Aspecto	Código 1	Código 2
Positivo	- Utiliza vectores, lo que facilita el manejo dinámico de memoria y simplifica la entrada/salida de datos.	- Utiliza arreglos de tamaño fijo, lo que puede ser más eficiente en términos de memoria en casos simples.
	- Muestra un mensaje de error si las dimensiones de las matrices no son compatibles para la multiplicación.	- Maneja correctamente la entrada de datos y muestra mensajes claros al usuario.
	- Uso de bucles anidados para calcular la multiplicación de matrices de manera eficiente.	- También utiliza bucles anidados para realizar la multiplicación de matrices.
Negativo	- No verifica límites de las dimensiones de las matrices (tamaño máximo).	- Utiliza arreglos de tamaño fijo, lo que puede ser limitante en aplicaciones donde las dimensiones de las matrices son desconocidas o cambian dinámicamente.
	- No incluye comentarios para explicar el propósito de cada sección del código.	- No utiliza la biblioteca de vectores de C++, lo que puede ser menos flexible en comparación con el uso de vectores.
Interesante	- Utiliza vectores bidimensionales para almacenar las matrices, lo que simplifica la gestión de memoria y la manipulación de matrices.	- Muestra mensajes claros y guía al usuario a través del proceso de entrada de datos.

	- Utiliza un enfoque más moderno y flexible al trabajar con matrices mediante la biblioteca de vectores de C++.	- Utiliza arreglos bidimensionales para almacenar las matrices y calcular la multiplicación de manera eficiente.