

✓ **Punto 4:** Árbol de procesos del pizarrón por copia exacta de código.

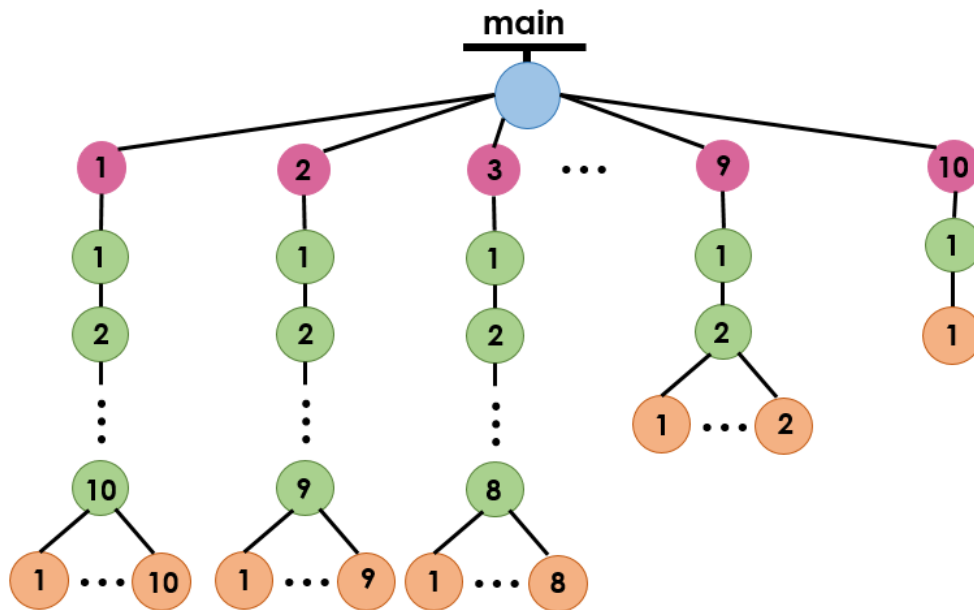


Figura 3: Árbol mostrado en el pizarrón

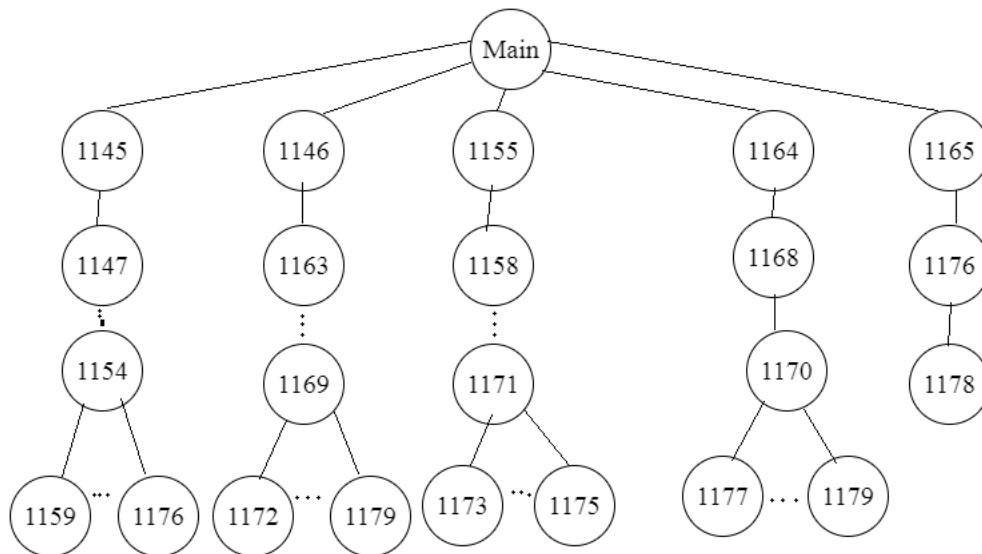


Figura 4: Árbol de procesos con su identificador

- ✓ **Punto 5:** Operaciones con matrices de 10x10 con creación de seis procesos por copia exacta de código y de forma secuencial. Medición de ambos tiempos.

