

## Indice

Introducción: .....	2
Primeros Pasos: .....	2
Simulador .....	4
Entrada de datos: .....	5
Formato de Entrada: .....	5
Salida de Datos: .....	5
Formato de Salida: .....	6
Graficador:.....	6
Sintaxis de comandos:.....	6
Manejando la interfaz: .....	7
Opciones y Configuración: .....	7
Lenguaje: .....	8

## Introducción:

El Simulador de requerimientos es un programa destinado a mostrar como se atienden los requerimientos en un disco duro según distintos tipos de algoritmos. Permite definir un entorno de simulación (ya sea tamaño de disco, algoritmos a utilizar, velocidad de simulación) y obtener resultados de forma gráfica, numérica\* e interactiva.

El usuario cuenta con la posibilidad de interactuar a través de una interfaz grafica, o si lo desea puede ejecutar el simulador desde la consola, obteniendo de forma rápida los resultados.

## Primeros Pasos:

### Requisitos:

Para ejecutar el programa es necesario tener instaladas las siguientes librerías y dependencias:

- Python 2.6.x o 2.7.x (no funciona con Python 3.x)
- Argparse (por defecto en Python 2.6.6 o superior)
- Setuptools(3)
- Tkinter\*(1)
- PyGame\*(1)
- Ttk\*(1)
- Matplotlib\*(2)
- Numpy\*(2)<sup>1</sup>

En el caso de usar el sistema operativo Linux, el paquete de dependencias que viene incluido junto con el programa puede no funcionar correctamente. Se recomienda obtener la mayor cantidad posible de paquetes mediante el universo.

Los siguientes paquetes pueden ser descargados e instalados directamente (requiere ser administrador del sistema). Setuptools, PyGame, Matplotlib, Numpy. El paquete pyTtk requiere instalación manual utilizando setuptools.

### Matplotlib:

```
sudo apt-get install python-matplotlib
```

### Numpy:

```
sudo apt-get install python-numpy
```

- 
1. <sup>1</sup> Necesario para la interfaz grafica
  2. Necesario para el Graficador
  3. Necesario para poder instalar la interfaz grafica y otros modulos

## PyGame:

*sudo apt-get install python-pygame*

## Para instalar pyTtk:

- 1) Descomprimos pyttk-0.3.2.tar.gz.
- 2) Posicionarse en la carpeta donde se descomprimio el paquete
- 3) Ejecutar el siguiente comando : `sudo Python setup.py install`
- 4) La instalación no debería demorarse en comenzar

*Se requieren permisos de administrador y setuptools para poder realizar la instalacion*

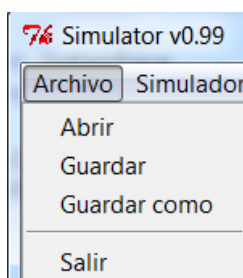
El programa se ejecuta mediante `__init__.py` ubicada en la carpeta raíz. El lenguaje predeterminado es el inglés, aunque es posible cambiarlo desde las opciones (ver. Opciones, lenguaje)

En la pantalla principal podemos ver los menus de selección:



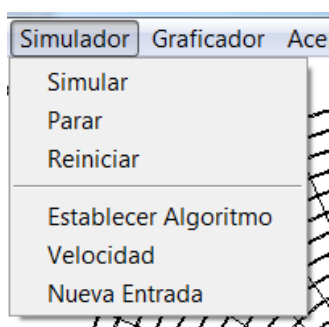
## **File:**

- Abrir: Abre un entorno existente
- Guardar: Guarda un entorno de simulación, cuando un entorno se guarda, se almacena todo el ambiente de simulación, ya sea la entrada, la salida actual, el algoritmo utilizado entre otros atributos. En el entorno no se guardan los atributos referidos a como se representa la simulación, como por ejemplo velocidad de simulación.
- Guardar Como: Siempre pregunta el nombre del archivo de salida.
- Salir: Sale del programa

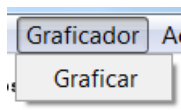


**Simulador:** Todos los comandos del simulador están en este menú (para conocer su funcionamiento vea Simulador)

- Simular: Inicia la simulación
- Parar: Para la simulación en curso
- Reiniciar: Reinicia el entorno de simulación y el grafico
- Nueva Entrada: Ingresar una nueva lista de requerimientos, y su sentido. (ver Simulador, entrada)
- Velocidad de Simulación: Velocidad del grafico. <sup>2</sup>
- Algoritmo: Selección de algoritmo



**Graficador:** Permite graficar la salida actual



## Simulador

El simulador es un sistema sencillo donde dada una entrada, se la reordena hacia una salida según el algoritmo de simulación elegido. La salida es siempre secuencial (el orden de lectura será siempre secuencial), el sucesor es el siguiente a representar en el gráfico (ver Salida.Formato de salida).

