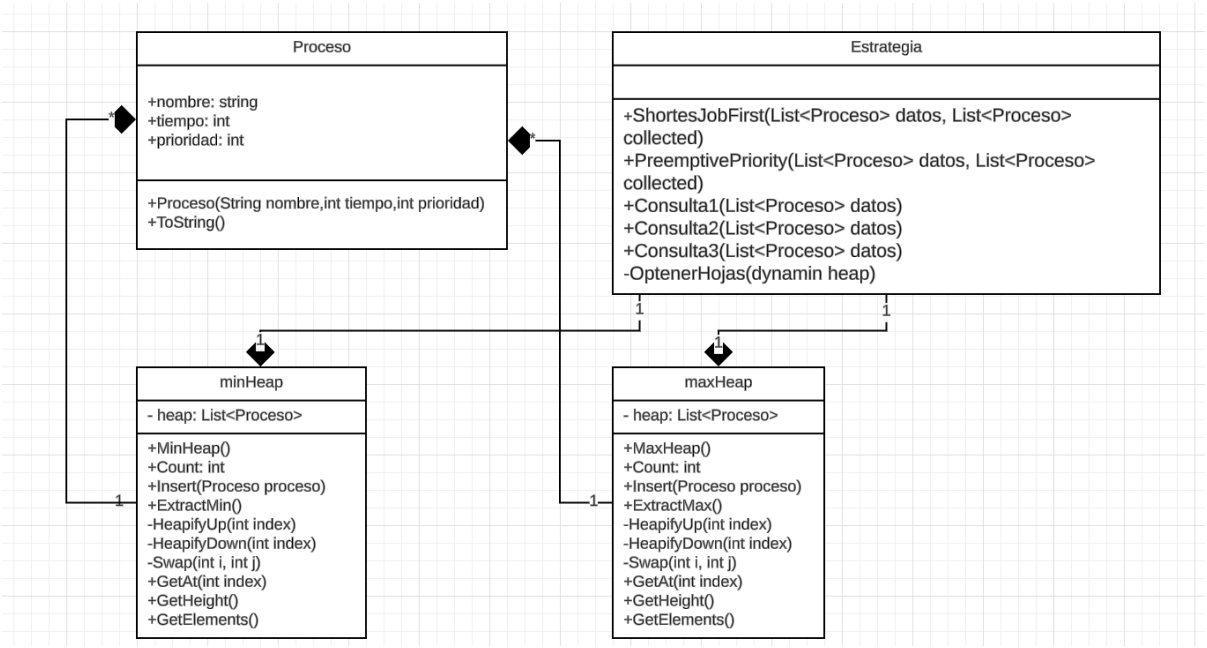
Trabajo Final

Informe

ASIGNATURA: Complejidad temporal, estructura de datos y algoritmos

Estudiante: Veloso, Ramiro Tomas

Diagrama UML



Detalles de implementación

Para la implementación de los métodos de la clase estrategia, se crearon dos clases la clase minHeap y la maxHeap. La clase minHeap ordena los elementos (procesos) manteniendo su propiedad de orden teniendo en cuenta el valor de tiempo de uso de CPU. Mientras que la estructura maxHeap ordena los procesos dentro de ella según valor de prioridad.

El problema principal al momento de realizar la implementación de los métodos “ShortesJobFirst” y “PreemptivePriority”, fue relacionar el atributo especifico de cada proceso según la planificación correspondiente. Para solucionar esa tarea opte por dos clases de estructuras de datos según se necesitase ordenar datos de menos a mayor o viceversa, y según el atributo mediante el cual discriminar a cada proceso.

Otra forma de resolverlo podría haber sido crear una única clase Heap y dos variantes que hereden su comportamiento y se diferencien según el método a implementar.

