## Dominando Matrices en C++: De la Teoría a la Práctica

### Objetivo Didáctico

Al finalizar esta guía, el estudiante será capaz de:

* Comprender el concepto de matrices y su importancia en la programación.
* Identificar problemas del mundo real que pueden ser resueltos eficientemente con matrices.
* Implementar soluciones en C++ utilizando matrices, incluyendo la declaración, inicialización y manipulación de datos.
* Utilizar bucles para recorrer y procesar elementos de una matriz.
* Desarrollar habilidades de pensamiento lógico y resolución de problemas aplicando matrices a diversos escenarios.

### Competencias a Desarrollar

* **Pensamiento Computacional:** Descomponer problemas complejos en pasos más pequeños y manejables.
* **Abstracción:** Identificar patrones y generalizar soluciones.
* **Modelado de Datos:** Representar información del mundo real utilizando estructuras de datos apropiadas (matrices).
* **Programación en C++:** Dominar la sintaxis y semántica de C++ para trabajar con matrices.
* **Lógica de Programación:** Diseñar algoritmos eficientes utilizando bucles y estructuras de control.
* **Resolución de Problemas:** Aplicar los conocimientos adquiridos para solucionar problemas prácticos.

### Introducción a las Matrices

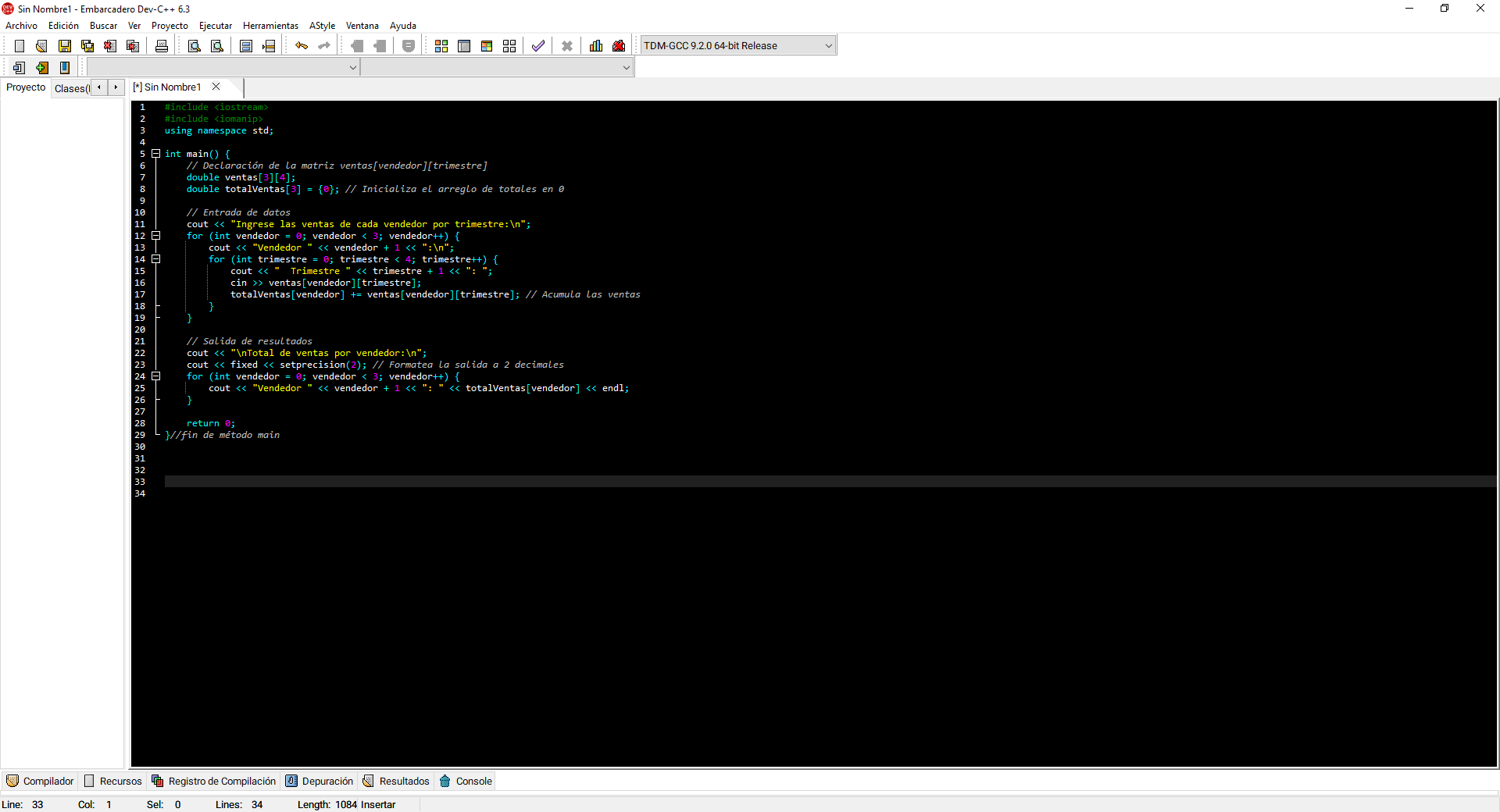
Una matriz es una estructura de datos que permite almacenar una colección de elementos del mismo tipo, organizados en filas y columnas. Las matrices son fundamentales en programación para representar tablas de datos, imágenes, gráficos y muchas otras estructuras. En C++, las matrices se implementan utilizando arreglos multidimensionales.

