# Taller 01 Plan de pruebas

Juan Sebastián Rodríguez Pabon

David Rodríguez Jurado

Julián Pérez Gomes

Daniel Galviz

Dirigido a: John Corredor

28 de agosto del 2024

Bogotá D.C

Estructura de datos

## Introducción

La compilación en g++ es esencial en la programación debido a que permite observar mediante consola todas las operaciones dentro del programa, permite documentar de mejormanera el programa en C++ debido a que está diseñado especialmente para ejecutar sus respectivos programas. En la elaboración del taller se realizará una detallada documentación acerca de la compilación en g++ de un código enfocado en Nodos, cada uno cumpliendo una funciónesencial en el programa.

En la compilación se compilarán archivos con extensiones *.c, *.cpp, *.cxx debido al formatode estos mismos, es decir que se puede realizar su compilación en g++ mediante una serie de comandos que se presentaran a continuación.

## Ejecución

Se realizará la ejecución por partes, a medida de la implementación de comandos, finalmente se mostrará el resultado final.

En la ejecución del programa, se usa el comando "ls" este comando permite visualizar archivos y directorios en el directorio base actual.

En la imagen se encuentra "ArchivoTaller01.zip" debido a que en el directorio base actual es una máquina virtual la cual aún no se han creado más archivos y "ArchivoTaller01.zip" es el único archivo creado hasta el momento. Esta función permite visualizar los archivos guardados en nuestro programa.

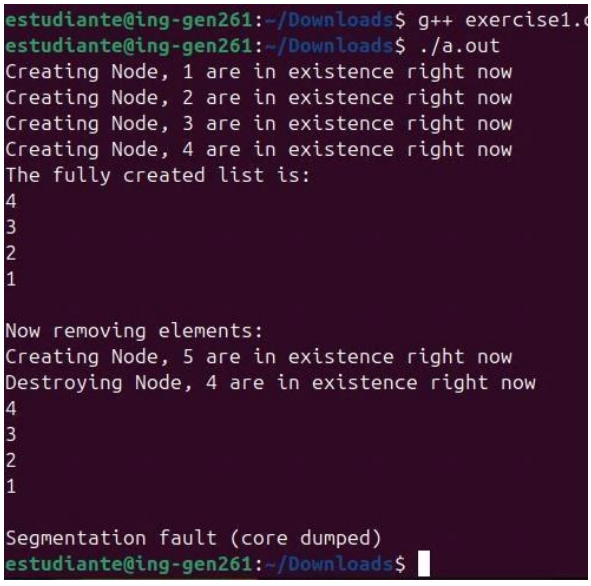
```
estudiante@ing-gen261:~/Downloads$ ls
ArchivoTaller01.zip  exercise1.cpp  Taller01
```

Estructura de datos

Una vez se haya guardado el archivo .cpp se usa el comando g++ seguido del nombre del programa anteriormente guardado. La función g++ se implementa con su objetivo de compilar el archivo guardado, compilar se refiere a ejecutar y verificar su correcto funcionamiento, en caso de contener errores se deberán de corregir y repetir el proceso.

Otra función ./a.out también se implementa para compilar el archivo guardado, si el archivo no contiene errores se podrá ejecutar correctamente.

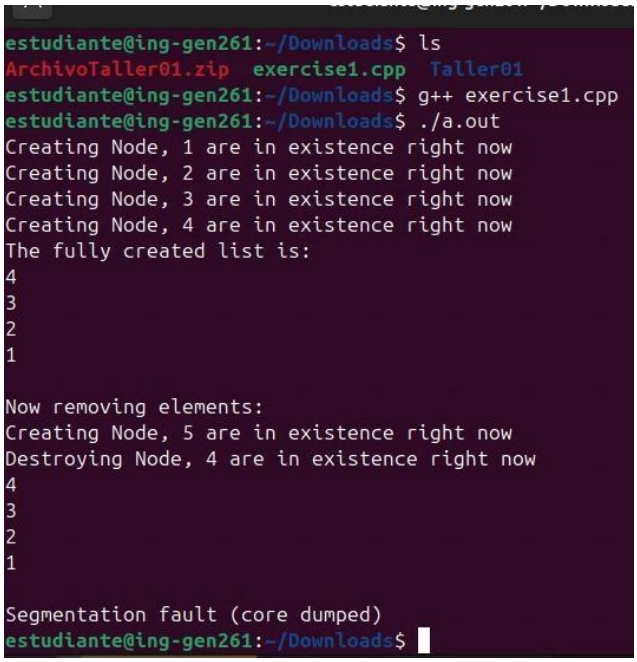
En la siguiente imagen se evidencia la compilación mediante el comando g++ explicado anteriormente. En la compilación se evidencia que en su ejecución se crearon listas enlazadas, añadiendo y creando Nodes uno por uno: Node 1. Node 2, Node 3 y Node 4. Después de la aquella creación se eliminan Nodes, eliminando el Node 4, este se elimina con el propósito de seguir con el propósito de una pila, ya que en una pila el ultimo nodo añadido es el primero en ser eliminado




