

**Nombre:** Luis Fernando González Chávez      **No. de Matrícula.:** ZAP408

**Materia:** Fundamentos de la Programación Grupo: Dev 22-1 Turno: Matutino

**Carrera:** Ingeniería en Desarrollo de Software Interactivo y Videojuegos

**Tema:** Memoria Estática y Dinámica **No:** R.1 16

**Fecha propuesta:** 02/Nov/2021      **Fecha de Entrega:** 28/Oct/2021

**Escuela:** Instituto Universitario Amerike

**Plantel** Guadalajara

**Calle:** C. Montemorelos

**No:** 3503

**Colonia:** Rinconada de la Calma

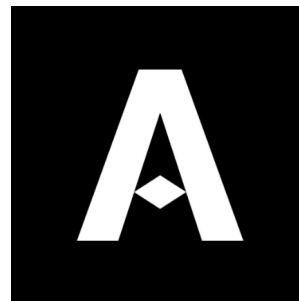
**C.P.:** 45080

**Teléfono:** 3336326100

**Ciudad:** Zapopan



Ferchus



## Firma del alumno (a)

## Firma de revisión fecha

Qué se evalúa:	10 pts.	7 pts.	4 pts.	
Entrega electrónica	Es en tiempo y forma al iniciar la clase. (1 pts.)	Después de 30 minutos de iniciada la clase. (.7 pts.)	Al minuto 40. (Posteriormente ya no se reciben). (.4pts.)	
Del formato.	Cumple con todos los elementos solicitados. (1 pts.)	No cumple con dos elementos solicitados. (.7 pts.)	No cumple con tres o más elementos solicitados. (.4pts.)	
La ortografía.	Tiene dos errores ortográficos. (1 pts.)	Tiene de tres a cuatro errores ortográficos. (.7 pts.)	Tiene cinco o más errores ortográficos. (.4pts.)	
Del tema y objetivo.	La teoría y ejemplos corresponden al tema tratado. (1 pts.)	La teoría o ejemplos no corresponden al tema tratado. (.7 pts.)	La teoría y ejemplos no corresponden al tema tratado. (.4pts.)	
El programa y los cálculos.	Los parámetros y componentes corresponden al 100% de lo planeado. (1 pts.)	El programa arroja un error o componente no corresponden al 100% de lo planeado. (7 pts.)	El programa arroja dos errores o componentes no corresponden al 100% de lo calculado. (.4pts.)	
Diagramas.	Los diagramas a bloques, de flujo y esquemáticos son acorde al de la práctica y siguen una secuencia lógica. (1 pts.)	Los diagramas a bloques, o de flujo o esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.7 pts.)	Los diagramas a bloques, de flujo y esquemáticos no son acorde al de la práctica y o no siguen una secuencia lógica. (.4pts.)	
La tabla de valores.	Los valores calculados y medidos presentan una desviación máxima del 10%. (1 pts.)	Los valores calculados y medidos presentan una desviación máxima del 15%. (.7 pts.)	Los valores calculados y medidos presentan una desviación máxima del 20%. (.4pts.)	
Las observaciones y conclusiones.	Son específicas y congruentes con la práctica. (1 pts.)	Las observaciones o conclusiones son específicas y congruentes con la práctica. (.7 pts.)	Las observaciones y las conclusiones no son específicas y congruentes con la práctica. (.4pts.)	
Bibliografía.	Es acorde al (los) tema (s) tratado (s) y está completa (1 pts.)	Es acorde a algún (os) tema (s) tratado (s), le falta algún elemento que la conforman (.7 pts.)	No es acorde al (los) tema (s) tratado (s), le faltan 2 elementos que la conforma (.4pts.)	
Fuentes de consulta.	Es acorde al (los) tema (s) tratado (s) (1 pts.)	Es acorde a algún (os) tema (s) tratado (s) (.7 pts.)	Es acorde a algún (los) tema (s) tratado (s) (.4pts.)	

