

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: Realizar un inventario de Juegos de Mesa No: R.1 15

Fecha propuesta: 26/Oct/2021 **Fecha de Entrega:** 20/Oct/2021

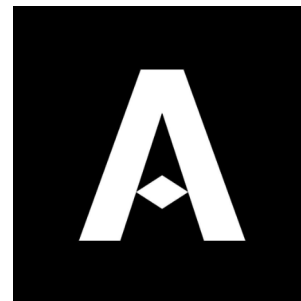
Escuela: Instituto Universitario Amerike **Plantel** Guadalajara

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: Realizar un inventario de Juegos de Mesa

No. T-15
Página 1

Índice

Teoría 3

Arreglos 3

Cálculos 3

Diagramas 4

Tabla (comparativa) 6

Bibliografía 7

Fuentes de consulta 7

Teoría

Arreglos: Una forma de representar una colección de variables, usualmente del mismo tipo, en una fila. Estos ocupan un área continua en la memoria. Las variables deben ser creadas manualmente; hay veces en las que quieres declararlas por separado y darles nombres a cada una, pero hay otras veces en las que solo quieres tener un grupo grande de variables. En estos casos, usarías un arreglo. Usualmente uno quiere asignar las variables en la pila de llamadas (stack), pero hay situaciones en las que uno quiere asignarlas en el montículo libre (heap). Dependiendo de cuál elijas será la forma en la que la declares.

Para declarar un arreglo en memoria stack:

```
tipoVariable nombreVariable[cantidadVariables];
```

Ejemplo: `int edad[10];` // Declaración de un arreglo en stack de 10 int llamados “edad”.

Para declarar un arreglo en memoria heap:

```
tipoVariable* nombreVariable = new tipoVariable[cantidadVariables];
```

Ejemplo: `int edad = new int[10];` // Declaración de un arreglo en heap de 10 int llamados “edad”.*

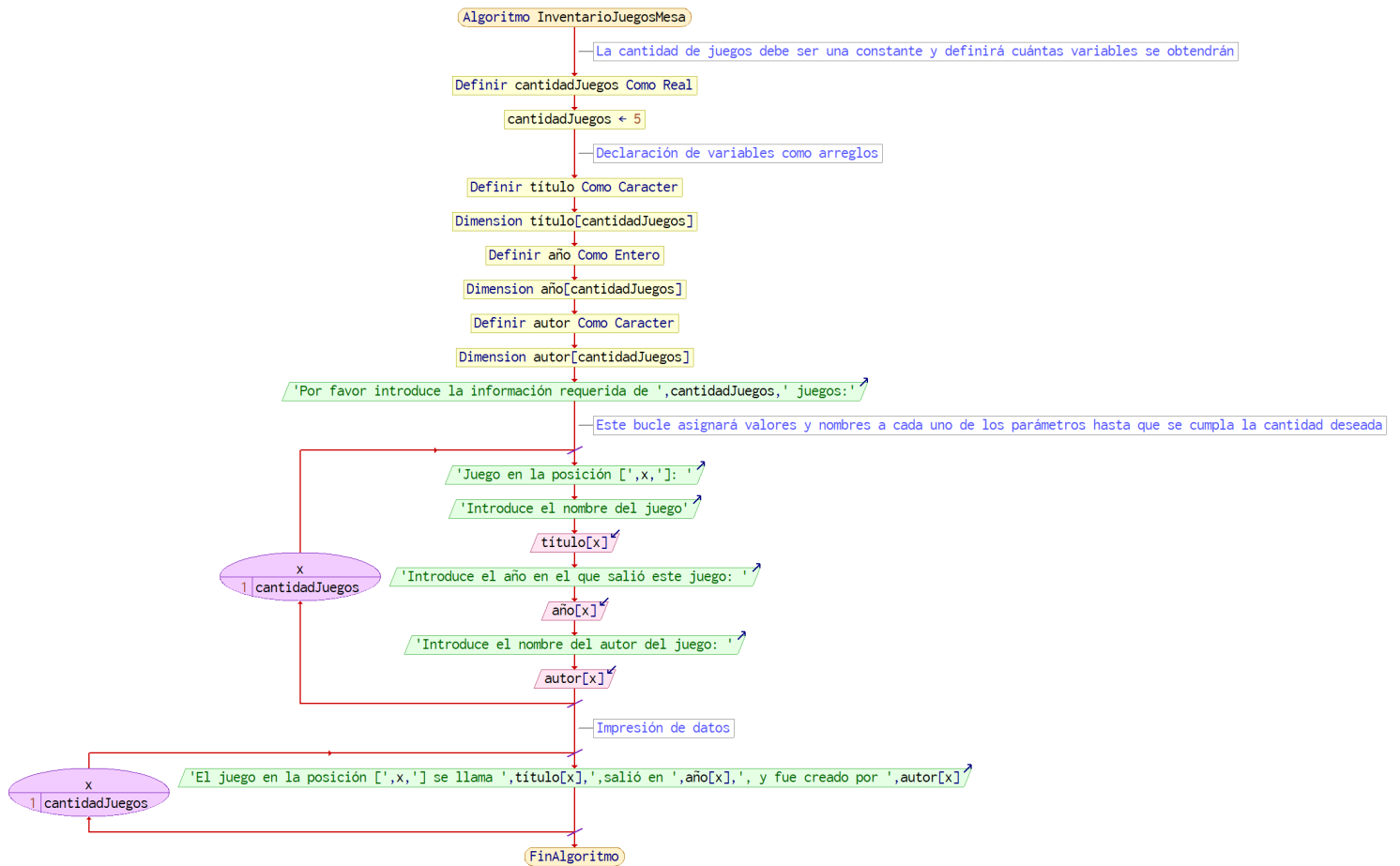
En esta práctica usaremos un bucle “for” para poder asignarle valores a todas las variables dentro de nuestros arreglos.

