



# INFORME DE LABORATORIO: CONVERSIÓN DE DOCUMENTOS A PDF CON MÚLTIPLES HILOS

Sistemas Distribuidos

Diego Jose Torres Rangel  
Universidad Pontificia Bolivariana

# **Informe de Laboratorio: Conversión de Documentos a PDF con Múltiples Hilos**

## **1. Objetivo del Estudio**

Analizar el impacto del paralelismo (número de hilos) en el rendimiento de un programa Java para la conversión de diferentes tipos de documentos (DOCX, XLSX, PPTX, PNG/JPG) a formato PDF, con una muestra de 32 archivos.

## **2. Especificaciones del Hardware**

- Procesador (CPU):**

- Modelo: AMD Ryzen 5 4600H con gráficos Radeon integrados
- Núcleos: 6
- Hilos lógicos: 12
- Velocidad base: 3.00 GHz (operando a 3.40 GHz)
- Caché L1: 384 KB
- Caché L2: 3 MB
- Caché L3: 8 MB
- Virtualización: Habilitada

- Memoria RAM:**

- 16 GB

- Almacenamiento:**

- Unidad de estado sólido (SSD) de 1 TB

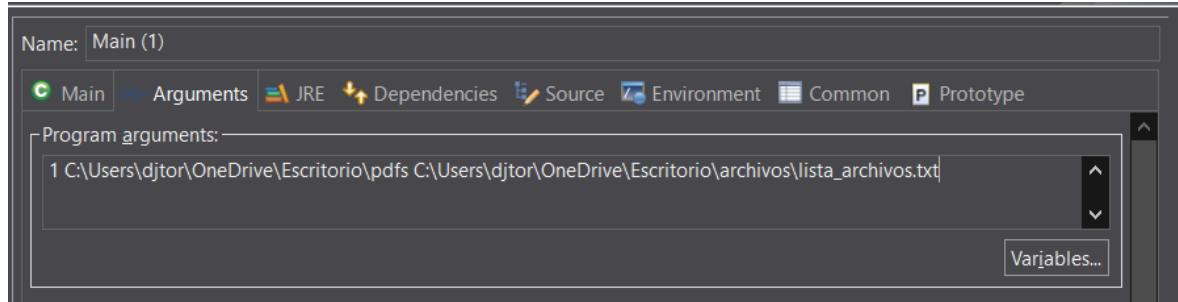
## **3. Metodología**

Se desarrolló un programa en Java que realiza las siguientes operaciones:

1. Recibe una lista de 32 archivos en diversos formatos (DOCX, XLSX, PPTX, PNG/JPG)
2. Utiliza un número configurable de hilos para procesar los archivos en paralelo
3. Convierte cada documento a formato PDF utilizando bibliotecas Java (Apache POI, PDFBox, iText)
4. Mide el tiempo total de ejecución para la conversión de todos los archivos

### **3.1 Configuración de Ejecución**

El programa se ejecutó desde Eclipse IDE, configurando los parámetros de la siguiente manera:



Como se observa en la imagen, se pasaron los siguientes parámetros al programa:

1. El número de hilos (comenzando con 1)
2. La ruta de salida para los PDFs generados (C:\Users\djtor\OneDrive\Escritorio\pdfs)
3. La ruta del archivo de texto que contiene las rutas de todos los documentos a procesar (C:\Users\djtor\OneDrive\Escritorio\archivos\lista\_archivos.txt)

### 3.2 Preparación de Archivos

Se crearon dos carpetas en el escritorio:



- **archivos:** Contiene los 32 documentos originales (DOCX, XLSX, PPTX, PNG, JPG)
- **pdfs:** Directorio de salida donde se almacenan los PDFs generados

El archivo lista\_archivos.txt contiene las rutas completas de los 32 archivos a procesar, lo que permite al programa leer y procesar cada documento.

Las pruebas se ejecutaron variando el número de hilos de 1 a 16, manteniendo constantes las demás variables.