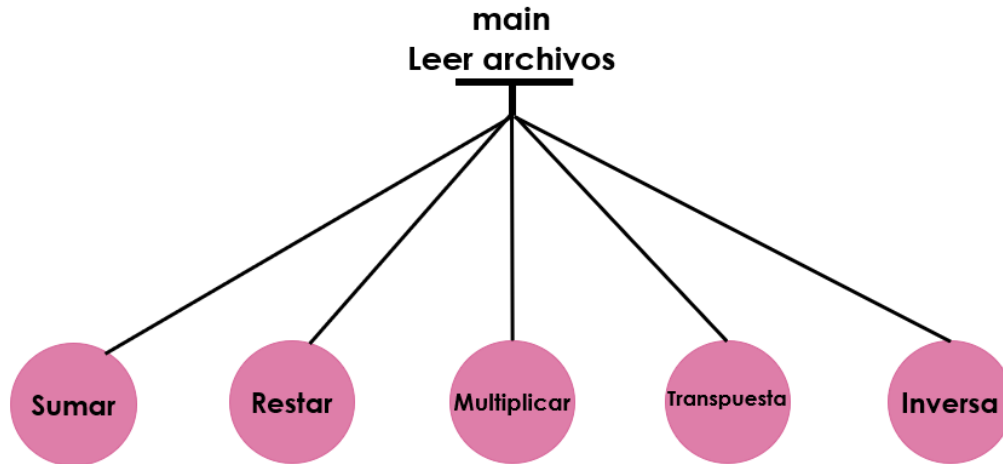


Figura 9: Árbol de los procesos punto 5

- Aplicación secuencial

```

enrike@enrike:~/Escritorio/P4-S0/5$ gcc operacionesMatrices.c tiempo.c -o o
enrike@enrike:~/Escritorio/P4-S0/5$ ./o
Ingrese el nuevo directorio: resultados1
MATRIZ 1
1.000 0.000 6.000 3.000 0.000 3.000 6.000 4.000 5.000 2.000
5.000 4.000 7.000 8.000 2.000 5.000 9.000 7.000 10.000 8.000
4.000 3.000 8.000 1.000 1.000 3.000 10.000 3.000 4.000 1.000
3.000 3.000 1.000 7.000 6.000 0.000 10.000 10.000 2.000 2.000
2.000 7.000 6.000 7.000 3.000 8.000 1.000 1.000 2.000 9.000
9.000 7.000 10.000 7.000 6.000 9.000 8.000 3.000 2.000 10.000
4.000 5.000 3.000 4.000 1.000 7.000 4.000 9.000 5.000 4.000
0.000 5.000 0.000 5.000 1.000 3.000 2.000 0.000 3.000 3.000
7.000 1.000 8.000 4.000 6.000 3.000 2.000 3.000 6.000 4.000
1.000 9.000 9.000 4.000 2.000 8.000 9.000 4.000 4.000 1.000

MATRIZ 2
8.000 5.000 6.000 8.000 10.000 5.000 10.000 10.000 3.000 2.000
2.000 10.000 1.000 10.000 3.000 8.000 0.000 6.000 9.000 5.000
8.000 10.000 3.000 5.000 1.000 3.000 2.000 9.000 7.000 7.000
10.000 4.000 10.000 6.000 10.000 7.000 9.000 9.000 6.000 2.000
9.000 7.000 10.000 0.000 4.000 1.000 6.000 5.000 5.000 2.000
8.000 2.000 0.000 9.000 7.000 1.000 1.000 8.000 10.000 8.000
2.000 8.000 10.000 1.000 1.000 9.000 6.000 10.000 6.000 1.000
1.000 2.000 6.000 10.000 0.000 0.000 0.000 4.000 3.000 5.000
  
```

Figura 10: Genera matrices aleatorias

- ✓ **Punto 7:** Creación de tres procesos por sustitución de código que ejecutarán: Expresión aritmética, cambio de permisos a archivos e inversas de dos matrices.

