

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO DE CLASE			
DOCENTE:	Ing. Rogelio Padilla G, MSc	SEMESTRE	
ESTUDIANTE:		FECHA:	
TEMA:	Tipos de datos simples		

1.1. TIPOS DE DATOS SIMPLES

VARIABLE: Celda de memoria que permite almacenar un dato

Ejemplo de dato:

- 5
- 'A'
- "Juan"
- 2.5
- Falso

Datos simples

Tipos variables

- Enteras ejemplo 2; -200;
- Carácter ejemplo 'x' ; 'X'; '2'; ('-2' no es un carácter)
- Cadena ejemplo "Maria";"Casa";"12345"; "-2"; "2";"";
- Decimal ejemplo 2.5 ; -0.25; 2;
- Booleanas ejemplo (verdadero o falso)(Si o no)(1 o 0)

Datos simples

- **Tipos variables**

Como se define una variable:

- El nombre debe ser alfanumérico
- Debe de comenzar con una letra
- Puede contener guion bajo

Ejemplo:

- Izmael_1
- Casa
- C
- A100
- Izmael1
- 9_de_octubre_y_boyaca

Recomendaciones para nombre de variables

- 4 caracteres
- Nombre tenga relación con su contenido
- Un nombre compuesto 1 letra de primera palabra y 3 letras la segunda palabra

Nombre(nomb)

Apellido(apel)

Edad(edad)

Estatuta(esta)

Estado civil(eciv)

Fecha de nacimiento(fnac)

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Declaraciones de variables

Seudo código	Dev c++
Entero nume;	int nume;
Nume=5;	nume=0;
Entero edad=0;	Int edad=0;
Decimal esta=0;	float esta=0;
Carácter sexo;	char sexo;
Cadena nomb;	char nomb[25];
Boolean band;	boolean band;

1.2. ARREGLOS

Datos simples (Tipos variables)

- Enteros
- Carácter
- Decimal
- Booleanos

Datos compuestos(TDA)

ARREGLO: Celdas de memoria que permite almacenar datos de un solo tipo de dato

- Arreglos
 - Arreglo unidimensional (Cadena)(VECTOR)
 - Arreglos bidimensionales (TABLAS)
 - Arreglos tridimensionales
- Estructura variables



1.2.1. ARREGLOS UNIDIMENSIONALES (VECTOR)

ARREGLO: Celdas de memoria que permite almacenar datos del mismo tipo

Arreglos unidimensionales o vectoriales

Demostrar el uso de scanf – cin – gets

J	U	A	N	
---	---	---	---	--

5	8	-6	200	-100
---	---	----	-----	------

Nume[5]

Entero nume[5];

int nume[5];

2
7
-5
3
8

Nume[2]=-5

Declarar un arreglo

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

- Entero int edad[100];
- Decimal suel[200];
- Char sexo[50];

1.2.2. ARREGLOS BIDIMENSIONALES (MATRIZ)

Arreglos bidimensional o matriz

J	U	A	N	
S	O	L		
P	A	Z		

PUEDO INGRESAR 15 CARACTERES ARREGLO BIDIMENSIONAL
 PUEDO INGRESAR 3 CADENAS ARREGLO UNIDEMENSIONAL

5	8	-6	200	-100
1	1	1	11	1
1	1	1	1	1

NUME[1][3]

Entero nume[3][5]; int nume[3][5];

Declarar un arreglo

- Entero int edad[100] [100];
- Decimal suel[200] [100];
- Char sexo[50] [100];

Arreglos tridimensionales

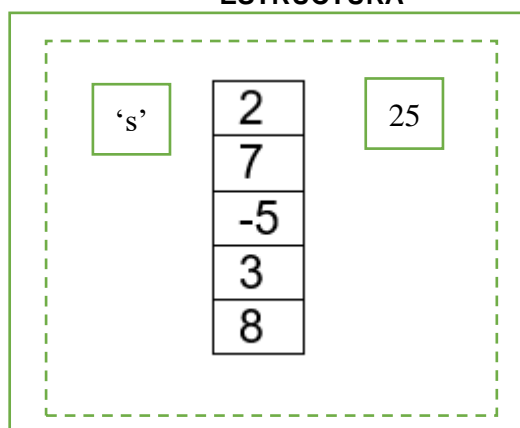
5	8	-6	200	-100
1	1	1	1	1
1	1	1	1	1

Entero nume[3][5][2]; int nume[3][5][2];

1.3. ESTRUCTURAS Y UNIONES

Estructura o Unión : Celdas de memoria que permite almacenar datos de diferente tipo

ESTRUCTURA



```
struct sper
{
    char eciv;
    int nume[10];
    int edad;
}pers[3];
```

pers.eciv='s';

pers[0].eciv='c';

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

```
pers[1].eciv='d';  
pers[2].eciv='v';
```

ESTRUCTURAS

Cuando uno declara una variable en un programa, está definiendo un espacio de memoria en la que se podrá almacenar un solo valor. Por ejemplo, cuando colocamos en un programa la sentencia.

```
int a;
```

estamos definiendo la variable entera a, la cual sólo podrá almacenar un valor en un instante dado. Este tipo de variables se denominan escalares.

