



**UNIVERSIDAD  
TECNOLÓGICA NACIONAL**  
FACULTAD REGIONAL  
RESISTENCIA

TPI INTEGRADOR:  
Simulador de Asignación de  
Memoria y Planificación de  
procesos

**INTEGRANTES:**

- Bangher, Matías Ezequiel
- Decoud, Santiago
- Gomez, Marcelo Edgardo
- Nuñez, Ian Lautaro
- Tourn, Miguel Agustín

**NOMBRE DEL GRUPO:** SI3M.exe

**ASIGNATURA:** Sistemas Operativos

**CARRERA:** Ingeniería en Sistemas de Información

**SEGUNDO CUATRIMESTRE**

**CICLO LECTIVO:** 2023

**UNIVERSIDAD Y REGIONAL:** UTN FRRe

**FECHA:** 21/11/2023

## **Índice**

<b>Introducción.....</b>	<b>3</b>
<b>Desarrollo (problemas surgidos y cómo lo solucionamos, como nos organizamos.....</b>	<b>3</b>
<b>Consideraciones que tuvimos como grupo:.....</b>	<b>3</b>
<b>¿Cómo nos fuimos organizando?.....</b>	<b>4</b>
<b>Ejecución del Simulador.....</b>	<b>5</b>
<b>Conclusión.....</b>	<b>8</b>

## **Introducción**

En el presente informe abordaremos la implementación de un simulador diseñado para emular la asignación de memoria con particiones fijas y planificación de los procesos, tratando el ciclo de vida completo de un proceso desde su ingreso al sistema hasta su finalización. A su vez, exploraremos el diseño, funcionamiento y aplicaciones prácticas del simulador, proporcionando una visión detallada de los complejos procesos de asignación de memoria y planificación de la CPU.

Para el desarrollo del código utilizamos el lenguaje “**Python**” con sus respectivas librerías: pandas, tkinter, time y os.

## **Desarrollo**

Para la implementación del simulador nos basamos en las siguientes consideraciones proporcionadas por la cátedra:

- ❖ Planificación a Corto Plazo.
- ❖ Gestión de memoria con Particiones Fijas y un máximo de 10 procesos.
- ❖ Grado de multiprogramación de 5.
- ❖ Política de asignación BEST-FIT.
- ❖ Algoritmo de Planificación de CPU: Round-Robin con  $q=2$ .
- ❖ Esquema de particiones de la memoria será:
  - 100k destinados al Sistema Operativo.
  - 250k para trabajos más grandes.
  - 120k para trabajos medianos.
  - 60k para trabajos pequeños.

## **Consideraciones que tuvimos como grupo**

- ❖ Si todos los procesos que se encuentran cargados en memoria terminaron de ejecutar, y sigue habiendo procesos esperando para ser admitidos a la memoria principal, no sucederá nada irrelevante hasta que nuestro delta de tiempo sea igual al TA (tiempo de arribo) del primer proceso cargado en cola de nuevos.
- ❖ Decidimos dar prioridad a la cola de LISTO/SUSPENDIDO para aquellos casos donde teníamos dos procesos en dicha cola y a su vez otro en cola de NUEVO, y estábamos en un delta de tiempo igual al tiempo de arribo del proceso de cola de NUEVO en dicho momento se liberaba una partición en memoria, por lo tanto podría entrar cualquier proceso de ambas colas (siempre y cuando sus tamaños no superen al de la partición liberada). Si se diese el caso que los procesos que estaban en LISTO/SUSPENDIDO, superan en tamaño a la partición liberada, se da prioridad al proceso cargado en cola de NUEVO.

- ❖ Tuvimos en cuenta que al ser cargado un proceso en CPU no tomamos como irrupción, es decir, si tenemos un proceso que fue cargado en un delta de tiempo, comenzamos a contar al delta de tiempo siguiente como irrupción. Para entender mejor, supongamos que tenemos un proceso que se cargó, y arranca a ejecutarse con  $t=0$ , por lo tanto al haber entrado a CPU no empieza a contar como irrupción, esto sucederá cuando  $t=1$ .
- ❖ Si vienen varios procesos que se encuentran en cola de nuevos, y el que se encuentra en cabeza de cola no puede entrar a memoria porque no hay partición libre o no hay lugar en la cola de Listo/Suspendido, los demás procesos que tengan el mismo tiempo de arribo que el primero no van a poder entrar ya que consideramos darle prioridad al cabeza de cola.

## ¿Cómo nos fuimos organizando?

En nuestro equipo de trabajo, optamos por utilizar la plataforma Trello como una herramienta para gestionar y organizar nuestras tareas para el proyecto. Con un equipo compuesto por 5 miembros, cada uno con roles específicos, Trello fue el epicentro de nuestra colaboración y seguimiento de tareas.

The screenshot shows a Trello board for the project 'SIMULADOR SI3M.exe'. The board is organized into three columns: 'Lista de tareas' (To Do), 'En proceso' (In Progress), and 'Hecho' (Done). Each column contains a list of tasks with assigned responsibilities.