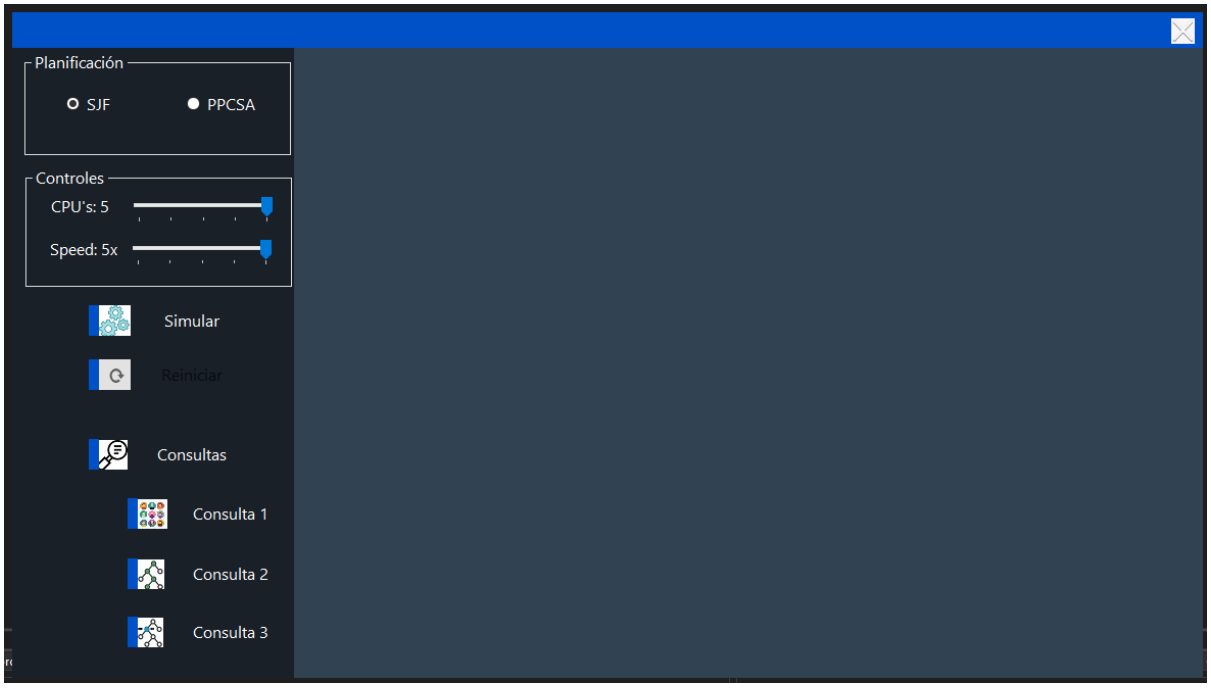
Para la implementación del método “Consulta1”, obtenemos a las hojas de las minHeap y maxHeap creadas en la simulación previamente corrida y las guardamos en una lista, a través del método “ObtenerHojas” que recorre una heap y usa el método “GetAt” para extraer el elemento según un índice. Lugo se realiza la impresión de los datos de acuerdo a una secuencia establecida.

Para el método “Consulta2”, se obtiene el valor de la altura mediante el método “GetHeight” implementado en las clases de las heaps, el cual realiza una operación matemática dada por la expresión altura=[log2 (n+1)], donde n es la cantidad de elementos de la heap.

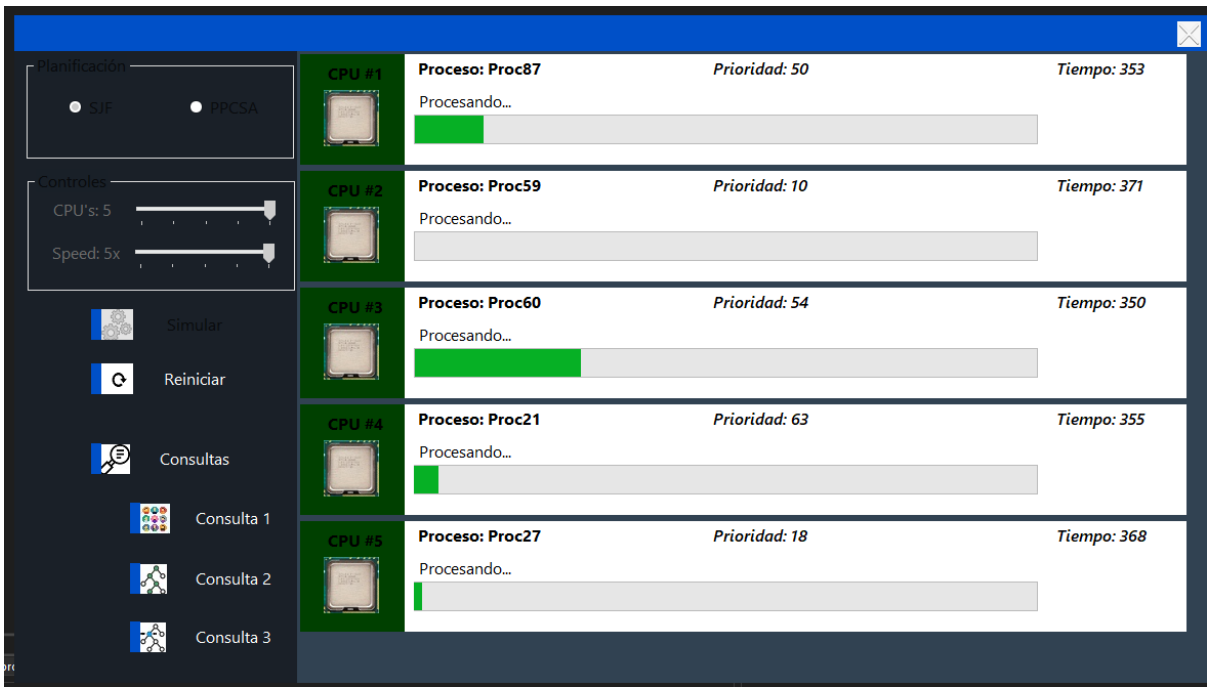
Para el método “Consulta3”, se imprimen los datos de las heaps, ordenados por niveles. Para ello se crea una lista de listas, que se va cargando utilizando un recorrido for y una condición que depende de una variable entera dada por la expresión matemática: nivel=floor(log2 (i+1)), donde i es el índice de la estructura iterativa.

