



INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO



Aplicaciones para comunicaciones en Red

***Practica 1***

Contador de aparición de palabras

Integrantes:

Cortez Enriquez Jovanny Wilver  
González López Emiliano

3CV6

## Introducción

En la creación de sistemas se debe de tomar en cuenta la concurrencia de sus funciones para que el sistema sea rápido, eficiente y factible. La concurrencia lo simulamos con la creación de hilos y su implementación varía según el lenguaje de programación utilizada.

### Concurrencia[1]

Número máximo de flujos de ejecución secuenciales -hilos- que podía estar ejecutando simultáneamente si dispusiera de un número ilimitado de procesadores y depende de cómo esté escrita la aplicación o sistema.

### Hilos [1]

Existen 3 tipos de hilos en unix que son:

- Hilos del núcleo:  
No están ligados con ningún proceso del usuario. son creados y destruidos según las necesidades internas del núcleo y comparten el código y los datos de éste, si bien cada uno de ellos tiene su propia pila y contexto de registros.
- Hilos del usuario:  
Una aplicación puede también expresar su concurrencia a través de hilos del usuario que son gestionadas por la aplicación, sin la intervención del núcleo, Esto se puede conseguir a través de bibliotecas como C-threads de Mach [Cooper & Draves, 1990] o pthreads de POSIX [Mueller, 1993] que proporciona las funciones necesarias para crear, sincronizar, planificar y distribuir hilos sin la necesidad del núcleo.
- Procesos ligeros:  
Es un hilo del usuario sostenido por el núcleo. Desde el punto de vista del usuario, un proceso se puede componer de varios procesos ligeros, cada uno de los cuales se apoya en un hilo del núcleo.

## **Problema**

Realizar un algoritmo que resuelva lo siguiente:

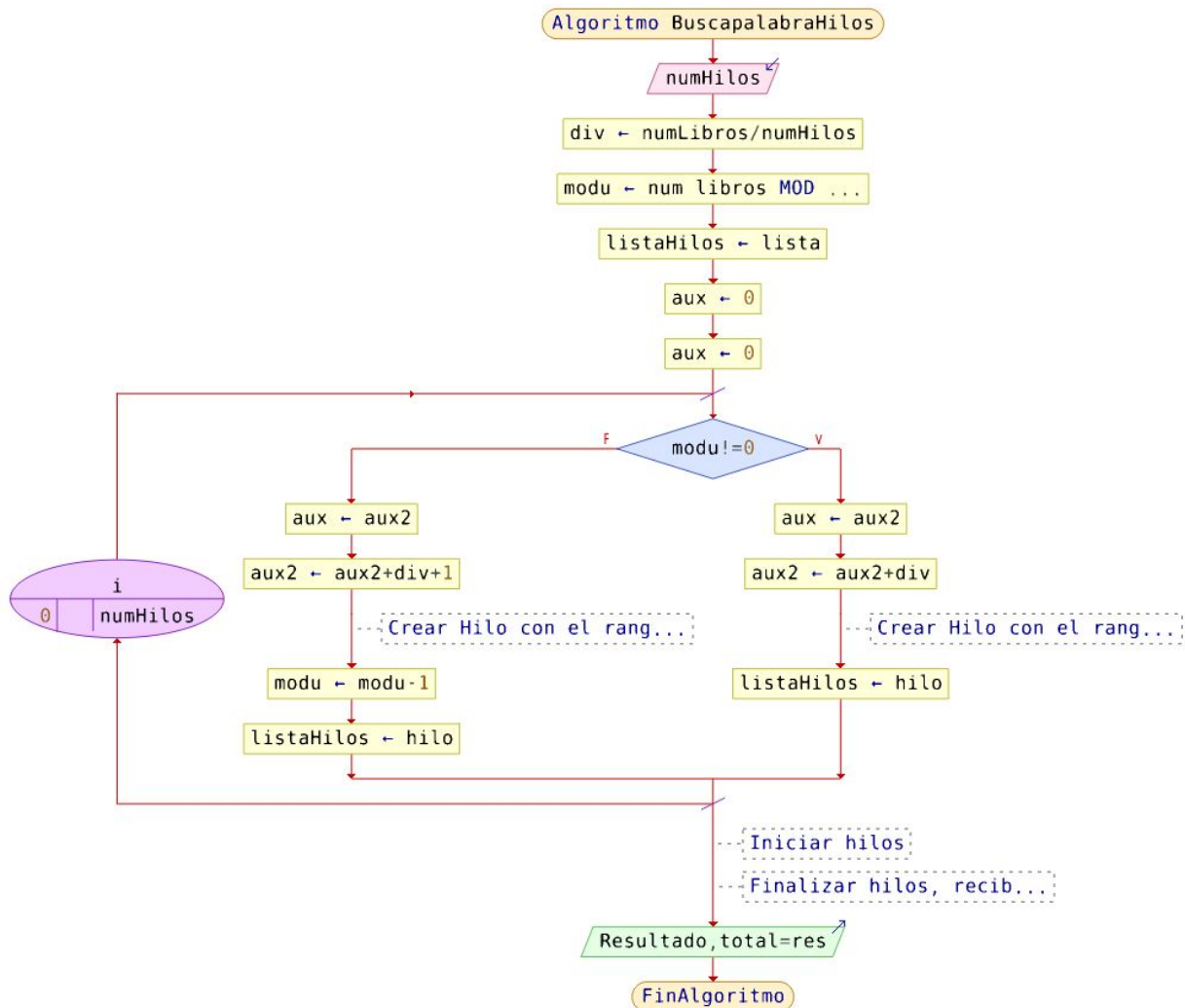
Dado  $n$  número de libros y  $m$  número de hilos el algoritmo debe distribuir los libros de manera proporcional a los hilos siguiendo 2 reglas:

