tiempo de retorno de cada proceso, al igual que el tiempo de espera y el tiempo de retorno promedio de todo el ejercicio.
2. En el área de texto del lado izquierdo se visualizan los pasos que cada proceso realizó en forma textual, especificando línea por línea lo que estaba ocurriendo durante la ejecución del algoritmo, es decir, que proceso entro, cuantos ms de tiempo ejecuto y cuando finalizo.
3. En el diagrama de Gantt se visualiza de forma grafica la ejecución de cada proceso, estando estos representados por cada fila del diagrama y evidenciando mediante el uso de diferentes colores su o sus intervalos de ejecución. Cabe aclarar que cada color se configuro automáticamente por parte del programa utilizando el formato RGB, de tal forma que cada parámetro de RGB (Es decir, el parámetro red, el parámetro green y el parámetro blue) consiste en un aleatorio que se ubica entre 1 y 255, posterior a esto a cada color se le asigno un numero de proceso utilizando un arreglo de una dimensión, de tal forma que en el momento que se ejecutaba un proceso, el color correspondiente a ese proceso era el que se ejecutaba utilizaba en el diagrama de Gantt.

4. Los parámetros utilizados, aunque se configuren automáticamente haciendo uso de números enteros aleatorios, corresponden a los parámetros utilizados realmente por cada algoritmo en su aplicación real. El tiempo de llegada de los procesos se tomo como 0 en los algoritmos de Round Robin, FCFS y JSF por efectos académicos, mientras que el SRTF si requería necesariamente que hubiera procesos con tiempos de llegada diferentes puesto que es si consiste en una variación del algoritmo JSF en donde pueden llegar procesos cuyo tiempo de ejecución es menor al tiempo de ejecución restante de un proceso que estaba antes.

Consideraciones finales

Como consideraciones personales, el proceso de realizar tanto este programa como el informe con los 8 ejercicios presentados anteriormente fue pedagógico para comprender el funcionamiento de los 4 algoritmos estudiados, al igual que nos permitió trabajar con un grupo de trabajo (Valga la redundancia) con el que nunca había trabajado ninguno de nosotros, lo cual también es valioso para nuestro proceso formativo.

Como aclaración final de este manual, en ningún momento se hace uso de librerías, todo viene propiamente con Java.