

11074 - Programación I

División Computación - Departamento de Ciencias Básicas



CONCEPTOS GENERALES C

Conceptos Generales

Organización del curso

TEORIA



Clases con contenidos teóricos: Vídeos en “Contenidos” (Aula Virtual).



Conceptos: Apuntes teóricos de la asignatura en “Contenidos”.

PRÁCTICA



Ejercicios: Enunciados en “Trabajos Prácticos” (Aula Virtual).

CONSULTAS



Presencial: Martes 9-12 y 18-21 hs en Luján. Jueves 18:00-21:00 Chivilcoy






Zoom: Miércoles 14-17 y Jueves 19-22 hs.






Servidor Discord: Programación I. En cualquier momento de la semana.

Evaluación del curso

- **HOMEWORKS**
-  **Github:** Ejercicios entregables con autocorrección mediante plataforma Github. (**Es necesario aprobarlos**).
- **PARCIAL**
-  **1º Parcial Presencial:** Miércoles 12/10 14 hs.
-  **2º Parcial Presencial:** Miércoles 23/11 14 hs.

Semana 1 – Material Relacionado

- **TEORIA**
-  **Teórica 1** - Introducción a C y Equivalencias con Python.
-  **Apunte** **Cap. 1** - Conceptos básicos.
- **PRÁCTICA**
-  **TP I** - Introducción a C.
-
- **ENTREGABLES**
- *No posee.*

Repaso de Conceptos Básicos

Conceptos Básicos

SOFTWARE

Conjunto de instrucciones y datos que definen el comportamiento de un programa de computadora.

PROGRAMACIÓN

La actividad de crear software.

CÓDIGO

FUENTE

Código escrito en un lenguaje de programación que, una vez compilado o interpretado por la computadora, genera un programa ejecutable.

INSTRUCCIÓN

Acción concreta que la máquina sabe ejecutar.

Conceptos Básicos

ALGORITMO

“Receta” o conjunto de pasos a seguir para resolver una tarea en particular.

LENGUAJE DE PROGRAMACIÓN

“Idioma” en el que se escriben los algoritmos para que la máquina pueda interpretarlos o compilarlos. Cada lenguaje tiene su propia **sintaxis**.

PARADIGMA DE PROGRAMACIÓN

Filosofía o ideales que definen cómo debe codificarse correctamente el software.

PROGRAMACIÓN ESTRUCTURADA/PROCEDURAL

Paradigma de programación que sostiene que un programa es un conjunto de algoritmos que se ejecutan para resolver un problema o brindar un servicio.

Conceptos Básicos

PROGRAMACIÓN

ESTRUCTURADA/PROCEDURAL

Paradigma de programación que sostiene que un programa es un conjunto de algoritmos que se ejecutan para resolver un problema o brindar un servicio.



¡El que vamos a utilizar en la asignatura!

Tipos de Lenguajes

Una característica importante de los lenguajes de programación es que pueden ser:

COMPILADOS

El código fuente es transformado a un lenguaje comprensible por la máquina para dar como resultado un programa ejecutable.

... o bien ...

INTERPRETADOS

El código fuente no se transforma, sino que es interpretado en tiempo real (por un *intérprete*) para poder ejecutarse.

Compiladores



El programador escribe el *código fuente* del programa utilizando un lenguaje de programación en particular.

El código fuente es analizado por un programa *compilador*, que se encarga de traducirlo a instrucciones legibles por la máquina.

Si el compilador no detecta errores, se genera una versión ejecutable (o compilada) del programa.

Intérpretes



El programador escribe el *código fuente* del programa utilizando un lenguaje de programación en particular.



El código fuente *no* se compila, sino que es interpretado en tiempo real por un *intérprete*.



El código fuente no se transforma para ser ejecutado, sino que es ejecutado por intermediación del intérprete.

Ejemplos de Lenguajes

COMPILADOS

- Ada.
- Basic.
- Pascal.
- C++, C#, **C**
- Java.

INTERPRETADOS

- JavaScript.
- PHP.
- Smalltalk.
- Python.
- Lisp.



¡El que vamos a utilizar en la asignatura!

