

Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-HUMANA.

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES.

TEMA: MODULO 2 CLASE 9 PLANIFICADOR SRT

NOMBRES DE LOS ESTUDIANTES:

Padilla Perez Jorge Daray | 216584703

Luis Ricardo Díaz Montes | 219293947

Ernesto Macias Flores | 221349941

NOMBRE DE LA MATERIA: Sistemas operativos

NOMBRE DEL PROFESOR: Ramiro Lupercio Coronel

Table of Contents

[Introducción 3](#_Toc161407292)

[Procesos 4](#_Toc161407293)

[Problemas que se encontraron en el desarrollo **¡Error! Marcador no definido.**](#_Toc161407294)

[Soluciones **¡Error! Marcador no definido.**](#_Toc161407295)

[Conclusión 8](#_Toc161407296)

[Bibliografia **¡Error! Marcador no definido.**](#_Toc161407297)

# Introducción

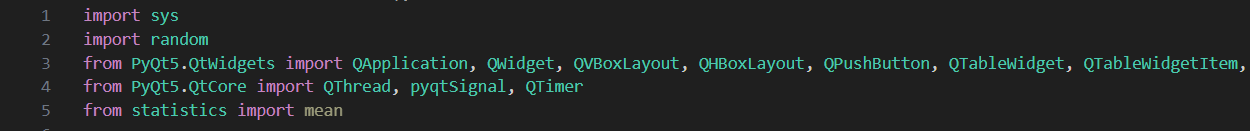
En esta práctica se realizó un planificador SRT (Shortest Remaining Time First) es una variante del algoritmo SJF (Shortest Job First) el cual ya se entregó la semana pasada. En este caso, el procesador asigna el tiempo de procesamiento al proceso que tenga la duración más corta, pero se actualiza constantemente para adaptarse a los cambios en la duración del proceso.

También en términos más sencillos, el planificador SRT prioriza la ejecución de los procesos más cortos en cada momento, lo que ayuda a minimizar el tiempo de espera y a mantener una respuesta ágil ante las variaciones en las necesidades de procesamiento. Es especialmente útil en sistemas multitarea y multiproceso, donde se busca una distribución eficiente del tiempo de CPU entre los procesos disponibles.

Como consecuencia se logro realizar el algoritmo de manera exitosa y como en la actividad pasada, también esta actividad se realizaron botones para Agregar, Terminar, Pausar. Los cuales son puntos clave en la realización de la practica solicitada.

# Procesos

## librerías



En este caso se utilizó la librería sys, random, PyQt5 para la interfaz y por ultimo se hizo uso de statistics.

## Interfaz

Captura de pantalla de computadora

Descripción generada automáticamente

Aquí se crea la clase proceso que incluye al proceso como se indica, y también se crea la ventana principal para la interfaz grafica del programa.

Texto

Descripción generada automáticamente

Aquí se aprecia mas el inicio de la interfaz.

## Trigger

Texto

Descripción generada automáticamente

Trigger de botón clickeado en el cual se esta esperando que se pulse un botón de los 4 que hay, y dependiendo el botón procede.

## Actualizar tabla

Texto

Descripción generada automáticamente

Aquí se actualiza la tabla en la cual se agrega la barra de carga en la columna, y también ordena por columna los tiempos de llegada.

## Lógica botones

Texto

Descripción generada automáticamente

En este caso se implementan funciones para cada botón en los cuales están agregar, eliminar procesos, pausar procesos, inicializar los proceso y falta continuar pero es casi lo mismo que pausa.

## Lógica principal FCFS

Texto

Descripción generada automáticamente con confianza media

Aquí se puede apreciar toda la lógica detrás del algoritmo FCFS

## Calcular promedio

Texto

Descripción generada automáticamente

Aquí se aprecia la función calcular promedio que como su nombre lo indica se utiliza para calcular el promedio de las columnas que se fueron agregando.

