

## PRACTICA 6

### Objetivo General

Reforzar los conocimientos sobre planificadores de procesos, mediante una simulación del algoritmo de Round Robin.

### Objetivos Particulares:

- Llevar a la simulación uno de los algoritmos más representativos de planificación de procesos mediante el uso del lenguaje Java.
- Explotar y poner a prueba todos sus conocimientos de los cursos anteriores de programación en una implementación seria (no un “*piece of cake*”).
- Todo debe recaer en una pequeña aplicación grafica que sea lo más general posible, modelando la llegada de 4 procesos con diferentes arribos y ciclos de ráfagas de uso de CPU,
- Cumplir un *checklists* de especificaciones a revisar que se mostraran mas adelante para obtener una calificación de la aplicación desarrollada (Sistemas Operativos orientado a la Ingeniería de Software)

### Documentos a entregar

Esta práctica se tiene que realizar de forma individual y la parte individual es en serio ya que se pedirán los códigos con anticipación y se comprobará en uno de los puntos del *checklists* que los códigos sean independientes.

En caso de que haya similitudes de más del 70% la practica será considerada como nula.

- El Proyecto Completo ya sea en NetBeans o Eclipse para desarrollar esta práctica (100 % de la calificación). Hay que recordar que todo debe implementarse de forma manual, de lo cual se tiene que explicar con detalle el día de la revisión (si se usa una herramienta para la parte visual eso se omitirá).
- Checklist impreso donde se depositará la calificación de la Practica.

### Plazo de entrega

La práctica se organiza de la siguiente forma en cuestión de la entrega:

1. Envío de proyecto, sábado 22 de junio del 2019.
2. Revisión en el laboratorio, lunes 24 de junio del 2019

**Nota. No se recibirá ninguna práctica fuera de ese horario, sin ninguna excepción.**

**Especificaciones de las partes que conforman la práctica:**

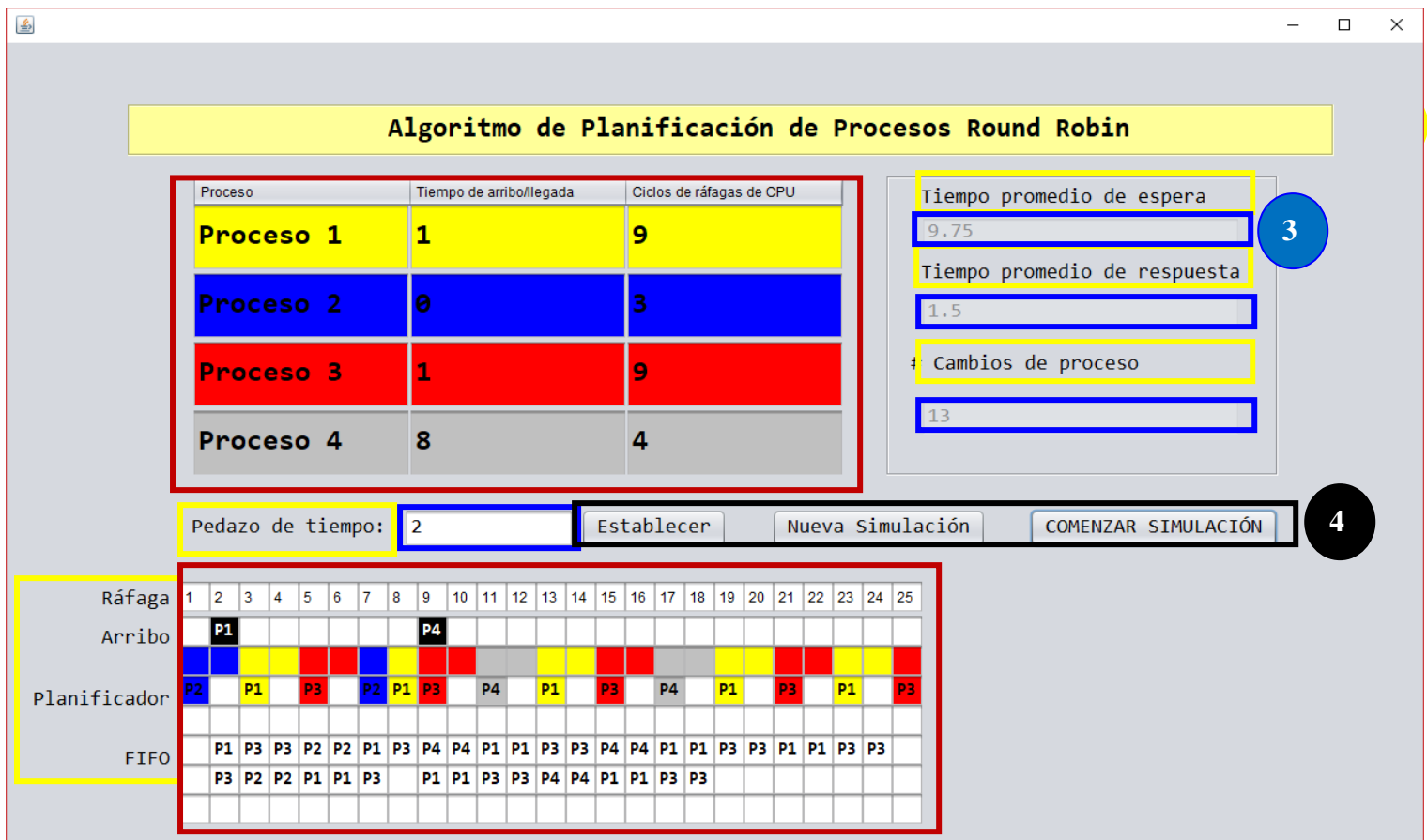
Para esta práctica se necesitan crear un Proyecto en NetBeans o Eclipse con el nombre de Practica6\_NombreCompleto.

