

Nombre: GersonRonaldoOrtizMarin NodeMatrícula 20171176

Materia: Fundamentos de la Programacion Grupo: Turno: Matuino

Carrera: Developer

Tema: Investigacion 3 No: T.1

Fecha propuesta: Fecha de Entrega:

Escuela: Plantel

Calle: No: Colonia: C.P.:

Teléfono: Ciudad:

Logotipo personal



Logotipo (de la escuela)



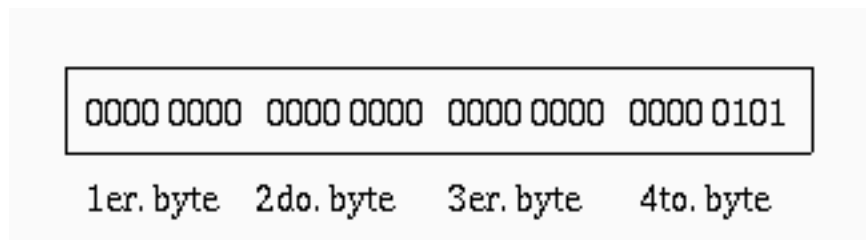
Firma del alumno (a)

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 20 minutos de iniciada la clase. (.7 pts.)	Al minuto 30. (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.	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 resumen.	Es congruente con el (los) tema (s) y aporta conceptos propios del alumno. (1.5pts.)	Es congruente con el (los) tema (s) y no aporta conceptos propios del alumno. (1 pts.)	No es congruente con el (los) tema (s) y no aporta conceptos propios del alumno. (.4pts.)	
Conocimientos.	Responde acertadamente las preguntas del tema tratado que se le formulan oralmente. (1.5pts.)	Responde acertadamente algunas preguntas del tema tratado que se le formulan oralmente. (.7 pts.)	No responde acertadamente las preguntas del tema tratado que se le formulan oralmente. (.4 pts.)	
Las preguntas.	Todas las preguntas formuladas son acordes con su nivel de estudio, cuentan con cálculos matemáticos y su respectiva respuesta. (1 pts.)	Una o dos preguntas formuladas no son acordes con su nivel de estudio o no cuentan con cálculos matemáticos, o su respectiva respuesta. (.7 pts.)	Tres o más preguntas formuladas no son acordes con su nivel de estudio o no cuentan con cálculos matemáticos o su respectiva respuesta. (.4pts.)	
Presentación y archivo electrónico.	Es congruente con el (los) tema (s) presenta una secuencia lógica y no tiene más de dos errores ortográficos. (1 pts.)	Es congruente con el (los) tema (s) presenta una secuencia lógica y no tiene más de tres a cuatro errores ortográficos. (.8 pts.)	No es congruente con el (los) tema (s) no presenta una secuencia lógica y tiene más de cinco errores ortográficos. (.4pts.)	
Bibliografía.	Es acorde al (los) tema (s) tratado (s) y está completa (.7 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) (.3 pts.)	Es acorde a algún (os) tema (s) tratado (s) (.3 pts.)	Es acorde a algún (os) tema (s) tratado (s) (.4 pts.)	

Almacenamiento de variables en la memoria.

Los números se almacenan en las variables. Una variable representa un trozo de la memoria del computador. La memoria está formada por una gran cantidad de bytes y cada byte está constituido por 8 bits. Un bit puede almacenar un 1 o un 0.



Los enteros

Una variable entera (int) está formada por 4 bytes, es decir 32 bits. Estos 32 bits representan el valor almacenado por esa variable en binario.

Una variable entera (int) siempre utiliza 32 bits, aún cuando el número sea pequeño. Por otra parte, no es capaz de almacenar números demasiado grandes (en valor absoluto).

usualmente se usa el tipo int para almacenar enteros, pero también existen otros tipos que también almacenan enteros pero en menos o más bits. La siguiente tabla muestra los tipos enteros presentes en Java:

tipo	número	rango
	de bits	representado
int	32	$[-2^{31}, 2^{31}-1]$
short	16	$[-2^{15}, 2^{15}-1]$
byte	8	$[-2^7, 2^7-1]$
long	64	$[-2^{63}, 2^{63}-1]$

Los reales

Una variable real (double) está formada por 8 bytes, es decir el doble de un entero. En ella se almacenan números reales en formato binario. Por ejemplo, el número:

1101.1010

signo	mantisa	exponente	en binario	en decimal
+	1000...	100	1 seguido de 99 ceros	2^{99}
+	1010...	0	0.1010	$0.5 + 0.125$ (0.625)

+	1000...	-3	0.0001	2^{-4} (0.0625)
---	---------	----	--------	-------------------

¿¿Como afecta??

Es importante saber cuanto llega a almacenar cada tipo de variables, pues eso se debe tener en cuenta para poder declarar las variables.

Si queremos ingresar un numero entero para que necesitamos declarar una variable con float que tenga mas espacio para decimales.

También en el caso de un bool o un char para si vamos a necesitar solo espacio para verdadero o falso no necesitamos un char con mayor espacio en texto

Nota: hay que tener en cuenta que, 8 bits son equivalentes a 1 byte y cada carácter posee 8 bits, es decir que cada carácter tiene un espacio de 1 byte.

Por ejemplo, el carácter A ocuparía 1 byte de memoria,

Una forma de saber cuanto espacio tiene cada variable:

```
#include <iostream.h>
#include <conio.h>
main ()
{
cout<<endl;
cout<<"Los tipos de datos enteros: '\int\' ocupan: "<<sizeof(int)<<" bytes de memoria \n\n";
cout<<"Los tipos de datos enteros cortos: '\short int\' ocupan: "<<sizeof(short int)<<" bytes de memoria \n\n";
cout<<"Los tipos de datos enteros largos: '\long int\' ocupan: "<<sizeof(long int)<<" bytes de memoria \n\n";
cout<<"Los tipos de datos flotantes: '\float\' ocupan: "<<sizeof(float)<<" bytes de memoria \n\n";
cout<<"Los tipos de datos entero con decimales: '\double\' ocupan: "<<sizeof(double)<<" bytes de memoria \n\n";
cout<<"Los tipos de datos entero con decimales largos: '\long double\' ocupan: "<<sizeof(long double)<<" bytes de memoria \n\n";
cout<<"Los tipos de datos caracter: '\char\' ocupan: "<<sizeof(char)<<" bytes de memoria \n\n";
getch();
}
```

Mi opinión:

La declaración de las variables tiene que tomar en cuenta los datos que vamos a ingresar y si la variable es la ideal para hacer mas funcional el programa, evitar problemas a futuro con otros programas y ahorrando espacio en la memoria.

Bibliografía:

<https://eperdomo89.wordpress.com/2009/11/11/espacios-de-memoria-por-tipo-de-variable/>

<http://recursostic.educacion.es/observatorio/web/fr/software/programacion/972-tipos-de-datos>

<https://www.youtube.com/watch?v=ouyEMxQl8>

https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=899:tipos-de-datos-en-c-declarar-variables-enteras-int-long-o-decimal-float-double-char-inicializacion-cu00510f&catid=82&Itemid=210