



**Instituto Politécnico  
Nacional**

**Escuela Superior de  
Cómputo**



Aplicaciones para Comunicaciones en Red

Docente: **Josué Rangel González.**

Práctica 1. ***Hilos***

Presentan:

**César Uriel Hernández Castellanos**

**Mauricio Joel Martínez Islas**

**Carlos Pimentel González**

Fecha de entrega: **5 de febrero de 2019**

Grupo: **3CV5**

# Índice

<b>1. Introducción.</b>	<b>3</b>
1.1. Definición de hilo . . . . .	3
1.2. Definición de concurrencia . . . . .	3
1.3. Definición de paralelismo . . . . .	3
1.4. Diferencias entre concurrencia y paralelismo . . . . .	3
1.5. Funcionalidad de los hilos . . . . .	3
1.6. Estados de un hilo . . . . .	3
1.7. Ventajas de los hilos contra procesos . . . . .	4
1.8. Sincronización de hilos . . . . .	5
1.9. Usos más comunes de hilos . . . . .	5
<b>2. Descripción del problema.</b>	<b>5</b>
2.1. Problema de tratamiento de imágenes con hilos. . . . .	5
2.2. Problema del producto de matrices con hilos. . . . .	6
<b>3. Desarrollo.</b>	<b>6</b>
3.1. El problema de tratamiento de imágenes con hilos. . . . .	7
3.2. El problema de multiplicación de matrices con hilos . . . . .	7
<b>4. Diagramas.</b>	<b>7</b>
4.1. Problema de tratamiento de imágenes con hilos. . . . .	7
4.2. El problema de multiplicación de matrices con hilos. . . . .	12
<b>5. Conclusiones.</b>	<b>14</b>

## Índice de figuras

1.	Ciclo de vida de un hilo [1] . . . . .	4
2.	Combinación RGB . . . . .	6
3.	Producto de matrices . . . . .	6
4.	Diagrama de flujo de la función principal . . . . .	8
5.	Diagrama de flujo de función que crea carpetas . . . . .	8
6.	Diagrama de flujo de la función encargada de recopilar las imágenes del sistema . . . . .	9
7.	Diagrama de flujo de la función encargada de contar el número de imágenes . . . . .	9
8.	Diagrama de flujo de la función que obtiene el nombre de cada imagen . . . . .	10
9.	Función que crea los hilos . . . . .	10
10.	Función que se ejecuta como un hilo . . . . .	11
11.	Función principal . . . . .	12
12.	Función que solicita la entrada del usuario input_Integer . . . . .	13
13.	Función que crea y rellena las matrices create_Matrix . . . . .	13

## **1. Introducción.**

### **1.1. Definición de hilo**

En sistemas operativos, un hilo, proceso ligero o subproceso es una secuencia de tareas encadenadas muy pequeña que puede ser ejecutada por un sistema operativo

### **1.2. Definición de concurrencia**

Se refiere a la habilidad de distintas partes de un programa, algoritmo o problema de ser ejecutado en desorden o en orden parcial, sin afectar el resultado final. Las operaciones pueden ser ejecutadas en múltiples procesadores, o ejecutados en procesadores separados físicamente o virtualmente.

### **1.3. Definición de paralelismo**

El paralelismo es una forma de computación en la cual varios cálculos pueden realizarse simultáneamente, basado en el principio de dividir los problemas grandes para obtener varios problemas pequeños, que son posteriormente solucionados en paralelo.

### **1.4. Diferencias entre concurrencia y paralelismo**

La concurrencia y paralelismo están muy relacionados, sin embargo no son lo mismo, por lo tanto, la concurrencia ayuda a procesar varias tareas al mismo tiempo, mientras que el paralelismo resuelve una única tarea de una forma mucho más eficiente.