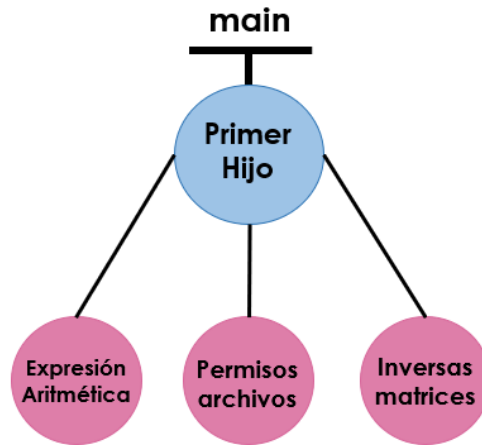


Figura 22: Árbol de los procesos punto 7

```

enrike@enrike:~/Escritorio/P4-S0/7$ gcc expresion.c -lm -o expresion
enrike@enrike:~/Escritorio/P4-S0/7$ gcc permisos.c -o permisos
enrike@enrike:~/Escritorio/P4-S0/7$ gcc inversa.c -lm -o inversa
enrike@enrike:~/Escritorio/P4-S0/7$ gcc 7.c -o 7
enrike@enrike:~/Escritorio/P4-S0/7$ ./7
Soy el hijo 1 ejecutando: /home/enrike/Escritorio/P4-S0/7/expresion
Ingrese la expresion: ((5+3)/8)
Resultado: 1.00
-----

Soy el hijo 2 ejecutando: /home/enrike/Escritorio/P4-S0/7/permisos
Ruta o nombre del archivo: inversa.c

Ingresa 1 para responde SI e ingresa 0 para responder NO

¿Propietario tiene permiso de: lectura? 1
¿Propietario tiene permiso de: escritura? 1
¿Propietario tiene permiso de: ejecucion/busqueda? 0

¿Grupo tiene permiso de: lectura? 1
¿Grupo tiene permiso de: escritura? 0
¿Grupo tiene permiso de: ejecucion/busqueda? 1

¿Otros tiene permiso de: lectura? 1
  
```

Figura 23: Proceso de expresión aritmética y cambio de permisos a archivos

```

¿Otros tiene permiso de: lectura? 1
¿Otros tiene permiso de: escritura? 0
¿Otros tiene permiso de: ejecucion/busqueda? 0

Permisos: 654
Permisos de inversa.c cambiados.
-----

Soy el hijo 3 ejecutando: /home/enrike/Escritorio/P4-S0/7/inversa
MATRIZ 1
5.000 7.000 6.000 5.000 0.000 0.000 4.000 2.000 2.000 7.000
5.000 1.000 6.000 6.000 7.000 3.000 1.000 9.000 8.000 7.000
5.000 10.000 5.000 10.000 3.000 10.000 3.000 5.000 0.000 1.000
5.000 3.000 7.000 9.000 6.000 5.000 9.000 8.000 7.000 10.000
5.000 2.000 0.000 9.000 6.000 5.000 0.000 6.000 4.000 6.000
2.000 9.000 5.000 5.000 9.000 8.000 5.000 10.000 0.000 3.000
10.000 3.000 4.000 6.000 1.000 10.000 0.000 9.000 5.000 6.000
6.000 8.000 8.000 4.000 7.000 1.000 9.000 5.000 5.000 2.000
9.000 7.000 10.000 3.000 2.000 8.000 9.000 5.000 5.000 7.000
6.000 4.000 8.000 10.000 10.000 9.000 7.000 9.000 5.000 1.000

MATRIZ 2
4.000 0.000 10.000 10.000 4.000 4.000 9.000 1.000 9.000 4.000
3.000 7.000 0.000 0.000 8.000 0.000 6.000 4.000 3.000 1.000
  
```

Figura 24: Genera dos matrices aleatorias

- ✓ **Punto 8:** Operaciones con matrices de 10x10 con creación de seis procesos por sustitución código.