**Nombre:** Luis Fernando González Chávez

**Tema:** Memoria Estática y Dinámica

**No.** T-16

**Página** 1

## Índice

Teoría 3

Matriz 3

Vector 3

Cálculos 3

Diagramas 4

Tabla (comparativa) 6

Bibliografía 7

Fuentes de consulta 7

## Teoría

Matriz: En la programación, una matriz es básicamente un arreglo de arreglos. Estas pueden tener múltiples dimensiones que determinan cómo se va a almacenar la información. Como sabemos, los arreglos no son nada más que punteros que nos llevan a un área en la memoria en la que ya está alocada un espacio, definido por su tamaño. Con esto, podemos decir que una matriz es simplemente una colección de punteros hacia otros punteros. Para llenarlos, uno puede usar igual un bucle de “for”, pero deberá estar también anidado.

```
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        matriz[i][j] = 0
    }
}
```

Vector: Los vectores son parte de la librería estándar de c++ y nos permiten hacer arreglos dinámicos que pueden cambiar de tamaño dependiendo de cuántos elementos tengan. Estos, a diferencia de los arreglos convencionales, nos permiten definir el tamaño durante la ejecución del programa, lo cual no sería posible debido a la necesidad de una constante a la hora de declararlas.

```
std::vector<tipoVariable> nombreVariable
```

## Cálculos

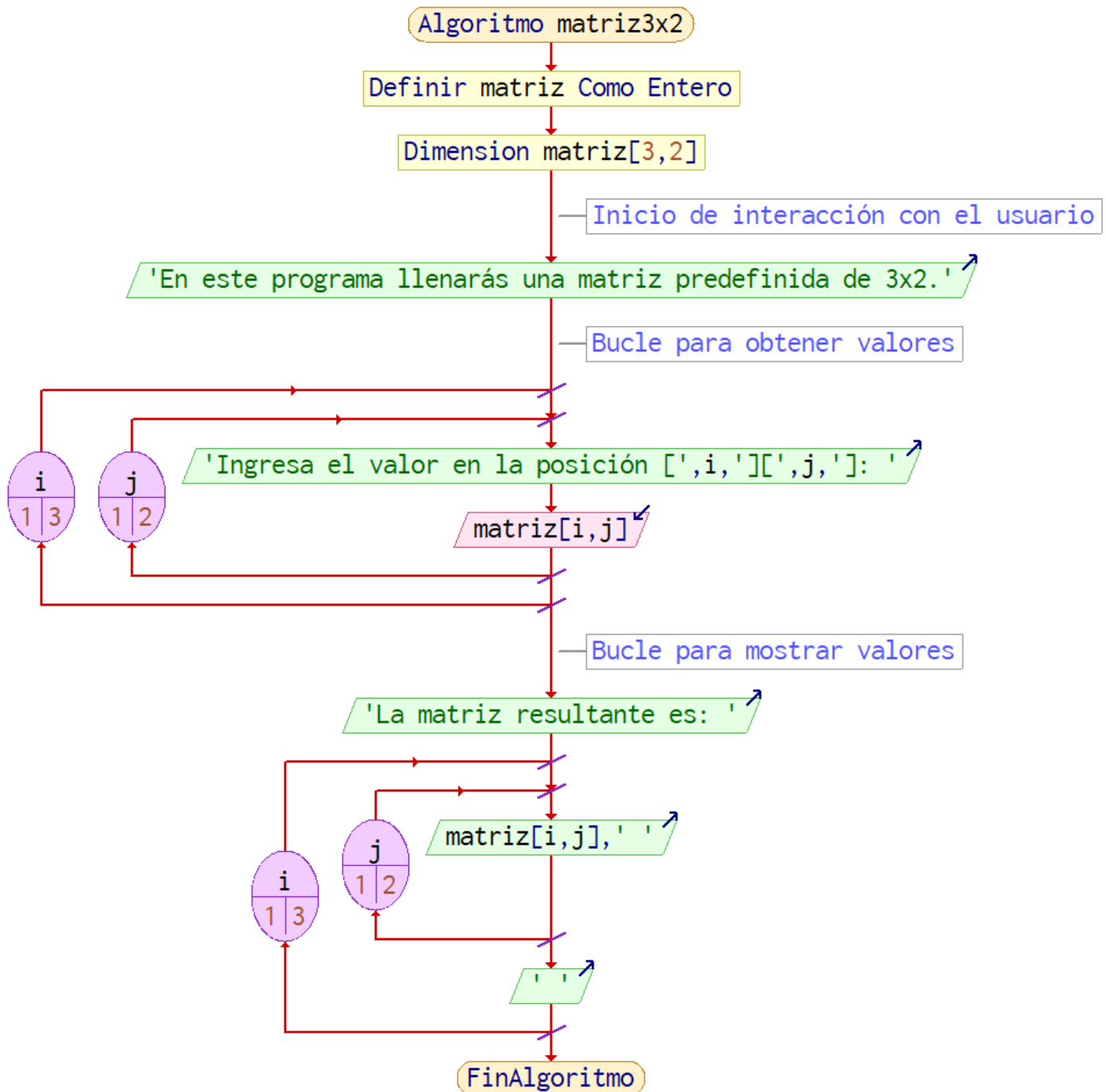
```
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        matriz[i][j] = 0
    }
}
```