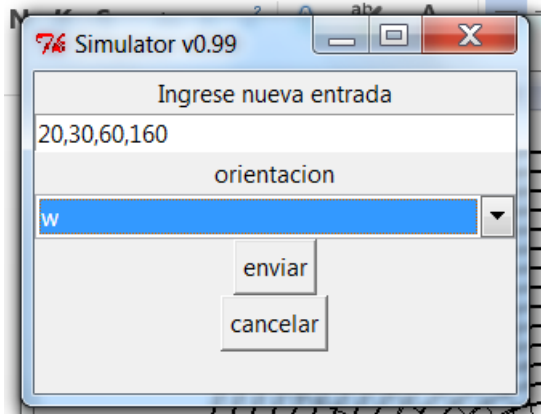
Es importante destacar que el simulador protege los datos y su estado interno cuando esta activo, esto significa que **NO es posible modificar la configuración ni la entrada del simulador mientras este esta en funcionamiento!**. Esto es para proteger la integridad de los datos y asegurar que la salida siempre corresponde a una determinada entrada. Para poder modificar el estado del simulador, es necesario pararlo. **Los datos de salida solo se pueden obtener mientras el simulador esta activo.** Cualquier intento de violar estas condiciones dara como resultado un **error o excepción** si se trabaja con el simulador por consola.

---

<sup>2</sup> Step by Step no permitido en Linux

### Entrada de datos:

La entrada de datos en el simulador debe corresponderse con un formato predefinido. Se dispone de un campo para ingresar los datos, y otro campo que indica la orientación inicial. Se debe tener en cuenta que la orientación inicial puede no ser requerida para ciertos algoritmos (es omitida). La orientación w (Oeste) indica que se estuvo atendiendo previamente valores mayores, y nos estamos moviendo hacia valores menores. La orientación e (Este) indica lo opuesto. Por defecto la orientación es 'Este' (e).



### Formato de Entrada:

El formato de la entrada es sencillo. Los valores están separados por comas. El primer valor es la posición inicial. El resto de los valores son los sectores del disco a atender.

Tenga en cuenta que la lista no puede superar la cantidad máxima de sectores del simulador (por defecto 512, ver Opciones). Si es menor o mayor dará como resultado error de entrada. Para ingresar un Pagefault, tan solo es necesario ingresar pf antes del valor deseado. *Ejemplo: pf40*

*Ejemplo de lista: 20, 30,160,500 (donde 20 es la posición inicial)*

*Las entradas al simulador son en formato String. En el caso de que la entrada sea errónea, el simulador elevara la excepción InputError.*

### Salida de Datos:

La salida de datos puede ser de dos formas, grafica o numérica<sup>3</sup>. La salida grafica es a través de la interfaz o del graficador. Los colores designados para los movimientos son:

