



**Tecnológico
de Monterrey**

Reporte Avance de Reto 1

Tomas Diaz Servin A01637531

Isaac Husny Tuachi A01027140

Programación de estructuras de datos y algoritmos fundamentales
(Gpo 100)

Prof. Jorge Rodriguez Ruiz

Preguntas

1. ¿Cuántos registros tiene tu archivo?

Nuestro archivo (Equipo1.csv) tiene 37,360 registros.

2. ¿Cuántos récords hay del segundo día registrado? ¿Qué día es este?

Hay 3338 registros el segundo día, el cual es 11/08/20.

3. ¿Alguna de las computadoras pertenece a Jeffrey, Betty, Katherine, Scott, Benjamin, Samuel o Raymond?

Nombre	
Jeffrey	Si
Betty	No
Kathrine	No
Scott	No
Benjamin	No
Samuel	No
Raymon	No

4. ¿Cuál es la dirección de la red interna de la compañía?

172.31.90 que después es precedido por la dirección de la computadora específica que se utiliza para acceder una página web.

5. ¿Alguna computadora se llama server.reto.com?

No, este tendría que ser el servidor central de la red.

6. ¿Qué servicio de mail utilizan de todos estos: gmail.com, outlook.com, protonmail.com, freemailserver.com?

En la compañía utilizan el servidor de freemailserver, ya que es un dominio al que los empleados acceden con frecuencia. Probablemente esta compañía les ofrece servicio gratis de correo ya teniendo un dominio registrado.

7. Considerando solamente los puertos destino ¿Qué puertos abajo del 1000 se están usando? Lista los puertos e investiga que aplicación/servicio lo utiliza generalmente.

53	Se utiliza para DNS para realizar transferencias de zona, esto para mantener coherencia entre la base de datos del DNS y el servidor.
67	Para enviar información dependiendo la aplicación utilizada
80	Regularmente se usa para comunicación HTTP, envía y recoba desde una interfaz web de cliente
135	Se usa para RPC y servicios como messenger
443	Para comunicación segura desde un navegador web.
465	Para correo electrónico SMTP en certificación SSL
965	Aplicaciones de datos
993	Correo IMAP con certificación SSL

Tipo de búsqueda y ordenamiento

El tipo de búsqueda que empleamos fue **búsqueda binaria** ya que es la mas rapida, ya que como contamos con el ram suficiente para no tener que considerar el espacio en memoria podemos enfocarnos en que el programa sea más eficiente en tiempo de compilación.

Al igual, empleamos el uso del método de ordenamiento **MergeSort** ya que se puede aplicar a archivos de cualquier tamaño, además de ser rápido con listas con un tamaño grande y tener un tiempo de compilación constante.

Complejidad

El número total de complejidad del código es de $O(n * \log n)$.

Function Name	NLOC	Complexity	Token #	Parameter #
Record::Record	10	1	60	
Record::imprimirRecord	3	1	13	
leerDatos	19	4	209	
compararPorFecha	9	3	48	
compararPorHora	9	3	48	
compararPoripFuente	9	3	48	
compararPorPuertoFuente	9	3	48	
compararPorNombreFuente	9	3	48	
compararPoripDestino	9	3	48	
compararPorPuertoDestino	9	3	48	
compararPorNombreDestino	9	3	48	
Sort::intercambiar	5	1	42	
Sort::imprimir	6	2	46	
MergeSort::sort	3	1	38	
MergeSort::sortAux	9	2	85	
MergeSort::merge	35	8	242	
busquedaBinaria	15	4	113	