1.- Cada hilo debe trabajar con al menos un archivo, de los cuales obtendrá el conteo de palabras indicadas (alegría, amor, enojo, ira, sueño y aburrimiento) + el total de palabras del archivo.

2.- El padre muestra la estadística final de aparición de palabras en todos los archivos.

## Desarrollo

### Diagrama de Flujo principal

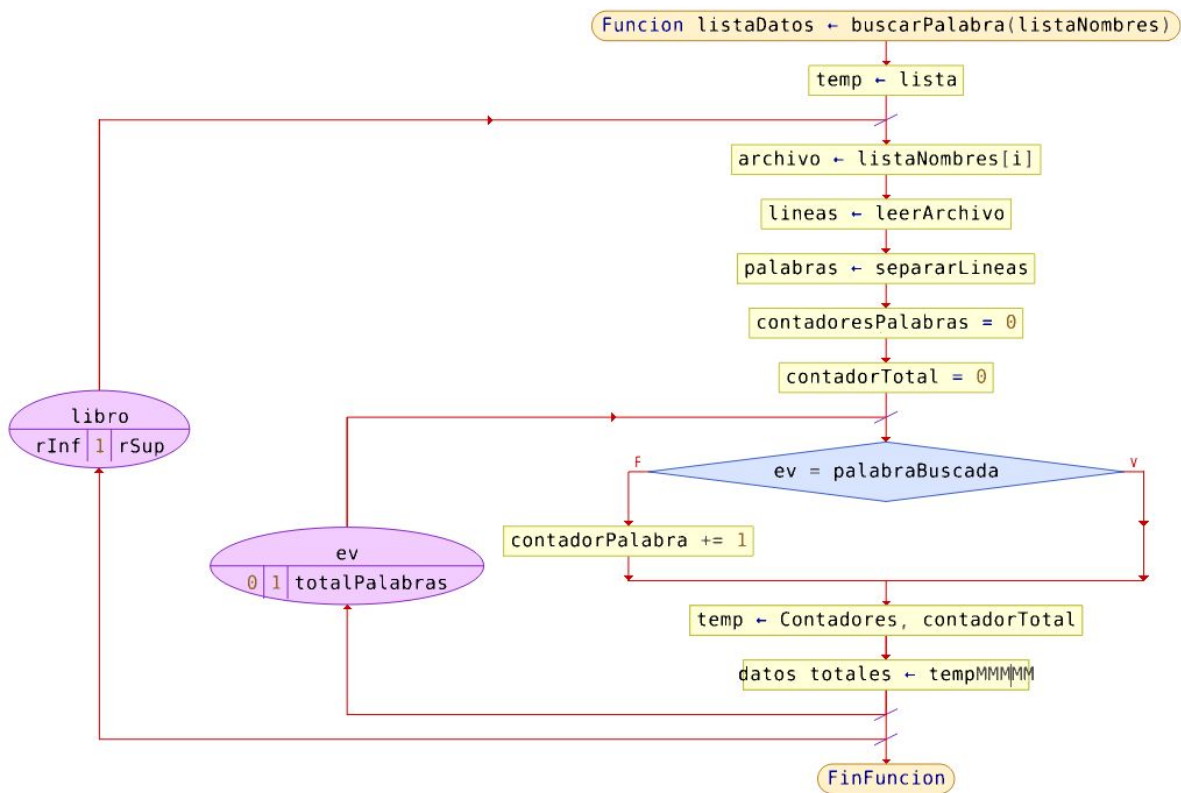


El flujo principal de la aplicación se basa en recibir el número de hilos con el que se va a trabajar evaluando que sean menos hilos que documentos totales, se crean dos variables una con la división entera de número de libros vs número de hilos, seguido a esto se crean 2 variables auxiliares que nos ayudarán a definir los rangos de salto de los documentos a evaluar en la función del hilo.

Posteriormente se ingresa a un ciclo que va desde 0 hasta el número total de hilos, cada iteración se evaluará si existen documentos extra (modulo), estos documentos extra se irán agregando de uno en uno al rango de salto hasta que no queden libros extra, lo que nos permitirá asignar hilos con la mejor distribución para cada uno.

Cuando creamos el hilo lo agregamos a una lista de hilos para su posterior iniciación y finalización.

## Diagrama función BuscaPalabra



La función que ejecutará cada hilo consiste en 2 ciclos anidados donde el primero va a recorrer los libros dado el nombre, para así poder abrir el archivo, leerlo y separar cada palabra. El ciclo inferior se encargará de evaluar cada palabra y compararla para saber si es una palabra que nos interesa, de ser así aumentará el contador de la palabra encontrada, posteriormente se registran los datos en la variable global `datos_totales`.

