

**INSTITUTO POLITÉCNICO NACIONAL****UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS  
AVANZADAS****Práctica No. 1. Ambiente de programación.**

**Unidad Temática:** I Introducción a la programación

**Lugar de realización:** Laboratorio de Cómputo

**Duración:** 8.5 hrs.

**Objetivo**

Conocer el ambiente del Lenguaje de programación en C.

**Resultados Esperados**

- Trabajo de investigación logrado en el pre-reporte
- Construcción de pequeños programas en Lenguaje C para conocer la estructura
- Identificación, selección y desarrollo de los ejercicios usando la variedad de operadores

**Pre-reporte.**

Para el buen desarrollo de la práctica el alumno entregará un reporte que incluya los siguientes elementos:

1. Tabla comparativa de tipo de datos. La cual debe incluir el nombre del tipo de dato, su tamaño, palabra reservada y rango de valores.
2. Tabla de operadores en orden según la precedencia. Comprenderá el operador, su clasificación (aritmético, lógico, relacional, etc), y ejemplo de sintaxis.
3. Descripción de los formatos controlados como `\n`, `\t`, `\r`, `\v`, `\\`, `\'` y `\a`

*"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en lo bello y maravilloso mundo del saber". (Einstein)*

## Material y Equipo.

- IDE de desarrollo en C, puede ser Dev-Cpp, CodeBlocks
- Editor de diagramas de flujo, se sugiere DFD
- Computadora

## Introducción.

### Proceso de un programa



Figura 1.1. Proceso de un Programa

### Estructura de un programa

La escritura de un programa consiste en el llamado a las directivas del procesador (como objetos externos) a través del uso de `#include`, el nombre de la librería puede ir entre signos de mayor que y menor que, o entre comillas.

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

La función principal de un programa es la función `main`, la cual tiene un valor de retorno del tipo genérico conocido como `void` y cuyos parámetros son recibidos entre paréntesis; las acciones van dentro de las llaves. Su sintaxis es la siguiente:

```
void main () {  
  
}
```

### Variables y Constantes

Los datos son el elemento principal con la que se alimenta un programa para generar los resultados; todos los datos en C se ajusta a un *tipo de dato* que define las especificaciones.

Para declarar una variable es necesario establecer su tipo de dato y su nombre.

Declaración:

```
Tipo_de_dato nombre_variable;
```

Ejemplo:

```
int num_salon;
```

Para inicializar a la variable se requiere un valor compatible con el tipo de dato. Éste puede cambiar durante la ejecución del programa.

Asignación:

```
Tipo_de_dato nombre_variable = valor ;
```

Ejemplo:

```
int num_salon = 302;
```

Es posible hacer una declaración con asignación, lo cual implica que la variable tendrá un valor inicial pero podría cambiar.

Una **constante simbólica** representa (sustituye) a una secuencia de caracteres, en vez de representar a un valor (dato almacenado en memoria). Para declarar una constante simbólica se utiliza la directiva `#define`, se considera una buena práctica declararla con mayúscula:

Declaración:

```
#define <constante> <secuencia_de_caracteres>;
```

Ejemplo:

```
#define NUMERO_E 2.718281;
```

Por otra parte, es posible declarar una variable indicando que su valor es inalterable. Para ello, se utiliza el cualificador `const` lo que implica que la variable no puede cambiar su valor durante la ejecución del programa:

Declaración:

```
const tipo_dato variable = valor;
```

Ejemplo:

```
const int temperatura = 15;
```

## Operadores

Existen varios tipos de operadores:

1. **Operadores aritméticos:** Sirven para realizar operaciones aritméticas básicas: +, -, \*, /, %
2. **Operadores de asignación:** Muestran de manera abreviada las operaciones aritméticas: =, +=, -=, \*=, /=, %=
3. **Operadores de incremento y decremento:** Muestran de manera abreviada la operación de incremento y decremento de una variable en una unidad: ++, --
4. **Operadores relacionales:** Se utilizan por lo general para comprobar una condición, en sentencias: ==, <, >, <=, >=, !=
5. **Operador ternario:** También conocido como operador condicional, recibe tres argumentos. Se evalúa la expresión del primer argumento, y si su resultado es

verdadero, entonces devuelve como resultado el segundo argumento, en caso contrario devuelve el tercero: `arg1 ? arg2 : arg3`

6. **Operadores lógicos:** Se utilizan para comparar expresiones. `&&`, `||`, `!`
7. **Operadores lógicos bit a bit:** Se utilizan para aplicar operaciones lógicas a números enteros bit a bit: `<<`, `>>`, `>>>`, `<<<`, `&`, `|`,

Estos operadores permiten las operaciones aritméticas y las comparaciones dentro de las sentencias de control.