### **1.5. Funcionalidad de los hilos**

Al igual que los procesos, los hilos poseen un estado de ejecución y pueden sincronizarse entre ellos para evitar problemas de compartición de recursos. Generalmente, cada hilo tiene una tarea específica y determinada, como forma de aumentar la eficiencia del uso del procesador

### **1.6. Estados de un hilo**

Existen tres estados para un hilo (son los mismos que existen para el modelo tradicional de un proceso):

1. En ejecución (utilizando el procesador)
2. Listo (ejecutable, detenido momentáneamente para dejar que se ejecute otro hilo)
3. Bloqueado (no se puede ejecutar hasta que ocurra algún evento)

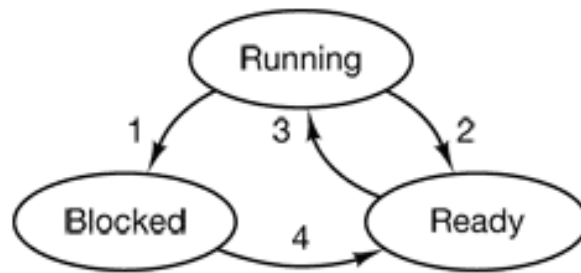


Figura 1: Ciclo de vida de un hilo [1]

De acuerdo con [1], los hilos pueden estar en uno de tres estados. Nótese que hay cuatro posibles transiciones posibles entre los estados. Describiéndolas:

1. Ocurre cuando el sistema operativo descubre que un hilo no puede seguir ejecutándose por el momento. En algunos sistemas, el proceso puede ejecutar una llamada al sistema, como **pause**, para llevarlo al estado de bloqueado.
2. Las transiciones 2 y 3 son causadas por el planificador de procesos. La transición 2 se da cuando el planificador decide que el hilo ya se ha ejecutado por el tiempo necesario y es hora de cederle tiempo de procesamiento a otro hilo.
3. Se da cuando todos los demás hilos se han procesado y ya es tiempo para que el primer hilo se vuelva a ejecutar otra vez
4. Ocurre cuando se da un evento externo por el cual estaba esperando un hilo. Si ningún otro hilo está ejecutándose en ese momento, ocurrirá la transición 3 y el hilo comenzará a ejecutarse. Si no, tendrá que esperar en el estado listo por un rato hasta que el procesador esté disponible y llegue su turno

### 1.7. Ventajas de los hilos contra procesos

Los hilos son generados a partir de la creación de un proceso, así es de que podemos decir que un proceso es un hilo de ejecución, conocido como Monohilo. Las ventajas de los hijos se dan cuando hablamos de Multihilos, que es cuando un proceso tiene múltiples hilos de ejecución los cuales realizan actividades distintas, que pueden o no ser cooperativas entre si. Los beneficios de los hilos se derivan de las implicaciones de rendimiento

- Se tarda mucho menos tiempo en crear un hilo nuevo en un proceso existente que en crear un proceso. Algunas investigaciones llevan al resultado que esto es así en un factor de 10.
- Se tarda mucho menos en terminar un hilo que un proceso, ya que cuando se elimina un proceso.
- Se tarda mucho menos tiempo en cambiar entre dos hilos de un mismo proceso.
- Se tarda mucho menos tiempo en cambiar entre dos hilos de un mismo proceso.

- Los hilos aumentan la eficiencia de la comunicación entre programas en ejecución. En la mayoría de los sistemas en la comunicación entre procesos debe intervenir el núcleo para ofrecer protección de los recursos y realizar la comunicación misma. En cambio, entre hilos pueden comunicarse entre sí sin la invocación al núcleo.

## **1.8. Sincronización de hilos**

Los hilos comparten el mismo espacio de direcciones y otros recursos que pueden ser archivos abiertos. Cualquier modificación a un recurso desde un hilo afecta el entorno del resto de los hilos del mismo proceso. Por lo tanto, es necesario sincronizar la actividad de los distintos hilos para que no interfieran unos con otros o corrompan estructuras de datos.

## **1.9. Usos más comunes de hilos**

Los usos más comunes son en tecnologías SMPP (Short message peer-to-peer protocol) y SMS (Short Message Service) para la telecomunicaciones, aquí hay bastantes procesos ejecutándose los cuales requieren de un servicio, algunos de sus usos se muestran a continuación:

- Trabajo interactivo y en segundo plano
- Procesamiento asíncrono
- Aceleración de la ejecución
- Estructuración modular de los programas

# **2. Descripción del problema.**

## **2.1. Problema de tratamiento de imágenes con hilos.**

- Diseñar un programa que realice una búsqueda de imágenes en todos los directorios del sistema, por cada imagen que encuentre deberá generar un hilo que se encargue de cargar la imagen y aplicarle los filtros RGB. Por lo tanto, a partir de cada imagen se generarán 3 más (RGB). Una vez aplicados los filtros el hilo se encargará de almacenar estas imágenes en un directorio llamado "ImágenesFiltradas"