Este proyecto debe contener un paquete con el nombre Parte\_Final, dentro de este paquete se deben crear la siguiente clase:

1. Round\_Robin.java

Esto se hace con el fin de tener un estándar en la revisión, por lo que eviten realizar más clases, como crear una Principal.java para solo poner el main. La clase Round\_Robin.java será explicada a continuación

**Round\_Robin.java:** Se propone la aplicación gráfica que se muestra en la Figura 1.



**Figura 1.** Aplicación gráfica para la Practica 5.

De la Figura 1 podemos señalar lo siguiente:

1. Se deben ocupar una serie de JLabel's para darle al usuario una forma mas fluida para el uso de GUI desarrollada. Las JLabel's deben tener un formato adecuado para el tamaño de la ventana, alineadas en la mayor parte posible, resaltadas del resto de los componentes (Negritas), usar un tipo y tamaño de letra estandar (ni muy grandes, ni muy chicas y tipos de letra entendibles). La GUI mostrada en la Figura 1 no es referencia de una GUI de 10, por lo que se podra hacer cosas mejores, pero que sean funcionales (algo que si tiene la GUI de la Figura 1)
2. El círculo de color rojo hace referencia a la area de simulación. Se podra utilizar un JTable, una matriz de JButton's o de JLabel's o de JTextField's según sea la decisión de cada alumno.

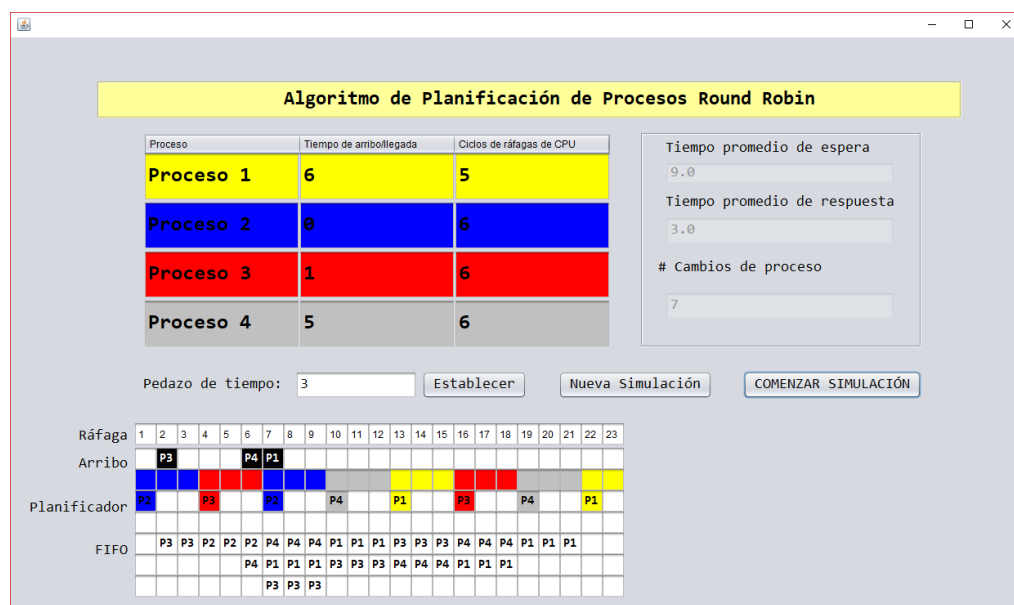
En la primera area de simulación (parte superior) se pondran los datos del proceso: nombre, arribo y numero de rafagas. Esto podra ser modificado en cualquier momento por el usuario para que funcione con cualquier combinación de valorer.