C

CARACTERÍSTICAS GENERALES

- El desarrollo de programas en un lenguaje requiere “palabras reservadas” (keywords) que conforman el “vocabulario” capaz de ser entendido por ese lenguaje. Son “reservadas” en el sentido de que el programador no puede asignarles otros significados (por ejemplo, en C la palabra reservada float indica un tipo de dato de coma flotante –real– y sólo se la puede emplear con esa connotación). Por su parte, **C tiene pocas palabras reservadas**: auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, restrict, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while.

C

CARACTERÍSTICAS GENERALES

Estructura de un programa en C

Un programa sencillo escrito en C puede presentar, entre otras, las siguientes partes:

- Directivas al preprocesador.
- Prototipos de funciones.
- La función `main()`.
- Las definiciones de las demás funciones elaboradas por el programador.

C

CARACTERÍSTICAS GENERALES

***Directivas al preprocesador**

El preprocesamiento es una etapa previa a la compilación, en la que se toma el código escrito por el programador y se lo transforma en un nuevo código, más conveniente para el compilador. Esta traducción se realiza de acuerdo con palabras especiales que el programador inserta en su código, denominadas directivas al preprocesador. Éstas tienen la siguiente forma:

`#<nombre de la directiva>`

Por ejemplo:

`#include <stdio.h>`

`#define MAX 30`

C

CARACTERÍSTICAS GENERALES

La primera indica al preprocesador que debe incluir el archivo de cabecera (header) **stdio.h** mientras que la segunda directiva indica al preprocesador que toda vez que encuentre la palabra MAX en el código fuente, la reemplace por el número 30.

***Prototipos de funciones**

Cuando se escribe un programa, si éste no es demasiado sencillo, en general se lo divide en una función principal y otras auxiliares. Durante el proceso de compilación, se requiere que cada función esté definida antes de cualquier llamada a ella, ya sea con su implementación completa o sólo con su nombre, parámetros recibidos y tipo de dato retornado, lo que se conoce como el prototipo de la función.

CARACTERÍSTICAS GENERALES

Como ejemplo, tenemos:

```
float f1(int, char); /* prototipo 1*/  
char f2(float, int, float); /* prototipo 2*/  
void f3(); /* prototipo 3*/  
int main() /* programa principal */  
{  
...  
}
```

Acá se observa un programa C con su función `main()` y, antes, la declaración de tres prototipos de funciones, `f1()`, `f2()` y `f3()`. Cada declaración comienza indicando el tipo de dato retornado por la función, luego su nombre y, entre paréntesis, la lista de tipos de parámetros.

C

CARACTERÍSTICAS GENERALES

***La función main()**

Esta función es un tanto “especial”, en cuanto que es el “punto de partida” por el cual comienza la ejecución del programa. A partir de aquí podrán invocarse otras funciones. Además, como parte del cuerpo de la función main(), se pueden encontrar:

- **Declaración de constantes** (por lo general estas declaraciones se hacen fuera del cuerpo del main() para que sean accesibles a otras funciones).
- **Declaración de tipos** (estas declaraciones suelen hacerse fuera del cuerpo del main() para que sean accesibles a otras funciones).
- **Declaración de variables** (son las variables locales del main()).
- **Sentencias** (instrucciones) que conforman la lógica de la función main().

CARACTERÍSTICAS GENERALES

***Declaración de constantes**

Muchas veces puede ser útil emplear en varias oportunidades dentro de un programa un valor numérico o un texto determinado. Para evitar tener que definirlo en cada punto en que se utiliza, emplearemos una referencia única en todo el programa, a la que llamaremos constante. Las constantes son inalterables (de ahí su nombre), en general se las nombra usando mayúsculas y se declaran como se muestra a continuación:

```
const <tipo de dato> <nombre de la constante> = <valor>;
```

C

CARACTERÍSTICAS GENERALES

Por ejemplo:

```
const float PI = 3.14;  
const char *MENSAJE = "Buenos días!";
```

También podríamos declarar una constante utilizando un tipo de directivas al procesador:

```
#define PI 3.14
```

En este caso, cada vez que el preprocesador reconozca la palabra PI en el código, la reemplazará por el número 3,14.

