

TALLER 01

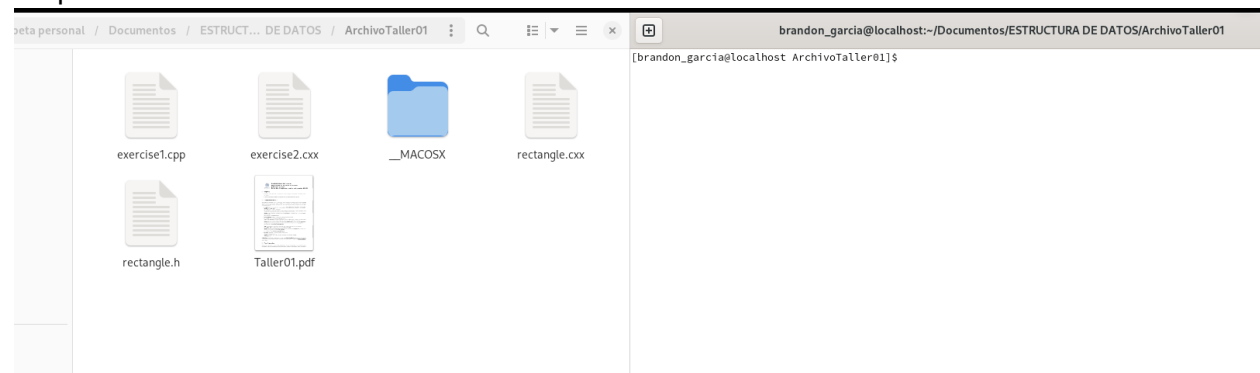
Nombre del estudiante: Diego Zabala y Brandon García

Curso: Estructura de datos

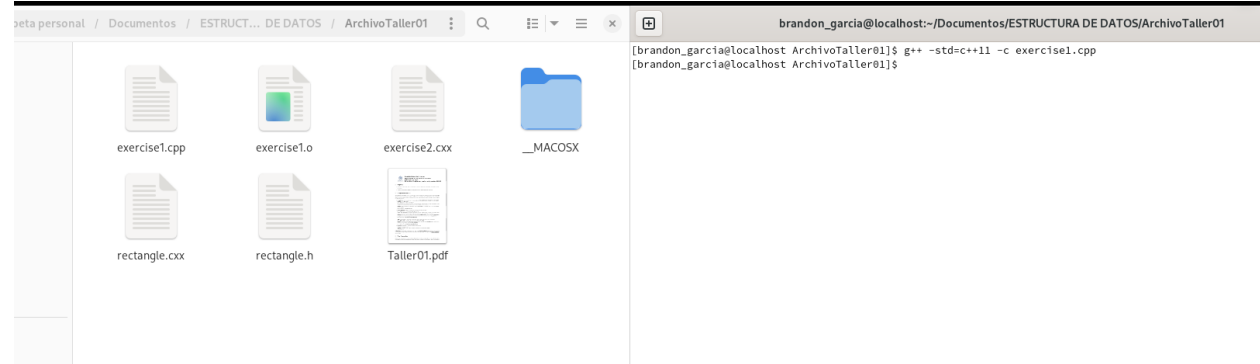
Fecha: 12 de febrero de 2025

Punto 1

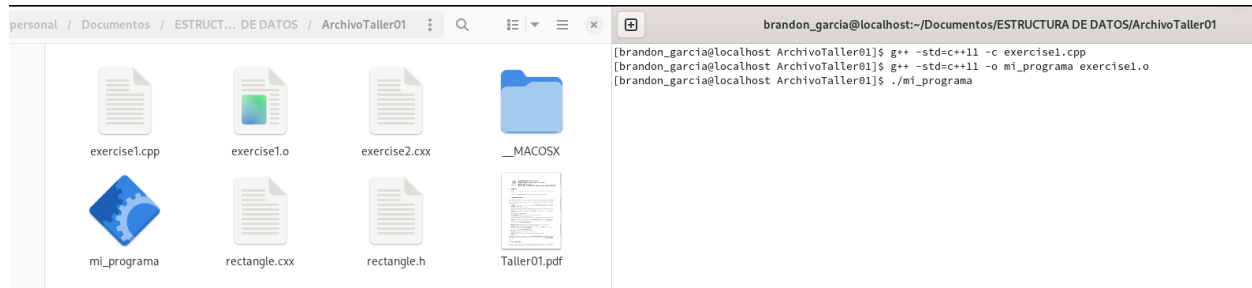
Para empezar, debemos descargar los archivos y guardarlos en una carpeta para que desde la terminal podamos ingresar a ellos para empezar a realizar los procesos respectivos