Los valores del arribo estaran en un rango de [0, 10], mientras que los ciclos estaran en un rango de [1 a 10], de manera estricta y seran llenados con un generador de numeros aleatorios (debe ser algo realmente aleatorio).

En la segunda area de simulación (parte inferior) se deben despegar los datos que se aprendieron en la explicación de la Semana 5, los cuales son; rafaga, arribo, casilla de proceso en uso, planificador, FIFO.

De lo anterior se debe implementar una FIFO, pieza central del funcionamiento del algoritmo Round Robin, La FIFO se debe implementar mediante la clase `LinkedList<>()`, no crear una clase más.

El desarrollo de la segunda parte de simulación podra crecer o disminuir de forma aleatoria en cada ejecución del programa, ver la Figura 2.



**Figura 2.** Ajuste automatico por cada ejecución

3. Para mostrar los resultados se ocuparán una serie de JTextField's para desplegar por una parte las medidas de rendimiento del algoritmo y por otra parte permitir el ingreso del pedazo de tiempo (quantum) para cada proceso.

Este ultimo JTextField (pedazo de tiempo) debe contener una serie de excepciones para evitar ingresar un dato que no sea entero, ver Figura 3 y un dato que no este en un rango de [1, 10], ver Figura 4.

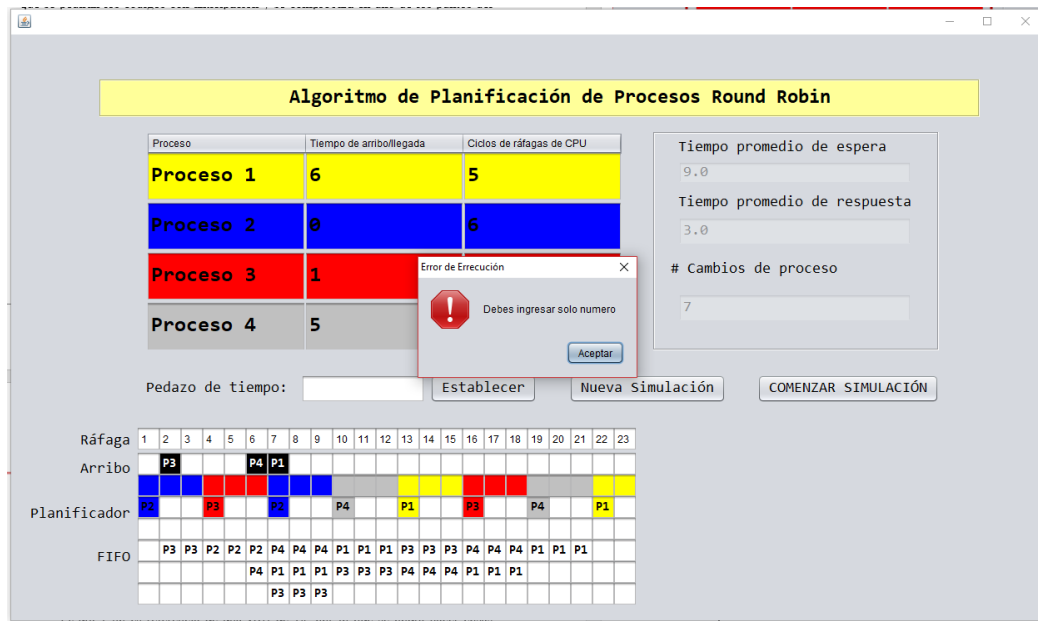


Figura 3. Ingreso de datos que no son numeros.