CARACTERÍSTICAS GENERALES

***Declaración de tipos y variables**

Los diferentes objetos de información con los que trabaja un programa C se conocen en conjunto, como datos. **Todos los datos tienen un tipo asociado.** Un dato puede ser un simple carácter, un valor entero o un número decimal, entre otros.

Lenguajes como C o Pascal son conocidos como **fuertemente tipados** (strongly-typed), en cuanto a que es obligatorio para el programador asignar un tipo determinado a cada dato procesado.

La asignación de tipos a los datos tiene dos objetivos principales:

C

CARACTERÍSTICAS GENERALES

- Detectar errores de operaciones en programas (por ejemplo, si una operación matemática requiere números enteros, el uso de números decimales sería un error que, eventualmente, podría detectar el compilador).
- Determinar cómo ejecutar las operaciones.

En C, una variable se declara indicando el tipo de dato que aloja y su nombre:

`<tipo de dato> <nombre de la variable>;`

Por ejemplo:

```
int numero_empleado; /* número de empleado */
```

```
float horas; /* horas trabajadas */
```

```
char apellidos[30]; /* apellidos del empleado */
```

C

CARACTERÍSTICAS GENERALES

Todas las variables de un programa C deben declararse antes de utilizarse, y es una buena práctica de programación utilizar nombres representativos para ese fin que sugieran lo que las variables representan (esto hace que el programa sea más legible y fácil de comprender).

También es buena práctica incluir comentarios breves que indiquen cómo se utiliza la variable.

El **tipo de un dato** determina la naturaleza del **conjunto de valores** que puede tomar una variable y el **conjunto de operaciones** permitidas sobre dicha variable.

CARACTERÍSTICAS GENERALES

Los tipos de datos básicos del lenguaje C son los siguientes:

<i>Tipo</i>	<i>Tamaño* (bytes)</i>	<i>Descripción del tipo</i>
char	1	Carácter o entero de un byte
int	4	Entero (con signo)
signed int	4	Entero (con signo)
unsigned int	4	Entero (sin signo)
short	2	Entero corto (con signo)
signed short	2	Entero corto (con signo)
unsigned short	2	Entero corto (sin signo)
long	8	Entero largo (con signo)
signed long	8	Entero largo (con signo)
unsigned long	8	Entero largo (sin signo)
float	4	Decimal de simple precisión
double	8	Decimal de doble precisión

CARACTERÍSTICAS GENERALES

Los tamaños de los tipos de datos dependen de la arquitectura (procesador). En este caso se considera una arquitectura de 32 bits.

La función `sizeof()` recibe como único parámetro el nombre de una variable, o el de un tipo de datos, y devuelve su tamaño en bytes.

Ejercicio: modificar el siguiente código para que muestre los tamaños de los tipos de datos con la función `sizeof`: <http://tpcg.io/FnrXP5>

ASCII :American Standard Code for Information Interchange. Estándar para codificación de caracteres, creado en 1963, de uso muy frecuente en el manejo de texto en computadoras.

C

INFORMACIÓN

- Instalar **gcc** en windows:
<https://www.youtube.com/watch?v=nkzkiLODxEU>
- **Visual Studio Code**. IDE ya conocido, que también podemos usar para escribir código en C.
- Otra opción: descargar e instalar **Geany** - Geany es un editor de texto de programación potente, estable y ligero que proporciona muchas características útiles sin atascar su flujo de trabajo. Se ejecuta en Linux, Windows y macOS, está traducido a más de 40 idiomas y tiene soporte incorporado para más de 50 lenguajes de programación. [Descargar Geany 1.38.0 »](#)

C

INFORMACIÓN

- **Git** - Herramienta de control de versiones que utilizaremos para el seguimiento de nuestro código
- **GitHub** - Implementación de Git en la nube, para tener repositorios en la Web.

// Enseñar no es transferir conocimientos, sino crear las posibilidades de su construcción; quien enseña aprende al enseñar y quien aprende enseña al aprender”



Paulo Freire