### Salida de datos

La función **printf** representa la salida estándar de la computadora, es decir, la pantalla. Lo utilizamos para imprimir datos en pantalla y se encuentra en la librería `"stdio.h"`.

Sintaxis: `printf("Hola mundo\n");`

### Entrada de datos

La función **scanf** representa la entrada estándar de la computadora, es decir, el teclado. Lo utilizamos para obtener datos del teclado y se encuentra en la librería `"stdio.h"`.

Sintaxis: `scanf("%d", &variable);`

## Desarrollo.

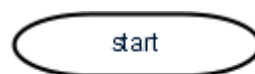
### Ejemplo:

Dado el valor de venta de un producto, hallar su IVA (16%) y el precio de venta.

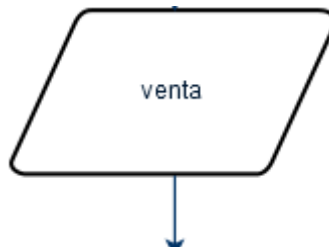
### Solución:

#### Diagrama de Flujo

1. Inicio del programa

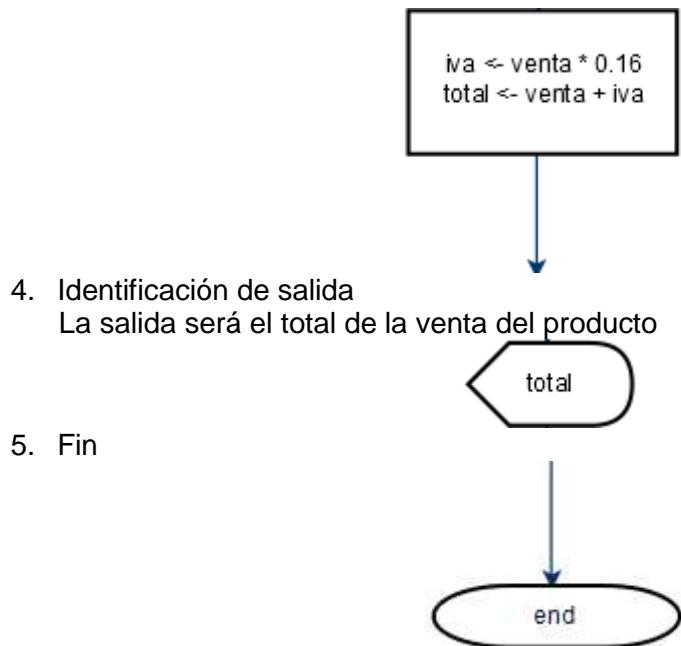


2. Identificación de entrada:  
La entrada ingresada por el usuario es: `venta`



3. Realizar cálculo:  
`iva = venta * 0.16`

```
total = venta + iva
```



4. Identificación de salida  
La salida será el total de la venta del producto

5. Fin

### Programa en C *Ver código 1iva.c anexo en disco*

- Inicio del programa  

```
#include <stdio.h>
int main(){
```
- Identificación de entrada:  
 Declarar las variables a utilizar en el programa  

```
float venta, iva, total;
```

 Solicitar el precio de venta del producto al usuario:  

```
scanf("%f", &venta);
```
- Realizar cálculo:  

```
iva = venta * 0.16;
total = venta + iva;
```
- Identificación de salida  
 Imprimir el resultado del total de la venta del producto  

```
printf("Total: %f", total);
```
- Fin  

```
}
```

## Programas

Realice los siguientes ejercicios:

- Escribir un programa que resuelva la siguiente expresión:

$$(x + y)^2(a - b)$$

Entrada:

Valor de las variables:  $x$ ,  $y$ ,  $a$  y  $b$ .

Salida:

Suponga los valores de entrada  $x=3$ ,  $y=2$ ,  $a=7$  y  $b=4$ , la salida sería:  
`res=75`

Procedimiento:

- 1) Declarar  $x$ ,  $y$ ,  $a$ ,  $b$  y otra variable para guardar el resultado (llamémosla `res`)
- 2) Solicitar con la función `scanf` los valores para las variables  $x$ ,  $y$ ,  $a$  y  $b$ .
- 3) Codificar la expresión solicitada y guardar resultado en `res`  
 Considere que  $z^2 = z * z$
- 4) Imprimir `res` con la función `printf`

2. Escribir un programa para convertir una medida dada en pies a sus equivalentes en: yardas, pulgadas, centímetros y metros (1 pie= 12 pulgadas, 1 yarda= 3 pies, 1 pulgada = 2.54 cm, 1 m = 100 cm). Leer el número de pies e imprimir el número de yardas, pies, pulgadas, centímetros y metros.