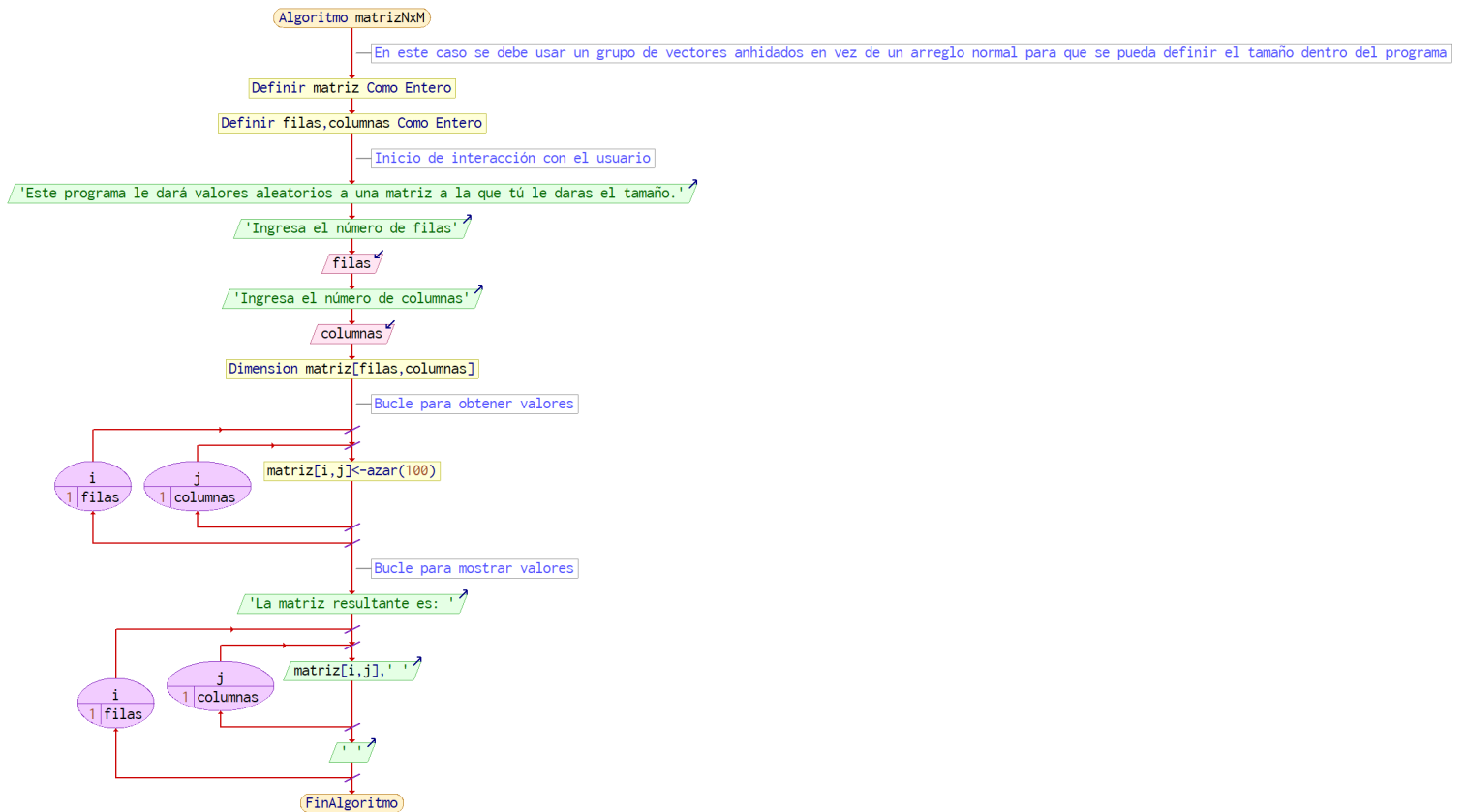
// En este caso “i” controlará el bucle para las filas, mientras que “j” controlará el bucle para las columnas. Esto seguirá hasta que todo esté lleno.

```
matriz2[i][j] = rand() % 100;
```

