

**TAREA N°1**  
***Programación en C: Aplicación de Listas Dinámicas.***

**Fecha de envío:** Lunes 24 de abril, a las 23:55 hrs.

**Modalidad:** Trabajo individual

**I. OBJETIVOS.**

El objetivo del presente laboratorio es evaluar tu capacidad para:

1. Llevar a cabo un programa completo en el lenguaje de programación C, utilizando todos los elementos básicos que provee el lenguaje, y sus tipos de datos básicos.
2. Aplicar las buenas prácticas en programación (orden, comentarios, identificadores representativos).
3. Ser capaz de cumplir con una interfaz concreta solicitada para el programa.
4. Diseñar e implementar una estructura lineal dinámica para resolver un problema que requiera búsqueda de información.

**II. ENUNCIADO.**

El mundialmente conocido festival de música **Lollapalooza**, ha decidido ofrecer una jornada extra de presentación de bandas musicales. Para atraer nuevamente a los fanáticos, la empresa que vende las entradas va a generar una súper promoción en la que las 50 primeras entradas que se vendan tendrán un descuento del 50% en su valor. Los interesados deberán inscribirse a través del sitio web a partir de las 00:00 una semana antes del evento, ingresando su RUT, su nombre y apellido, y la cantidad de entradas que desea comprar. Solamente se puede comprar un máximo de dos entradas por persona con este descuento especial.

La información ingresada en el sitio web se registrará en un archivo de texto. Cada una de sus líneas contendrá cada inscripción realizada y su formato será:

`Rut,nombre y apellido,número de entradas solicitadas`

En la figura 1, se muestra un ejemplo del contenido de un archivo que posee este formato de entrada. Observa que los datos están separados por una coma, sin espacios extra.

12333444-2,Maria Reyes,2
13444555-6,Pedro Diaz,1
19000111-3,Jose Herrera,1
13444555-6,Pedro Diaz,2

**Figura 1:** Ejemplo del formato en el archivo de entrada

Debes construir un programa en C, que reciba el nombre de un archivo de texto con la información recién descrita, y que genere dos archivos de salida: El primero, con el listado de las personas que lograron adquirir alguna(s) de las 50 entradas en promoción, y el segundo, con el listado de las personas que quedaron en lista de espera (fuera de la promoción), quienes serán participantes de una siguiente promoción del evento. Para esto, tu programa debe:

1. Respetar el orden de inscripción que aparece en el archivo de entrada.
2. No vender más de 2 entradas por persona: una persona se puede inscribir en la página web cuantas veces quiera lo que tendrá la siguiente penalización:
  - a. Si la persona solicita en total, 1 ó 2 entradas, no habrá penalización, y se le asignarán la cantidad de entradas solicitadas (no superando las 50). Si no alcanzó la promoción porque se habían vendido ya las primeras 50 entradas, será ingresada a la lista de espera.
  - b. Si la persona solicita entre 3 y 5 entradas, se le asignarán 2 y será considerada penalizada, por lo que no será agregada a la lista de espera.
  - c. Si la persona solicita más de 5 entradas, no se le asignará ninguna y tampoco será agregada a la lista de espera.

NOTA: Se considera como total de entradas solicitadas a la suma de todas las que pidió una persona en todas las inscripciones en el sitio web.

3. El nombre del primer archivo de salida (la lista de asignaciones) debe ser el mismo que el de entrada, pero deberás cambiar la extensión a: **“.asg”**.
4. En el archivo de la lista de asignaciones, deberá aparecer la siguiente información:  
Rut, nombre y apellido, número de entradas asignadas
5. Las 2 últimas líneas del archivo de asignaciones deberán contener el número de entradas asignadas y el número de entradas por asignar, **en caso de que la cantidad de inscripciones válidas no alcance las 50 entradas en oferta**, con el siguiente formato:  
Asignadas: total asignadas  
Pendientes: total pendientes
6. El nombre del segundo archivo de salida (la lista de espera) debe ser el mismo que el del archivo de entrada, pero deberás cambiar la extensión a: **“.esp”**.
7. En el archivo de la lista de espera, deberá aparecer la siguiente información:  
Rut, nombre y apellido.
8. La lista de espera solo debe ser generada en caso de que todas las entradas **hayan sido asignadas**, pero aún queden personas válidamente inscritas (es decir, que **no se les haya penalizado**, siguiendo el criterio de los puntos 2.a y 2.b). Si todas las personas que no

alcanzaron las 50 entradas fueron penalizadas, o si las inscripciones válidas no alcanzaron las 50 entradas disponibles, **omita generar este archivo**.

