universidad cenfotec
La U de la informática

Fundamentos de Programación

Variables, tipos de datos, valores y estándares.

Objetivos

universidad cenfotec
La U de la informática

- Aprender la diferencia entre variable, tipo de dato y valor.
- Aprender el estándar para definir variables en Java.
- Aprender los diferentes tipos primitivos en Java.
- Representar en el lenguaje de programación Java cualquier algoritmo con estructuras secuenciales (operaciones básicas).

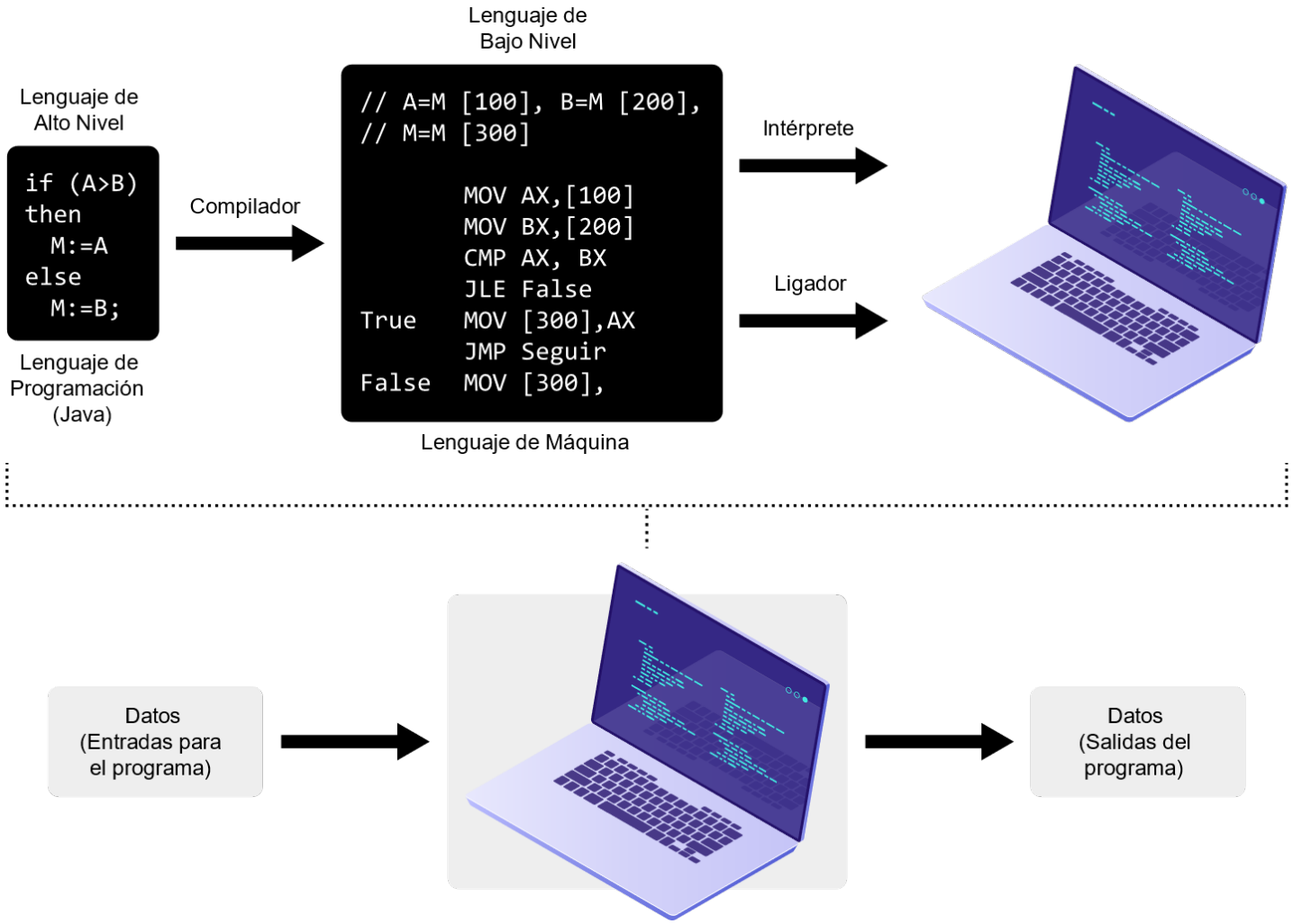
Repaso

universidad cenfotec
La U de la informática

Es importante que recuerde lo siguiente:

- Conceptos básicos de computación como compilador, intérprete y lenguaje de programación.
- Definición de algoritmo y programa.
- Los pasos necesarios para crear un programa estructurado.
- Conceptos básicos de Java.

