

Emiliano Cabrera Ruiz - A01025453

Programacion de estructuras de datos y algoritmos fundamentales  
22 de septiembre del 2021

## Reflexión

### Preguntas

```
1. ¿Cuántos registros tiene tu archivo?
38511
2. ¿Cuántos records hay del segundo día registrado? ¿Qué día es?
Datos en el primer día (10/8/2020): 3331
Datos en el segundo día (11/8/2020): 3285
3. ¿Alguna de las computadoras pertenece a Jeffrey, Betty, Katherine, Scott, Benjamin, Samuel o Raymond?
Si
Samuel
Benjamin
Jeffrey
Raymond
4. ¿Cuál es la dirección de la red interna de la compañía?
192.168.29.0
5. ¿Alguna computadora se llama server.reto.com?
No
6. ¿Qué servicio de mail utilizan?
Mail          Cantidad
protonmail.com 9506
7. ¿Qué puertos abajo del 1000 se están usando?
Puerto  Cantidad
53       8577
67       681
80       1909
135      124
443      7872
465      3806
965      4866
993      9506
```

1.- 38511 registros

2.- 3285 registros del día 11-08-2020

3.- Hay computadoras pertenecientes a Samuel, Benjamin, Jeffrey y Raymond

4.- 192.168.29.0

5.- No hay computadoras llamadas server.reto.com

6.- De acuerdo al puerto que se utiliza, 993, se utiliza protonmail.com

7.- Se utilizan los puertos 53, 67, 80, 135, 443, 365, 965 y 993

Puerto	Uso
53	DNS, Domain Name System, ayuda a manejar y organizar los nombres de las distintas direcciones IP sobre la red.
67	DHCP, Dynamic Host Config Protocol, da direcciones IP y configuraciones a maquinaria sobre una red.

80	HTTP, HyperText Transfer Protocol, datos que cruzan la World Wide Web
135	RPC, Remote Procedure Call, que permite ejecutar instrucciones en otro hardware
443	HTTPS, un protocolo utilizado en linea encriptado sobre TLS o SSL
465	SMTP autenticado con SSL para cifrar comunicaciones como los correos
965	Servicios de RPC en macOS
993	Utilizado para recibir correos mediante el IMAP encriptado sobre SSL

### Algoritmos

La utilización de algoritmos de ordenamiento no fue necesario ya que el .csv proporcionado venía ordenado de manera cronológica, por lo que se necesitaron únicamente algoritmos de búsqueda. De estos se utilizó inicialmente la búsqueda secuencial, dado el ordenamiento ya hecho y la necesidad de obtener datos de acuerdo a su fecha en la segunda pregunta.

```
// Sequential search
int sequentialSearch( std::vector<Registry> d, bool (*condition)(Registry r) ){
    for(int i = 0; i < d.size(); i++){
        if( condition(d[i]) ) return i;
    }
    return -1;
}

// Sequential search overloading
int sequentialSearch( std::vector<Registry> d, bool (*condition)(Registry a, Registry b), Registry r ){
    for(int i = 0; i < d.size(); i++){
        if( condition(d[i], r) ) return i;
    }
    return -1;
}
```

Tras ver el tamaño de la base de datos y la naturaleza de los requerimientos de las demás preguntas, el mismo algoritmo de búsqueda secuencial fue utilizado. Para obtener los datos de puertos, nombres y demás que se pedían, se necesitaba acceder a todos los registros sin dejar alguno fuera, por lo que concluimos que un algoritmo con complejidad  $O(n)$  era el adecuado.

```
// Binary search
int binarySearch(std::vector<Registry>d, bool (*condition)(Registry r), int start, int end){
    int med = start + (end - start)/2;

    if (end <= start) return -1;
    if (condition(d[med])) return med;

    int left = binarySearch(d, *condition, start, med-1);
    if (left != -1) return left;

    int right = binarySearch(d, *condition, med+1, end);
    if (right != -1) return right;

    return -1;
}
```

Originalmente se había implementado un algoritmo de búsqueda binaria, con complejidad de  $O(\log n)$ , para la búsqueda de pertenencia de las computadoras, pero se vió que como se llamaría múltiples veces tomaría más tiempo que una búsqueda secuencial y por lo que mi camarada y yo optamos por no utilizarlo.

### **Aportaciones**

De una manera muy simplificada, la repartición de trabajo fue semi-equitativa. Mi compañero Do Hyun se encargó de la creación de clases de Reader y Registro junto con sus métodos correspondientes, mientras que yo me enfoqué en los algoritmos de búsqueda y el manejo de vectores de elementos para cada objeto creado. El desarrollo de este último fue en conjunto, ya que ocupaba que mi colega me informara de antemano de los métodos de las clases para que pudiera yo utilizarlos apropiadamente en los vectores correspondientes.

Para el desarrollo en conjunto del programa principal, el main, nos coordinamos mediante llamadas de Zoom o Discord y el manejo de versiones se llevó a cabo en un repositorio de Github, con el fin de evitar errores o fallas que fueran a comprometer el programa a futuro. Esto nos permitió implementar las partes del código que se utilizarían para responder las preguntas con eficiencia y con errores mínimos.

## Referencias

- A list of SMTP and POP3 server. A List of SMTP and POP3 Servers. (n.d.). Recuperado de <https://www.arclab.com/en/kb/email/list-of-smtp-and-pop3-servers-mailserver-list.html>.
- Struct TM. cplusplus.com. (n.d.). Recuperado de <https://www.cplusplus.com/reference/ctime/tm/>.
- Wikimedia Foundation. (2021, September 21). List of TCP and UDP Port Numbers. *Wikipedia*. Recuperado de [https://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers).