



**UNIVERSIDAD TECNOLÓGICA NACIONAL**  
**FACULTAD REGIONAL RESISTENCIA**

**SISTEMAS OPERATIVOS**  
**SIMULADOR DE ASIGNACIÓN DE MEMORIA Y**  
**PLANIFICACIÓN DE PROCESOS**

**Equipo de Cátedra**

- Ing. Liliana CUENCA PLETSCH
- Dr. Sergio GRAMAJO
- Ing. ROA Jorge Alejandro

**Alumnos**

- Acosta, Gonzalo Exequiel
- Arrejin, Sixto Feliciano
- Maciel Meister, Tobias Alejandro
- Orrego, Nilson
- San Lorenzo, André Leandro

## Introducción

Este documento describe el diseño, implementación y características del simulador de asignación de memoria y planificación de procesos desarrollado por el grupo Lambda. Este simulador aborda los requisitos planteados en la consigna proporcionada por la cátedra, originándose como una herramienta esencial para poder comprender el ciclo completo de un proceso en el entorno dinámico de un sistema operativo.

A lo largo del documento, se detallarán aspectos importantes del desarrollo del simulador, desde los algoritmos empleados hasta la gestión eficiente de la memoria.

El desarrollo del simulador se llevó a cabo en el lenguaje de programación Go, un lenguaje en crecimiento en los últimos años. Elegimos Go porque está basado en C y nos pareció una opción eficiente para un proyecto en donde la rapidez y eficiencia son importante como ser un Sistema Operativo, a pesar de que solo se trata de un simulador. Por otra parte, se trataba de un lenguaje del cual ninguno tenía mucha experiencia, así que también nos sirvió como desafío para introducirnos en el mismo.

---

## Características del simulador

### **Cantidad de procesos admitidos**

Si bien el enunciado del trabajo especifica que para facilitar la implementación se establece un máximo de 10 procesos, la solución propuesta fue pensada de manera general para trabajar con una cantidad indeterminada de procesos, por lo que es posible cargar más de 10 procesos en el simulador. Esto proporciona flexibilidad para probar diferentes escenarios de simulación. Esto se hizo con la idea de que fuera posible simular escenarios más complejos en caso de ser necesario.

### **Algoritmos empleados**

Los algoritmos implementados en la solución son:

- Algoritmo **Round-Robin** para la planificación a corto plazo, con un quantum de 2 unidades de tiempo. Esto significa que luego de que un proceso ingresa al procesador, posee dos unidades de tiempo para ejecutarse y luego deberá desalojarlo, yendo al final de la cola de listos en caso de que necesite más tiempo.

En caso de que un proceso deba desalojar el procesador en un determinado tiempo y en ese mismo instante otro deba ingresar a la cola de listos, se le dará prioridad al que ingresa en la cola de listos. Esto quiere decir que el que acaba de salir del procesador se colocará último en la cola de listos, luego del que recién fue admitido.

- Se utilizó el algoritmo **Best-Fit** para la asignación de procesos a particiones de memoria, optimizando así el uso de las particiones.

Este algoritmo también es utilizado en la lógica de liberación de memoria cuando se debe sacar un proceso de memoria (SwapOut) para el ingreso de otro, la elección se hace mediante BestFit, de manera que el que entra, lo hace a la mejor partición el mismo, minimizando la fragmentación interna.

Cuando un proceso ingrese a la cola de listos, se intentará asignarlo a una partición de memoria en caso de que existiese espacio disponible, haciendo recorrido de cola.

## Memoria y particiones

La memoria cuenta con un total de 530 Kb, de los cuales 100 Kb están destinados al sistema operativo y el resto se divide en 3 particiones donde ingresarán los procesos a la memoria. El esquema de particiones será el siguiente:

- 100 kB destinados al Sistema Operativo.
- 250 kB para los trabajos más grandes.
- 120 kB para los trabajos medianos.
- 60 kB para los trabajos pequeños.

## Otras características

El simulador permite un grado de multiprogramación igual a 5, esto quiere decir que no se puede trabajar con más de 5 procesos simultáneamente. Esto quiere decir que la cola de listos tiene un tamaño máximo de 4, ya que sumado a estos procesos de la cola de listos, tenemos el que se encuentra en el procesador en un determinado momento. Si existiesen más procesos que por tiempo de irrupción deben ingresar y ya se está trabajando con 5 procesos, simplemente no se lo admite y se esperará hasta que finalice un proceso con los que se está trabajando, para admitirlo al que estaba esperando.