A continuación, aparecen dos ejemplos mostrando lo que debe lograr tu programa para una entrada dada que cubren todos los aspectos aquí explicados.

### Ejemplos:

En la figura 2(a) aparece un ejemplo de archivo de entrada llamada “**datos.txt**”. En la figura 2(b) aparece la salida exacta que debe generar tu programa, en un archivo llamado “**datos.asg**”. Como puedes observar, Maria Reyes y José Herrera solicitaron a lo más 2 entradas, por lo que se le asignó la cantidad solicitada. Rocío Morales, en cambio, pidió un total de 5 entradas, por lo que tuvo penalización y obtuvo solo dos entradas.

Observa que Juan Pardo no aparece en el archivo de salida: esto se debe a que solicitó un total de 18 entradas, por lo que tuvo la mayor penalización, no obteniendo entrada.

Finalmente, observa además que cada persona puede escribir con mayúsculas, minúsculas y con o sin espacios su nombre. Esto no debe ser relevante, pues el verdadero identificador de cada persona es el rut ingresado.

```
12333444-2,Maria Reyes,2
20165233-4,Rocio Morales,2
20165233-4,RocioMorales,2
13444555-6,Pedro Diaz,1
19000111-3,Jose Herrera,1
13444555-6,Pedro Diaz,2
12432008-k,Juan Pardo,3
20165233-4,rocio morales,1
12432008-k,Juan pardo,5
12432008-k,juan Pardo,6
12432008-k,juanpardo,4
```

(a)

```
12333444-2,Maria Reyes,2
20165233-4,Rocio Morales,2
13444555-6,Pedro Diaz,2
19000111-3,Jose Herrera,1
Asignadas: 7
Pendientes: 43
```

(b)

**Figura 2:** Ejemplo de lo que debe obtener tu programa: (a) Archivo llamado “datos.txt”, con la información de las personas inscritas en la página web; (b) Archivo llamado “datos.asg” con la información que debe generar tu programa.

En la figura 3(a), aparece un segundo ejemplo de archivo de entrada, también llamado “**datos.txt**”. Este nuevo ejemplo toma un caso en el que las 50 entradas son asignadas en la figura 3(b) en el archivo “**datos.asg**”. Sin embargo, aún quedan personas válidamente inscritas, por lo cual se genera una lista de espera en otro archivo llamado “**datos.esp**”, como se ve en la figura 3(c). Observa también que Federico Santis queda fuera de la lista de espera al haber sido penalizado.

12333444-2,Maria Reyes,2	12333444-2,Maria Reyes,2
20165233-4,Rocio Morales,2	20165233-4,Rocio Morales,2
13444555-6,Pedro Diaz,2	13444555-6,Pedro Diaz,2
19000111-3,Jose Herrera,2	19000111-3,Jose Herrera,2
12432008-k,Juan Pardo,2	12432008-k,Juan Pardo,2
23748506-5,ManuelMartinez,2	23748506-5,ManuelMartinez,2
7395526-k,EustaquioBarros,2	7395526-k,EustaquioBarros,2
20184756-7,EusebioLillo ,2	20184756-7,EusebioLillo ,2
21854748-0,Armando Casas,2	21854748-0,Armando Casas,2
17484564-9,JohnnySabia,2	17484564-9,JohnnySabia,2
9264783-k,Ines Perado,2	9264783-k,Ines Perado,2
15894369-2,JaimeGonzalez,2	15894369-2,JaimeGonzalez,2
31415926-5,TrinidadQuintana,2	31415926-5,TrinidadQuintana,2
8979323-8,Miguel Henriquez,2	8979323-8,Miguel Henriquez,2
19454746-6,Claudia Logan,2	19454746-6,Claudia Logan,2
15384654-8,EstebanQuito,2	15384654-8,EstebanQuito,2
20846758-9,AstridMendez,2	20846758-9,AstridMendez,2
22754894-5,Estela Vial,2	22754894-5,Estela Vial,2
9374853-3,Jorge Gonzalez,2	9374853-3,Jorge Gonzalez,2
18394538-0,Esther Eotipo,2	18394538-0,Esther Eotipo,2
7326734-3,Alonso Bulnes,2	7326734-3,Alonso Bulnes,2
14784964-k,FantasMontt,2	14784964-k,FantasMontt,2
12948945-9,BernardoNeira,2	12948945-9,BernardoNeira,2
24985869-2,YamilaUrrutia,2	24985869-2,YamilaUrrutia,2
13876037-5,FernandoGomez,2	24985869-2,YamilaUrrutia,2
20758493-k,Camila Narea,2	13876037-5,FranciscoGotte,2
2349573-5,Aquiles Vera,1	
20474853-0,Alberto Mora,2	
9386624-k,FedericoSantis,5	
16934659-4,IgnacioNunez,1	
	Asignadas: 50
	Pendientes: 0