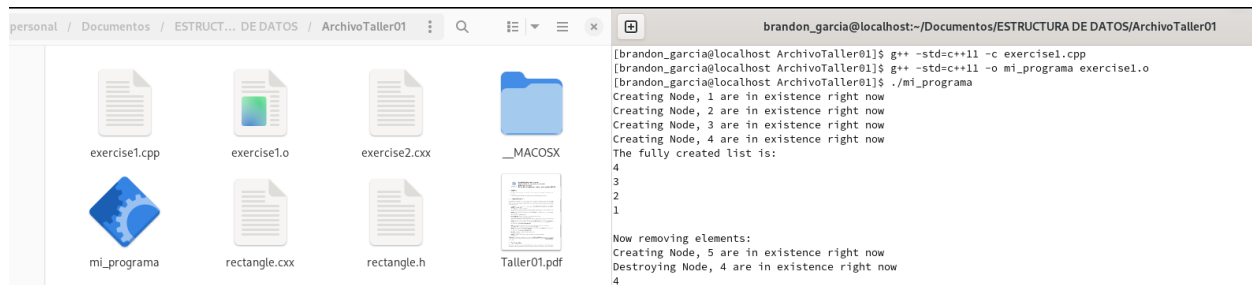
Despues de compilar



A hora debemos encadenarlo para que más adelante se pueda ejecutar el programa en este caso exercise1.o y ahora el ejecutable pasara a llamarse mi_programa



Ahora para ejecutar utilizamos el comando `./` y el nombre del ejecutable



Ahora ya tenemos el programa con su respectiva ejecución, dado que tenemos un fallo hemos tomado la decisión de arreglar el problema debido a un uso incorrecto de los punteros además de un error en cuanto a cómo borrar los nodos, ahora mostraremos la respectiva ejecución con el código con las correcciones (en la izquierda el código original y en la derecha el código corregido)

```

// returns 0 on success, -1 on failure
int remove(const T &item_to_remove) {
    Node<T> *marker = head_;
    Node<T> *temp = 0; // temp points to one behind as we iterate

    while (marker != 0) {
        if (marker->value() == item_to_remove) {
            if (temp == 0) { // marker is the first element in the list
                if (marker->next() == 0) {
                    head_ = 0;
                    delete marker; // marker is the only element in the list
                    marker = 0;
                } else {
                    head_ = new Node<T>(marker->value(), marker->next());
                    delete marker;
                    marker = 0;
                }
            }
            return 0;
        } else {
            temp->next(marker->next());
            delete temp;
            temp = 0;
            return 0;
        }
    }
    marker = 0; // reset the marker
    temp = marker;
    marker = marker->next();
}

return -1; // failure