Imágenes de las pantallas del sistema

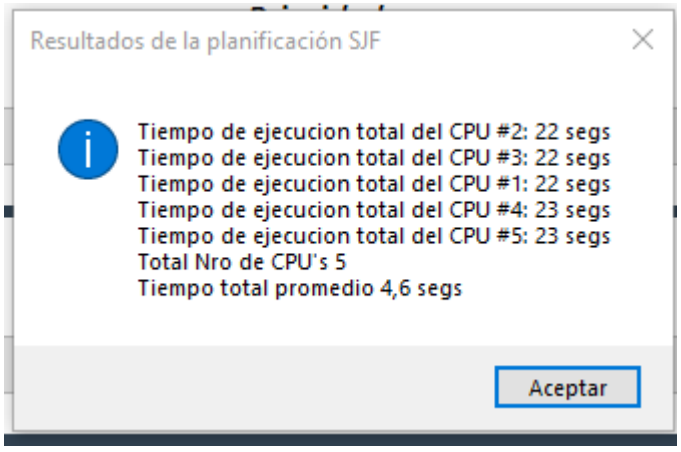
Carga del archivo con la colección de procesos



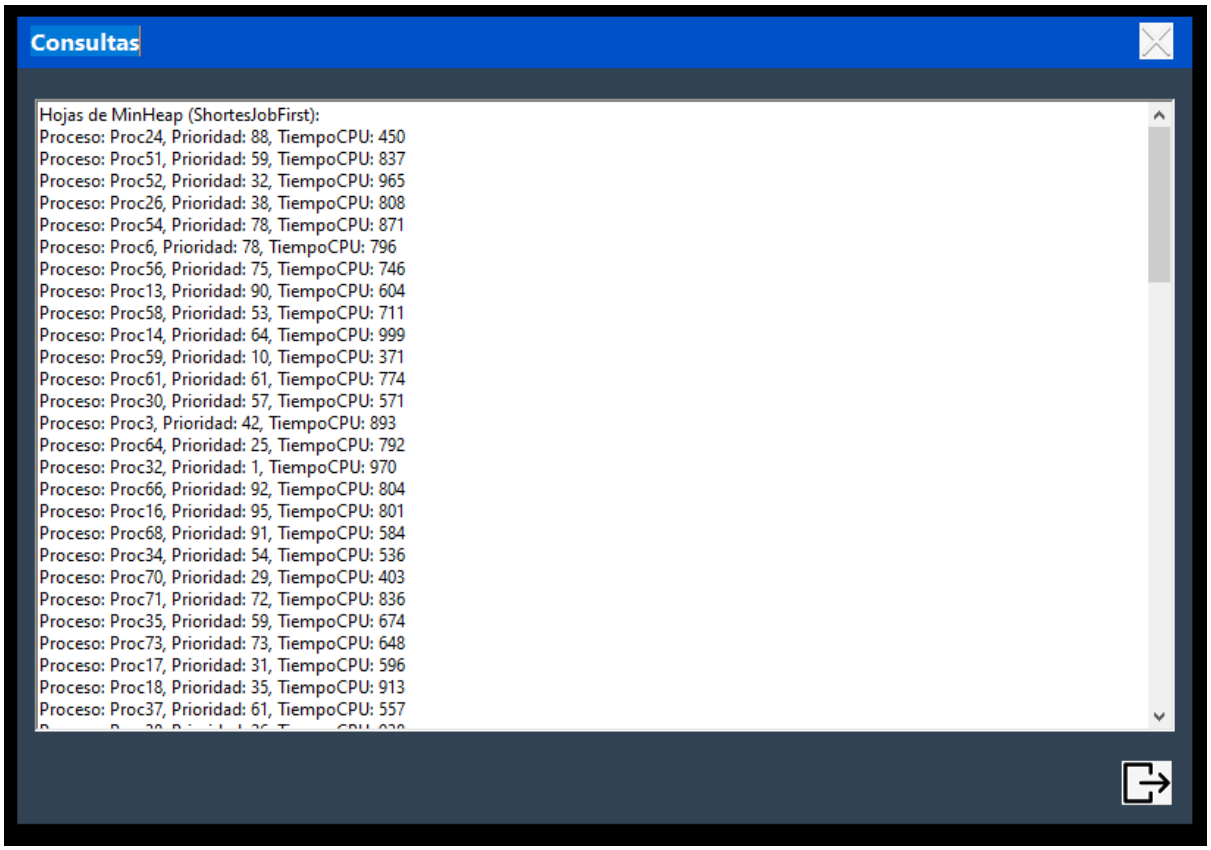
