

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: IF-ELSE **No:** R.1 12

Fecha propuesta: 05/Oct/2021 **Fecha de Entrega:** 04/Oct/2021

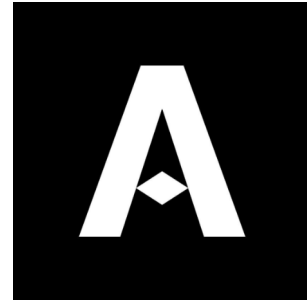
Escuela: Instituto Universitario Amerike **Plantel** Guadalajara

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: IF-ELSE

No. T-12

Página 1

Índice

Teoría 3

if 3

else 3

Cálculos 4

Diagramas 5

Tabla (comparativa) 6

Bibliografía 8

Fuentes de consulta 8

Teoría

if: “Una declaración con una derivación condicional; si se cumple, el código dentro de “if” se ejecuta, si no, puede pasar a otra declaración “if”, a una declaración “else” o simplemente sigue con el código.

Como funcionan estas declaraciones es que primero chequea si la condición se cumple, y luego va a otra porción de la memoria para empezar a ejecutar instrucciones desde ahí, lo cual ralentiza el programa. Si el objetivo de tu código es que el programa corra lo más rápido posible, deberás buscar formas de reducir e incluso eliminar este tipo de declaraciones y sus derivados como “else” y “else if” (que en realidad son dos declaraciones separadas, no una sola, pues “else if” no es una palabra reservada).” (Luis Fernando González Chávez, 2021)

Para usarlo, escribiremos la declaración de la siguiente forma:

```
if (condición/es) instrucción_si_se_cumple/n;
```

O si serán varias acciones si se cumple la condicional, se escribe:

```
if (condición)
{
    instrucciones_si_se_cumple/n;
}
```

Es importante recordar que para las condicionales se usa ‘==’ y no ‘=’, pues una compara dos valores o variables mientras que la otra ingresa un valor en una variable.

Para usar más de una condicional, usas ‘&&’ para que múltiples se tengan que cumplir o ‘||’ para que con que una se cumpla se tome como verdadero.

else: “Viene después de una declaración “if” y sirve para controlar una rama de condiciones. Se ejecuta si la condición del “if” no se cumplió y terminó siendo falsa.” (Luis Fernando González Chávez, 2021)

Al igual que ‘if’, ‘else’ puede indicar una o varias instrucciones que se llevan a cabo si la condicional anterior fue falsa y se escriben de la misma manera pero sin la condicional. Además, se puede agrupar con un ‘if’ de la forma ‘else if’ para que cheque otra condicional.

Puede haber múltiples 'else if', pero hay que tener el impacto que tiene en el rendimiento de nuestro programa.

Cálculos

1. if (miEntero == enteroCorrecto)
2. if (miFloat == floatCorrecto)
3. if (miDouble == doubleCorrecto)
4. if (miChar == charCorrecto)
5. Como quiero que haya un retraso de cinco segundos, pero la función 'Sleep()' funciona en milisegundos, multiplico $5 * 1000 = 5000$.