```

```

// returns 0 on success, -1 on failure
int remove(const T &item_to_remove) {
    Node<T> *marker = head_;
    Node<T> *temp = nullptr; // temp mantiene la referencia al nodo anterior

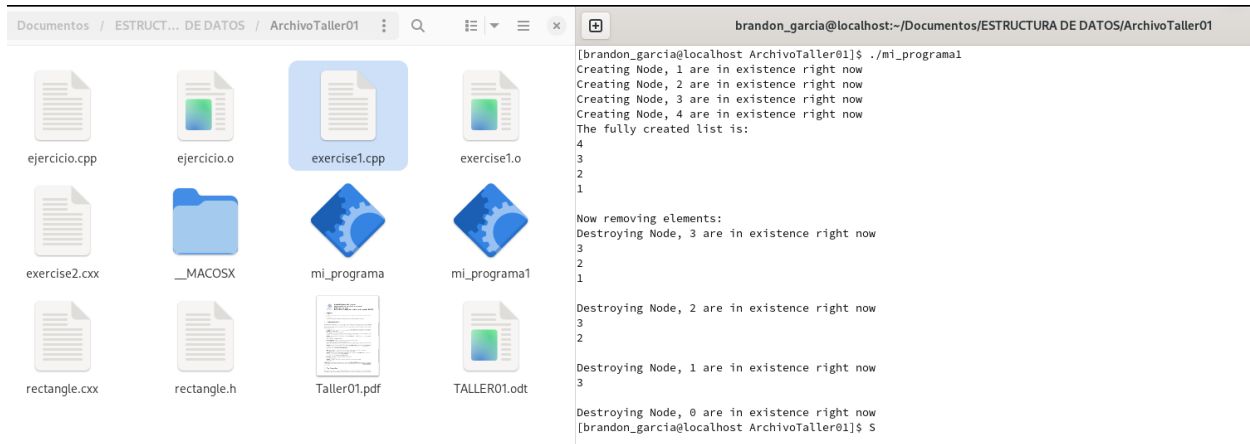
    while (marker != nullptr) {
        if (marker->value() == item_to_remove) {
            if (temp == nullptr) { // El nodo a eliminar es el primero
                head_ = marker->next();
            } else {
                temp->next(marker->next());
            }
            delete marker; // Eliminamos correctamente el nodo
            return 0;
        }
        temp = marker;
        marker = marker->next();
    }

    return -1; // Si no se encuentra el elemento, devolvemos -1
}

void print() {
    Node<T> *marker = head_;
    while (marker != nullptr) {
        cout << marker->value() << endl;
        marker = marker->next();
    }
}

private:
    ...

```



Punto 2: Compilación y Depuración

1. Se accedió al directorio donde se encuentra el código fuente utilizando el comando:

- **cd Descargas/ArchivoTaller01**

Para verificar la existencia de los archivos, se listaron los contenidos con:

- **ls**

Se confirmó que `exercise2.cxx` estaba presente.

2. Se compiló el programa usando:

- **g++ -std=c++11 -g -o exercise2 exercise2.cxx**
 - a. `-std=c++11`: Especifica la versión de C++ a utilizar.
 - b. `-g`: Permite la inclusión de Información de depuración.
 - c. `-o exercise2`: Define el nombre del archivo compilado.

3. Se inició el depurador con:

- **gdb exercise2**

Luego, dentro de `gdb`, se ejecutó el programa con:

- **run**

El programa solicitó la entrada de datos para definir un rectángulo (posición, ancho y alto). Calculó y mostró el perímetro, el área y la distancia al origen. La ejecución finalizó sin errores de sintaxis, pero con errores en la lógica del algoritmo.

```
user@user-System-Product-Name: ~/Descargas/ArchivoTaller01
user@user-System-Product-Name:~$ cd Descargas/ArchivoTaller01
user@user-System-Product-Name:~/Descargas/ArchivoTaller01$ ls
exercise1.cpp exercise2.cxx rectangle.cxx rectangle.h Taller01.pdf
user@user-System-Product-Name:~/Descargas/ArchivoTaller01$ g++ -std=c++11 -g -o
exercise2 exercise2.cxx
user@user-System-Product-Name:~/Descargas/ArchivoTaller01$ gdb exercise2
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from exercise2...
(gdb) run
```

Se ejecutó el programa exercise2 dentro de gdb para su depuración. Al iniciar, se habilitó la descarga de información de depuración (debuginfod). Luego, el programa solicitó la entrada de datos para definir un rectángulo (posición, ancho y alto). Calculó y mostró el perímetro, el área y la distancia al origen. Finalmente, el programa terminó su ejecución sin errores de sintaxis, pero con errores en la lógica del algoritmo

```
user@user-System-Product-Name: ~/Descargas/ArchivoTaller01
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from exercise2...
(gdb) run
Starting program: /home/user/Descargas/ArchivoTaller01/exercise2

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
Downloading separate debug info for system-supplied DSO at 0x7ffff7fc3000
[Depuración de hilo usando libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posición del rectángulo: 15
Ingrese coordenada Y de la posición del rectángulo: 10
Ingrese ancho del rectángulo: 6
Ingrese alto del rectángulo: 3

Perímetro del rectángulo: 15
Área del rectángulo: 9
Distancia del rectángulo al origen de coordenadas: 18.0278
[Inferior 1 (process 25274) exited normally]
(gdb)
```

