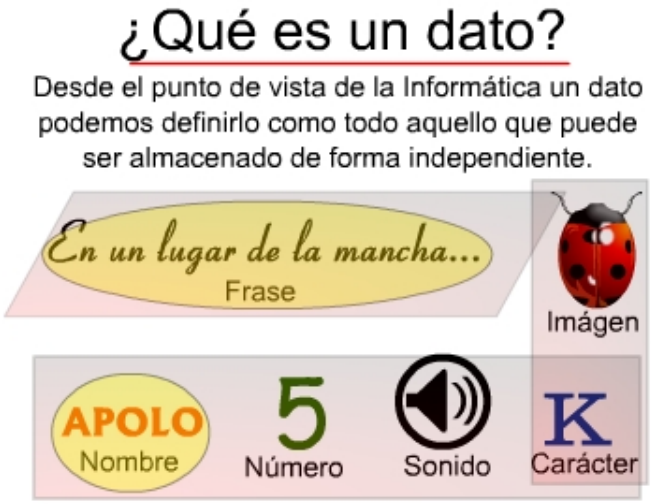
Índice de Contenidos

En esta unidad veremos una serie de conceptos básicos para la programación, tales como datos, tipos y estructuras de datos. Estos conceptos deben ser dominados por el programador a fin de incorporarse con ciertas garantías a un equipo de programación. Hay que especificar que los programas y aplicaciones informáticas trabajan con datos, que pueden ser almacenados para posteriormente ser utilizados en la obtención de información útil, empleada para adoptar las decisiones más adecuadas en cada situación.

2.1.- Estructuras Básicas de datos.

Son objetos de un programa todos los datos y resultados manipulados por las instrucciones de un programa.



Todo objeto tiene tres atributos:

- Nombre: con el que se identifica al objeto.
- Tipo: conjunto de valores que puede tomar.
- Valor: elemento del tipo que se le asigna.

Ejemplo: Nombre: sueldo
Tipo: entero
Valor: 70000

2.2.- Identificadores.

Son palabras creadas por el programador para dar nombre a los objetos y demás elementos que necesita declarar en un programa.

Existen unas normas generales para su empleo:

- Pueden estar constituidos por letras, dígitos y el carácter subrayado (_), aunque algunas herramientas permiten otros caracteres especiales.
- Deben comenzar por una letra y en algunos compiladores también por (_).
- No deben tener espacios en blanco.
- El número máximo de caracteres que se pueden emplear depende del compilador utilizado.

- El nombre asignado conviene que tenga relación con la información que contiene, pudiéndose emplear abreviaturas que sean significativas.

2.3.- Datos básicos.

Dato es toda información que se aporta a un programa. O también toda información característica propia de cualquier entidad.

Los programas procesan datos a fin de obtener resultados.

Como ya **vimos** en la unidad 1, podemos definir **dato** como un conjunto de símbolos que representan valores, hechos, objetos o ideas de forma adecuada para ser **tratados**. Incluso, en el ámbito de la informática, podemos definir dato de forma similar, como cualquier "objeto" manipulable por la computadora. **Un dato puede ser un carácter leído desde teclado, un número, la información almacenada en un CD, una foto almacenada en un fichero, una canción, el nombre de un alumno almacenado en la memoria del ordenador, etc.**

Los ordenadores son máquinas creadas para procesar automáticamente la información, y para ello esa información no se almacena ni representa al azar, sino que ha de **organizarse y estructurarse de forma adecuada para que pueda almacenarse, procesarse y recuperarse de la forma más eficiente posible**. Éste será el principal problema del que nos ocuparemos en esta unidad.