Por otro lado, cuando definimos un arreglo (estático o dinámico) estamos definiendo un conjunto de variables, todas identificadas por un mismo nombre, diferenciándose cada una por uno o más índices. Por ejemplo, cuando escribimos la sentencia

```
int nume[5];
```

estamos definiendo las variables enteras nume[0], nume[1], nume[2], nume[3] y nume[4]. Sin embargo, a pesar de ello, los arreglos tienen un inconveniente, que todos elementos de un arreglo tienen el mismo tipo de dato. Esto quiere decir que, en el ejemplo, en el arreglo a sus elementos solo pueden albergar valores enteros.

Si quisiéramos almacenar por medio de un programa una lista de personas en donde por cada una tenemos un código, un nombre y un sueldo, para poder hacerlo tendríamos que definir un arreglo por cada uno de los datos, esto es un arreglo de enteros para el código, un arreglo de cadenas de caracteres para el nombre y uno de punto flotante para el sueldo. El inconveniente sería que los datos de una persona no se pueden manejar, así, como una unidad. Los arreglos reciben el nombre de **variables estructuradas**.

Los registros o "estructuras" como se denominan en el lenguaje C, permiten definir variables que pertenecen también a la categoría de variables estructuradas, en este sentido al declarar una estructura estaremos definiendo, como en el caso de los arreglos, un conjunto de datos, todos identificados por el mismo nombre, pero con la diferencia que cada elemento del conjunto puede ser de diferente tipo. La forma en que se diferenciará un elemento de otro del conjunto ya no se hará por medio de índices, sino que se le asignará a cada elemento un nombre. Cada elemento

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

de una estructura se denominará campo. Las estructuras son un tipo primitivo de datos, su evolución ha dado lugar a las "Clases" definidas en la programación orientada a objetos (POO) e implementadas en C++.

RESUMEN DE STRUCTURAS

Una estructura contiene varios datos. La forma de definir una estructura es haciendo uso de la palabra clave struct. Aquí hay ejemplo de la declaración de una estructura:

```
struct Scliente
{
    int clie_codi;
    double clie_suel;
    char clie_nomb[25];
} variable;
```

O

```
struct Scliente
{
    int clie_codi;
    double clie_suel;
    char clie_nomb[25];
};
```

"variable" es una instancia de "Scliente" y no es necesario ponerla aquí. Se podría omitir de la declaración de "Scliente" y más tarde declararla usando:

```
struct Scliente variable;
```

También es una práctica muy común asignarle un alias o sinónimo al nombre de la estructura, para evitar el tener que poner "struct Scliente" cada vez. C nos permite la posibilidad de hacer esto usando la palabra clave typedef, lo que crea un alias a un tipo:

```
typedef struct
{
    int clie_codi;
    double clie_suel;
    char clie_nomb[25];
} Structura;
```

La estructura misma no tiene nombre (por la ausencia de nombre en la primera línea), pero tiene de alias "Structura". Entonces se puede usar así:

```
Structura variable;
```

UNIONES

Es bastante parecida a la estructura la única diferencia es que la estructura genera espacios diferentes para los datos, mientras que la unión utiliza el mismo espacio

```
#include<stdio.h>
#include<conio.h>
```

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

```
union dato {
    int x;
    char letra;
    float z;
};

int main()
{
    union dato d;
    d.x=10;
    printf("Impresion del entero:%i\n",d.x);
    printf("Impresion del caracter:%c\n",d.letra);
    printf("Impresion del float:%0.2f\n",d.z);
    d.letra='A';
    printf("\n\nImpresion del entero:%i\n",d.x);
    printf("Impresion del caracter:%c\n",d.letra);
    printf("Impresion del float:%0.2f\n",d.z); //El dato se ha modificado
    d.z=5.25;
    printf("\n\nImpresion del entero:%i\n",d.x);
    printf("Impresion del caracter:%c\n",d.letra);
    printf("Impresion del float:%0.2f\n",d.z); //El dato se ha modificado nuevamente
    getch();
    return 0;
}
```

PUNTERO

Un puntero no es más que una variable, en la cual se almacena una dirección de memoria. Esto parece muy simple, Al ser una dirección de memoria, le podemos decir a un puntero que en ese lugar donde apunta queremos almacenar un valor, por ejemplo, un número.