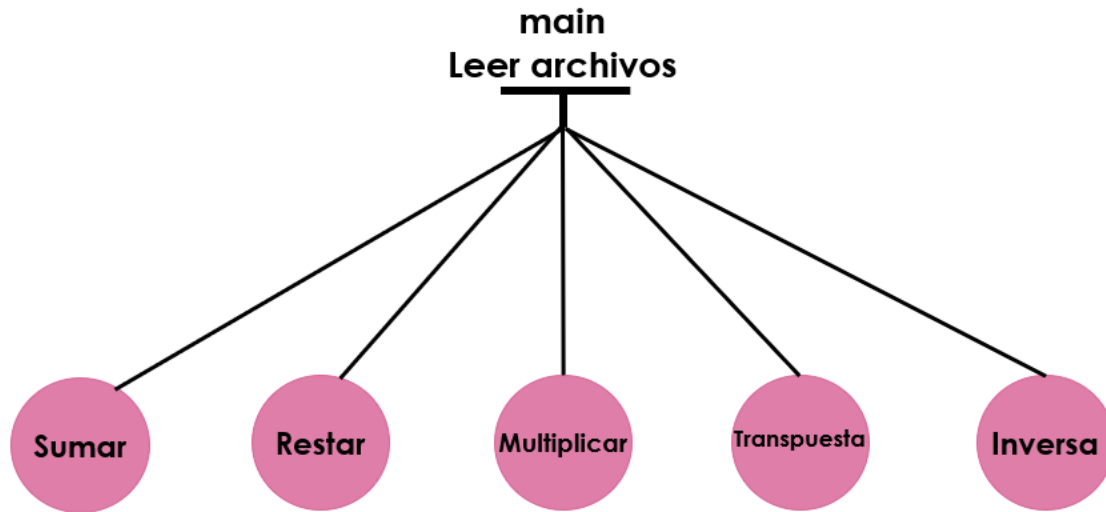


Figura 27: Árbol de los procesos punto 8. Es el mismo que el del punto 5

```

enrike@enrike:~/Escritorio/P4-S0/8$ gcc suma.c -o suma
enrike@enrike:~/Escritorio/P4-S0/8$ gcc resta.c -o resta
enrike@enrike:~/Escritorio/P4-S0/8$ gcc multiplicacion.c -o multiplicacion
enrike@enrike:~/Escritorio/P4-S0/8$ gcc transpuesta.c -o transpuesta
enrike@enrike:~/Escritorio/P4-S0/8$ gcc inversa.c -o inversa
enrike@enrike:~/Escritorio/P4-S0/8$ gcc leerArchivos.c -o leerArchivos
enrike@enrike:~/Escritorio/P4-S0/8$ gcc 8.c -o 8
enrike@enrike:~/Escritorio/P4-S0/8$ ./8
Soy el hijo 1 ejecutando: /home/enrike/Escritorio/P4-S0/8/suma
MATRIZ 1
9.000  0.000  9.000  6.000  8.000  4.000  9.000  8.000  2.000  3.000
8.000  10.000  5.000  9.000  6.000  3.000  8.000  4.000  9.000  4.000
4.000  9.000  10.000  4.000  0.000  9.000  7.000  3.000  2.000  9.000
9.000  0.000  7.000  5.000  5.000  4.000  8.000  3.000  10.000  8.000
4.000  5.000  7.000  9.000  1.000  0.000  2.000  9.000  4.000  9.000
0.000  9.000  6.000  8.000  0.000  4.000  4.000  7.000  7.000  7.000
5.000  5.000  5.000  1.000  9.000  8.000  3.000  6.000  1.000  0.000
3.000  5.000  5.000  8.000  2.000  6.000  8.000  2.000  2.000  10.000
0.000  0.000  6.000  4.000  9.000  7.000  8.000  2.000  3.000  4.000
7.000  7.000  8.000  0.000  6.000  6.000  8.000  8.000  10.000  9.000

MATRIZ 2
8.000  0.000  2.000  1.000  8.000  2.000  7.000  3.000  4.000  8.000
2.000  2.000  6.000  9.000  5.000  4.000  5.000  2.000  5.000  6.000
7.000  10.000  0.000  4.000  10.000  7.000  8.000  8.000  4.000  7.000
4.000  10.000  7.000  4.000  0.000  2.000  4.000  6.000  5.000  6.000
3.000  5.000  7.000  7.000  3.000  1.000  10.000  6.000  3.000  4.000
0.000  10.000  1.000  0.000  1.000  1.000  7.000  9.000  7.000  9.000
3.000  9.000  9.000  8.000  1.000  9.000  8.000  5.000  4.000  2.000
10.000  5.000  6.000  6.000  2.000  7.000  7.000  10.000  3.000  10.000
1.000  1.000  8.000  2.000  1.000  9.000  3.000  7.000  6.000  8.000
5.000  9.000  5.000  1.000  5.000  6.000  0.000  2.000  9.000  2.000
  
```

Figura 28: Proceso Hijo 1: Ejecuta la suma y genera archivo de resultados

✓ **Punto 7:** Árbol de procesos que imprimen su identificador.