Entrada:

Valor de la variable: `pies`

Salida:

Suponga el valor de entrada `pies=6`, la salida sería: `pulgadas= 72`,  
`yarda=2`, `centímetros=182.88` y `metros=1.82`

Procedimiento:

- 1) Declarar `pies`, `pulgadas`, `yardas`, `centímetros` y `metros`.
- 2) Solicitar con la función `scanf` el valor para la variable `pies`.
- 3) Escriba el código para realizar la operación de conversión a pulgadas, guardando el resultado en la variable `pulgadas`. Recuerde que en el operador de asignación la variable de la izquierda guarda el valor dado por la expresión en el lado derecho.
- 4) Escriba el código para realizar la operación de conversión a yarda, guardando el resultado en la variable `yardas`
- 5) Escriba el código para realizar la operación de conversión a centímetros, guardando el resultado en la variable `centímetros`
- 6) Escriba el código para realizar la operación de conversión a metros, guardando el resultado en la variable `metros`
- 7) Imprimir con la función `printf` las variables `pulgadas`, `yardas`, `centímetros` y `metros`

3. Calcular el volumen y área de un cono. El usuario ingresará el radio ( $r$ ) y la altura ( $h$ ) el programa calculará el área [en una sola línea de código] y el volumen [en otra línea]. Para ello requerirá obtener el lado ( $g$ ) y usar  $\pi$  como una constante.

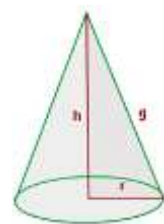


Figura 1.2. Cono

$$V = \frac{1}{3} \pi r^2 h$$

$$A = \text{areaLateral} + \text{areaDeBase}$$

$$A = 2\pi r \frac{g}{2} + \pi r^2$$

Entrada:

Valor de las variables: radio y altura.

Salida:

Suponga los valores de entrada radio=5 y altura=12, la salida sería: área=282.744 y volumen=314.16

Procedimiento:

- 1) Determine el tipo de dato necesario y declare las variables: radio, altura, lado, área, volumen
- 2) Declare como una constante  $\pi$
- 3) Solicitar con la función `scanf` el valor de las variables radio y altura.
- 4) Con Teorema de Pitágoras obtenga la fórmula de para obtener el lado; escriba la fórmula:  

$$g =$$
- 5) Escriba el código que permita calcular el volumen; recuerde observar la precedencia de operadores. Evite hacer uso de paréntesis innecesarios.
- 6) Escriba el código que permita calcular el lado g y el volumen; recuerde nuevamente observar la precedencia de operadores.
- 7) Imprimir el volumen y pareo con la función `printf`

4. Escribir un programa que convierte un número dado de segundos en su equivalente de horas, minutos y segundos.

Entrada:

Valor de la variable: segundos.

Salida:

Suponga el valor de entrada segundos\_totales=9010, las salidas serían: horas=2, minutos:30 y segundos:10

Procedimiento:

- 1) Determine el tipo de dato necesario y declare las variables: segundos\_totales, horas, minutos y segundos
- 2) Solicitar con la función `scanf` el valor de la variable segundos\_totales.
- 3) Usando operador ternario ( ? : ) verifique si la división por 60 del número de segundos\_totales es menor a 60
  - a. Si es menor a 60: el cociente es el número de minutos y el residuo el número de segundo (el residuo lo obtiene con el operador de módulo %)
  - b. Si es mayor o igual a 60, divida nuevamente por 60 para obtener las horas. Determine como obtener los minutos y segundo bajo esta circunstancia.
- 4) Imprima las horas, minutos y segundos con una sola función `printf`

5. Una temperatura Celsius (centígrados) puede ser convertida a una temperatura en grados Fahrenheit con la siguiente fórmula:

$$F = \frac{9}{5}C + 32$$

Escribir un programa que lea la temperatura en grados Celsius y convierta a Fahrenheit y viceversa. El usuario ingresará el valor de los grados y una letra: la cual indica que tipo de grados está ingresando, dicha letra puede ser una 'c' si los grados ingresados son Celsius o una 'f' si los grados ingresados son Fahrenheit.

Por ejemplo puede ingresar: 25f

**Entrada:**

Valor de las variables: `grados` y `letra`.