Diagramas

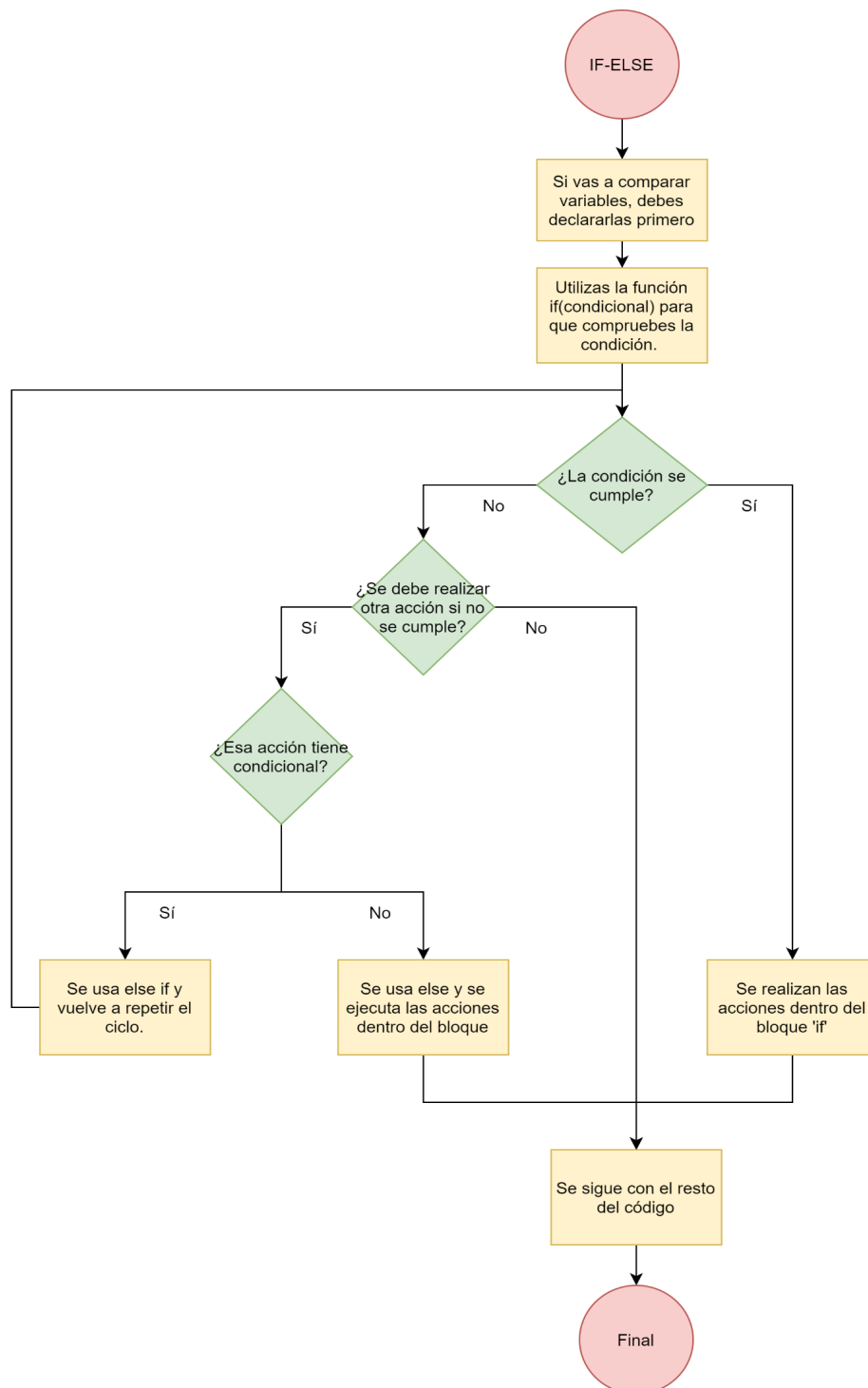


Tabla (comparativa)

Función	Expectativa	Resultado	Conclusión
if(miString == stringCorrecto)	<p>Si se cumple: Mostrará aprobación y seguirá con el código.</p> <p>Si no se cumple: Hará todo lo del bloque else.</p>	<p>Cuando se cumple muestra aprobación y sigue con el código.</p> <p>Cuando no se cumple, ejecuta todo lo que está en el bloque else.</p>	Funciona correctamente
if(miEntero == enteroCorrecto)	<p>Si se cumple: Mostrará aprobación y seguirá con el código.</p> <p>Si no se cumple: Hará todo lo del bloque else.</p>	<p>Cuando se cumple muestra aprobación y sigue con el código.</p> <p>Cuando no se cumple, ejecuta todo lo que está en el bloque else.</p>	
if(miFloat == floatCorrecto)	<p>Si se cumple: Mostrará aprobación y seguirá con el código.</p> <p>Si no se cumple: Hará todo lo del bloque else.</p>	<p>Cuando se cumple muestra aprobación y sigue con el código.</p> <p>Cuando no se cumple, ejecuta todo lo que está en el bloque else.</p>	
if(miDouble == doubleCorrecto)	<p>Si se cumple: Mostrará aprobación y seguirá con el código.</p> <p>Si no se cumple: Hará todo lo del bloque else.</p>	<p>Cuando se cumple muestra aprobación y sigue con el código.</p> <p>Cuando no se cumple, ejecuta todo lo que está en el bloque else.</p>	

if(miChar == charCorrecto)

Si se cumple:
Mostrará
aprobación y
seguirá con el
código.

Cuando se cumple
muestra aprobación
y sigue con el
código.

Si no se cumple:
Hará todo lo del
bloque else.

Cuando no se
cumple, ejecuta todo
lo que está en el
bloque else.

Funciona correctamente

Bibliografía

Stroustrup, B. (2013). 2.2 The Basics. En The C++ Programming Language (4th ed., p. 273). Addison-Wesley Professional.

Fuentes de consulta

Chernikov, Y. (2017, mayo 21). CONDITIONS and BRANCHES in C++ (if statements). YouTube.

<https://www.youtube.com/watch?v=qEgCT87KOfc>

GeeksforGeeks. (2017, 20 mayo). switch vs if else.

<https://www.geeksforgeeks.org/switch-vs-else/#:~:text=A%20switch%20statement%20is%20usually,a%20set%20of%20nested%20ifs.&text=Check%20the%20Testing%20Expression%3A%20An,enumerated%20value%2C%20or%20String%20object>

Microsoft. (2020a, octubre 2). if-else statement (C++). Microsoft Docs.

<https://docs.microsoft.com/en-us/cpp/cpp/if-else-statement-cpp?view=msvc-160>

Microsoft. (2020b, 7 diciembre). C++ *program termination*. Microsoft Docs.

<https://docs.microsoft.com/en-us/cpp/cpp/program-termination?view=msvc-160>