**Análisis Comparativo: Fiabilidad de Hilos (FH) vs. Inicialización Temprana (IT)**

Basado en los datos proporcionados, se comparan dos patrones de diseño de concurrencia en un programa idéntico, midiendo tiempos en nanosegundos (ns). **Menor tiempo = mejor rendimiento**.

**1. Resumen de Resultados**

| **Métrica** | **FH (ns)** | **IT (ns)** | **Diferencia (FH - IT)** | **Ganancia de IT** |
| --- | --- | --- | --- | --- |
| **Obtener FileLogger (Promedio)** | 2,495,405 | 1,616,564 | +878,841 | **34.8% más rápido** |
| **100 Escrituras (Promedio)** | 295,199,292 | 309,164,386 | -13,965,094 | 4.7% más lento |

**2. Análisis por Componente**

**A. Obtener FileLogger**

* **IT es significativamente mejor**:
  + Reducción promedio de **35%** en tiempo para obtener el logger.
  + Todos los hilos muestran mejoras:
    - Thread 4: **37% más rápido** (528,131 ns vs. 332,115 ns).
    - Thread 1: **32% más rápido** (692,431 ns vs. 471,414 ns).

**B. 100 Escrituras de Mensajes**

* **FH es ligeramente mejor**:
  + IT es **4.7% más lento** en escrituras.
  + Posible causa:
    - La inicialización temprana podría añadir overhead al gestionar recursos preasignados.
    - Posibles sub procesos del equipo alteraron los resultados en el momento de la prueba.

**3. Análisis por Hilo**

| **Thread** | **Obtener Logger (FH vs. IT)** | **100 escrituras (FH vs. IT)** |
| --- | --- | --- |
| **Thread 1** | 692k ns vs. 471k ns (**↓32%**) | 56.9M ns vs. 54.1M ns (**↓5%**) |
| **Thread 2** | 662k ns vs. 431k ns (**↓35%**) | 69.5M ns vs. 79.3M ns (**↑14%**) |
| **Thread 3** | 612k ns vs. 381k ns (**↓38%**) | 81.0M ns vs. 85.3M ns (**↑5%**) |
| **Thread 4** | 528k ns vs. 332k ns (**↓37%**) | 87.6M ns vs. 90.4M ns (**↑3%**) |

* **Patrón claro**:
  + IT **mejora consistentemente** la obtención del logger (30-38% más rápido).
  + FH **mejora ligeramente** las escrituras (excepto Thread 2, donde IT es 14% peor).

**4. Conclusiones**

1. **Inicialización Temprana (IT) es superior** para:
   * Reducir tiempos de inicialización (**+35% eficiencia**).
   * Escenarios donde la creación de recursos es crítica (ej. servicios de alto tráfico).
2. **Fiabilidad de Hilos (FH) es mejor** para:
   * Procesos con muchas escrituras (**4.7% más rápido**).
   * Cargas de trabajo donde la inicialización no es el cuello de botella.
3. **Recomendación**:
   * Usar **IT** si el programa requiere acceso rápido al logger.
   * Usar **FH** si el foco está en la velocidad de escritura masiva.
   * **Híbrido**: Inicializar temprano el logger (IT) pero gestionar escrituras con FH.