Estructura de datos

Se evidencia que el archivo se ejecutó correctamente y hace referencia a 4 nodos loscuales se conforma por una lista enlazada genérica, con la capacidad de insertar y eliminar nodos.

Al juntar los comandos mencionados anteriormente, el programa tendrá la siguiente estructura:




Estructura de datos

A continuación, se realizará la ejecución del ejercicio 2 ejecutando el archivo llamado rectangle.h. Para su correcta compilación se realizará con exactitud el proceso mencionado anteriormente.

En la ejecución del programa se usa el comando ls este comando nos permite visualizar archivos y directorios en el directorio base actual.

En la imagen se encuentra el archivo “ArchivoTaller01.zip” en la carpeta llamada “exercise2.cxx” y el programa “rectangle.h” el cual es programa para ejecutar.


En la siguiente imagen se ejecuta el archivo “exercise2.cxx” por medio del comando g++ seguido del nombre del programa anteriormente guardado. La función g++ se implementa con su objetivo de compilar el archivo guardado, compilar se refiere a ejecutar y verificar su correcto funcionamiento, en caso de contener errores se deberán de corregir y repetir el proceso.

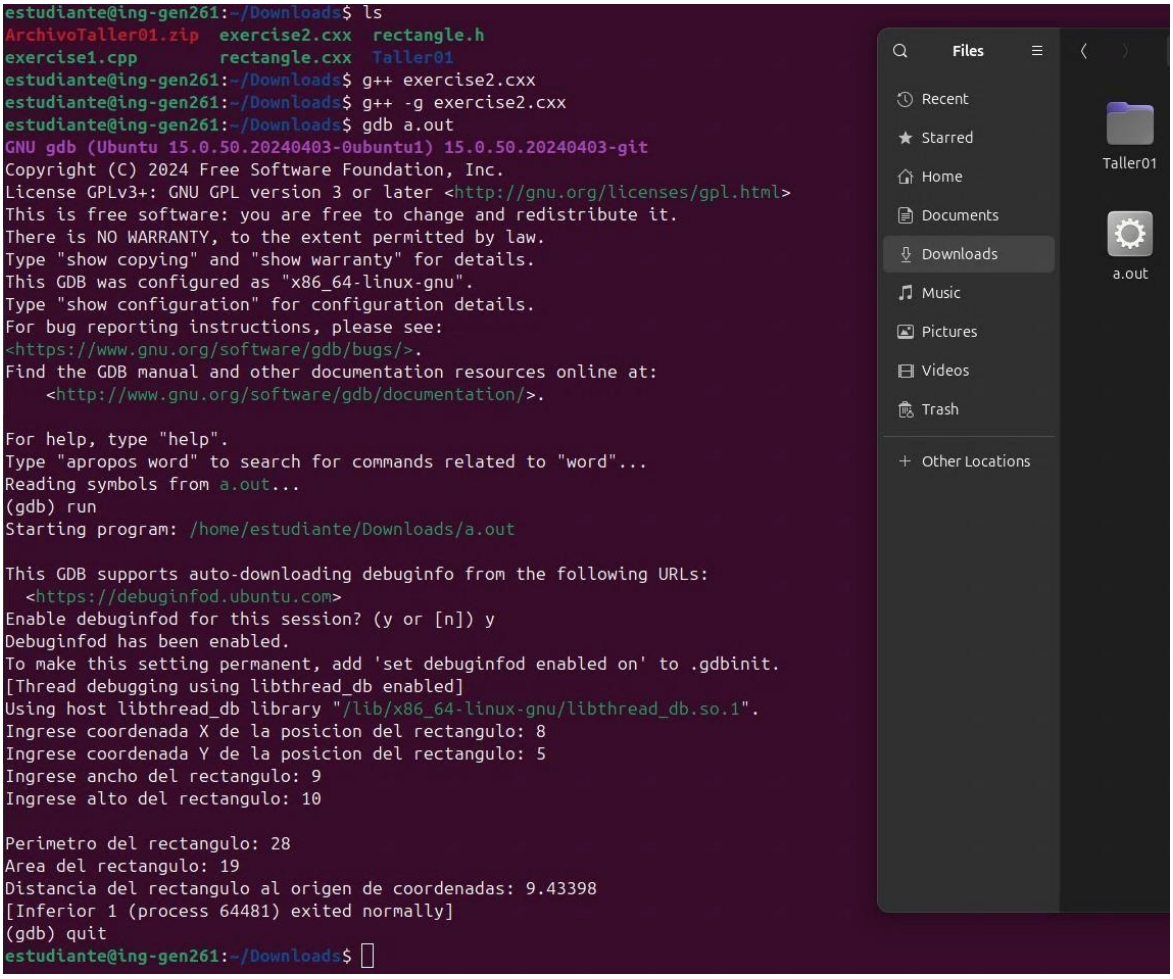
El comando -g se utiliza para incluir información de depuración en el archivo ejecutable