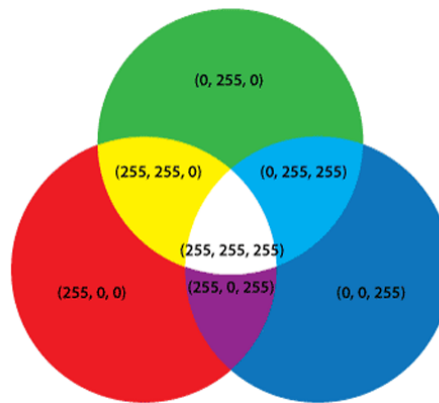


Figura 2: Combinación RGB

## 2.2. Problema del producto de matrices con hilos.

- Diseñar un programa que realice el producto de dos matrices, con la característica especial que la matriz resultante será calculada por medio de cómputo concurrente. El usuario ingresará el tamaño de las matrices, así como el número de hilos a utilizar, cada uno de los hilos calculará una fila o más de la matriz resultante.

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \times \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{pmatrix}$$

Figura 3: Producto de matrices

## 3. Desarrollo.

Se decidió utilizar el lenguaje Python, que, de acuerdo con [2], *desde una perspectiva, se puede decir que Python es un lenguaje de scripting de muy alto nivel. Desde otra perspectiva, puedes decir que es un lenguaje de programación de alto nivel que ha sido implementado de una manera que pone énfasis en la interactividad. Python comparte características con lenguajes de scripting, pero también comparte algunas características con lenguajes de programación más tradicionales.* Nosotros, nos aprovechamos de la naturaleza de 'alto nivel' que posee Python para poder escribir programas que en otros lenguajes (e.g. el lenguaje C) hubieran sido tardados y bastante laboriosos.

Es, de una manera preferir el diseño y la abstracción en lugar de la codificación. Obviamente tiene sus sacrificios, pero para nuestros propósitos, termina siendo una herramienta muy útil.

### 3.1. El problema de tratamiento de imágenes con hilos.

Como podemos observar en la figura ??, lo que se hace es crear las carpetas utilizando una función de Python que nos permite correr comandos de `shell` para ahorrar trabajo. Seguido de eso, recopilamos todos los archivos png y jpg (o los que queramos especificar dentro de la tupla) y nuevamente, utilizando `subprocess.call()`, metemos las imágenes a al directorio `ImagenesRecopiladas`. Ya recopiladas en el directorio, obtenemos una lista de las rutas de cada imagen dentro de `ImagenesRecopiladas` anteponiendo el nombre del directorio al nombre de archivo de cada imagen.

Después, obtenemos la lista de los nombres de cada archivo y se la pasamos, junto con nuestra lista de rutas y el número de imágenes, a la función que se encarga de crear cada hilo.

Lo que hace cada hilo es abrir cada imagen, convertirla a RGB y guardar tres imágenes nuevas (una para el filtro rojo, verde y azul) en el directorio `ImagenesFiltradas`

### 3.2. El problema de multiplicación de matrices con hilos

Primeramente, el usuario ingresa el tamaño de las matrices (aquí hay una pequeña validación para comprobar si se le está dando un número o no) y se crean las dos. Después, lo que se hace es pedir cuántos hilos van a haber para ejecutar la operación (también es validado).

Ya con el número de hilos y el número de columnas de la primera matriz, lo que se hace es obtener el número de operaciones (el número operaciones siendo el número de filas de la primera matriz), y el residuo de la división se le asigna al último hilo.

Ahora, ya se empiezan a recorrer las filas de la primera matriz y se ejecutan los hilos, los cuales se encargan de multiplicar una fila por la segunda matriz. Al último, se ejecuta el hilo que se iba a encargar del residuo de filas/hilos.

Al último, sólo se espera a que terminen los hilos y se imprime el resultado.

## 4. Diagramas.

### 4.1. Problema de tratamiento de imágenes con hilos.

En la figura que se muestra a continuación es posible observar los pasos necesarios para dar solución al problema, los cuales se describirán a detalle.