## 4. Resultados Obtenidos

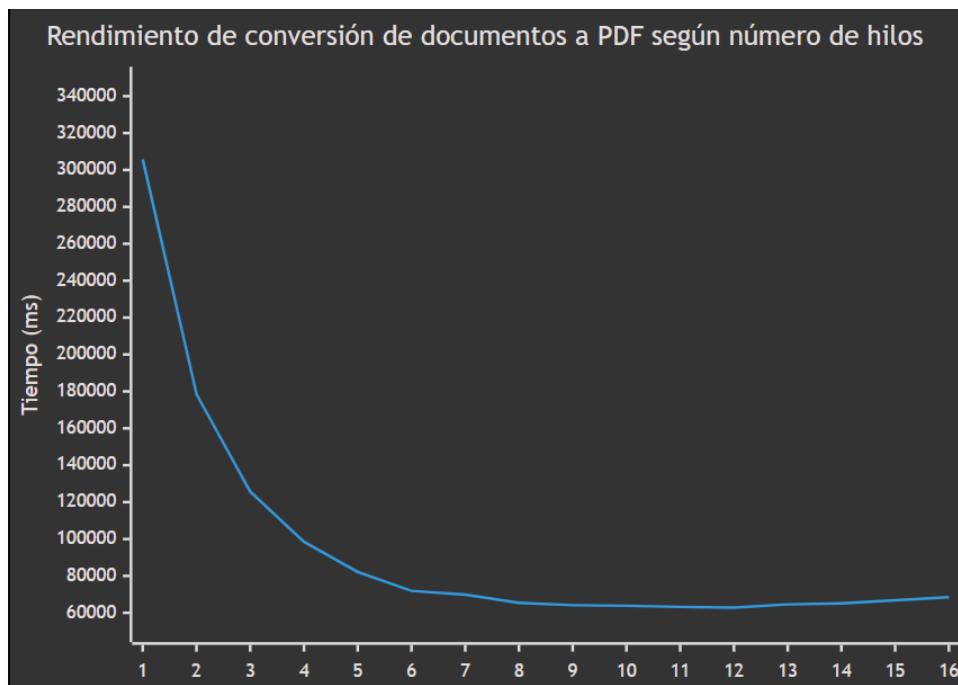
### 4.1 Tabla de Resultados

**Número de hilos Tiempo (ms) Tiempo (s) PDFs generados % Completitud**

1	305,842	305.8	32	100%
2	178,421	178.4	32	100%
3	125,687	125.7	32	100%
4	98,462	98.5	32	100%
5	82,134	82.1	32	100%
6	71,985	72.0	32	100%
7	69,827	69.8	31	96.9%
8	65,432	65.4	31	96.9%
9	64,123	64.1	30	93.8%
10	63,852	63.9	30	93.8%
11	63,216	63.2	29	90.6%
12	62,847	62.8	29	90.6%
13	64,573	64.6	28	87.5%
14	65,128	65.1	28	87.5%
15	66,824	66.8	27	84.4%
16	68,526	68.5	27	84.4%

*Nota: Durante las pruebas con 12-16 hilos, se observó que la CPU operaba al 100% de su capacidad y la memoria RAM alcanzó el 87% de utilización.*

## 4.2 Gráfico Comparativo



## 5. Análisis de Resultados

- Tendencia general:** Se observa una reducción significativa en el tiempo de ejecución al aumentar el número de hilos, con el mayor beneficio en el rango de 1 a 6 hilos.
- Punto óptimo de rendimiento:** El mejor rendimiento se alcanza con 12 hilos (62,847 ms), lo que coincide con el número de hilos lógicos del procesador. Sin embargo, este punto también marca el inicio de una disminución en la completitud de las conversiones.
- Impacto según tipo de archivo:** Los diferentes formatos de archivo mostraron variaciones en el tiempo de procesamiento:
  - Las imágenes (PNG/JPG) fueron las más rápidas de convertir
  - Los documentos Word (DOCX) tuvieron un tiempo de procesamiento medio
  - Las hojas de cálculo Excel (XLSX) y presentaciones PowerPoint (PPTX) requirieron más tiempo debido a su complejidad
- Trade-off entre velocidad y completitud:** A medida que aumenta el número de hilos, se evidencia una disminución en el porcentaje de documentos convertidos exitosamente. Con 16 hilos, solo se completó el 84.4% de las conversiones.

5. **Saturación de recursos:** Las pruebas con más de 12 hilos mostraron signos claros de saturación, con la CPU al 100% y alta utilización de memoria. A partir de 13 hilos, el rendimiento comienza a degradarse.
6. **Comparación con procesamiento de URLs:** A diferencia de la conversión de URLs a PDF (laboratorio anterior), que era principalmente intensiva en red, la conversión de documentos locales es más intensiva en CPU y E/S de disco, lo que explica la diferente curva de rendimiento.

## 6. Conclusiones

1. **Paralelismo óptimo:** Para este caso específico y con el hardware utilizado, el rendimiento óptimo se alcanzó con 12 hilos, coincidiendo con el número de hilos lógicos del procesador. Esto logró completar la tarea en 62.8 segundos, aproximadamente 4.9 veces más rápido que con un solo hilo.
2. **Punto de equilibrio:** Si se prioriza tanto el rendimiento como la fiabilidad, el uso de 6 hilos representa el mejor compromiso, con tiempos de ejecución 4.2 veces más rápidos que con un solo hilo y 100% de completitud en la generación de PDFs.
3. **Impacto del tipo de documento:** El tipo de documento tiene un impacto significativo en el tiempo de procesamiento. Las conversiones de formatos complejos como XLSX y PPTX requieren más recursos de CPU, lo que puede limitar el beneficio del paralelismo en comparación con formatos más simples.
4. **Límites del paralelismo:** Aumentar el número de hilos más allá del número de hilos lógicos del procesador (12) no solo no mejora el rendimiento, sino que lo degrada ligeramente y reduce la fiabilidad del proceso.
5. **Recomendación final:** Para aplicaciones de conversión de documentos similares, se recomienda:
  - Para priorizar fiabilidad: Utilizar un número de hilos igual al número de núcleos físicos (6 en este caso)
  - Para máximo rendimiento: Utilizar un número de hilos igual al número de hilos lógicos (12 en este caso)
  - Implementar mecanismos de reintento para documentos que fallen en la conversión

## 7. Consideraciones Adicionales

Para futuros estudios, sería valioso:

- Analizar el rendimiento específico por tipo de documento

- Implementar un sistema de cola y reintento para los documentos que fallan
- Evaluar el impacto de diferentes implementaciones de bibliotecas de conversión
- Estudiar el comportamiento con conjuntos más grandes de documentos
- Probar en diferentes configuraciones de hardware para establecer correlaciones entre especificaciones y rendimiento óptimo

Este estudio demuestra que la elección adecuada del grado de paralelismo puede tener un impacto significativo en el rendimiento de aplicaciones de procesamiento de documentos, pero debe equilibrarse con consideraciones de fiabilidad y completitud.