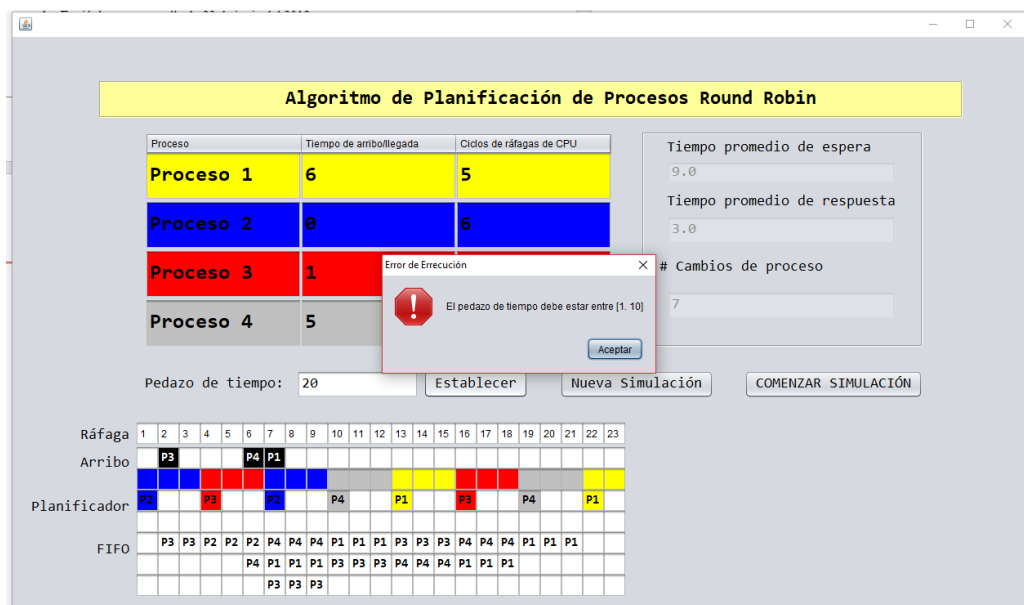
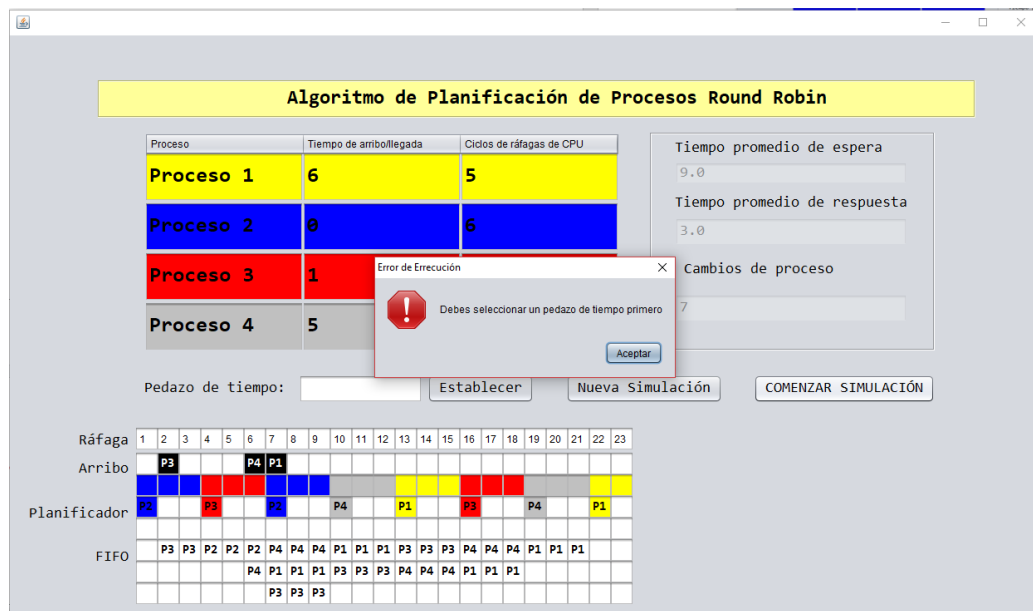


Figura 4. Ingreso de datos que no cumplen el rango establecido.