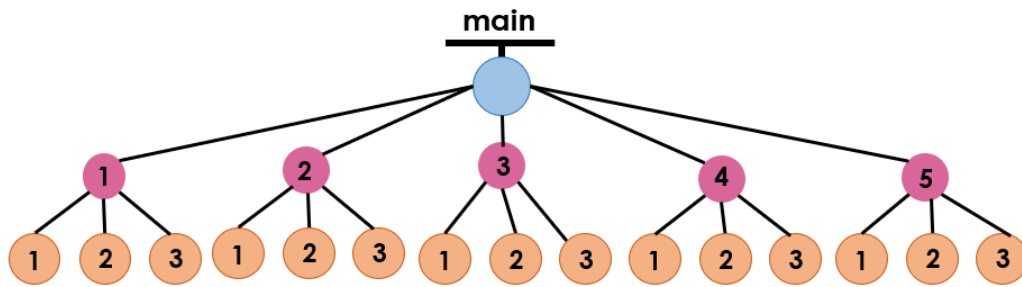


Figura 36: Árbol de los procesos punto 7

```

C:\Users\YaKerTaker\Google Drive\5º SEMESTRE\Sistemas-Operativos\Practica4\Windows\7>gcc 7_nieto.c -o nieto
C:\Users\YaKerTaker\Google Drive\5º SEMESTRE\Sistemas-Operativos\Practica4\Windows\7>gcc 7_hijo.c -o hijo
C:\Users\YaKerTaker\Google Drive\5º SEMESTRE\Sistemas-Operativos\Practica4\Windows\7>gcc 7_padre.c -o padre
C:\Users\YaKerTaker\Google Drive\5º SEMESTRE\Sistemas-Operativos\Practica4\Windows\7>gcc 7.c -o 7
C:\Users\YaKerTaker\Google Drive\5º SEMESTRE\Sistemas-Operativos\Practica4\Windows\7>7 padre
Soy el hijo-padre
ID del Proceso (7056)
ID del Proceso (6852)
ID del Proceso (1060)
Soy el proceso hijo
Creando nietos
ID del Proceso (5408)
Soy el proceso hijo
Creando nietos
ID del Proceso (7812)
ID del Proceso (2836)
Soy el proceso hijo
Creando nietos
ID del Proceso (11176)
ID del Proceso (4396)
Soy el proceso nieto
ID del Proceso (2952)
Soy el proceso hijo
Creando nietos
Soy el proceso nieto
Soy el proceso nieto
ID del Proceso (11084)
ID del Proceso (2736)
Soy el proceso hijo
Creando nietos
ID del Proceso (4792)
ID del Proceso (1516)
Soy el proceso nieto
ID del Proceso (7508)
Soy el proceso nieto
Soy el proceso nieto
ID del Proceso (9972)
Soy el proceso nieto
ID del Proceso (2528)
Soy el proceso nieto
ID del Proceso (4040)
ID del Proceso (9592)
Soy el proceso nieto
Soy el proceso nieto
ID del Proceso (2328)
ID del Proceso (9292)
Soy el proceso nieto
Soy el proceso nieto
ID del Proceso (2616)
Soy el proceso nieto
Soy el proceso nieto
Soy el proceso nieto
C:\Users\YaKerTaker\Google Drive\5º SEMESTRE\Sistemas-Operativos\Practica4\Windows\7>
  
```

| | inversa_1.txt | inversa_2.txt | multiplicacion.txt | resta.txt | suma.txt | transpuesta_1.txt | transpuesta_2.txt |
|----|---------------|---------------|--------------------|-----------|----------|-------------------|-------------------|
| 1 | 249.000 | 175.000 | 306.000 | 117.000 | 240.000 | 241.000 | 218.000 |
| 2 | 280.000 | 186.000 | 224.000 | 195.000 | 229.000 | 227.000 | 160.000 |
| 3 | 323.000 | 250.000 | 296.000 | 218.000 | 324.000 | 361.000 | 254.000 |
| 4 | 299.000 | 147.000 | 270.000 | 250.000 | 284.000 | 190.000 | 212.000 |
| 5 | 342.000 | 257.000 | 281.000 | 228.000 | 272.000 | 290.000 | 173.000 |
| 6 | 246.000 | 174.000 | 241.000 | 226.000 | 250.000 | 185.000 | 148.000 |
| 7 | 233.000 | 154.000 | 265.000 | 193.000 | 255.000 | 252.000 | 196.000 |
| 8 | 359.000 | 243.000 | 307.000 | 273.000 | 307.000 | 290.000 | 247.000 |
| 9 | 305.000 | 234.000 | 331.000 | 231.000 | 298.000 | 269.000 | 228.000 |
| 10 | 191.000 | 138.000 | 155.000 | 99.000 | 117.000 | 121.000 | 100.000 |