### Ejemplos Prácticos

#### Ejemplo 1: Suma de Ventas por Vendedor

**Descripción:** Una empresa tiene 3 vendedores y registra las ventas de cada vendedor durante 4 trimestres del año. Se desea calcular el total de ventas por vendedor.

**Código en C++:**



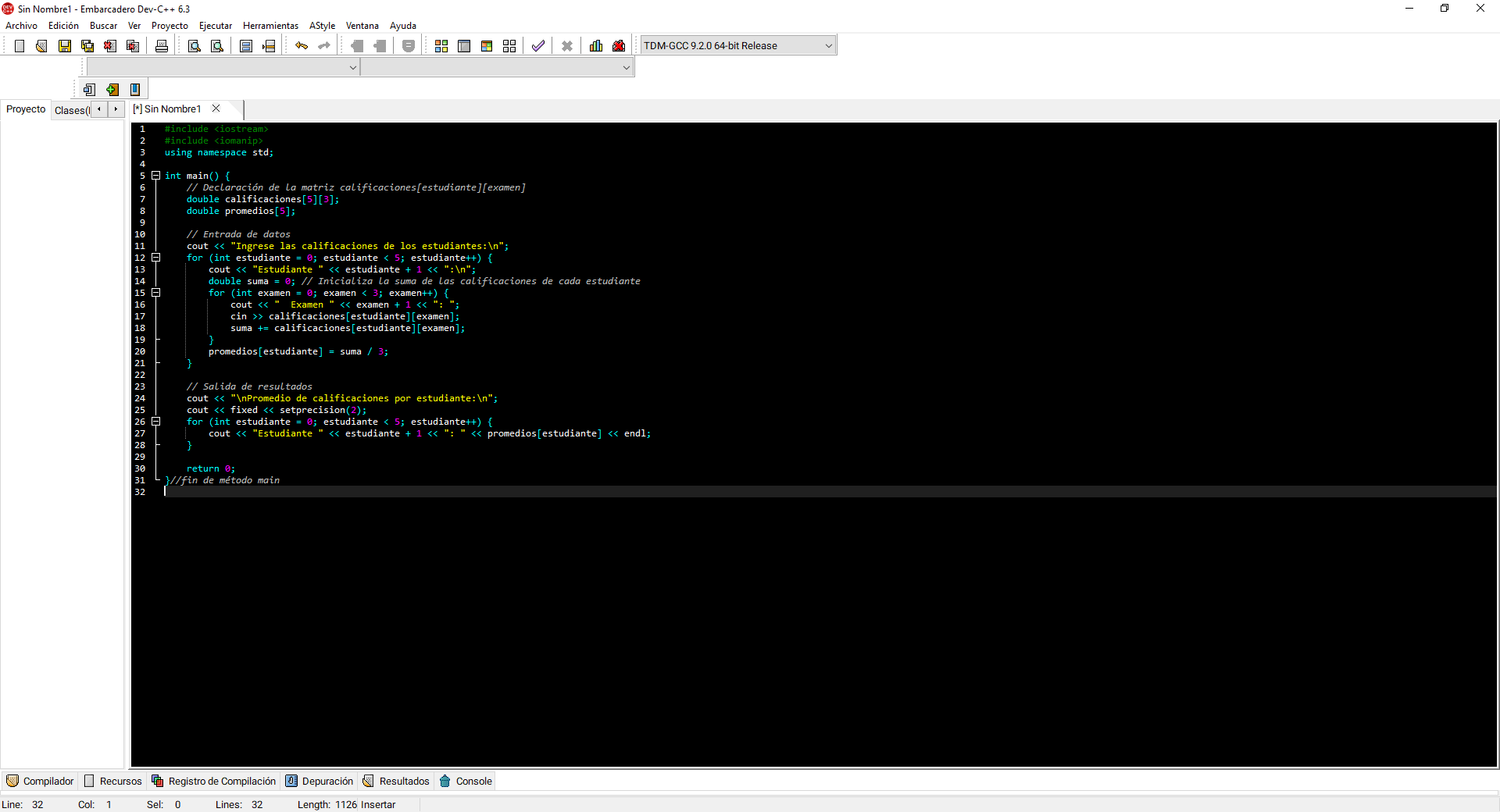
**Explicación:**

* Se declara una matriz ventas de 3x4 para almacenar las ventas de cada vendedor en cada trimestre.
* Se declara un arreglo totalVentas para almacenar la suma de las ventas de cada vendedor.
* Se utilizan dos bucles for anidados para recorrer la matriz ventas, pedir al usuario que ingrese las ventas de cada vendedor por trimestre, y acumular las ventas en el arreglo totalVentas.
* Se utiliza un bucle for para mostrar el total de ventas de cada vendedor.

#### Ejemplo 2: Promedio de Calificaciones por Estudiante

**Descripción:** Un profesor tiene las calificaciones de 5 estudiantes en 3 exámenes. Se desea calcular el promedio de cada estudiante.

**Código en C++:**



**Explicación:**

* Se declara una matriz calificaciones de 5x3 para almacenar las calificaciones de los estudiantes en los exámenes.
* Se declara un arreglo promedio para almacenar el promedio de cada estudiante.
* Se utilizan dos bucles for anidados para recorrer la matriz calificaciones, pedir al usuario que ingrese las calificaciones y calcular el promedio de cada estudiante.
* Se utiliza un bucle for para mostrar el promedio de cada estudiante.