Hay muchas formas de como cubrir el punto anterior, todas son validas siempre y cuando no salgan de lo establecido por este punto. Para obtener los resultados presentados en la Figura 3 y 4. Se utiliza la interacción con el actionPerformed() del JButton Establecer pero se puede utilizar algo mas novedoso.

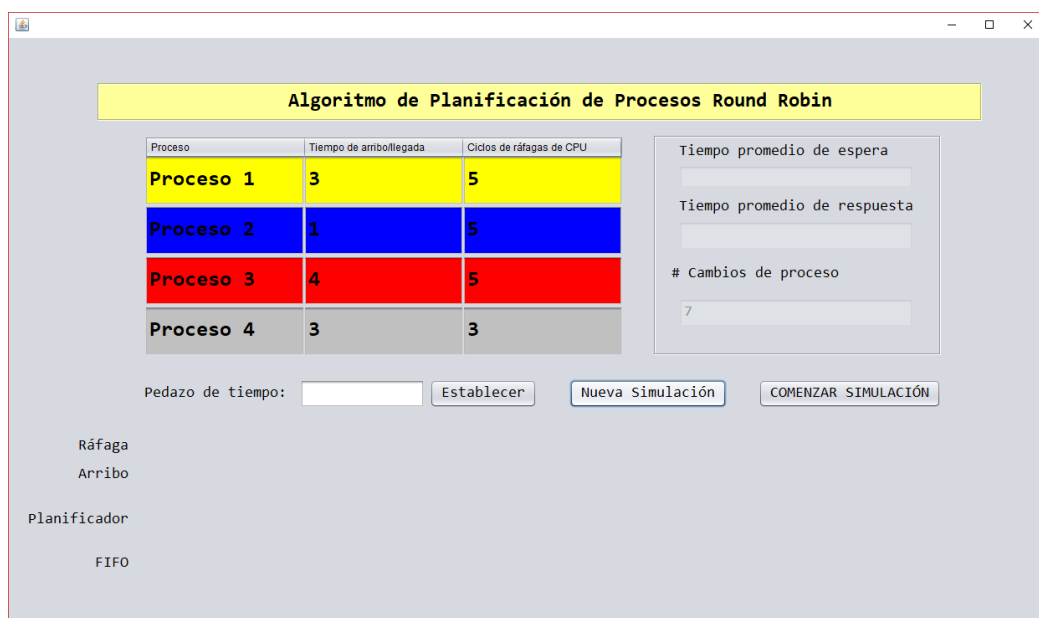
Para los otros tres JTextField's se deben mostrar los resultados del tiempo promedio de espera y respuesta así como los cambios de contexto o procesos.

- Finalmente se necesita implementar un serie de JButton's para darle funcionalidad a la GUI (muy basico pero mas que suficiente). El funcionamiento del JButton Establecer permite realizar la simulación, ya que sin este dato el JButton COMENZAR SIMULACIÓN no realizará nada y mandará un ventana emergente, ver la Figura 5.