*Punto negro: Posición inicial*

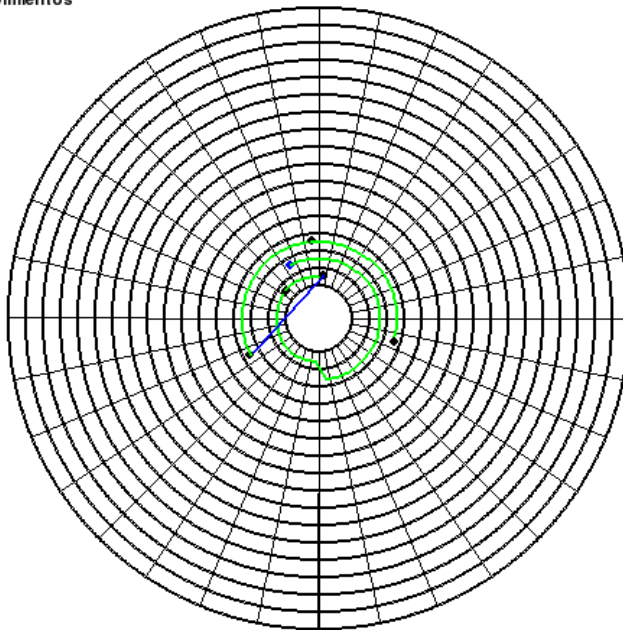
*Línea Verde: Transición normal*

*Línea Roja: Transición prioritaria (Page Fault)*

*Línea Azul: Transición directa (no se procesan sectores)*

---

<sup>3</sup> Todavía no implementado



### Formato de Salida:

El formato numérico de la salida es una lista ordenada de forma secuencial, donde cada elemento representa un sector. Junto con la lista se devuelve la cantidad de pagefaults atendidos, los cuales siempre serán los primeros valores de la lista. La posición inicial se representa por separado. El valor clave -1 indica que existe una transición directa entre el valor anterior y el posterior.

### Graficador:

El graficador es una unidad independiente que permite expresar el resultado de una simulación mediante una representación en ejes cartesianos de la lista de salida. El modulo permite ser ejecutado por línea de comandos y funcionar sin tener que atravesar por la interfaz, o también puede ser llamado utilizando la interfaz mediante la opción 'Graficador' ubicada en la barra de menus.

### Sintaxis de comandos:

Como ya se menciono previamente el graficador soporta ser ejecutado mediante línea de comandos.

En primer lugar debemos ubicar el nombre del modulo 'plotter.py'. Hay 3 campos que son obligatorios y deben ser completados por orden:

```
>>> Python plotter.py DATA_INPUT ALGORITMO DISK_SIZE opciones**
```

### Opciones:

-o seleccion : Orientacion de inicio, entradas validas 'e', 'w'

-s output : Guardar imagen de grafico en carpeta raíz. Output debe ser un nombre de archivo.

-p : Mostrar grafico

*Para conocer los comandos actualizados de cada versión del graficador es necesario ejecutar*

```
>>>Python plotter.py -h
```

## Manejando la interfaz:

**Velocidad de la simulación:** La velocidad de simulación se divide en 5 opciones ( en Windows). Una de ellas (step by step o paso a paso) permite al usuario controlar manualmente el tiempo de cada nuevo movimiento. Para utilizar este sistema es necesario seleccionar la opción (antes de simular) y luego con la tecla 's' se puede adelantar cada paso. Notese que no se permite acceder a ninguna otra opción del entorno mientras se ejecuta la simulación paso a paso. Para cancelar se debe presionar la tecla 'c'.

**Cantidad de accesos:** La cantidad de accesos define la suma de las diferencias entre cada sector accedido. Mediante este contador podemos saber la cantidad de accesos que hubo y así comparar la eficiencia de los algoritmos para cada caso.

## Opciones y Configuración:

El simulador cuenta con un set de opciones modificables. Estas opciones se encuentran en el archivo config.ini. **Cambiar opciones sin el previo conocimiento puede modificar el funcionamiento del programa de forma contraproducente y podría dar resultados erróneos en la simulación.**

*El simulador automáticamente crea un archivo de configuración por defecto. Si desea restablecer los valores predeterminados simplemente elimine el archivo config.ini.*

### Opciones:

lang: define el lenguaje del simulador. Por defecto eng (ingles)

disk\_size: tamaño del disco a simular, por defecto 512, tenga en cuenta que el tamaño del disco debe ser preferentemente potencia de dos. (768, 1024,2048).

plotter: Graficador habilitado o deshabilitado. Por defecto 1.

display\_size: por defecto (640,480). Debe ser un valor adaptado a la pantalla del usuario, con el formato (www,hhh).

separator: define la cantidad de radios que ocupara el circulo separador del centro, recomendado y predeterminado 2.

ratio: Relacion de aspecto entre el tamaño de la pantalla y el tamaño del disco. 0.85 predeterminado.

speed: velocidad. Por defecto 1.

## Lenguaje:

Dentro de la carpeta Lang se encuentran los archivos de lenguaje. Estos se componen con el siguiente formato: clave\_de\_texto#texto<sup>4</sup>

El usuario puede crear todos los lenguajes que quiera siempre y cuando no se agreguen claves de mas ni existan claves sin un valor asignado.

---

<sup>4</sup> Existen algunos problemas de compatibilidad entre Linux y Windows con los archivos de texto, se recomienda hacer backup antes de modificar un archivo de configuración o de lenguaje