### Ejercicios Propuestos

1. **Suma de Filas y Columnas:**
   * Crea un programa que declare una matriz de 4x4.
   * Pide al usuario que ingrese los elementos de la matriz.
   * Calcula e imprime la suma de los elementos de cada fila y de cada columna.
2. **Búsqueda de un Elemento:**
   * Crea un programa que declare una matriz de 5x5.
   * Pide al usuario que ingrese los elementos de la matriz.
   * Pide al usuario que ingrese un número a buscar.
   * Busca el número en la matriz e imprime su posición (fila y columna) si lo encuentra. Si no lo encuentra, imprime un mensaje indicando que no está en la matriz.
3. **Transpuesta de una Matriz:**
   * Crea un programa que declare una matriz de 3x2.
   * Pide al usuario que ingrese los elementos de la matriz.
   * Calcula la matriz transpuesta (intercambia filas por columnas) y la imprime.
4. **Multiplicación de Matrices:**
   * Crea un programa que declare dos matrices de 2x2.
   * Pide al usuario que ingrese los elementos de las matrices.
   * Realiza la multiplicación de las dos matrices y muestra el resultado.
5. **Juego del Gato (Tic-Tac-Toe):**
   * Crea un programa que implemente el juego del gato para dos jugadores.
   * Utiliza una matriz de 3x3 para representar el tablero de juego.
   * Permite a los jugadores ingresar sus movimientos (fila y columna).
   * Verifica si hay un ganador o si el tablero está lleno (empate).