

Laura Catalina Riveros Florez – 67001165

Julian Andres Cuadros Muñoz - 67001239

Rodolfo Angel Gilart Pulido – 67001245

Jhonatan Camilo Garrido Gomez – 67000762

Actividad 1 teorico-practica.

TEORICO.

1. ¿Que es un archivo .cpp y para que sirve?
2. ¿Que es un archivo .h y para que sirve?
3. ¿Qué es la separación por capas (lógica, datos, interfaz) y para qué sirve?
4. ¿Qué es la compilación g++ y para qué sirve?

PRACTICO.

1. Crear un proyecto con al menos 2 clases en archivos separados (.h + .cpp)
2. Compilar con línea de comandos usando g++.

SOLUCIÓN.

Parte Teórica.

1. Archivo .cpp

Un archivo .cpp es un tipo de archivo generado por el lenguaje de programación C++ donde se almacena el código fuente. Este archivo contiene las líneas de código con la lógica para llevar a cabo el funcionamiento del programa desarrollado.

Estos archivos trabajan de la mano con un compilador como g++ el cual convierte el código escrito por los desarrolladores y lo traduce a código máquina, donde este es procesado por la unidad de procesamiento central (CPU) de la computadora con el fin de entender y ejecutar la lógica de programación.

Las ventajas de usar archivos .cpp en C++ son las siguientes:

- Alto rendimiento para tareas que requieren velocidad y control sobre los recursos de la máquina.
- Control total de la memoria haciendo uso de punteros y la asignación dinámica.
- Altamente orientado a la programación orientada a objetos para modelar las necesidades del cliente o el mundo por medio de clases y objetos.

- Debido a la portabilidad que estos archivos poseen pueden compilarse en sistemas operativos como Windows, Linux y MacOS.
- Amplia variedad de bibliotecas disponibles para C++ con enfoque en el desarrollo de redes, matemáticas y inteligencia artificial.

2. Archivo .h:

Un archivo .h (o archivo de encabezado), es un componente fundamental en la programación en c++. Su función principal es almacenar declaraciones de funciones, variables, constantes, macros y estructuras de datos que pueden ser compartidas entre múltiples archivos de código fuente. Esto permite modularizar el código, facilitando su mantenimiento, reutilización.

Algunas funciones clave de los archivos .h son:

- Contienen prototipos de funciones, clases, estructuras y variables (extern), que son accesibles desde cualquier archivo que los incluya.
- Incluyen otros encabezados, para garantizar que las dependencias necesarias estén disponibles en los archivos de implementación.
- Usan protectores de inclusión para evitar duplicaciones en la compilación, que generarían errores de redefinición.
- Almacenan macros y plantillas de clases/funciones, que el preprocesador expande antes de la compilación.

Importancia en C++:

Los encabezados definen *que hace* el código, mientras que los .cpp definen el *cómo lo hace*. Esto agiliza la compilación y reduce los errores.

Facilita la creación de bibliotecas.

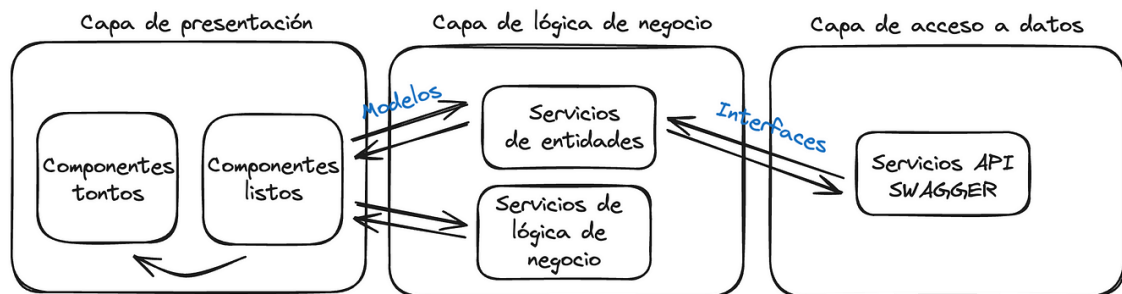
En proyectos grandes, agrupan funcionalidades relacionadas.

3. ¿Qué es la separación por capas (lógica, datos, interfaz) y para qué sirve?

La separación por capas es un patrón de diseño que permite a los desarrolladores establecer un modelo arquitectónico a la hora de crear aplicaciones demasiado complejas. Su objetivo principal es dividir una aplicación en secciones lógicas con distintas funcionalidades, cada una con una responsabilidad bien definida. Las capas más comunes son la Capa de Presentación (interfaz del usuario), la Capa Lógica y la capa de Acceso a Datos.