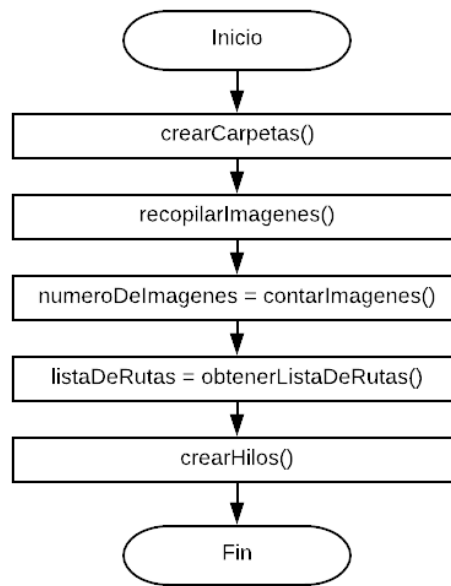


Figura 4: Diagrama de flujo de la función principal

En la figura siguiente se observa el proceso para crear las carpetas que nos serán de ayuda durante la ejecución del programa, por lo que se hizo uso de la librería `os`, con la finalidad de poder ejecutar comandos

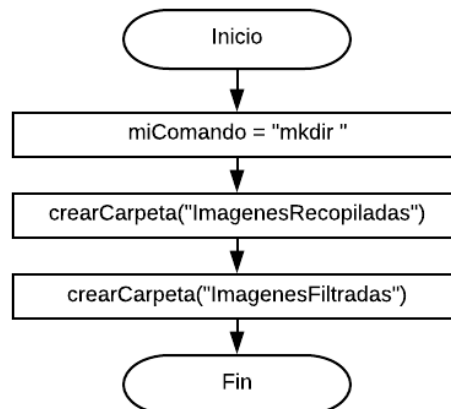


Figura 5: Diagrama de flujo de función que crea carpetas

En la imagen de a continuación se describe la función encargada de recopilar las imágenes del sistema, esto por medio de comandos.

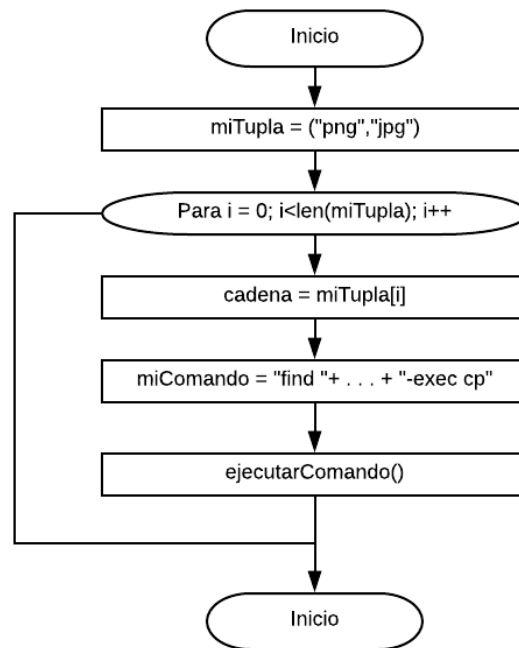


Figura 6: Diagrama de flujo de la función encargada de recopilar las imágenes del sistema

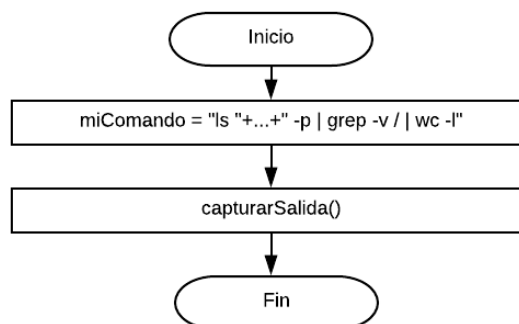


Figura 7: Diagrama de flujo de la función encargada de contar el número de imágenes

