# A. Contexto

A lo largo del desarrollo del buscador bibliográfico hemos ido adoptando las decisiones más propicias para alcanzar una elevada eficiencia computacional: listas ordenadas para conseguir una rápida unión e intersección; tablas de dispersión para acceder de forma inmediata a los libros; árboles que optimizan el uso de memoria y la búsqueda de palabras, etc. Ahora ha llegado el momento de poner las cartas sobre la mesa.

Y es que todas las decisiones de diseño, hasta las más pequeñas, pueden suponer una importante diferencia. Por ejemplo, en un caso con unos 80 libros y 32000 palabras, podemos encontrar ejecuciones: desde los 0,37 segundos del grupo S33, hasta los 8,93 segundos del T8; desde los 1,19 Mbytes de memoria del T37, a los 16,11 Mbytes del G130.

En definitiva, el objetivo de esta práctica es realizar un estudio de eficiencia exhaustivo, riguroso y completo de nuestro buscador bibliográfico, combinando los resultados del análisis teórico con los de un estudio experimental. Los alumnos deben tomar conciencia de la importancia de la eficiencia en el desarrollo de programas, y del necesario rigor matemático y estadístico en el proceso de medición.

# B. Descripción de la práctica

Es bien conocido que la **eficiencia** –en el sentido más amplio– no es un concepto absoluto, sino que debe entenderse como la relación entre **productos obtenidos** y **recursos consumidos** en la consecución de los mismos. La eficiencia de un programa se puede analizar tanto de forma teórica como experimental. Pero, ¿cuáles son los productos y cuáles los recursos?

## B.1. Análisis de factores y recursos

Existen dos recursos básicos en computación: el tiempo y la memoria. En nuestro caso concreto, podemos desglosarlos en los siguientes **recursos**, que serán el objetivo del análisis:

- Tiempo total de inserción de los libros en el buscador (que denotaremos con $\mathbf{t_i}$), es decir, el tiempo de los comandos "**i**".
- Tiempo de las búsquedas con AND ($\mathbf{t_a}$).
- Tiempo de las búsquedas con OR ($\mathbf{t_o}$).
- Memoria ocupada por el programa (**m**).

Se debe estudiar la variación de estos recursos en función de los siguientes **factores**:

- Número total de libros introducidos (**n**).
- Número promedio de párrafos por libro (**c**), sumando los párrafos de todo el libro.
- Número promedio de palabras por párrafo (**p**).
- ¿Existen otros factores relevantes en el programa? Por ejemplo, ¿qué ocurre con el número de apariciones? ¿Y el número de apariciones por párrafo? ¿Existe alguna relación con **p**?

## B.2. El proceso de análisis

Vamos a concretar aquí el trabajo a realizar en esta práctica. De manera simplificada, el estudio consistirá en analizar cómo varían los recursos en nuestro buscador cambiando uno de los factores y manteniendo fijos los demás. Y así para cada factor de interés. En concreto, el análisis constará de las siguientes fases:

**A1. Diseño del proceso.** Determinar de forma precisa los aspectos a medir: (1) recursos a considerar, (2) factores relevantes en cada recurso, (3) intervalos de análisis para cada recurso. Por ejemplo, podemos concretar que el tiempo $\mathbf{t_a}$ se refiere a la media de 10 búsquedas con AND donde aparecen 8 palabras en cada una; o podemos decidir que nos interesa analizar casos donde el número de libros esté en torno a los cien mil.

Después, especificar los tamaños concretos de los factores y el número de pruebas por tamaño, para conseguir un análisis fiable y preciso. Se entiende que cuanto mayor sea el número de pruebas, más completo y fiable será el estudio. En estas pruebas, un factor se va aumentando y todos los demás se mantienen fijos. Por ejemplo, podríamos establecer las tres siguientes pruebas[1]:

| | 1. Estudio del factor n | 2. Estudio del factor c | 3. Estudio del factor p |
|---|---|---|---|
| **n** | 1, 2, 3, 4, 5, 6, 7, 8 | 5 | 5 |
| **c** | 6000 | 2000, 4000, 6000, 8000 | 6000 |
| **p** | 5000 | 5000 | 2500, 5000, 7500, 10000 |

Deben estar claramente indicadas las condiciones en las que se aplican los casos de prueba: el ordenador, sistema operativo, las opciones de compilación, etc. Es importante garantizar que el S.O. no tiene otros procesos que interfieran en las medidas.