Otro aspecto es que todos aquellos procesos que excedan el tamaño de la partición más grande de memoria, no se ejecutarán, y se hará la simulación con los restantes procesos.

Se considera una única cola de listos, que es tanto para los procesos que se encuentran en memoria principal como en disco. Si bien se controla que en memoria principal no se encuentren más de 3 procesos (uno por partición disponible), la cola de listos no distingue de la ubicación física del proceso. El simulador se encarga de mostrar en todo momento en donde se ubica cada proceso. Otra alternativa es tener dos colas: una de *listos* y otra de *listos y suspendidos*, para diferenciar a procesos que están en memoria principal de aquellos que están en disco respectivamente. No obstante se consideró mejor la opción implementada debido a su simplicidad, tal como fue acordado con los profesores de la cátedra en los diferentes coloquios.

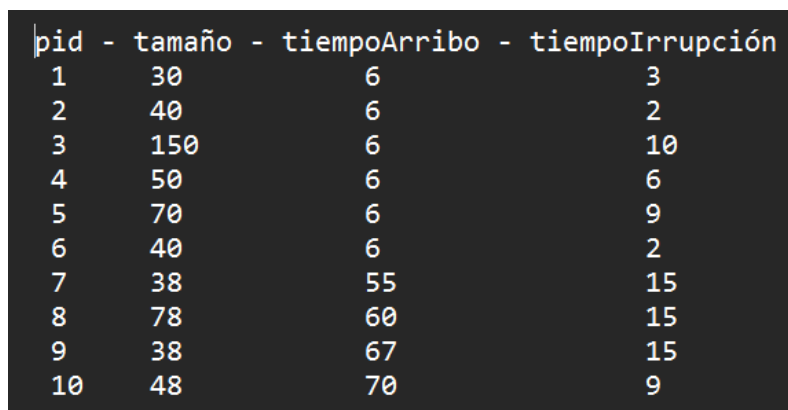
## Funcionamiento del simulador

### Carga de procesos

El simulador desarrollado permite la carga de procesos de dos maneras, de forma manual y por medio de un archivo de texto (con extensión “.txt”).

**Aclaraciones:** La función de cargar procesos a través de archivos funciona correctamente para sistemas operativos Windows, no obstante en sistemas Unix como ciertas versiones de macOS o Linux puede presentar problemas. La función de carga manual funciona correctamente para Windows y macOS.

En el caso de la carga de procesos mediante un archivo de texto, este debe tener el formato que se observa en la siguiente imagen:



pid	- tamaño	- tiempoArriba	- tiempoIrrupción
1	30	6	3
2	40	6	2
3	150	6	10
4	50	6	6
5	70	6	9
6	40	6	2
7	38	55	15
8	78	60	15
9	38	67	15
10	48	70	9

Donde:

- pid: Identificador único del proceso.
- tamaño: Tamaño del proceso en kilobytes.
- tiempoArriba: Tiempo de arribo del proceso.
- tempIrrupción: Tiempo de irrupción del proceso.

Cada fila de la tabla debe contener la información esencial sobre un proceso, los cuales se indican en la primera fila, el nombre de las columnas. A la hora de leer el archivo, el SO se salta la primera fila donde se indican los datos contenidos y pasa a leer los procesos a partir de la segunda en adelante. Cada línea del archivo a partir de la segunda representa un proceso. La misma debe contener 4 valores, separados por al menos un espacio. En caso de contener más datos, al ejecutarlo se informará que se trata de un formato incorrecto.

Es importante aclarar que los PIDs de los procesos deben ser únicos. Para la carga manual, si intentamos ingresar un PID existente se nos informará el problema y se nos solicitará que ingresemos uno válido. En el caso del archivo, se informará del problema y se volverá al inicio del simulador. A continuación se muestran ejemplos de los errores informados.

Por otra parte, no es necesario que se encuentren ordenados de acuerdo a su tiempo de arribo, ya que el simulador se encarga de ordenarlo antes de comenzar la simulación.

### **Información presentada**

En cada punto de control, (cada vez que un proceso cumple un quantum o si termina de ejecutarse), se muestra al usuario en la pantalla como se encuentra la memoria principal, las colas de listos, la cola de procesos nuevos, los procesos terminados y que proceso se está ejecutando. Además, se informa en caso de que se haya realizado un Swap Out o si un proceso pasa a la cola de finalizados. Esta información se muestra mediante distintas tablas según se correspondan que contienen el detalle de esto, las cuales se podrán apreciar en la siguiente sección. Se consideró que para el caso de los procesos finalizados era mejor mostrar únicamente su PID, así la información se puede ver en una sola pantalla y se facilita la interpretación.