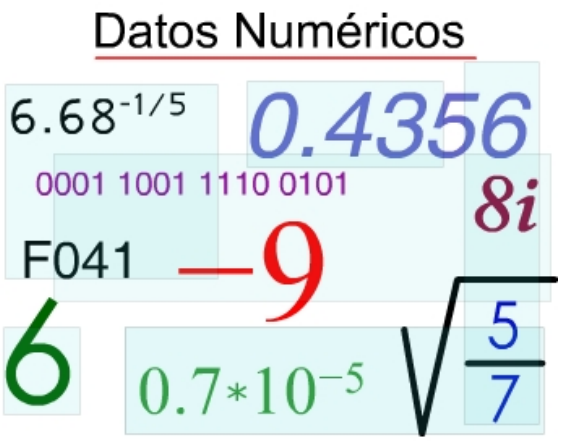
En esta unidad analizaremos:

- **Los tipos básicos de datos que incluyen la mayoría de los lenguajes de programación**, (aunque con diferencias significativas entre unos y otros) .
  Por ejemplo, la edad de un trabajador la podremos representar con un tipo básico, concretamente un tipo **entero**, ya que la edad es un valor numérico positivo y sin decimales (que es justo lo que define el tipo entero, como se verá más adelante).

La siguiente clasificación contempla las diversas formas de representación de la información para su almacenamiento y tratamiento.

2.3.1.- Numéricos.

Se utilizan para contener magnitudes y se clasifican en enteros y reales:



- Enteros: se emplean para representar números enteros cuyo rango o tamaño dependen del lenguaje y del ordenador utilizados. Los datos de este tipo se expresan mediante una serie de dígitos (de 0 a 9), pudiendo estar precedidos del signo (+ o -).

  Ejemplo: -82, +345

- Reales: se emplea para representar los números con parte decimal o los números muy grandes o muy pequeños que no pueden ser contenidos en un entero. Se pueden representar de dos formas:
  - ➥ Punto decimal: emplea los dígitos del 0 al 9 con su signo correspondiente y un punto para separar la parte entera de la decimal. Ejemplo: -82.75, +345.87
  - ➥ Científica o exponencial: Utiliza el formato "mantisaEcaracterística", donde:
    - Mantisa es un número real.
    - E representa la base decimal.
    - Característica es el exponente correspondiente a un número entero con su signo. Ejemplos: 2.5E3, -0.75E-2

2.3.2.- No numéricos.

- Carácter: Se emplea para representar un símbolo dentro de un código definido por el fabricante del ordenador, de tal forma que cada uno de ellos se corresponde con un número entero sin signo según un determinado código.

La representación interna depende del código utilizado. Los códigos mas empleados son los que utilizan 8 bits. Ejemplo: 'A' en ASCII es 65 y en binario es 01000001.
En algunos lenguajes (como en COBOL) se incluye como tipo de datos básico el tipo alfanumérico, que es una cadena de caracteres formada por un número determinado de caracteres y en otros lenguajes (como en C) se considera este tipo de dato como un vector de caracteres.

- Lógico: se emplea para representar dos valores opuestos, Verdadero o Falso, True o False, V o F, 1 o 0 . Internamente se considera 1 como verdadero y 0 como falso.

2.3.3.- Datos estructurados.

Se pueden considerar, de forma general, las siguientes características:

- Internos: son los que residen en la memoria principal del ordenador.
  - ➥ Estáticos: son aquellos cuyo tamaño queda definido en la compilación del programa y no se puede modificar durante la ejecución del mismo.
  - ➥ Dinámicos: son aquellos cuyo tamaño puede ser modificado durante la ejecución del programa.
    - Lineales: son los que pueden estar enlazados con un solo elemento anterior y un solo elemento posterior.
    - No lineales: son los que pueden enlazarse con más de un elemento anterior y más de un elemento posterior.
- Externos: son los que residen en un soporte externo a la memoria principal, es decir memoria auxiliar.
- Compuestos: son los formados por el programador en base a los tipos de datos básicos y derivados, pudiendo ser internos o externos.

2.3.4.- Constantes.

