

## Actividad 2.3

**TC1031.501**

**Programación de Estructuras de  
Datos y Algoritmos Fundamentales**

**Profesor Baldomero Olvera Villanueva**





**Integrantes:**

Matías Kochlowski – a01625364







Esteban Sánchez - a01251440

Michelle Andrea Arceo Solano - a01625268

# Actividad Integral de Estructura de Datos Lineales

bitacora.txt			
			
06-01	0:14:39	895.33.752.33:5974	Failed password for illegal user test
06-01	0:17:32	901.18.919.12:5807	Failed password for illegal user guest
06-01	0:22:14	268.82.665.50:6202	Failed password for root
06-01	0:31:39	745.41.553.21:4925	Failed password for admin
06-01	0:42:50	497.97.988.31:6636	Illegal user
06-01	0:49:25	608.37.179.94:6715	Failed password for admin
06-01	0:55:42	335.95.645.32:6284	Illegal user
06-01	1:18:23	10.43.466.53:6937	Failed password for illegal user test
06-01	1:30:57	930.68.543.89:5825	Failed password for illegal user test
...	...	...	...

En equipos de tres personas, hacer una aplicación que:

-  1 Abra el archivo de entrada llamado "bitacora. " lealo y almacene los datos en una lista doblemente ligada.
-  2 Ordene la información por ip para la realización de las búsquedas.
-  3 Solicite al usuario las ips de inicio y fin de búsqueda de información.
-  4 Despliegue los registros correspondientes a esas IPs.
-  5 Almacenar en un archivo el resultado del ordenamiento.
-  6 Realizar una investigación y reflexión \*nombrada "ReflexAct2.3.pdf") en forma individual sobre:
  - la importancia y eficiencia del uso de las listas doblemente ligadas en una situación problema de esta naturaleza.



- 1
- Abra el archivo de entrada llamado "bitacora.txt", lealo y almacene los datos en una lista doblemente ligada.

bitacora.txt

Jun-01	0:14:39	895.33.752.33:5974	Failed password for illegal user test
Jun-01	0:17:32	901.18.919.12:5807	Failed password for illegal user guest
Jun-01	0:22:14	268.82.665.50:6202	Failed password for root
Jun-01	0:31:39	745.41.553.21:4925	Failed password for admin
Jun-01	0:42:50	497.97.988.31:6636	Illegal user
Jun-01	0:49:25	608.37.179.94:6715	Failed password for admin
Jun-01	0:55:42	335.95.645.32:6284	Illegal user
...	...	...	...

```
$ ~ wc -l bitacora.txt
16806
```

→

16806 Registros

N = 16806

txt - String

Oct 30 21:39:26 679.57.853.40:5668 Failed password for root  
Oct 30 21:52:57 164.48.60.79:6597 Failed password for illegal user test  
Oct 30 22:06:30 774.16.162.29:4184 Illegal user



Lista Doblemente Ligada

{  
  "679.57.853.40:5668,10-30T21:39:26,Failed password for root ",  
  "164.48.60.79:6597,10-30T21:52:57,Failed password for illegal user test ",  
  "774.16.162.29:4184,10-30T22:06:30,Illegal user "  
}

Este dato es útil para el cálculo de la complejidad tiempo de los algoritmos (de eficiencia del uso de las listas doblemente ligadas).

Nos permitirá realizar un evaluación.