Dentro de las principales razones para desarrollar un sistema por medio de capas es que permite una mayor separación de responsabilidades entre el grupo de trabajo. Esto reduce la complejidad del código, ya que cada miembro tiene un único enfoque establecido. Asimismo, si dentro del proceso de desarrollo surge errores, estos pueden ser

identificados rápidamente debido a la separación por capas sin tener afectaciones la una con la otra. Esto permite una mayor depuración del código y reduce el riesgo de generar nuevos errores.



- **Capa de Presentación o Interfaz**

Esta capa es la que permite la interacción del usuario final con el sistema debido a que es la parte visible de toda la aplicación. Su función principal es mostrar la información al usuario y capturar el ingreso de los datos mediante clics en botones, menús o textos de campos.

- **Capa de Lógica de Negocio (También conocida de dominio o reglas de negocio)**

Esta capa contiene las reglas y procesos que permiten definir el comportamiento central de la aplicación. Allí, se procesan los datos, se realizan los cálculos y se generan las validaciones para tomar las respectivas necesidades según las necesidades del negocio.

- **Capa de Acceso a Datos**

Esta capa permite interactuar con el sistema para el almacenamiento de los datos de la aplicación. Esto se puede ver evidencia cuando se suele usar bases de datos como SQL server, MySQL, PostgreSQL, entre otras. De igual manera esta capa permite guardar, recuperar, actualizar y eliminar datos de forma continua.

4. ¿Qué es la compilación g++ y para qué sirve?

g++ es el compilador oficial de GNU para C++ cuya función es traducir el código fuente escrito en C++ como archivos .cpp o .h a programas ejecutables, manejando todo el proceso de compilación: preprocesamiento, compilación, ensamblado y enlazado.

El compilador tiene múltiples pasos:

- **Reprocesador:** Resuelve directivas `#include`, `#define` (macros), y `#ifdef` (condicionales).
- **Compilación:** Convierte el código preprocesado en assembly.
- **Ensamblado:** Traduce assembly a código objeto binario.

- Enlazado: Combina múltiples (. o) y bibliotecas en un ejecutable final.

Parte Práctica:

Código ejecutado con g++ en terminal de windows

```
Microsoft Windows [Versión 10.0.26100.4351]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\rgila>Maraton C++
"Maraton" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Users\rgila>cd Maraton C++

C:\Users\rgila\Maraton C++>g++ -o maraton.exe src/Equipo.cpp src/Problema.cpp src/main.cpp

C:\Users\rgila\Maraton C++>maraton.exe

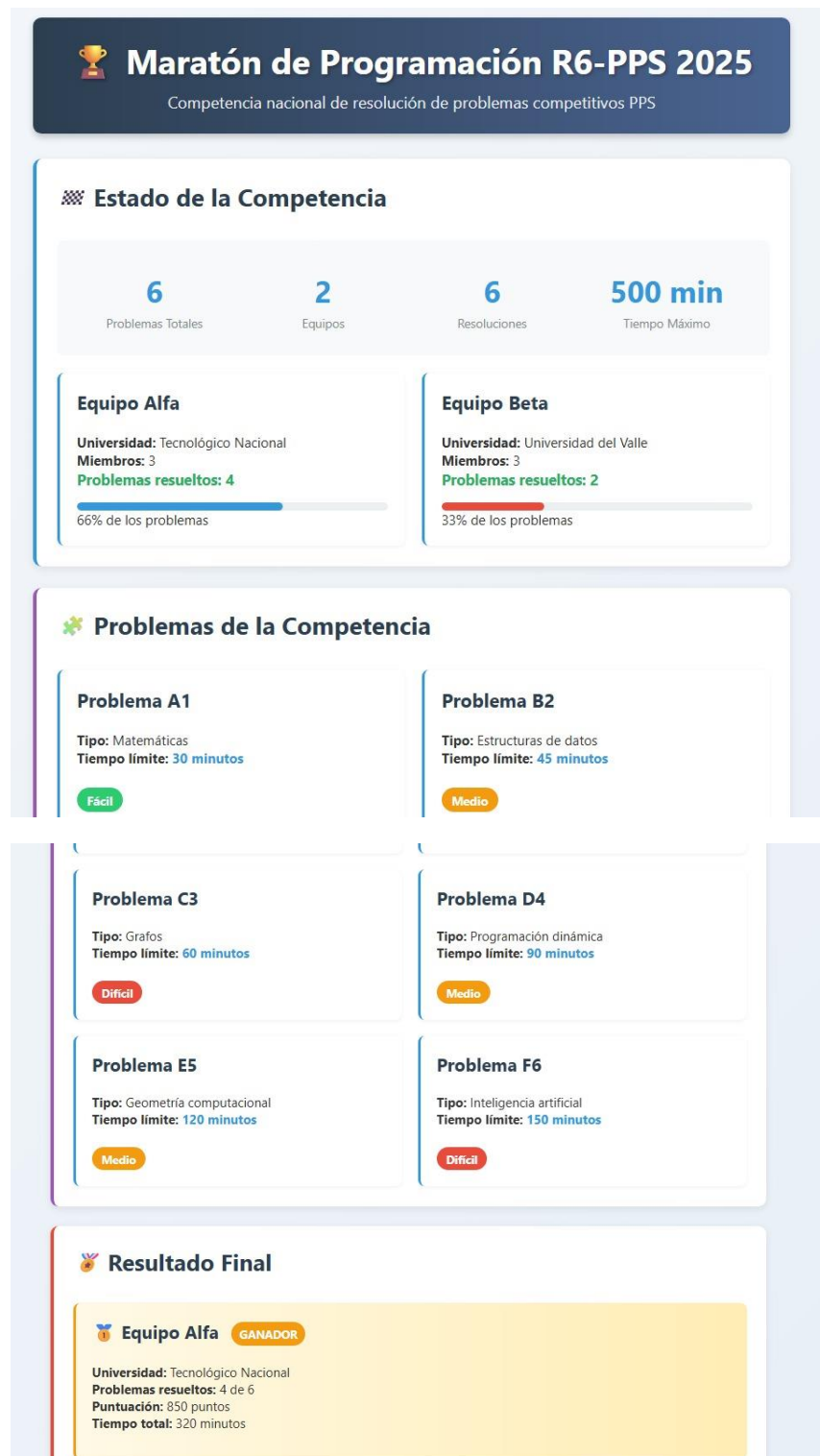
=== ESTADO DE LA MARATON ===
Equipo: Alfa | Universidad: Tecnológico Nacional | Problemas resueltos: 4
Equipo: Beta | Universidad: Universidad del Valle | Problemas resueltos: 2

=== PROBLEMAS DISPONIBLES ===
Problema A1 | Dificultad: Fácil | Tiempo: 30 min
Problema B2 | Dificultad: Medio | Tiempo: 45 min
Problema C3 | Dificultad: Difícil | Tiempo: 90 min
Problema D4 | Dificultad: Medio | Tiempo: 35 min
Problema E5 | Dificultad: Medio | Tiempo: 40 min
Problema F6 | Dificultad: Difícil | Tiempo: 150 min

=== RESULTADO FINAL ===
Ganador: Alfa!

C:\Users\rgila\Maraton C++>|
```

Interfáz web:





BIBLIOGRAFIA

1. FileFormat. (2024). *H - Formato de archivo de encabezado C/C++. Recuperado de <https://docs.fileformat.com/es/programming/h/>
2. Microsoft Learn. (2024). *Archivos de encabezado (C++). Recuperado de <https://learn.microsoft.com/es-es/cpp/cpp/header-files-cpp?view=msvc-170>
3. GeeksforGeeks. (2024). *Header Files in C/C++ and its uses. Recuperado de <https://www.geeksforgeeks.org/header-files-in-c-cpp-and-its-uses/>
4. **Microsoft Learn.** (2024). *Arquitectura de aplicaciones en capas*. Microsoft. Recuperado de <https://learn.microsoft.com/es-es/dotnet/architecture/modern-web-apps-azure/layered-application-architecture>
5. Free Software Foundation. (2024). GCC, the GNU Compiler Collection. Recuperado de <https://gcc.gnu.org/>
6. GNU Project. (2023). G++ Command Options (Using the GNU Compiler Collection). Recuperado de <https://gcc.gnu.org/onlinedocs/gcc-13.2.0/gcc/>
7. **Techopedia.** (s.f.). *What is a C++ File?* Recuperado de <https://www.techopedia.com/definition/32688/c-plus-plus-file>
8. Stroustrup, B. (2013). *The C++ Programming Language* (4th ed.). Addison Wesley. (Capítulo 2: "Compilation and Execution").