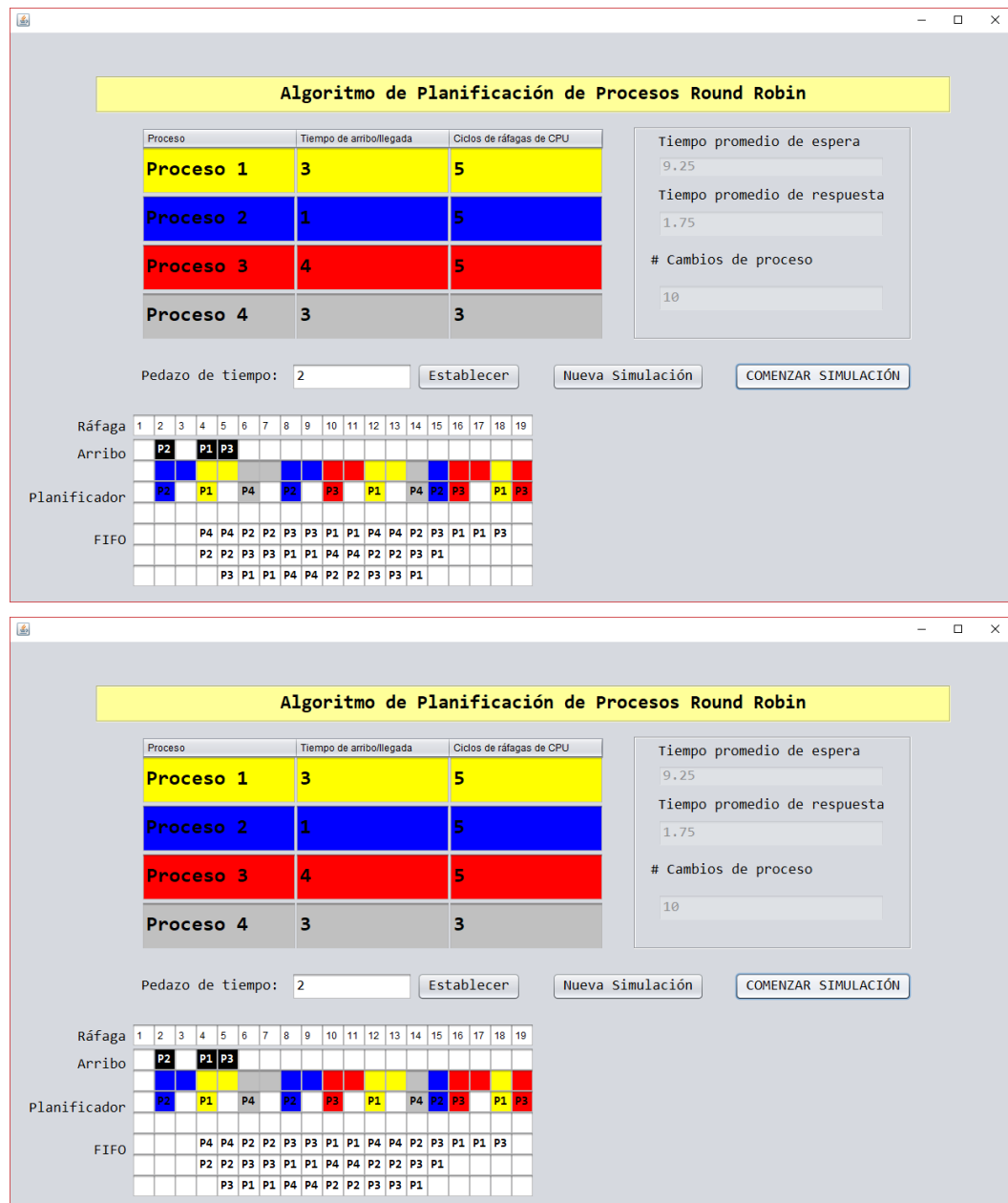
**Figura 5.** Restricción de Comenzar Simulación en base al JButton Establecer.

El JButton Nueva Simulación, cumple el funcionamiento de su nombre al 100% ya que permite restablecer a una configuración inicial la GUI, sin tramas de procesos y valores de medidas de rendimiento, y con nuevos valores en los datos de los procesos, ver Figura 6.



**Figura 6.** Reinicio de configuración mediante el JButton Nueva Simulación.

Finalmente el JButton COMENZAR SIMULACIÓN, permite realizar el funcionamiento de la aplicación las veces que el usuario tenga abierto el GUI, para ello se muestran un par de ejecuciones que se pueden ver en la Figura 7.



**Figura 7.** Funcionalidad general de la GUI desarrollada.

No se dan mayor detalle a la funcionalidad del programa ya que eso debe ser de aporte personal. Sin embargo, se invita a asistir a las dos sesiones de ayudantía, impartidas los miércoles y viernes de 8:30 a 10 en el cubículo T-163 o también conocido como Sala de Juntas del Departamento de Ingeniería Eléctrica.

El checklist se adjuntará en el apartado de la Práctica, por lo que será como referencia para cubrir cada uno de los puntos establecidos en el mismo.

Lo que se valorará más para esta práctica es el esfuerzo individual, así que cualquier parcialidad será muy bienvenida y sin ninguna discriminación.