Cálculos

```
for (int i = 0; i < cantidadJuegos; i++)
```

// Debemos seguir con el bucle hasta que se le den valores a cada una de las variables dentro de los arreglos.

Diagramas



Algoritmo InventarioJuegosMesa
// La cantidad de juegos debe ser una constante y definirá cuántas variables se obtendrán
Definir cantidadJuegos Como Real
cantidadJuegos ← 5
// Declaración de variables como arreglos
Definir título Como Caracter
Dimension título[cantidadJuegos]
Definir año Como Entero
Dimension año[cantidadJuegos]
Definir autor Como Caracter
Dimension autor[cantidadJuegos]
Escribir 'Por favor introduce la información requerida de ',cantidadJuegos,' juegos:'
// Este bucle asignará valores y nombres a cada uno de los parámetros hasta que se cumpla la cantidad deseada
Para x Desde 1 Hasta cantidadJuegos Con Paso
Escribir 'Juego en la posición [' ,x,']: '
Escribir 'Introduce el nombre del juego'
Leer título[x]
Escribir 'Introduce el año en el que salió este juego: '
Leer año[x]
Escribir 'Introduce el nombre del autor del juego: '
Leer autor[x]
// Impresión de datos
Para x Desde 1 Hasta cantidadJuegos Con Paso
Escribir 'El juego en la posición [' ,x,'] se llama ',título[x],' ,salió en ',año[x],' , y fue creado por ',autor[x]
FinAlgoritmo

Tabla (comparativa)

Acción	Expectativa	Resultado	Conclusión
Uso del bucle: <code>for (int i = 0; i < cantidadJuegos; i++)</code>	Nos permitirá asignarle valores a cada una de las variables y terminará el bucle.	Nos permite asignar todos los valores y termina cuando esto sucede.	Funciona correctamente.
<code>getline(cin >> ws, string[i]);</code>	Obtendrá una entrada por parte del usuario. "ws" nos permitirá que el usuario introduzca una string con espacios en blanco y no los tomará en cuenta para terminar con la función.	Las strings se obtienen de manera correcta, sin importar que tengan espacios en blanco.	Funciona correctamente
<code>getline(cin >> ws, título[i]);</code>	Obtendrá un int por parte del usuario y lo registrará en el espacio de memoria correspondiente dentro del arreglo.	Obtiene un int por parte del usuario y lo registra en el espacio de memoria correspondiente dentro del arreglo.	Funciona correctamente.
Uso del bucle: <code>for (int j = 0; j < cantidadJuegos; j++)</code>	Este bucle mostrará todos los juegos de mesa que ingresamos con su título, año de salida, autor y posición en el arreglo.	Este bucle muestra todos los juegos de mesa que ingresamos con su título, año de salida, autor y posición en el arreglo.	Funciona correctamente

Bibliografía

Stroustrup, B. (2013). 7.3 Arrays. En *The C++ Programming Language* (4th ed., p. 174). Addison-Wesley Professional.

Fuentes de consulta

Chernikov, Y. (2017a, agosto 19). *Arrays in C++*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=ENDaJi08jCU>

Chernikov, Y. (2017b, agosto 31). *How to CREATE/INstantiate OBJECTS in C++*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=Ks97R1knQDY>

Chernikov, Y. (2017c, septiembre 1). *The NEW Keyword in C++*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=NUZdUSqsCs4>

Chernikov, Y. (2017d, septiembre 3). *OPERATORS and OPERATOR OVERLOADING in C++*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=mS9755gF66w>

Chernikov, Y. (2017e, septiembre 8). *Object Lifetime in C++ (Stack/Scope Lifetimes)*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=iNuTwvD6cil>

Chernikov, Y. (2017f, septiembre 11). *SMART POINTERS in C++ (std::unique_ptr, std::shared_ptr, std::weak_ptr)*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=UOB7-B2MfwA>

Chernikov, Y. (2017g, diciembre 10). *Stack vs Heap Memory in C++*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=wJ1L2nSIV1s>

Chernikov, Y. (2018a, febrero 22). *Why I don't «using namespace std»*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=4NYC-VU-svE>

Chernikov, Y. (2018b, febrero 28). *Namespaces in C++*. YouTube. Recuperado 19 de octubre de 2021, de <https://www.youtube.com/watch?v=ts1Eek5w7ZA>

Microsoft. (2021a, agosto 3). *Arrays (C++)*. Microsoft Docs. Recuperado 19 de octubre de 2021, de <https://docs.microsoft.com/en-us/cpp/cpp/arrays-cpp?view=msvc-160>

Microsoft. (2021b, agosto 3). *new Operator (C++)*. Microsoft Docs. Recuperado 19 de octubre de 2021, de <https://docs.microsoft.com/en-us/cpp/cpp/new-operator-cpp?view=msvc-160>