Son datos cuyo valor no cambia durante la ejecución del programa. Las constantes pueden ser:

- Enteras
- Reales
- Alfanuméricas (es un carácter válido encerrado entre apóstrofos ').
- Cadena de caracteres:

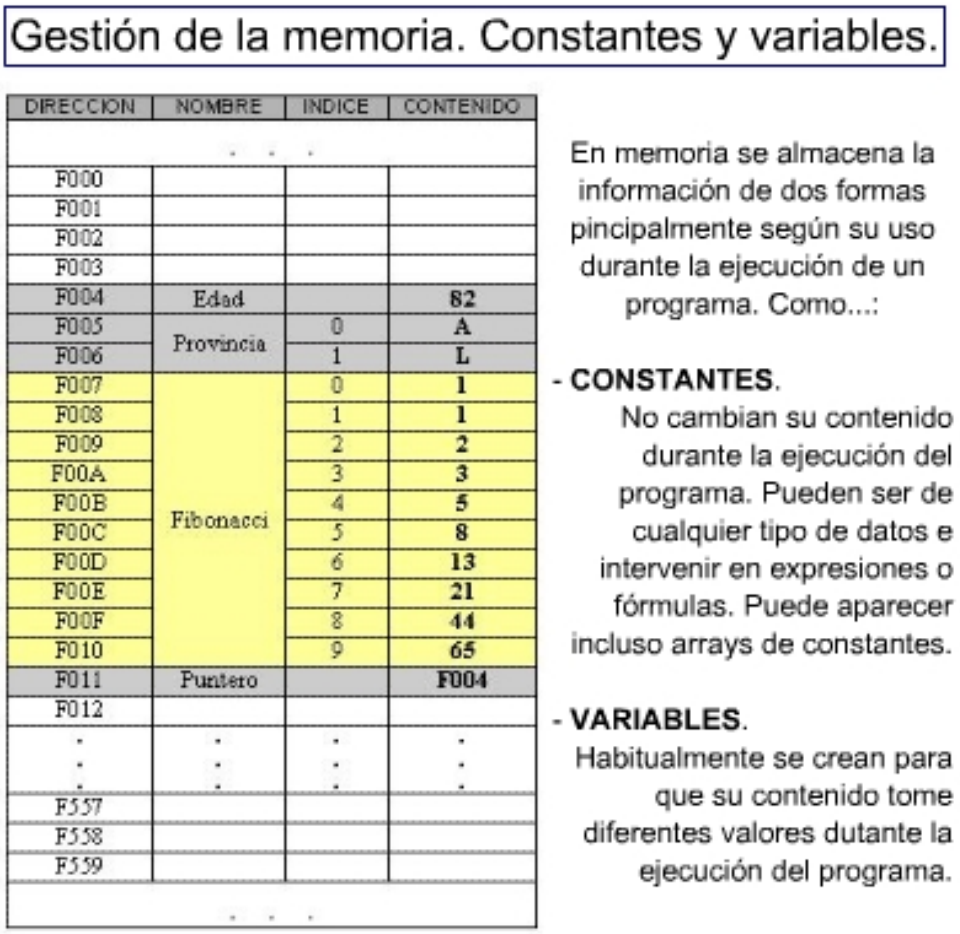
Se pueden expresar de dos formas:

- De forma explícita mediante su valor.
  Por ejemplo: 87.5 'A'
- Utilizando un identificador para definir la constante en memoria, asignándole un valor. El compilador asignará la memoria con el tamaño correspondiente al tipo de dato expresado por su valor.
  Por ejemplo: Pi = 3.1416, descuento = 10.

2.3.5.- Variables.

Son datos cuya información puede ser variable durante la ejecución del programa. Estos datos deben ser definidos con un identificador y un tipo de dato.

El identificador es elegido por el programador y permite referenciar la variable para su uso en el programa pudiéndose modificar su valor. El tipo de dato permite determinar el tamaño de la variable en memoria. Según el tipo de dato que almacenan las variables pueden ser:



- **Numéricas:** nombre-variable = valor
  Ejemplo: suma = 80

- **Alfanuméricas:** nombre-variable = 'comentario'
  Ejemplo: apellido = 'Pérez'
- **Lógicas:** nombre-variable = valor lógico
  Ejemplo: A = verdadero

Antes de utilizar una variable en el programa, esta debe contener un valor que puede ser asignado inicialmente o bien durante la ejecución del programa.