(a)

(b)

20758493-k,Camila Narea  
2349573-5,Aquiles Vera  
20474853-0,Alberto Mora  
16934659-4,IgnacioNunez

(c)

**Figura 3:** Segundo ejemplo de lo que debe obtener tu programa: (a) Archivo llamado “datos.txt”, con la información de las personas inscritas en la página web; (b) Archivo llamado “datos.asg” con las entradas asignadas; (c) Archivo llamado “datos.esp” con los datos de la lista de espera.

### III. CONSIDERACIONES EN LA REVISIÓN.

#### Sobre el código:

1. Tu programa **debe seguir exactamente** las **reglas e interfaces** explicadas en este documento.
2. Debes construir **una función para cada tarea que realice tu código**, cuidando los tipos de datos de las entradas y de las salidas.
3. Debes usar identificadores representativos para tus variables, parámetros de entrada y funciones.
4. Debes comentar cada una de tus funciones, estructuras y tipos de datos que definas, indicando una descripción de la labor que lleva a cabo, sus entradas, y sus salidas. En la figura 3 aparece un ejemplo de cómo debes hacerlo.

```
/*  
Entrada: número entero que se revisará.  
Salida: número entero que corresponde al número de divisores.  
Proceso: función que recibe un entero y calcula y retorna el total  
de divisores que posee.  
*/  
int numeroDivisores(int n)  
{  
    int i=2, cont=0;  
  
    while (i < n)  
    {  
        if (n%i == 0)  
        {  
            cont++;  
        }  
        i++;  
    }  
    return cont+2;  
}
```

**Figura 4:** Ejemplo de cómo debes comentar una función en esta tarea.

9. Tu código debe estar correctamente *indentado* (uso de sangrías para cada sub-bloque de instrucciones), esto incluye el correcto alineamiento de las llaves (“{” y “}”) que enmarcan tales bloques. Se penalizarán códigos desordenados en este aspecto.
10. Tu código no puede presentar más de 1 línea en blanco. Se penalizarán códigos que no cumplan con este punto.
11. Puedes trabajar con el IDE y compilador de C que más te acomode. No obstante, tu código debe poder ser compilado y ejecutado sin problemas en Windows.
12. El sistema debe ser **robusto**, se penalizarán los errores no manejados de cualquier tipo.
13. Debes usar listas dinámicas para resolver esta tarea. Si se ocupa otro tipo de solución, la nota será **1.0**.

#### IV. SOBRE LA ENTREGA, ATRASOS Y FALTAS A LA ÉTICA.

1. Debes subir tu trabajo **al aula virtual de tu curso de cátedra** dentro de la plataforma <https://canvas.unab.com/>, en una casilla que se habilitará especialmente para esto.
2. **NO SE ACEPTARÁN TRABAJOS ATRASADOS.**
3. Tu trabajo es **individual**, no se aceptarán trabajos en grupo.
4. Debe subir un archivo con el código fuente de tu programa.
5. El nombre de tu archivo debe ser “**paterno\_materno\_nombre.c**”
6. Si el programa no se puede ejecutar, tendrá la nota mínima: **1.0**. Si tu programa funciona, se evaluará su ejecución, y también el código fuente.
7. Ante el escenario de existir sospecha de copia (con otros compañeros, o desde internet), será interrogado acerca de su trabajo, para aclarar dudas de su entendimiento y autoría. Si se confirma la sospecha, el trabajo será evaluado con nota **1.0**.
8. Las consultas las debe realizar directamente a los profesores de taller, presencialmente en clases o a los correos y en los horarios que ellos establezcan.