**A2. Estudio teórico.** Estudiar de forma teórica las funciones que indican la variación de los recursos en función de los diferentes factores, esto es, analizar en papel $\mathbf{t_i(n, c, p)}$, **m(n, c, p)**, etc. Ojo, no se trata de hacer un análisis del código línea por línea, sino de forma más global y abstracta, teniendo en cuenta las estructuras de datos y los algoritmos que intervienen en cada operación.

El estudio teórico debe conducir a la obtención de los órdenes de complejidad de los recursos en función de los factores considerados. Usar las notaciones asintóticas adecuadas (O, Ω, Θ, o-pequeña). Por ejemplo, después del análisis teórico podemos llegar a la conclusión de que el tiempo de carga de los libros, $\mathbf{t_i(n, c, p)}$, pertenece a un orden de $O(\mathbf{c}^2+\mathbf{p}\cdot\log \mathbf{n})$ (es sólo una suposición). Hacer lo mismo para los 4 recursos, $\mathbf{t_i}$, $\mathbf{t_a}$, $\mathbf{t_o}$, **m**.

Estos órdenes de complejidad se usarán más adelante, dentro del contraste teórico/experimental.

**A3. Creación y ejecución de los casos de prueba.** Ya tenemos establecidos los elementos que hay que analizar, las pruebas, los tamaños y las condiciones. Ahora hay que llevarlo a cabo en el ordenador. Básicamente, cada prueba consiste en: llamar al generador de casos con los tamaños establecidos; ejecutar nuestro buscador con ese caso de prueba; extraer el tiempo y la memoria de esa ejecución; y guardarlo todo en una tabla.

En este paso será necesario programar: (1) para crear los casos de prueba con los tamaños especificados en el paso **A1**; (2) para modificar el buscador, haciendo que muestre el tiempo y la memoria gastados después de cada operación; y (3) para automatizar la ejecución de los casos y el almacenamiento de los resultados.

Para el punto (1), se ofrece a los alumnos un generador de casos de prueba (ver la web indicada abajo). Este generador es configurable, permite especificar muchos parámetros sobre los casos (número de libros, de capítulos por libro, de párrafos por capítulo, etc.), y puede ser modificado y usado por los alumnos si lo creen conveniente. Por ejemplo, podría ser conveniente que las búsquedas analizadas fueran siempre las mismas, en lugar de generarlas de forma aleatoria.

Para el punto (2), se ha dejado también información en la web sobre la medición del tiempo y la memoria de un programa. Estas mediciones pueden hacerse, a su vez, de dos formas diferentes: mediante un programa externo (al estilo del **time** de Linux), o añadiendo en el código del buscador funciones para medir el tiempo y la memoria.

Para el punto (3), hay que tener en cuenta que puede ser necesario repetir el proceso de medición del orden de 600 veces (digamos, 4 recursos por 3 factores por 50 pruebas por

[1] Ojo, esto es sólo un ejemplo ilustrativo, y no muy acertado por lo disparatado de los valores. Los alumnos deben decidir justificadamente qué tamaños pueden ser más razonables y realistas, y hacer un número suficiente y mucho mayor de pruebas para cada estudio.

estudio). Por lo tanto, aconsejamos automatizar lo máximo posible el proceso de ejecución y extracción de tiempos y memoria, por ejemplo, mediante el uso de shell scripts.

**A4. Análisis estadístico y contraste teórico/experimental.** Aprovechando los amplios conocimientos del alumno en estadística, se analizarán los resultados obtenidos de manera formal y rigurosa. Sugerimos las siguientes técnicas: representaciones gráficas, ajustes de regresión, medidas de bondad de ajuste, etc.

Deberá haber una gráfica de cada factor con cada recurso: tiempo de carga en función del número de libros, memoria en función del número de palabras por párrafo, etc. En la documentación, todas las gráficas deben estar comentadas, de forma concisa pero señalando lo más relevante de cada prueba.

Se usarán las funciones teóricas, obtenidas en el paso **A2**, para realizar ajustes de regresión de esas funciones con los datos experimentales, indicando también las medidas de bondad del ajuste. Por ejemplo, si la función teórica es un $O(\mathbf{n}^2)$, la gráfica debe tener forma de parábola, y la regresión será con una función del tipo: $c_1 \cdot n^2 + c_2 \cdot n + c_3$. Para ello, se usarán las herramientas estadísticas/matemáticas adecuadas que dominen los alumnos. Se pide que, como mínimo, se haga un ajuste de regresión de alguna de las funciones.

Además del análisis, se deben mostrar tabularmente los valores obtenidos y usados (es decir, los datos en crudo).

**A5. Cuestiones puntuales.** Esta parte de la práctica es opcional y cuenta un 20% del total. Cada cuestión corresponde a un 10%. Ojo, no se trata de preguntas triviales para contestarlas de forma teórica. Ambas implican modificaciones, algunas pruebas adicionales y análisis de resultados, que deberán documentarse en la memoria.