Pantalla inicial del sistema



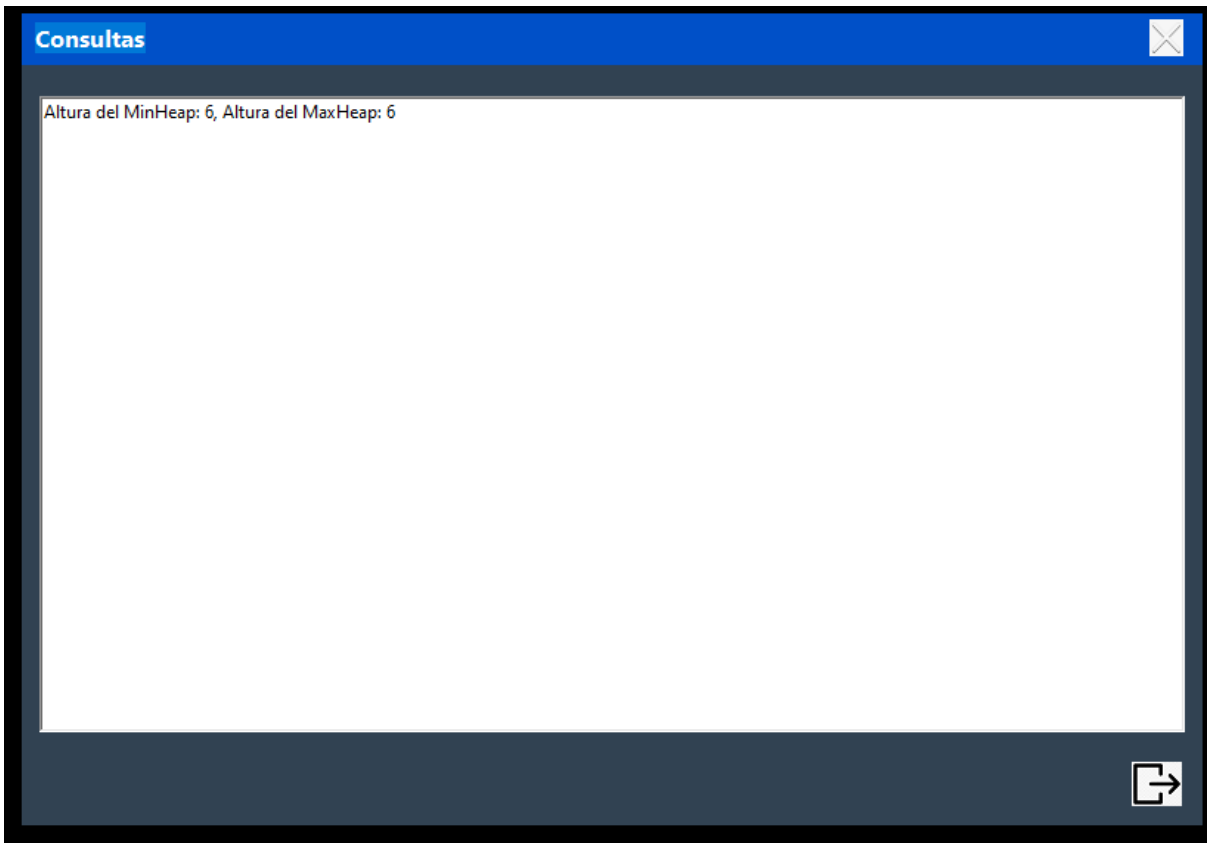
Ejemplo de simulación de la ejecución de procesos bajo uno de los algoritmos de planificación