Lista de tareas	En proceso	Hecho
Corrección de errores finales al código	Crear .exe Responsable : Matias	Código: Ya permite operar sobre un archivo .csv Resp: Miguel
Confección del informe	Crear carpeta para zippear Responsable: Matias	Escribir el README de como ejecutar el programa. Responsable : MATIAS
+ Añada una tarjeta	Crear procesos de prueba para enviar en carpeta . Resp: matias	Testear el simulador . responsable : Marcelo
	+ Añada una tarjeta	Ordenar que los informes por consola tengan logica. resp: Ian
		Realización del informe del simulador Resp: Santiago
		Calculo de TRP y TEP de cada proceso Resp: Santiago
		Calculos de TRP y TEP (promedio de todos los procesos) . Resp : Marcelo
		Codigo RoundRobin Resp: Miguel
		Código: Hacer una interfaz amigable para el usuario.(Todo por consola) Resp: Ian
		Codigo: Lograr una coherencia entre el RoundRobin y Bestfit Resp: Miguel
		Código Bestfit: Mique
		+ Añada una tarjeta

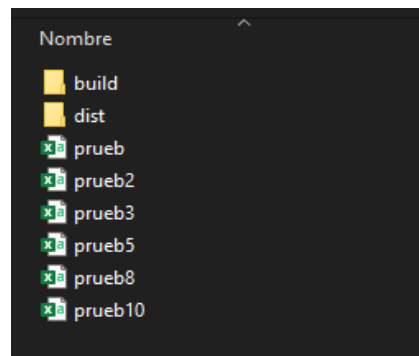
## Ejecución del Simulador

Una vez que ejecutemos el programa se nos abrirá el explorador de archivos para seleccionar el .csv donde está toda la información cargada respecto a los procesos.

```
SIMULADOR DE ASIGNACION DE MEMORIA
Y PLANIFICACION DE PROCESOS
++Hecho por el grupo SI3M.exe++

A continuacion se abra su explorador de archivos...
Solo apareceran los archivos .csv
```

Seleccionamos el archivo.



Una vez ingresados el .csv, la ejecución iniciará mostrando todos los procesos discriminados por ID, Tiempo de Arribo, Tiempo de Interrupción y Tamaño como así también que fueron admitidos al sistema. Aclarar que, por cada instancia de tiempo que vaya pasando, se le solicitará al usuario presionar la tecla Enter para continuar.

```
* ----- *
| - Procesos cargados para realizar la simulacion - |
* ----- *
| ID      | TA      | TI      | TAM (kb) |
| p1      | 0       | 3       | 50       |
| p2      | 0       | 5       | 100      |
| p3      | 1       | 3       | 150      |
| p4      | 4       | 4       | 100      |
| p5      | 4       | 3       | 100      |
| p6      | 7       | 4       | 100      |
| p7      | 7       | 2       | 50       |
| p8      | 7       | 2       | 25       |
* ----- *

-----
ESTADO DE LA COLA DE NUEVO |
-----
['p1', 'p2', 'p3', 'p4', 'p5', 'p6', 'p7', 'p8']

Presione enter para continuar...
```

Por cada instancia de tiempo que vaya pasando, el programa irá mostrando la siguiente información:

- ❖ Procesos cargados en memoria.
- ❖ Estado del procesador, es decir, si se está ejecutando o no un proceso.
- ❖ Información respecto al tiempo restante por ejecutarse para cada proceso.
- ❖ Cola de Listo/Suspendido y de Listo.

```

-----
Tiempo: 0 |
-----

El proceso p1 se cargo en memoria

* -----*
| - MEMORIA - |
* -----*
| Id Proceso | Particion | Dir inicio | Frag. Int (kb) |
|-----|
| p1 | 1 | 0x7F2A9C203C10 | 10 |
|-----|
| | 2 | 0xABF8D764EEF2 | 0 |
|-----|
| | 3 | 0x215B3F8A72C5 | 0 |
|-----|

El proceso p2 se cargo en memoria

* -----*
| - MEMORIA - |
* -----*
| Id Proceso | Particion | Dir inicio | Frag. Int (kb) |
|-----|
| p1 | 1 | 0x7F2A9C203C10 | 10 |
|-----|
| p2 | 2 | 0xABF8D764EEF2 | 20 |
|-----|
| | 3 | 0x215B3F8A72C5 | 0 |
|-----|

-----
ESTADO DEL PROCESADOR |
-----

El proceso p1 entra a ejecucion

Presione enter para continuar...|

```

Una vez finalizada la simulación de asignación de memoria y planificación de procesos, mostrará un informe estadístico referido a los tiempos de espera y de retorno de cada proceso, como así también un promedio de ambas.

```
*-----*
```

INFORMACION ESTADISTICA DE TIEMPOS DE ESPERA Y RETORNO DE LOS PROCESOS		
*-----*		
Proceso	Tiempo de Espera (u.t)	Tiempo de Retorno (u.t)
p1	4	7
p2	6	11
p3	6	9
p4	9	13
p5	11	14
p6	13	17
p7	13	15
p8	17	19

```
-----
```

TIEMPO DE RETORNO PROMEDIO
-----
13.125 unidades de tiempo

TIEMPO DE ESPERA PROMEDIO
-----
9.875 unidades de tiempo

Simulacion finalizada.

Presione enter para continuar...|

## **Conclusión**

Terminada la creación del simulador, logramos vincular las ideas teóricas sobre la gestión de memoria y la planificación de la CPU con una aplicación práctica.

Algunos de los problemas que se nos presentó tienen que ver con la implementación del algoritmo Round Robin, no pudimos comprender en su totalidad como es su funcionamiento hasta que tuvimos la segunda entrega con los profesores. Cabe destacar que en ciertos casos tuvimos que usar consideraciones propias para resolver ciertas dudas que se nos fueron presentando en la etapa de testeo.

Otras cuestiones tienen que ver con la formalidad en el código; también cabe recalcar que utilizamos Programación Imperativa (no orientada a objetos debido a que solo tenemos un conocimiento parcial de esta). Debido a esta programación, la realización de este informe se vió afectada, ya que por esto mismo el código no queda tan legible y/o formal; esto se debe a la poca experiencia programando del grupo

En cuanto a organización supimos administrar tiempos para el rol principal de cada participante, esto fue posible gracias a la utilización de Trello.

También agregar que hemos empleamos la lógica usada en la materia de Algoritmos y Estructura de Datos, utilizando estructuras de datos, tales como arreglos, listas, pilas, colas.