### **Finalización de la simulación**

Al finalizar la simulación permite apreciar un breve informe estadístico con, tiempo de retorno y espera para cada proceso y los respectivos tiempos promedios.

### **Guía de uso**

1. Ejecutar el archivo *SimuladorSOLambda.exe* o el comando *./SimuladorSOLambda* para macOS.
2. Seleccionar el modo de cargar los procesos
  - a. Manual

Luego de seleccionar el modo manual se solicita a continuación ingresar la cantidad de procesos que desea cargarse y los datos de cada uno (la identificación, tamaño, tiempo de arribo y tiempo de irrupción).

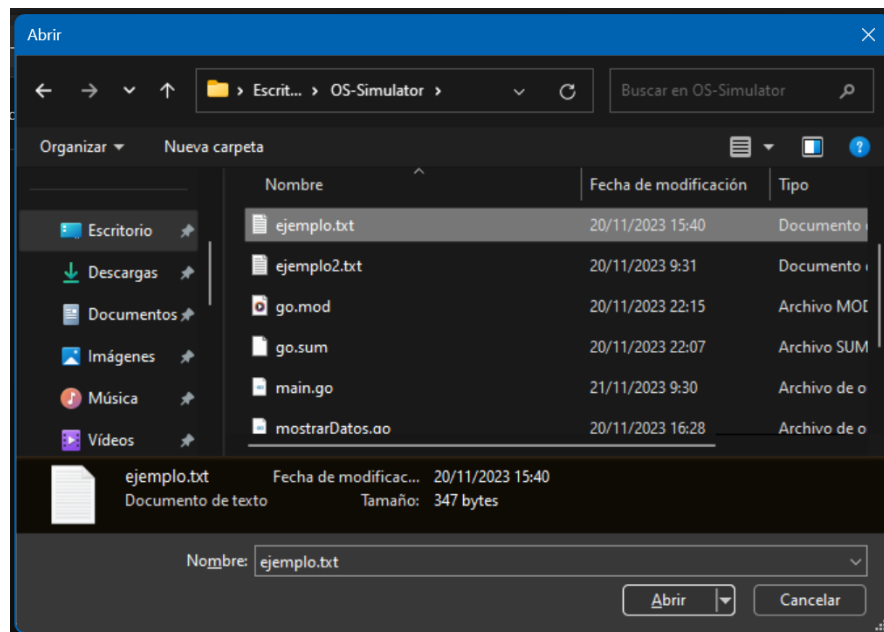
```
Seleccione una opción:
1. Ingresar procesos manualmente.
2. Cargar procesos desde un archivo.
1
• Ingrese la cantidad de procesos: 2
• Ingrese el PID para el proceso 1: 1
• Ingrese el TAMAÑO (Kb) para el proceso 1: 200
• Ingrese el TIEMPO DE ARRIBO para el proceso 1: 1
• Ingrese el TIEMPO DE IRRUPCIÓN para el proceso 1: 10
• Ingrese el PID para el proceso 2: █
```

**Aclaraciones:** los datos ingresados deben ser superiores a 0 (es decir un valor entero positivo) a excepción del tiempo de arribo el cual si puede ser 0. Además el tamaño de cada proceso no debe superar los 250Kb en cualquier caso, si incumplimos alguna condición se nos mostrará el mensaje de error, solicitandonos que ingrese nuevamente los datos, como se observa a continuación:

```
Seleccione una opción:
1. Ingresar procesos manualmente.
2. Cargar procesos desde un archivo.
1
• Ingrese la cantidad de procesos: 3
• Ingrese el PID para el proceso 1: -32
Error al leer el PID. Por favor, ingrese un número entero positivo.
• Ingrese el PID para el proceso 1: 1
• Ingrese el TAMAÑO (Kb) para el proceso 1: 0
Error al leer el tamaño. Por favor, ingrese un número entero positivo.
• Ingrese el TAMAÑO (Kb) para el proceso 1: 50
• Ingrese el TIEMPO DE ARRIBO para el proceso 1: -10
Error al leer el tiempo de arribo. Por favor, ingrese un número entero mayor o igual a 0.
• Ingrese el TIEMPO DE ARRIBO para el proceso 1: 5
• Ingrese el TIEMPO DE IRRUPCIÓN para el proceso 1: 0
Error al leer el tiempo de irrupción. Por favor, ingrese un número entero positivo.
• Ingrese el TIEMPO DE IRRUPCIÓN para el proceso 1: -5
Error al leer el tiempo de irrupción. Por favor, ingrese un número entero positivo.
• Ingrese el TIEMPO DE IRRUPCIÓN para el proceso 1: 4
• Ingrese el PID para el proceso 2: 1
Error: El PID ya está en uso. Por favor, ingrese un PID único.
• Ingrese el PID para el proceso 2: 2
• Ingrese el TAMAÑO (Kb) para el proceso 2: █
```