2.3.6.- Expresiones.

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales.

Cada expresión toma un valor que se determina tomando los valores de las variables y constantes implicadas y la ejecución de las operaciones existentes.

Por tanto una expresión es un conjunto de datos (operándoos) y operadores, con unas reglas especificas de construcción. Según sea el resultado que producen y los operadores que utilizan se clasifican en:

- Numéricas: son las que producen resultados de tipo numérico. Se construyen mediante operadores aritméticos.
  Ejemplo: suma + 5 2*Pi*R
- Alfanuméricos: son las que producen resultados alfanuméricos. Se construyen mediante operadores alfanuméricos.
  Ejemplo: "Juan" + "López"
- Lógicas: son las que producen resultados verdadero o falso. Se construyen mediante los operadores lógicos y/o relaciónales.
  Ejemplo: A > 0 AND B < 5

2.3.7.- Funciones.

Las operaciones que se requieren en los programas exigen en numerosas ocasiones, además de las operaciones aritméticas básicas, un número determinado de operadores espaciales que se denominan “funciones internas”, incorporadas o estándar.

Cada lenguaje de programación tiene sus propias funciones, entre las comunes y más utilizadas están las siguientes:

- Sqrt(x) raíz cuadrada de un número positivo
- abs(x) valor absoluto
- cos(x) coseno
- sin(x) seno
- ln(x)logaritmo neperiano
- log10(x) logaritmo decimal

2.3.8.- Operadores.

Los operadores son símbolos que sirven para conectar los datos haciendo diversas clases de operaciones. En función de las operaciones a realizar los operadores se clasifican según la siguiente tabla:

OPERADOR	SÍMBOLO	SIGNIFICADO
Paréntesis	( )	Paréntesis
Aritméticos	**, ^ * / div, \ mod + -	Potencia Producto División División entera Módulo (reto de división entera) Suma Resta
Alfanuméricos	+ -	Concatenación Concatenación eliminando espacios
Relaciónales	==, = < > < <= > >=	Igual a Distinto a Menor que Menor o igual que Mayor que Mayor o igual que
Lógicos	NOT, no AND, y OR, o	Negación Conjunción Disyunción

Nota: Sirva esta tabla de operadores como referencia general. En particular, cada lenguaje de programación utiliza su propia simbología para estos operadores. Por ejemplo en Java el operador lógico AND se representa mediante &&, el operador lógico mediante ||, el operador MOD se representa mediante %.

En las operaciones lógicas se determina su resultado por medio de las tablas de la verdad:

- **Negación:**

A	NOT A
F	V
V	F

- **Conjunción:**

A	B	A AND B
F	F	F
F	V	F
V	F	F
V	V	V

- **Disyunción:**

A	B	A OR B
F	F	F
F	V	V
V	F	V
V	V	V

2.3.8.1.- Orden de prioridad de los operadores.

Dentro de las expresiones hay que tener un orden de prioridad de los operadores que depende del lenguaje de programación utilizado, pero que de forma general se puede establecer de mayor a menor prioridad de la siguiente forma:

1. Paréntesis (Comenzando por los mas internos).
2. Signo.
3. Potencia.
4. Producto, división, módulo.
5. Suma, resta.
6. Concatenación.

7. Relaciónales.
8. Negación.
9. Conjunción.
10. Disyunción.

La evaluación de los operadores de igual orden se realiza siempre de izquierda a derecha.

2.3.9.- Constantes simbólicas.

Una constante simbólica es un nombre que sustituye una secuencia de caracteres. Los caracteres pueden representar una constante numérica, una constante de caracteres o una constante de cadena.
Cuando se compila un programa, cada aparición de una constante simbólica, es reemplazada por su correspondiente secuencia de caracteres.
Las constantes simbólicas se suelen definir al comienzo del programa.

Práctica propuesta a resolver por el alumno.

Práctica 2-1.