**Primera cuestión**: el tipo Aparición almacena el ISBN de los libros; si, en su lugar, almacenara un puntero al libro, podríamos obtener de forma directa los datos del libro, y se evitaría el acceso a la tabla de dispersión de libros. ¿Qué mejora se conseguiría en las operaciones de búsqueda almacenando punteros a libros en lugar de ISBN?

**Segunda cuestión**: hasta ahora hemos aplicado el buscador a casos de tamaño pequeño o mediano. Pero ¿qué pasará si lo aplicamos a bases de libros enormes, como la del proyecto Gutenberg? Predecir cuántos recursos consumirá nuestro buscador con 20000 libros, 12 capítulos por libros, 30 párrafos por capítulo y 60 palabras por párrafo. No es necesario ejecutar esos tamaños, se puede hacer a través de una estimación basada en tamaños menores, por ejemplo, mediante un ajuste de regresión.

**A6. Conclusiones.** Por último, habrá que extraer las conclusiones más relevantes del estudio realizado, en cuanto a: valorar la eficiencia obtenida, puntos fuertes y débiles del programa, qué cosas se podrían mejorar, fiabilidad del estudio, resultados del análisis de eficiencia y del contraste teórico/experimental, etc. No olvidar incluir las valoraciones personales de la práctica, la estimación del tiempo tardado y el reparto del trabajo.

Se debe documentar el trabajo y las decisiones tomadas en cada una de estas fases, los programas creados y los resultados obtenidos.

## C. Memoria de la práctica

La memoria de la práctica deberá contener obligatoriamente los siguientes apartados.

### C.0. Portada

Nombre de los alumnos, titulación y **e-mail** de cada uno.

### C.1. Proceso de análisis de eficiencia

Documentación de cómo se han llevado a cabo los pasos descritos en el apartado **B.2**:

- A1. Diseño del proceso.
- A2. Estudio teórico.
- A3. Creación y ejecución de los casos de prueba.
- A4. Análisis estadístico y contraste teórico/experimental.
- A5. Cuestiones puntuales.
- A6. Conclusiones del análisis de eficiencia.

### C.2. Listado del código

En caso de haber creado algún programa para la ejecución automática o semiautomática de las pruebas, incluir aquí el listado. Si se ha modificado el generador de casos o el buscador, incluir sólo las partes que hayan cambiado.

### C.3. Informe de desarrollo

Debe contener lo siguiente:

- Organización temporal de las fases de resolución de esta práctica.
- Cómo ha sido la coordinación y el reparto del trabajo entre los miembros del grupo.
- Conclusiones y valoraciones personales de la práctica.

## D. Evaluación de la práctica

### D.1. Obligatorio

Para aprobar la práctica se requiere que:

- Los programas que sean desarrollados se puedan ***compilar/ejecutar sin errores*** en las máquinas del laboratorio de prácticas.
- La memoria de la práctica debe contener ***todos los puntos*** indicados en el apartado **C**. La memoria debe ser entregada en el plazo que se establezca.
- Todos los ***datos*** (tiempos y memoria usada) deben ser ***ciertos***. La invención o manipulación de los mismos puede suponer no superar la práctica.

### D.2. Criterios de valoración

La práctica se puntuará de acuerdo con los siguientes criterios:

- **Rigor matemático y estadístico**, tanto en el estudio teórico, como en el diseño de casos y en el análisis de resultados.
- Utilización adecuada de las **herramientas** matemáticas y estadísticas que sean necesarias.
- Grado de **automatización** del proceso. Se valorará positivamente la implementación de programas que automaticen las pruebas, frente a la ejecución manual.
- Obtención de **conclusiones** relevantes y acertadas.

### D.3. Otras cuestiones

La práctica se deberá realizar en grupos de **dos alumnos**, en los mismos grupos que los creados para la práctica 1.

Se puede encontrar información adicional en la web para realizar la práctica en:

- **Generador de casos de prueba**: http://dis.um.es/~ginesgm/files/doc/generador08.zip
- **Medida del tiempo y la memoria**: http://dis.um.es/~ginesgm/medidas.html

Se establece como fecha tope de entrega de esta práctica el viernes 18 de abril de 2008. Antes de entregarla, comprobar que están todos los apartados indicados en la sección **C**. Para los alumnos que no entregaron a tiempo la práctica 1, podrán hacer la entrega tardía ese mismo día; en caso contrario, no se podrá entregar la práctica hasta la convocatoria de junio o de septiembre.