b. Cargar los procesos mediante un archivo

Si elegimos la segunda opción, se nos abre el explorador de archivos, donde podemos elegir el archivo de entrada para ejecutar la simulación. El archivo de entrada debe cumplir con todas las condiciones detalladas anteriormente. Esto se observa a continuación:



### 3. Inicio del simulador

Después de seleccionar el método de carga de los procesos nos aparecerá el siguiente mensaje:

```
Inicio del Sistema Operativo - Presione ENTER para continuar
• Estos son los procesos ELIMINADOS por exceder el tamaño de memoria: -
```

Donde se nos indica si hay algún proceso que no se ejecutará debido a que excede el tamaño de la memoria. Para comenzar se debe presionar la tecla ENTER.

### 4. Ejecución del simulador

Durante la ejecución se verá la siguiente información en pantalla:



----- TIEMPO: 6 -----

PROCESADOR	1
------------	---

MEMORIA				
Dir. Inicio	Tamaño (kB)	Estado	Fragm. Interna (kB)	Proceso
0x1D6	250	Ocupado	100	Proceso 3
0x15E	120	Ocupado	80	Proceso 2
0x64	60	Ocupado	30	Proceso 1

COLA DE LISTOS				
PID	Tiempo de Arribo	Tamaño (kB)	Tiempo Restante	Cargado
2	6	40	2	Memoria
3	6	150	10	Memoria
4	6	50	6	Disco
5	6	70	9	Disco

COLA DE NUEVOS			
PID	Tiempo de Arribo	Tamaño (kB)	Tiempo de Irrupción
6	6	40	2
7	55	38	15
8	60	78	15
9	67	38	15
10	70	48	9

• Esta es la cola de procesos FINALIZADOS: -

- El tiempo de ejecución del simulador
- Un cuadro con la etiqueta “Procesador” que nos indica que proceso se está ejecutando en el procesador en este instante.
- Un cuadro de “Memoria” que nos indica el estado de cada partición:
  - La dirección de inicio de cada partición
  - El tamaño de cada partición
  - El estado (ocupado o libre)
  - La fragmentación interna
  - El proceso que está alojado en esa partición (El que se encuentra en color morado es aquel que está en el procesador en ese momento).
- Un cuadro de “Cola de Listos” que nos brinda información sobre los procesos que se encuentran esa cola con los siguiente datos:
  - El PID del proceso
  - El tiempo de arribo previsto del mismo

- iii. El tamaño
- iv. Tiempo restante que le queda de ejecución (en un principio es igual al tiempo de irrupción)
- v. Si está en memoria o disco
- e. Un cuadro de “Cola de nuevos”, estos son los procesos que todavía no ingresaron a la cola de listos, que cuenta con la siguiente información
  - i. El número de identificación del proceso
  - ii. El tiempo de arribo
  - iii. El tamaño
  - iv. Tiempo de irrupción
- f. Un sector que nos indica la lista de procesos finalizados.

Es importante destacar que para el caso de la cola de listos los procesos que se encuentran más arriba son los que se encuentran próximos a ingresar al procesador, es decir que son los primeros que están esperando para ejecutarse. Para el caso de la cola de listos, los mismos están ordenados de acuerdo a su tiempo de arribo.

Para continuar la ejecución del simulador se debe presionar la tecla ENTER.

## 5. Presentación del informe estadístico

Al terminar de ejecutarse todos los procesos, se mostrar un cuadro estadístico como el que se encuentra a continuación:

Se termino de procesar todo - Fin de la Simulación

CUADRO ESTADÍSTICO		
PID	Tiempo de Retorno	Tiempo de espera
10	25	18
7	24	19
4	21	15
3	19	14
50	16	12
1	12	8

Promedio	Valor
P. Tiempo Retorno	19.5
P. Tiempo Espera	14.33

Esta tabla se encuentra ordenada de acuerdo al **tiempo de retorno decreciente**.

## 6. Para cerrar la ventana del simulador se debe presionar la tecla ENTER.