## 2 Ordene la información por fecha para la realización de las búsquedas.

bitacora\_AZ.csv

```
ipAddress,dateTime,errorMessage
1.14.815.77:4402,10-27T18:48:25,Failed password for admin
1.16.249.84:5139,09-19T12:46:04,Failed password for root
1.26.288.72:4768,07-26T22:52:23,Failed password for illegal user guest
1.29.485.70:4613,08-13T20:00:40,Failed password for illegal user guest
1.36.64.55:6486,09-13T06:15:34,Failed password for illegal user test
1.36.968.91:4437,06-20T15:08:11,Failed password for root
1.37.180.51:5825,08-02T08:48:25,Failed password for root
1.47.156.72:6475,08-06T06:56:34,Failed password for admin
1.47.233.24:5247,09-27T08:51:14,Illegal user
1.50.342.29:5546,09-05T18:55:39,Illegal user
1.55.405.18:6734,06-01T04:48:20,Failed password for illegal user guest
1.6.378.65:6772,06-05T04:12:47,Failed password for root
1.61.436.96:4201,09-13T23:30:55,Illegal user
1.62.197.95:5645,10-14T20:40:34,Illegal user
1.65.639.78:4398,10-03T19:11:43,Illegal user
1.65.936.39:6297,06-12T09:22:48,Failed password for illegal user guest
1.68.311.92:4927,07-12T11:58:38,Failed password for root
1.74.592.32:5365,09-04T12:21:09,Failed password for illegal user test
1.77.465.37:5059,10-01T10:40:08,Failed password for admin
1.81.892.90:4688,06-14T09:10:33,Illegal user
1.93.577.53:4679,06-13T14:43:51,Failed password for admin
1.94.523.85:5883,07-12T10:13:03,Illegal user
....
```

```
// Función para insertar posterior
void insercionOrdenada(Node **head, Node **tail, std::string key) {
    Node *p = new Node();
    p->data = key;
    p->next = NULL;
    // 1er Nodo insertado en LDL
    if ((*head) == NULL) {
        (*head) = p;
        (*tail) = p;
        (*head)->prev = NULL;
        return;
    }
    // Si nuevo nodo tiene valor menor
    if ((p->data) < ((*head)->data)) {
        p->prev = NULL;
        (*head)->prev = p;
        p->next = (*head);
        (*head) = p;
        return;
    }
    // Si nuevo nodo tiene valor mayor
    if ((p->data) > ((*tail)->data)) {
        p->prev = (*tail);
        (*tail)->next = p;
        (*tail) = p;
        return;
    }
    // Encontrar nodo previo a p
    Node *temp = (*head)->next;
    while ((temp->data) < (p->data))
        temp = temp->next;
    // Insertar nuevo nodo antes de p
    (temp->prev)->next = p;
    p->prev = temp->prev;
    temp->prev = p;
    p->next = temp;
}
```



- 3 Solicite al usuario las IPs de inicio y fin de búsqueda de información.

```
$ ~ ./bitacora
```

```
Esta base contiene 16,806 registros de intentos fallidos de ingreso al sistema desde  
01-Jun hasta 30-Oct.
```

```
Para realizar una búsqueda, por favor ingrese el rango de IPs deseado en formato  
0.0.0.0:0000:
```

```
Ingrese IP (in). Desde: 0.0.0.0:0000
```

```
Ingrese IP (fi). Hasta: 999.999.999.999:9999
```

```
Buscando resultados entre 0.0.0.0:0000 y 999.999.999.999:9999
```



## 4 Despliegue los registros correspondientes a esas IPs

```
$ ~ ./bitacora
```

```
./bitacora
```

Esta base contiene 16,806 registros de intentos fallidos de ingreso al sistema desde 01-Jun hasta 30-Oct y que detalla IPs y puertos origen de requests.

Para realizar una búsqueda, por favor ingrese el rango de IPs deseado en formato 0.0.0.0:0000

Ingrese IP y Puerto (0.0.0.0:0000). Desde: 1.6.378.65:6772

Ingrese IP y Puerto (999.999.999.999:9999). Hasta: 1.81.892.90:4688

Buscando resultados entre 1.6.378.65:6772 y 1.81.892.90:4688.

```
1.6.378.65:6772,06-05T04:12:47,Failed password for root
1.61.436.96:4201,09-13T23:30:55,Illegal user
1.62.197.95:5645,10-14T20:40:34,Illegal user
1.65.639.78:4398,10-03T19:11:43,Illegal user
1.65.936.39:6297,06-12T09:22:48,Failed password for illegal user guest
1.68.311.92:4927,07-12T11:58:38,Failed password for root
1.74.592.32:5365,09-04T12:21:09,Failed password for illegal user test
1.77.465.37:5059,10-01T10:40:08,Failed password for admin
```

```
.....
```