Generado, depurar hace referencia a diagnosticar fu funcionalidad general y corregir errores.

Se usa el comando gdb y a.out para depurar y después ejecutar el programa.

Al ejecutar el programa se muestran propiedades del software y sus respectivas configuraciones internas implementadas para poder ejecutar el código y después lo importante es que se puede evidenciar propiedades de un rectángulo, es decir las entradas del código; coordenadas (X,Y) Ancho y altura. Gracias a estos datos se pueden realizar las operaciones y funciones como lo son: Perímetro del rectángulo, Área del rectángulo, distancia del rectángulo al origen de coordenadas. El programa no detecta errores y con todos sus comandos mencionados anteriormente quedaría así:

Estructura de datos




Estructura de datos

**Plan de pruebas**

De acuerdo con el programa se buscar realizar una prueba para cada función, la cual sebusca un resultado esperado. Se realizarán las pruebas para: Perímetro del rectángulo, área del rectángulo y distancia del rectángulo al origen.

| Plan de pruebas: función Perímetro del rectángulo | | | |
|---|---|---|---|
| **Descripción de caso** | **Valores de entrada** | **Resultado esperado** | **Resultado obtenido** |
| 1: Alto como el doble de 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 | 10 | 10 |

| Plan de pruebas: función Área del rectángulo | | | |
|---|---|---|---|
| **Descripción de caso** | **Valores de entrada** | **Resultado esperado** | **Resultado obtenido** |
| 1: Alto como el doble de 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 |

| Plan de pruebas: función Distancia del rectángulo al origen | | | |
|---|---|---|---|
| **Descripción de caso** | **Valores de entrada** | **Resultado esperado** | **Resultado obtenido** |
| 1: números positivos | x = 15, y = 32 | 35.34 | 35.34 |
| 2: un número 0 | x = 0, y = 32 | 32 | 32 |
| 3: números iguales | x = 15, x = 15 | 21.21 | 21.21 |

Estructura de datos

Realizando las operaciones, se comprueba que los resultados esperados son los resultados obtenidos, sin embargo, hay discrepancia porque se debería de cambiar lainicialización de variable int debido a que este solo recibe valores enteros, no recibe valores flotantes, si se realizara este cambio, los resultados no generarían problema.

## Conclusión

La correcta ejecución del programa utilizando el compilador g++, se pudo comprobar la funcionalidad de un sistema compuesto por dos archivos de código, los cuales implementan diversas funciones orientadas a estructuras lineales. Estas funciones demostraron un manejo eficiente de operaciones, como el cálculo del perímetro, el área y ladistancia de un rectángulo al origen, utilizando las estructuras lineales definidas en el código. Este proceso validó el correcto funcionamiento de las funciones implementadas en un entorno C++.