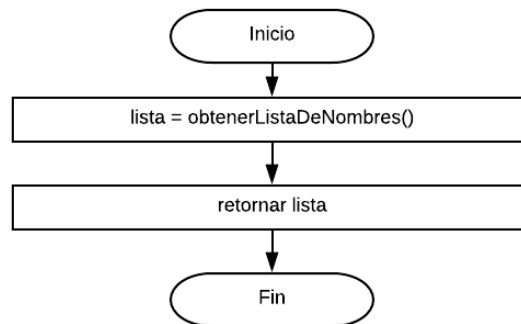


Figura 8: Diagrama de flujo de la función que obtiene el nombre de cada imagen

A continuación se muestra el diagrama de flujo de la función que crea los hilos de acuerdo al número de imágenes recopiladas por la función descrita anteriormente

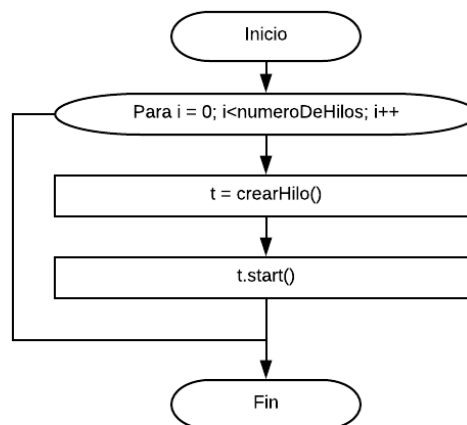


Figura 9: Función que crea los hilos

En la figura siguiente se muestra la función que funge como hilo, esta función se encarga de extraer la matriz asociada a la imagen, además de producir 3 imágenes más con los filtros RGB correspondientes, por último la función almacena las imágenes en el directorio "ImágenesFiltradas"