## 5 Almacenar en un archivo el resultado del ordenamiento.

output/resultados1.6.378.65:6772a1.81.892.90:4688.csv

```
ipAddress,dateTime,errorMessage
1.6.378.65:6772,06-05T04:12:47,Failed password for root
1.61.436.96:4201,09-13T23:30:55,Illegal user
1.62.197.95:5645,10-14T20:40:34,Illegal user
1.65.639.78:4398,10-03T19:11:43,Illegal user
1.65.936.39:6297,06-12T09:22:48,Failed password for illegal user guest
1.68.311.92:4927,07-12T11:58:38,Failed password for root
1.74.592.32:5365,09-04T12:21:09,Failed password for illegal user test
1.77.465.37:5059,10-01T10:40:08,Failed password for admin
```

output/bitacora\_AZ.csv

```
ipAddress,dateTime,errorMessage
1.14.815.77:4402,10-27T18:48:25,Failed password for admin
1.16.249.84:5139,09-19T12:46:04,Failed password for root
1.26.288.72:4768,07-26T22:52:23,Failed password for illegal user guest
1.29.485.70:4613,08-13T20:00:40,Failed password for illegal user guest
1.36.64.55:6486,09-13T06:15:34,Failed password for illegal user test
1.36.968.91:4437,06-20T15:08:11,Failed password for root
1.37.180.51:5825,08-02T08:48:25,Failed password for root
1.47.156.72:6475,08-06T06:56:34,Failed password for admin
1.47.233.24:5247,09-27T08:51:14,Illegal user
1.50.342.29:5546,09-05T18:55:39,Illegal user
1.55.405.18:6734,06-01T04:48:20,Failed password for illegal user guest
1.6.378.65:6772,06-05T04:12:47,Failed password for root
1.61.436.96:4201,09-13T23:30:55,Illegal user
1.62.197.95:5645,10-14T20:40:34,Illegal user
1.65.639.78:4398,10-03T19:11:43,Illegal user
1.65.936.39:6297,06-12T09:22:48,Failed password for illegal user guest
1.68.311.92:4927,07-12T11:58:38,Failed password for root
1.74.592.32:5365,09-04T12:21:09,Failed password for illegal user test
1.77.465.37:5059,10-01T10:40:08,Failed password for admin
1.81.892.90:4688,06-14T09:10:33,Illegal user
1.93.577.53:4679,06-13T14:43:51,Failed password for admin
1.94.523.85:5883,07-12T10:13:03,Illegal user
```

\*\*\*\*\*



- 6 Considerando una situación problema de esta naturaleza, realizar una investigación y reflexión sobre:
- la importancia y eficiencia del uso de las listas doblemente ligadas.

Las listas ligadas (estructura de datos lineal) están ordenados por apuntadores (una segunda lista que los ubica) sin cambiar su lugar en memoria.

Esto hace que se requiera más espacio (para esta segunda lista de apuntadores) en comparación a reordenar una sola lista (reordenar una sola lista cada vez que se agregar un elemento no es óptimo).

Estructura de datos lineales	Eficiencia (Worst Case Scenario)
Listas Ligadas [1]	$O(n)$ , al no especificar longitud de datos, pueden guardar la cantidad de información que la memoria puede almacenar, por cada información se crea un nodo y se insertan los datos ahí mismo.
Pila [1]	$O(n)$ , se agregan y se eliminan por el mismo lado por lo que tiene una estructura LIFO (Last-In, First-Out).
Cola [1]	$O(n)$ , se insertan los datos de un lado y se eliminan por el otro.
Deque [1]	$O(n)$ , se pueden insertar y eliminar por cualquiera de sus dos lados. (Mejor uso con listas doblemente ligada)

La lista doblemente ligada (a diferencia de la simple) tiene doble vínculo (a su termino previo y a su posterior), ofreciendo la posibilidad de recorrer la lista de un lado o del otro.



# Bibliografía

---

<https://www.geeksforgeeks.org/doubly-linked-list/>

<https://www.geeksforgeeks.org/insertion-sort-doubly-linked-list/>