**Salida:**

Suponga el valor de entrada `18c`, equivalente a  $18^{\circ}\text{C}$  cuya la salida sería:  $99.8^{\circ}\text{F}$

**Procedimiento:**

- 1) Obtenga la fórmula para convertir de grados Fahrenheit a Celsius
- 2) Determine el tipo de dato necesario para `grados`
- 3) Determine el tipo de dato necesario para `letra`
- 4) Solicitar con una sola función `scanf` el valor de ambas variables
- 5) Usando operador ternario ( `? :` ) verifique si la letra es `f` o `c`
  - a. Si es una `c` use la fórmula mostrada para obtener los grados Fahrenheit
  - b. Si es una `f` use la fórmula obtenida para calcular los grados Celsius
- 6) Imprima los grados calculados con la función `printf`

### Problema de Aplicación

Haciendo uso de variables y operadores (no usar sentencias de control). Resuelva el siguiente problema:

**Nivel en pozos.** Para la medición de nivel en pozos profundos de agua, se establece el uso de transmisores de presión diferencial sumergible. Éste mide la presión hidrostática que ejerce la columna de agua en el sensor que proporciona una salida de voltaje directamente proporcional a la presión. Se requiere el diseño e implementación de un programa que obtenga el volumen de agua o Diesel (según sea el líquido contenido) de un pozo de forma cilíndrica de  $d$  metros de diámetro (el usuario da el valor de  $d$ ). La altura del agua se calcula con la siguiente fórmula:

$$h = p / (\rho * g)$$

tal que,

$p$  = presión hidrostática [bar] (dada por el sensor, el usuario ingresa el valor)

$\rho$  = densidad del fluido [ $\text{kg} / \text{m}^3$ ]

$g$  = aceleración gravitacional o fuerza [ $\text{m} / \text{s}^2$ ]

$h$  = altura de la columna de líquido [m]

donde  $1 \text{ bar}$  corresponde aproximadamente a  $10 \text{ metros}$  de agua dada su densidad de fluido de  $1000 \text{ kg} / \text{m}^3$ , mientras que el Diesel tiene una densidad de fluido de  $820 \text{ kg} / \text{m}^3$ .

### Propuesta de Solución.

- Diseñe con un diagrama de flujo la solución
- Considere que tiene dos posibles líquidos que el usuario debe elegir, para distinguir cuál fórmula aplicar use el operador condicional
- Cuide la precedencia de operadores
- Usará la fórmula del volumen de un cilindro
- Guíe al usuario a través de instrucciones impresas en pantalla

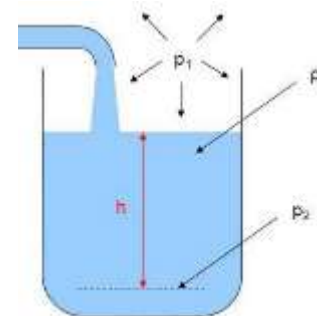


Figura 1.3. Pozo con variables para presión hidrostática

### Proyectos (opcionales).

Escriba un programa que calcule la resistencia de cierto voltaje y cierta corriente.  $R=V/I$



Codificar un programa que convierta una cantidad de pesos a su equivalente en monedas de \$10, \$5, \$2 y \$1.

Codificar un programa que obtenga el monto a pagar de tenencia con refrendo, según el valor del carro. Tomando en cuenta que al exceder determinado monto no se tiene subsidio.

### Evaluación de la Práctica.

Sección	Elemento a Evaluar
<b>Pre-reporte</b>	Comparación de datos y operadores
<b>Ejercicio 1</b>	Escritura básica de un programa, <code>printf</code> , <code>scanf</code> y uso de variables
<b>Ejercicio 2</b>	Identificación de tipo de datos de variables y operaciones básicas
<b>Ejercicio 3</b>	Uso de operadores, precedencia de operadores y constantes
<b>Ejercicio 4</b>	Uso de operador ternario y módulo
<b>Ejercicio 5</b>	Uso mixto de la función <code>scanf</code> , el operador ternario y el uso de caracteres
<b>Problema</b>	Interpretación del problema, diseño de solución, uso de operadores, manejo de precedencia de operadores, uso de las funciones <code>printf</code> y <code>scanf</code>

### Bibliografía.

Cairo Osvaldo, Metodología de la programación, Ed. Alfaomega 3ª Impresión, México 2005, ISBN: 970-15-0057-01, Págs: 3- 476.

Joyanes Aguilar Luis, Fundamentos de Programación, Ed. Mc Graw Hill, 4ª Impresión, España, 2008, ISBN: 8448161114, Págs: 47-73, 74-75, 76-101, 113-141 y 151 -534.

Marcelo Villalobos Ricardo, Fundamentos de programación C++ más de 100 algoritmos codificados, Ed. MACRO, 1ª Impresión, Lima, Perú, 2008, ISBN: 978-603-4007-99-4.

**NOTA:** Presentar el reporte en un documento (PDF o DOC), el código fuente y la impresión de la pantalla de ejecución.

**"Puedes apoyarte de tus compañeros y profesor para aclarar tus dudas"**