Decir cuál de los siguientes identificadores son válidos. Si no lo son explicar porqué:

a) Renta	b) Registro1	c) 1registro
d) Dos pulgadas	e) nombreapellidos	f) nombre_apellidos
g) 123nombre	h) C3P7	i) Nombre y Apellidos
j) impuesto$	k) bienvenido#5	

Práctica propuesta a resolver por el alumno.

Práctica 2-2.

Determinar cuáles de los siguientes valores numéricos son constantes válidas. Si es válida especificar de que tipo es.

a) 0.5	b) 9.3E-121registro	c) 0.8E+0.8
d) 'B'	e) 27,82	f) 1234
g) "nombre"	h) "apellidos	

Práctica propuesta a resolver por el alumno.

Práctica 2-3.

Supongamos que a, b, c son variables enteras que tienen asignadas los valores siguientes a = 8, b = 3, c =-5. Determinar el valor de cada una de las siguientes expresiones:

a) a + b + c
b) a mod b
c) a * b/c
d) a * (c mod b)
e) 2 * b + 3 * (a – c)
f) a * (b/c)
g) (a * c) mod b

Objetivos

Elementos básicos generales de un programa persigue que el alumno alcance los siguientes objetivos:

- Comprender la importancia de los datos en la programación de ordenadores, ya que todos los programas utilizan datos y operaciones sobre datos.
- Entender la necesidad de contar con diferentes tipos de datos para representar la información.
- Conocer y comprender cada uno de los tipos de datos básicos que existen.
- Asignar de forma correcta el tipo más adecuado para cada dato.
- Conocer las particularidades de cada tipo de dato.

Enlaces de interés

☞ Existen diferentes maneras de representar internamente los distintos tipos de datos, si quieres profundizar un poco más en este tema, visita el siguiente enlace, en la página 7 puedes encontrar la notación complemento a 2 para representar los números enteros:

Elementos de Programación.
http://www.conectados.org/doc/Apuntes/1_Elementos_de_Programacion/temai.pdf

☞ Al igual que con los números enteros, los números reales tienen diferentes representaciones internas, entre ellas está la representación interna en coma flotante, en la notación IEEE 754, de la que podrás obtener más información en el siguiente enlace:

Representación interna de los números reales. Coma flotante.
http://www.unalmed.edu.co/~daristiz/guias/fisica_computacional/semana_4/aritmetica.pdf

☞ Reciben el nombre de booleanos en honor del matemático inglés George Boole, que ideó una herramienta matemática denominada álgebra de Boole binaria, y que es la base matemática para la construcción de circuitos lógicos, y que permite aplicar la lógica matemática a la construcción de ordenadores. Para conocer algo más sobre Boole, sigue el siguiente enlace.

BOOLE
http://www-etsi2.ugr.es/alumnos/mlii/Boole.htm

☞ Para una lectura más detallada de los códigos de E/S puedes visitar la siguiente página:

Elementos_de_Programación.
http://www.conectados.org/doc/Apuntes/1_Elementos_de_Programacion/temai.pdf

☞ De entre los distintos códigos de E/S, hemos destacado dos por su uso mayoritario, ASCII y Unicode, puedes saber algo más de ellos en los siguientes enlaces:

Código ASCII
http://es.wikipedia.org/wiki/ASCII

UNICODE
http://www.unicode.org/

Glosario de términos

ASCII

American Standard Code for Information Interchange (Código Estadounidense Estándar para el Intercambio de Información) es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y otras lenguas occidentales. Define 128 códigos posibles, dividido en 4 grupos de 32 caracteres, (7 bits de información por código), aunque utiliza menos de la mitad, para caracteres de control, alfabéticos (no incluye minúsculas), numéricos y signos de puntuación. Su principal ventaja, aparte de constituir un estándar, consiste en la ordenación alfabética de los códigos.

Clase

Una clase es un tipo definido por el usuario que determina las estructuras de datos y las operaciones asociadas con ese tipo. Una clase es una colección de objetos similares y un objeto es una instancia de una definición de una clase.