Punto 3: Plan de Pruebas

3.A Pruebas de la Función Perímetro del Rectángulo

Plan de Pruebas: función perímetro del rectángulo			
Descripción caso	Valores de entrada	Resultado esperado	Resultado obtenido
1. Alto como el doble del ancho	Ancho=2 Alto=4	16	8
2. Alto igual a ancho	Ancho= 3 Alto=3	18	9
3. Un numero en cero	Ancho=5 Alto=0	0	10

Se realizó un depurado del programa y se encontraron errores en la lógica:

- Falta de paréntesis en las operaciones, lo que afectaba el orden de las sumas.
- Intento de cálculo del área y perímetro cuando uno de los valores era 0.

Se implementó una validación con do-while para mejorar la captura de datos y se corrigieron los cálculos.

```
~/2025-02-10-file-1$ nano rectangle.cxx
~/2025-02-10-file-1$ g++ -std=c++11 -g -o exercise2 exercise2.cxx
~/2025-02-10-file-1$ gdb exercise2
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from exercise2...
```

```

(gdb) break main
Breakpoint 1 at 0x55555555350: file exercise2.cxx, line 10.
(gdb) run
Starting program: /home/user/2025-02-10-file-1/exercise2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at exercise2.cxx:10
10      int main () {
(gdb) n
14          cout << "Ingreso coordenada X de la posicion del rectangulo: ";
(gdb) n
15          cin >> rect1.posX;
(gdb) n
Ingreso coordenada X de la posicion del rectangulo: 15
16          cout << "Ingreso coordenada Y de la posicion del rectangulo: ";
(gdb) n
17          cin >> rect1.posY;
(gdb) n
Ingreso coordenada Y de la posicion del rectangulo: 10
18          cout << "Ingreso ancho del rectangulo: ";
(gdb) n
19          cin >> rect1.width;
(gdb) n
Ingreso ancho del rectangulo: 10
20          cout << "Ingreso alto del rectangulo: ";
(gdb) n
21          cin >> rect1.height;
(gdb) n
Ingreso alto del rectangulo: 5
22          cout << "\nPerimetro del rectangulo: " << perimeterRect( rect1 ) << endl;
(gdb) s

perimeterRect (rect=...) at /home/user/2025-02-10-file-1/rectangle.cxx:6
6          float perim = 0.0;
(gdb) n
7          _      perim = 2.0 *rect.width + rect.height;

```

Plan de Pruebas: función perímetro del rectángulo			
Descripción caso	Valores de entrada	Resultado esperado	Resultado obtenido
1. Alto como el doble del ancho	Ancho=2 Alto=4	12	12
2. Alto igual a ancho	Ancho= 3 Alto=3	12	12
3. Un numero en cero	Ancho=5 Alto=0	0	0

```

Type "apropos word" to search for commands related to "word"...
Reading symbols from exercise2...
(gdb) run
Starting program: /home/user/Descargas/ArchivoTaller01/exercise2

This GDB supports auto-downloading debuginfo from the following URLs:
  <https://debuginfod.ubuntu.com>
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
Downloading separate debug info for system-supplied DSO at 0x7ffff7fc3000
[Depuración de hilo usando libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posicion del rectangulo: 5
Ingrese coordenada Y de la posicion del rectangulo: 3
Ingrese ancho del rectangulo: 2
Ingrese alto del rectangulo: 0
Ingrese alto del rectangulo: 4

Perimetro del rectangulo: 12
Area del rectangulo: 6
Distancia del rectangulo al origen de coordenadas: 5.83095
[Inferior 1 (process 43792) exited normally]
(gdb)

```

3.B Pruebas de la Función Área del Rectángulo

Plan de Pruebas: función área del rectángulo			
Descripción caso	Valores de entrada	Resultado esperado	Resultado obtenido
1. Alto como el doble del ancho	Ancho=2 Alto=4	8	6
2. Alto igual a ancho	Ancho= 3 Alto=3	9	6
3. Un numero en cero	Ancho=5 Alto=0	0	5