Figura 41: Archivos de resultados generados

- Aplicación con seis procesos por copia exacta de código

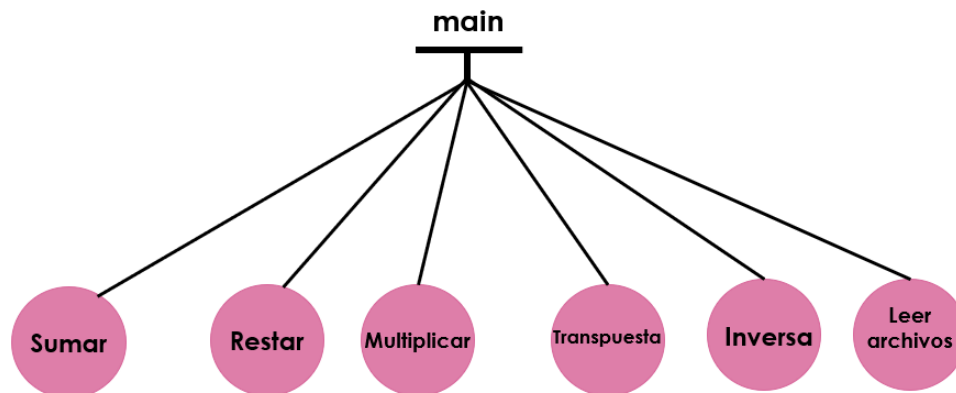


Figura 42: Árbol de procesos para el punto 8

```

C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>gcc suma.c -o suma
C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>gcc resta.c -o resta
C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>gcc multiplicacion.c -o multiplicacion
C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>gcc transpuesta.c -o transpuesta
C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>gcc inversa.c -o inversa
C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>gcc leerArchivos.c -o leerArchivos
C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>gcc 8.c -o 8
C:\Users\YaKerTaker\Google Drive\5to SEMESTRE\Sistemas-Operativos\Practica4\Windows>8

Suma de Matrices, ID del Proceso (8260)
4.000 0.000 2.000 4.000 1.000 3.000 2.000 6.000 1.000 6.000
3.000 6.000 5.000 10.000 9.000 10.000 5.000 5.000 1.000 1.000
10.000 10.000 9.000 5.000 9.000 5.000 2.000 7.000 6.000 10.000
4.000 10.000 5.000 7.000 4.000 3.000 7.000 8.000 10.000 7.000
1.000 8.000 0.000 7.000 6.000 4.000 9.000 6.000 3.000
6.000 7.000 1.000 3.000 4.000 10.000 5.000 7.000 7.000 4.000
10.000 7.000 6.000 9.000 6.000 8.000 6.000 9.000 4.000 8.000
9.000 10.000 4.000 6.000 9.000 4.000 10.000 6.000 0.000 4.000
6.000 0.000 3.000 6.000 5.000 8.000 10.000 6.000 7.000 7.000
8.000 2.000 2.000 5.000 1.000 5.000 7.000 3.000 1.000 3.000
  
```

Figura 43: Proceso Hijo 1: Ejecuta la suma y genera archivo de resultados

| Sistema Operativo | Forma | Tiempo |
|-------------------|--------------|------------|
| Linux | Secuencial | 0.8645 seg |
| | Con procesos | 0.7959 seg |
| Windows | Secuencial | 8.2770 seg |
| | Con procesos | 8.4970 seg |

Figura 50: Tiempos

En el sistema operativo Linux, el usar procesos disminuye el tiempo, sin embargo en Windows no pasa eso, al contrario, el tiempo es mayor con procesos.

5. Conclusiones

Como equipo pudimos llegar a varias conclusiones gracias al desarrollo de esta práctica y todas las clases tomadas en el salón de clases.

Los procesos son creados y destruidos por el sistema operativo; éste se debe hacer cargo de la comunicación entre procesos.

El mecanismo por el cual un proceso crea otro proceso se denomina bifurcación.

El sistema operativo es el responsable de determinar las pautas de intercalado y asignación de recursos a cada proceso.

Esta práctica nos ayudó a reafirmar los conocimientos vistos en la clase, como lo es la creación de procesos por sustitución de código.

En el equipo entramos en un pequeño debate para saber cual era la mejor forma para crear a los procesos. La mayoría coincidió en la creación de procesos por copia exacta de código, gracias a la facilidad de uso y la forma de ver el árbol de procesos, brindando una forma más accesible para programarlo, pero claramente estuvo la contra parte que era la creación de procesos por sustitución de código, donde se encontró que la mejor parte de la creación de procesos por este método es el hecho de que se pueden separar los procesos, cada uno en un archivo y juntarlos en un proceso "padre", a diferencia de los procesos por copia exacta de código.

Finalmente se llegó a la conclusión de que no hay una mejor o peor forma de creación de procesos sino que lo mejor es estudiar bien el problema y determinar cual método de creación de procesos es el mejor para esa situación.