Código de E/S

Los códigos de E/S o códigos externos asocian a cada carácter (alfabético, numérico o especial) una determinada combinación de bits.

Constante

Una constante es una zona de memoria del ordenador donde se almacena un dato cuyo valor no se puede modificar. Una vez que una constante se declara no se puede modificar dentro del programa.

Dato

En la vida común se usa como sinónimo de información, pero en informática el término dato es más restrictivo que el de información. Se define como un conjunto de símbolos que representan valores, hechos, objetos o ideas de forma adecuada para ser objeto de tratamiento.

Instancias de clase

Cada vez que se construye un objeto de una clase, se crea una instancia de esa clase. En general, los términos objetos e instancias de una clase se pueden utilizar indistintamente.

Interface

Interfaz. Conexión e interacción entre hardware, software y el usuario. El diseño y construcción de interfaces constituye una parte principal del trabajo de los ingenieros, programadores y consultores. Los usuarios "conversan" con el software. El software "conversa" con el hardware y otro software. El hardware "conversa" con otro hardware. Todo este "diálogo" no es más que el uso de interfaces.

Mantisa

Cifras decimales significativas en la representación en coma flotante de los números reales. Nº = mantisa * $Base^{exponente}$ Ejemplo: 5000 = .5 *10^4

Memoria RAM

RAM es el acrónimo inglés de *Random-Access Memory* (memoria de acceso aleatorio). Memoria en la que se puede tanto leer como escribir. Se trata de una memoria volátil, es decir, pierde su contenido al desconectar la energía eléctrica. Se utiliza normalmente como memoria temporal para almacenar resultados intermedios y datos similares no permanentes.

Número Binario

Número representado en el sistema de numeración binario, sistema de numeración en el que todas las cantidades se representan utilizando como base el número dos, con lo que disponemos únicamente de las cifras: cero y uno ('0' y '1').

Operador

Los operadores son elementos que relacionan de diversas formas, los valores contenidos en una o más variables o constantes. Como resultado de esta relación se obtiene un valor, que puede ser lógico o numérico. A los elementos que se relacionan (variables o constantes) se les conoce como OPERANDOS y los elementos relacionadores se les llama OPERADORES.

Overflow

Es una situación de «overflow» aquélla en la que hemos sobre pasado las capacidades de que disponemos cuando, por ejemplo, el ordenador nos indica que no dispone de memoria suficiente para responder a un requerimiento concreto que le hemos pedido. Un ejemplo más gráfico es el de una elemental calculadora de 8 dígitos que no puede mostrar en su pantalla el resultado de una operación en cientos de millones.

Sistema Operativo

Un sistema operativo (SO) es un conjunto de programas o software para permitir comunicarse el usuario con un ordenador y gestionar sus recursos de manera cómoda y eficiente. Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos.

Tabla Verdad (o tabla de verdad)

Dada una expresión lógica, compuesta por variables y operadores lógicos, la tabla verdad de esa expresión indica cual es el resultado de evaluar la expresión para cada una de las posibles combinaciones de valores que pueden darse para las variables que forman parte de la expresión. Es una tabla que para cada posible combinación de valores (verdadero o falso) de las variables que forman la expresión, asigna el resultado de evaluar esa expresión sustituyendo las variables por los valores de esa combinación. Evidentemente, el resultado de evaluar la expresión será a su vez Verdadero o Falso.

Tipo de Dato

Podemos definir un tipo de dato a partir de los valores permitidos y las operaciones que se puedan llevar a cabo sobre estos valores.

Underflow

Es un estado de «underflow» aquel que resulta de la pretensión de operar con cantidades demasiado pequeñas.

Unicode

Unicode es una norma de codificación de caracteres. Su objetivo es asignar a cada posible carácter de cada posible lenguaje un número y nombre único, a diferencia de la mayor parte de las normas de codificación, que sólo definen los caracteres necesarios para un idioma o zona geográfica.

Variable

Una variable corresponde a un área reservada en la memoria principal del ordenador, longitud fija o variable, en donde el programa puede almacenar valores