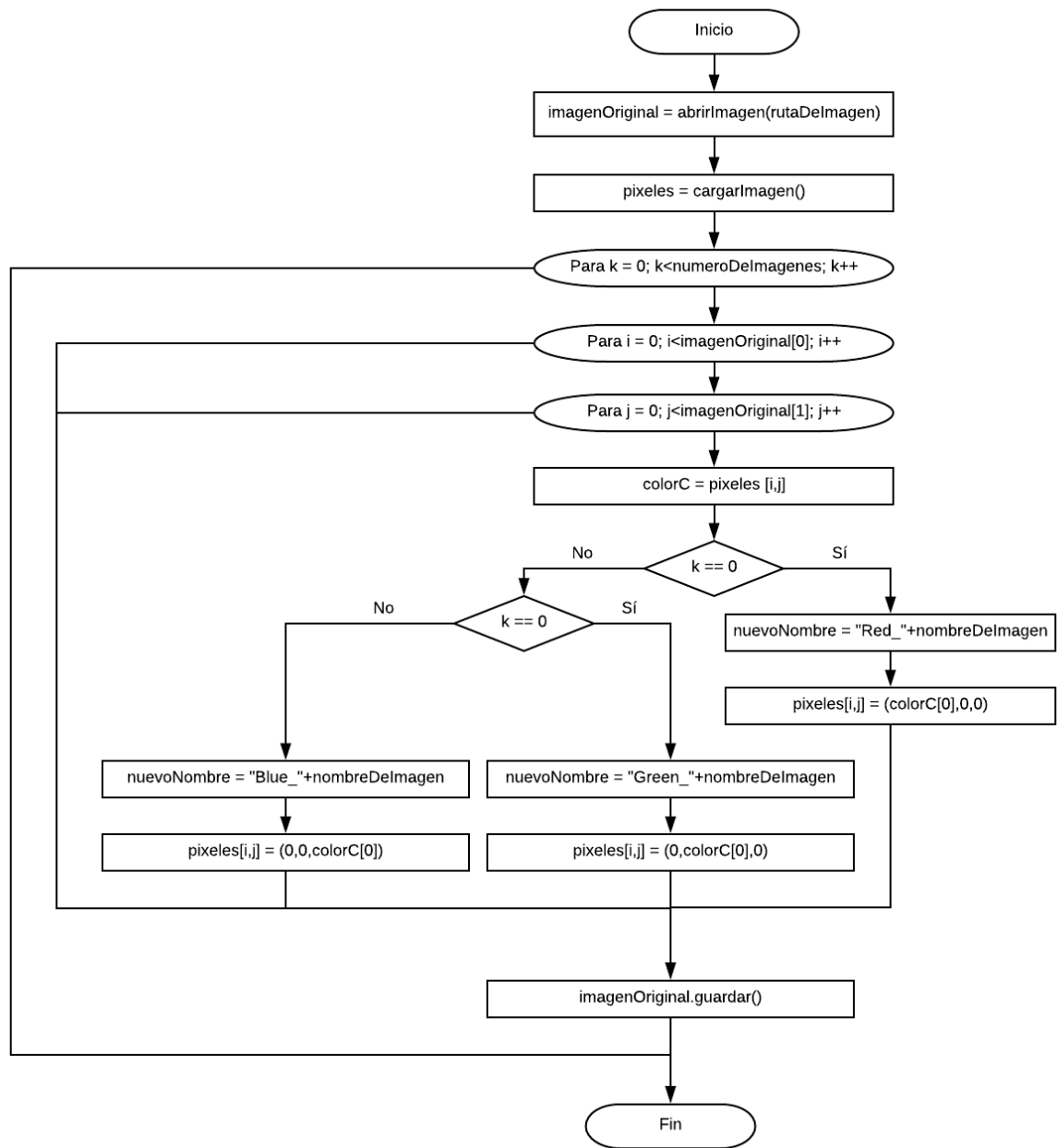


Figura 10: Función que se ejecuta como un hilo

## 4.2. El problema de multiplicación de matrices con hilos.

En esta primer figura podemos ver el comportamiento principal del programa. La función llamada en los hilos, es de una biblioteca de Python que se encarga de hacer la operación entre matrices.

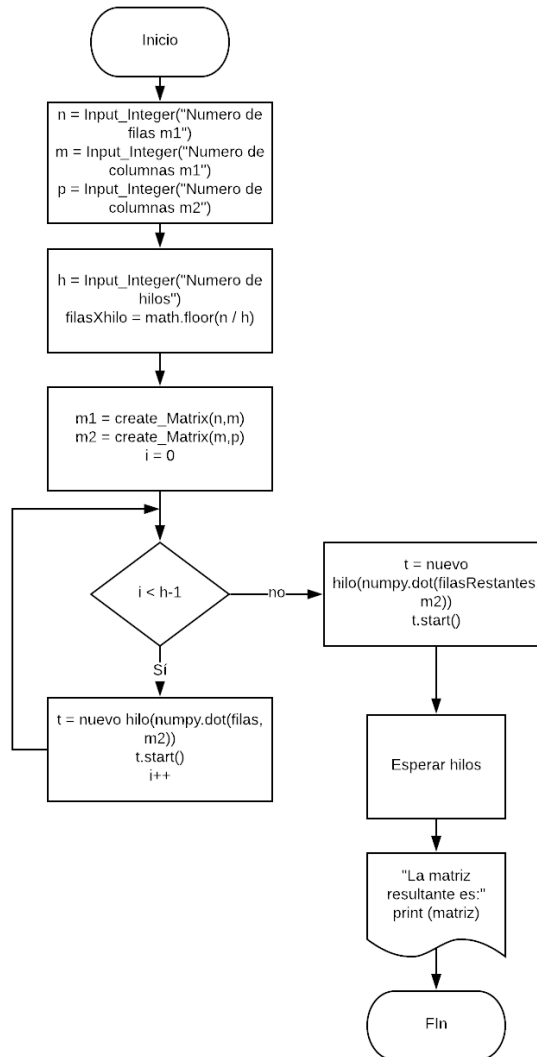


Figura 11: Función principal

En este siguiente diagrama de flujo, se muestra la función que se encarga de solicitar la entrada al usuario.