// Le da un valor aleatorio entre 0 y 99 a la variable en la posición [i][j]

## Diagramas





## Tabla (comparativa)

Acción	Expectativa	Resultado	Conclusión
Implementación de las medidas de validación de entero	Evitará que el usuario pueda terminar con el programa gracias a un fallo a la hora de ingresar los enteros.	Evita que el usuario pueda terminar con el programa gracias a un fallo a la hora de ingresar los enteros.	Funciona correctamente.
Petición de datos de parte del usuario.	Se le mostrará la posición en la que irá el valor ingresado.	Se muestra la posición en la que irá el valor ingresado.	Funciona correctamente.
Bucles anidados.	Empezarán con la primera fila y solo seguirán con la siguiente al terminar las columnas de la misma.	Empieza con la primera fila y solo sigue con la siguiente al terminar las columnas de la misma.	Funciona correctamente.
Representación de matriz.	Las funciones <i>std::right</i> y <i>std::setw()</i> nos permitirán tener una matriz mucho más organizada en la consola.	Las funciones <i>std::right</i> y <i>std::setw()</i> nos permiten tener una matriz mucho más organizada en la consola.	Funciona correctamente.
<code>std::vector&lt;std::vector&lt;int&gt;&gt; matriz2(n, std::vector&lt;int&gt;(m, 0));</code>	Al definir ya un tamaño para los vectores, no nos saldrá el error "vector subscript out of range".	Al definir ya un tamaño para los vectores, no nos sale el error "vector subscript out of range" y el programa corre correctamente.	Funciona correctamente.

## Bibliografía

Stroustrup, B. (2013). 31.4 Containers. En *The C++ Programming Language* (4th ed., p. 902). Addison-Wesley Professional.

## Fuentes de consulta

Chernikov, Y. (2017, 20 septiembre). *Dynamic Arrays in C++ (std::vector)*. YouTube. Recuperado 27 de octubre de 2021, de <https://www.youtube.com/watch?v=PocJ5jXv8No>

GeeksforGeeks. (2021a, febrero 17). *Vector in C++ STL*. Recuperado 27 de octubre de 2021, de <https://www.geeksforgeeks.org/vector-in-cpp-stl/>

GeeksforGeeks. (2021b, octubre 20). *2D Vector In C++ With User Defined Size*. Recuperado 27 de octubre de 2021, de <https://www.geeksforgeeks.org/2d-vector-in-cpp-with-user-defined-size/>

*std::left, std::right, std::internal*. (2021, 5 octubre). Cppreference.Com. Recuperado 26 de octubre de 2021, de <https://en.cppreference.com/w/cpp/io/manip/left>

*std::setw*. (2021, 31 marzo). Cppreference.Com. Recuperado 26 de octubre de 2021, de <https://en.cppreference.com/w/cpp/io/manip/setw>