## Conclusiones

### Cortez Enriquez Jovanny Wilver

Crear hilos en el lenguaje python no es complicado si lo comparamos con los thread de el lenguaje c aunque la lógica es parecida. Lo difícil de esta práctica fue repartir el número de archivos que debe manejar cada hilo de manera que no sobrara ninguno. La solución fue repartir los sobrantes de uno en uno a los n primeros hilos. La lectura del archivo es igual complicado debido a que se debe poner el estar de escritura, en este caso debe ser el estándar UTF8 en archivos de texto. python nos facilitó con funciones de split y strip para poder obtener palabra por palabra y compararlos con la lista de palabras dadas.

### González López Emiliano

Con esta práctica pude entender de mejor manera los hilos gracias a la facilidad que tiene el lenguaje python comparado con C, pero la lógica es bastante parecida por lo cual me ayudó a comprender toda la lógica detrás, también pude comprender la versatilidad que implica dividir la carga de trabajo al reducir notablemente el tiempo de ejecución de un programa que tardaría bastante más. En cuanto al desarrollo de la práctica la parte más difícil fue la creación del algoritmo con el cual haríamos la mejor distribución, aunque el problema parecía trivial, llegar a su solución no fue tan fácil. Ya programando nos ayudamos de las bondades del lenguaje python, ya que nos permite trabajar de una manera sencilla con textos e hilos.

## Referencias

[1] Francisco M. Márquez. (2004). UNIX programación avanzada. México: Alfaomega. p.169