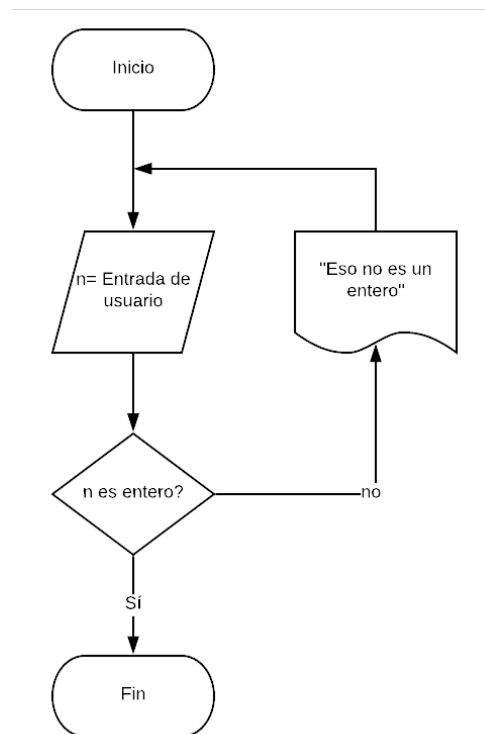


Figura 12: Función que solicita la entrada del usuario `input_Integer`

Finalmente, esta la función que crea y rellena las matrices con números al azar, y la función que se encarga, de realizar la operación de matrices, la cual sirve para los .

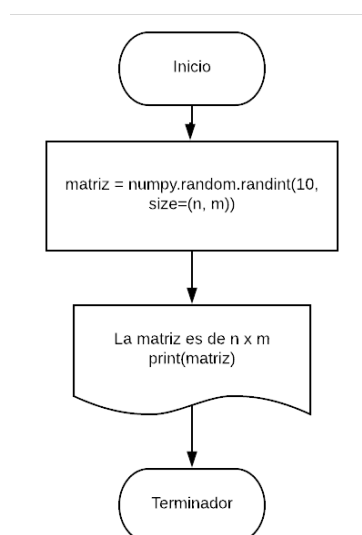


Figura 13: Función que crea y rellena las matrices `create_Matrix`

## 5. Conclusiones.

### Hernández Castellanos César Uriel

La programación concurrente tiene bastante campo de aplicación, desde los sistemas operativos hasta la tecnología que usamos de manera cotidiana.

En la elaboración de la presente práctica obtuvimos conceptos que nos darán la posibilidad de desarrollar aplicaciones utilizando hilos, con la finalidad de poder realizar más de una tarea a la vez y administrando correctamente los recursos del ordenador.

La utilidad de los hilos resultan evidente. Por ejemplo, un navegador web puede descargar un archivo de un sitio, y acceder a otro sitio al mismo tiempo. Si el navegador puede realizar simultáneamente dos tareas, no tendrá que esperar hasta que el archivo haya terminado de descargarse para poder navegar a otro sitio.

En conclusión la programación con hilos está presente en la mayor partes de las aplicaciones informáticas que usamos cotidianamente sin importar el lenguaje en que se desenvuelven, el concepto de Thread seguirá siendo el mismo

### Martínez Islas Mauricio Joel

Se han descrito millares de problemas en donde los hilos llegan a 'salvar el día', ¿Qué nos dice esto? La búsqueda de lo universal, lo holístico. Pese a que sea contradictorio dentro de temas como la computación, la naturaleza humana es capaz de filtrarse a través de cualquier tipo de conocimiento y mostrarse en momentos como éstos. Desde procesos, hasta la organización de los sistemas operativos que rigen el mundo moderno, es claro que se quiere retornar a lo que alguna vez fuimos.

La unificación total. No sabemos si mediante la excesiva segmentación de todo lo que vayamos descubriendo día a día vaya a converger a la unificación, pero creemos que si somos capaces de encontrar las unidades mínimas de nuestro ser, podremos construir la esencia de la vida. Quizás ahí sea a donde nos dirijamos.

### Pimentel González Carlos

Con la programación concurrente podemos, realizar tareas mientras nuestro hilo principal se dedica a otra cosa. Simplemente tenemos esta práctica donde cada hilo se dedican a realizar diferentes cosas, sin tener que esperar a que termine por ejemplo, de realizarse la multiplicación de matrices fila por fila, se hace por pedazos. Cada hilo realizando la un pedazo de la multiplicación y al final uniéndose.

Esto, junto con la gran evolución en los procesadores, que cada vez son mas rápidos, hacen que esta concurrencia ocurra tan rápido que pareciera casi paralelismo. Esto beneficia mucho a la experiencia de usuario.

## Referencias

- [1] T. Andrew S., *Modern Operating Systems, 3rd Edition*. Vrije Universiteit, Ámsterdam, Países Bajos: Pearson Education International, 2009.

- [2] B. Venners. (ene. de 2003). The Making of Python. A Conversation with Guido van Rossum, Part I, dirección: <https://www.artima.com/intv/python.html>.