

Tarea 2 INFO088 - Taller de Estructuras de Datos y Algoritmos

Instituto de Informática, Universidad Austral de Chile.

Noviembre 2025

Resumen

El objetivo de este trabajo es profundizar en el uso de estructuras de datos de la STL mediante la simulación de un sistema simplificado de resolución de nombres, inspirado en el *Domain Name System* (DNS). En lugar de utilizar dos vectores paralelos, la implementación solicitada emplea ahora un `std::map` que "mapea" directamente nombres de dominio a direcciones IP. Se pide describir las operaciones fundamentales (construcción y búsqueda), medir el tiempo de construcción de la estructura y el rendimiento de búsqueda con la nueva implementación, ademas de realizar un análisis teórico y experimental comparando comportamientos relevantes.

Descripción del problema

Se requiere simular un sistema básico de resolución de nombres de dominio (**DNS**) con una estructura **mapa** donde la clave es el nombre de dominio (ej.: "`www.uach.cl`") y el valor es la dirección IP asociada (ej.: "`146.83.181.2`").

Los requisitos específicos son:

- Construir la estructura **DNS** a partir de un conjunto de datos (pares dominio–IP), conjunto que puede ser simulado por usted.
- Utilizar operaciones de **búsqueda** de `std::map` orientado a resolución de dominio → IP.
- Utilizar operaciones de **inserción** de `std::map` con el objeto de construir la estructura.
- Medir el **tiempo de construcción** de la estructura (tiempo total para insertar n pares dominio–IP).
- Medir tiempos de búsqueda (múltiples repeticiones aleatorias) usando la implementación con `std::map`
- Comparar con la solución anterior basada en vectores (búsqueda secuencial y binaria)

Para cada requisito, usted debe investigar como implementarlo con `std::map`

Experimentación

Se proponen pruebas con distintos tamaños de estructuras:

- Conjuntos pequeños ($n = 200,000$ pares dominio–IP).
- Conjuntos medianos ($n = 1,000,000$ pares).

- Conjuntos grandes ($n = 10,000,000$ pares).

Si no es posible alcanzar esos valores, usted podrá reducirlos y deberá indicar explícitamente los valores usados en su informe.

Para cada caso de tamaño n realizar:

1. **Medir tiempo de construcción:** tiempo total (en ms o s) para insertar los n pares
2. Realizar REP búsquedas aleatorias (con $REP = 100,000$):
 - Registrar el tiempo promedio de las búsquedas.
3. Generar gráficos:
 - **Gráfico con dos líneas:** cantidad de pares (eje x) vs tiempo (eje y) para:
 - **construcción** con la nueva implementación.
 - **ordenamiento** de la estructura implementada en la tarea anterior.
 - **Gráfico comparativo de 3 líneas:** cantidad de pares vs tiempo promedio de búsqueda para:
 - nueva implementación con `std::map`.
 - búsqueda secuencial y búsqueda binaria con vectores (implementación de la tarea 1).

Requisitos de Entrega

Documento. [60 %] (Máximo 14 páginas, todo incluido) El informe técnico deberá tener:

- **Abstract:** Resumen breve del documento, estructura de datos, operaciones, experimentos, resultados, etc.
- **Introducción:** Contexto y problemática, se deben introducir los conceptos claves tales como DNS, `std::map` y la estructura que implementa, etc.
- **Metodología** incluyendo pseudocódigos de construcción y búsqueda con `std::map`, descripción de cómo se generaron los datos y explicación de los métodos utilizados.
- **Resultados experimentales** con gráficos solicitados anteriormente y una descripción breve de cada figura.
- **Análisis asintótico:** describir la complejidad de las operaciones de `std::map` utilizadas (para inserción y construcción).
- **Conclusiones** contrastar análisis teórico y resultados experimentales, discutiendo ventajas/desventajas de `std::map` frente a soluciones alternativas como vectores en el contexto de un resolvedor de nombres.

Código Fuente. [30 %] Basándose en el repositorio guía, la implementación en C++ deberá incluir:

- `main.cpp` que reciba argumento la cantidad de pares, ejecute construcción y experimentación, y produzca resultados/CSV.
- `dns.cpp/dns.h` donde se implementa el resovedor de nombres basado en `std::map`:
- `experimentacion.cpp` con rutinas que miden tiempos (usando `std::chrono`) y generan ficheros con los datos para graficar.
- Se evaluara estructura modular y código documentado. Uso de la STL y manejo cuidadoso de memoria.

Correcciones Tarea 1 [10 %]

- Entregar Tarea 1 con las correcciones correspondientes.

Entrega y revisión

- **Fecha de entrega:** Miércoles 26 de Noviembre 2025
- **Revisión:** Mediante reuniones presenciales la semana del 1 de diciembre.
- **Inscripción:** Selección de horario en hoja de cálculo compartida mediante correo electrónico el día Viernes 28 de Noviembre.

Formato

A continuación el pseudocódigo de una solución, este es el formato de pseudocódigo que se espera para las rutinas de los algoritmos de búsqueda, inserción y eliminación que se piden para cada estructura:

Input: un arreglo de n enteros $A[0 \dots n - 1]$

Output: u , el elemento más repetido en A

```
masRepetido(A, n) {
    u = A[0]
    occ = 0
    for i = 0 to n-1 do {
        count = 0
        x = A[i]
        for j = 0 to n-1 do
            if (A[j] == x) then
                count = count+1
        if (count > occ) then {
            occ = count
            u = x
    }
```

```
    }  
    return u  
}
```