Se encontró un error en el algoritmo del cálculo del área, ya que se realizaba una suma en lugar de una multiplicación. Luego de la corrección, los valores obtenidos fueron correctos.

```

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from exercise2...
(gdb) break main
Breakpoint 1 at 0x134f: file exercise2.cxx, line 10.
(gdb) n
The program is not being run.
(gdb) run
Starting program: /home/user/2025-02-10-file-1/exercise2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at exercise2.cxx:10
10      int main () {
(gdb) n
14          cout << "Ingrese coordenada X de la posicion del rectangulo: ";
(gdb) n
15          cin >> rect1.posX;
(gdb) n
Ingrese coordenada X de la posicion del rectangulo: 4
16          cout << "Ingrese coordenada Y de la posicion del rectangulo: ";
(gdb) n
17          cin >> rect1.posY;
(gdb) n
Ingrese coordenada Y de la posicion del rectangulo: 3
18          cout << "Ingrese ancho del rectangulo: ";
(gdb) n
19          cin >> rect1.width;
(gdb) n
Ingrese ancho del rectangulo: 2
20          cout << "Ingrese alto del rectangulo: ";
(gdb) n
21          cin >> rect1.height;
(gdb) n
Ingrese alto del rectangulo: 4
22          cout << "\nPerimetro del rectangulo: " << perimeterRect( rect1 ) << endl;
(gdb) n

Perimetro del rectangulo: 8
23          cout << "Area del rectangulo: " << areaRect( rect1 ) << endl;
(gdb) s
areaRect (rect=...) at /home/user/2025-02-10-file-1/rectangle.cxx:13
13      float area = 0.0;
(gdb) n
14      area = rect.width + rect.height;
(gdb) █

```

Se corrige el problema y la nueva tabla de casos es

Plan de Pruebas: función área del rectángulo			
Descripción caso	Valores de entrada	Resultado esperado	Resultado obtenido
1. Alto como el doble del ancho	Ancho=2 Alto=4	8	8
2. Alto igual a ancho	Ancho= 3 Alto=3	9	9
3. Un numero en cero	Ancho=5 Alto=0	0	0

3.C Pruebas de la Función Distancia del Rectángulo al Origen

Plan de Pruebas: función distancia del rectángulo al origen			
Descripción caso	Valores de entrada	Resultado esperado	Resultado obtenido
1. Números positivos	X=15, y=32	35.34	35.34
2. Un numero en cero	X=0, y =32	32	32
3. Un numero en cero	X=15, y=15	21.21	21.21

Se verificó que la función de cálculo de distancia operaba correctamente en todos los casos de prueba.

```
Starting program: /home/user/2025-02-10-file-1/exercise2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posicion del rectangulo: 15
Ingrese coordenada Y de la posicion del rectangulo: 32
Ingrese ancho del rectangulo: 2
Ingrese alto del rectangulo: 4

Perimetro del rectangulo: 12
Area del rectangulo: 8
Distancia del rectangulo al origen de coordenadas: 35.3412
[Inferior 1 (process 1071) exited normally]
(gdb) run
Starting program: /home/user/2025-02-10-file-1/exercise2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posicion del rectangulo: 0
Ingrese coordenada Y de la posicion del rectangulo: 32
Ingrese ancho del rectangulo: 3
Ingrese alto del rectangulo: 3

Perimetro del rectangulo: 12
Area del rectangulo: 9
Distancia del rectangulo al origen de coordenadas: 32
[Inferior 1 (process 1072) exited normally]
(gdb) run
Starting program: /home/user/2025-02-10-file-1/exercise2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Ingrese coordenada X de la posicion del rectangulo: 15
Ingrese coordenada Y de la posicion del rectangulo: 15
Ingrese ancho del rectangulo: 5
Ingrese alto del rectangulo: 0

Perimetro del rectangulo: 10
Area del rectangulo: 0
Distancia del rectangulo al origen de coordenadas: 21.2132
[Inferior 1 (process 1120) exited normally]
(gdb) █
```

Conclusión

Se logró compilar, ejecutar y depurar el programa con éxito. Se detectaron y corrigieron errores en la lógica de los cálculos, asegurando resultados correctos en las funciones de perímetro, área y distancia al origen del rectángulo.