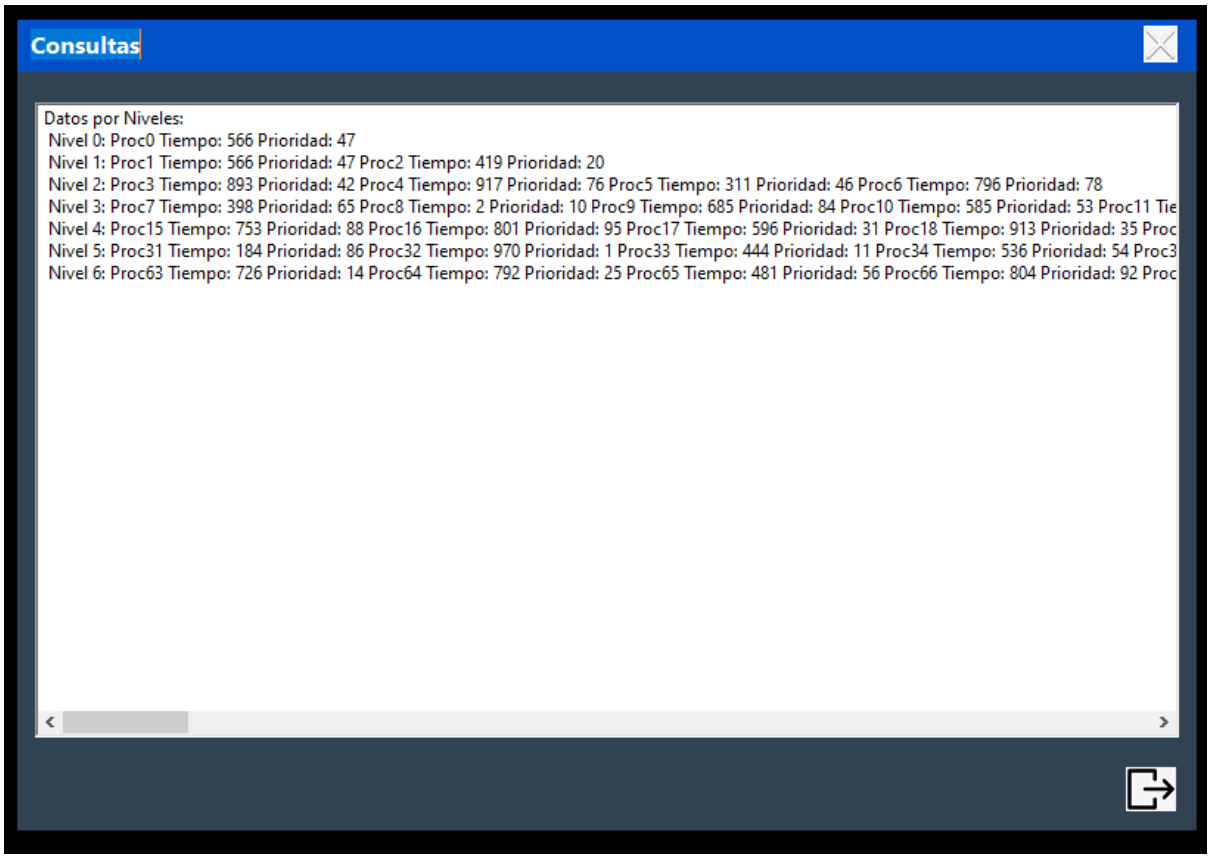
Resultado de la simulación



Resultado de la consulta 1



Resultado de la consulta 2



Resultado de la consulta 3

Ideas de mejora

La inclusión de otros algoritmos de planificación (ej., Round-Robin) o simulaciones en tiempo real.

Reflexión Final

Este proyecto proporciona una experiencia práctica integrando algoritmos de planificación y estructuras de datos avanzadas, permitiendo abordar problemas complejos relacionados con la gestión de procesos en sistemas operativos.

Personalmente me brindo la posibilidad de apreciar lo útiles que son algunas de las estructuras de datos que describimos en la materia y como poder usarlas de manera práctica. Además de proponerme un reto para encontrar la manera adecuada de aplicarlas con mi limitación en cuanto a conocimientos de programación.