# Conclusión

## Padilla Perez Jorge Daray 216584703.

Se implementó el algoritmo SRT para la planificación de procesos en un entorno simulado

utilizando PyQt5 en Python. Esta implementación ofrece una oportunidad para explorar y

comprender cómo funciona el algoritmo SRT en un sistema operativo, distribuyendo

equitativamente el tiempo de CPU entre los procesos activos.

## Ernesto Macias Flores 221349941.

La interfaz gráfica desarrollada proporciona una plataforma interactiva para observar el progreso

de los procesos en tiempo real. Al mostrar información detallada sobre cada proceso, como su

tiempo de llegada, tiempo de ejecución y estado actual, la interfaz ayuda a los usuarios a

comprender mejor cómo se administran los recursos del sistema durante la ejecución de

múltiples tareas.

## Luis Ricardo Díaz Montes 219293947.

A diferencia de otros algoritmos de planificación de procesos, como el FCFS, el algoritmo SRT el

cual se parece mucho al SJF garantiza una mayor equidad en la asignación de recursos de CPU al

dividir el tiempo de ejecución en pequeños intervalos para cada proceso. Esto mejora la capacidad

de respuesta del sistema y reduce el tiempo de espera percibido por los usuarios, lo que resulta en

una experiencia de usuario más fluida y satisfactoria.

## General

el planificador SRT es una estrategia inteligente para administrar los recursos de la CPU de manera eficiente. Al priorizar los procesos más cortos, logra minimizar el tiempo de espera y mantener una respuesta ágil en sistemas multitarea. La adaptabilidad constante a las condiciones cambiantes de los procesos es su mayor fortaleza. En resumen, el planificador SRT es como un maestro de ceremonias que dirige el espectáculo, asegurando que todos los actos tengan su momento en el escenario.

# Bibliografía

1. Tanenbaum, A. S., & Bos, H. (2015). Sistemas operativos modernos (4.ª ed.). Pearson.

[Sistemas Operativos Modernos de Tanenbaum (comprimido).pdf - Google Drive](https://drive.google.com/file/d/0B1AdoZ2fGwgfNjJjYTRjZjctZTUwYy00NGFiLTk0MzEtMmZmMzJhMDNjNTUw/view?resourcekey=0-jeNZJzvqM6aRFkMg5fS_-Q)

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10.ª ed.). Wiley. [Operating System Concepts 10th : Abraham Silberschatz, Peter B. Galvin, and Greg Gagne : Free Download, Borrow, and Streaming : Internet Archive](https://archive.org/details/operatingsystemconcepts10th)
2. Stallings, W. (2018). Operating Systems: Internals and Design Principles (9.ª ed.). Pearson. [[PDF] Operating Systems: Internals and Design Principles | Semantic Scholar](https://www.semanticscholar.org/paper/Operating-Systems%3A-Internals-and-Design-Principles-Stallings/e9cbb8b4fb36105b01c7ff653b27685f3afbbf02)
3. Abraham, S. (2019). Operating Systems: Principles and Practice (2.ª ed.). Recursive Books. [Operating system principles : Silberschatz, Abraham : Free Download, Borrow, and Streaming : Internet Archive](https://archive.org/details/operatingsystemp0000silb)
4. Deitel, H. M., & Deitel, P. J. (2018). Operating Systems (3.ª ed.). Pearson. [Operating systems (Deitel) (3rd edition)(1).pdf - Free Download PDF (kupdf.net)](https://kupdf.net/download/operating-systems-deitel-3rd-edition-1-pdf_58aa0f676454a7ee1bb1e8f5_pdf)
5. Goscinski, A. M., & Brock, M. T. (2017). Cloud Computing: Principles, Systems and Applications (1.ª ed.). Springer. [Cloud Computing: Principles, Systems and Applications | SpringerLink](https://link.springer.com/book/10.1007/978-3-319-54645-2)