Repaso: Conceptos básicos de computación



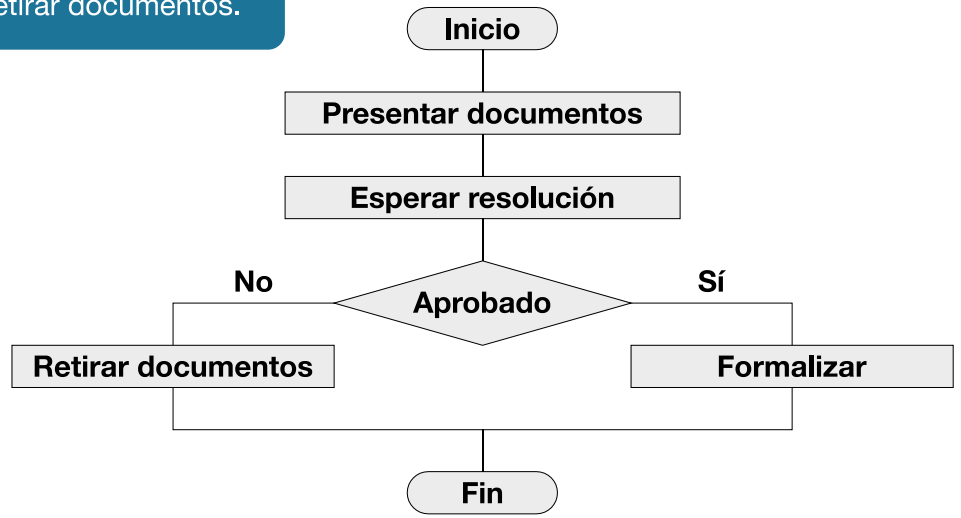
Repaso: Algoritmos

Un **algoritmo** es una secuencia de instrucciones que, realizados con fidelidad, darán como resultado una tarea realizada.

Los algoritmos son **finitos**, lo que significa que estas instrucciones tienen un inicio y un final.

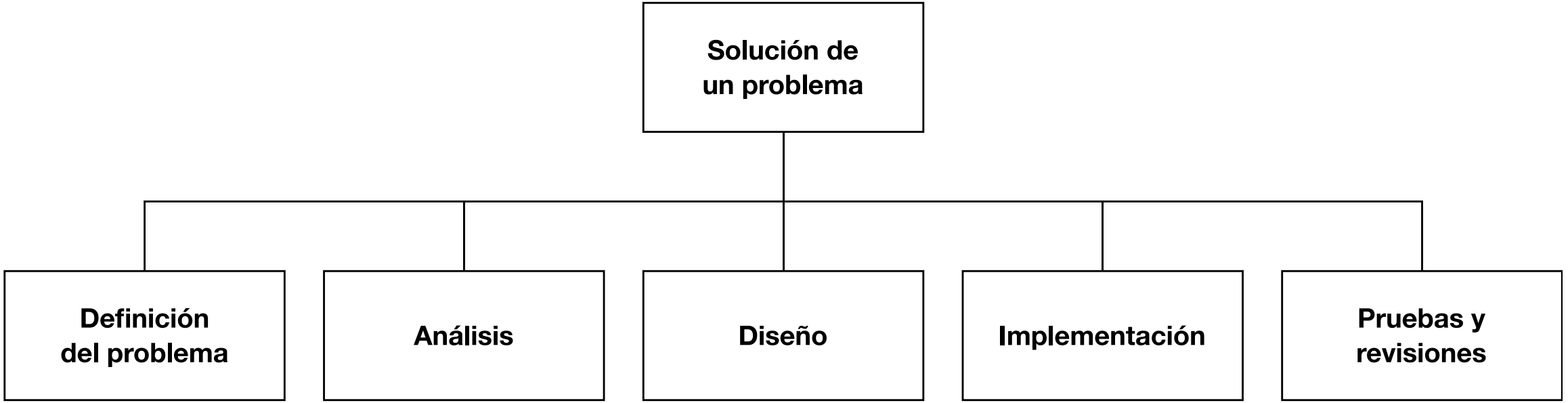
Solicitar un préstamo:

1. Presentar documentos.
2. Esperar resolución.
3. Aprobaron el préstamo:
   - **Sí:** Formalizar.
   - **No:** Retirar documentos.



Repaso: Pasos para crear un programa estructurado

universidad cenfotec
La U de la informática



DOCUMENTACIÓN

Repaso: Conceptos básicos de Java

universidad cenfotec
La U de la informática

```java
package Semana0;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintStream;

public class Intro {

    static BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    static PrintStream out = System.out;
    public static void main(String[] args) throws IOException {
        String nombre = in.readLine();
        out.println("Hola " + nombre);
    }
}
```

En el código puede ver los conceptos vistos anteriormente de **`package`**, **`import`**, **`class`**, **`main`**, **llaves**, entre otros.


Variables, tipos de datos, valores y estándares

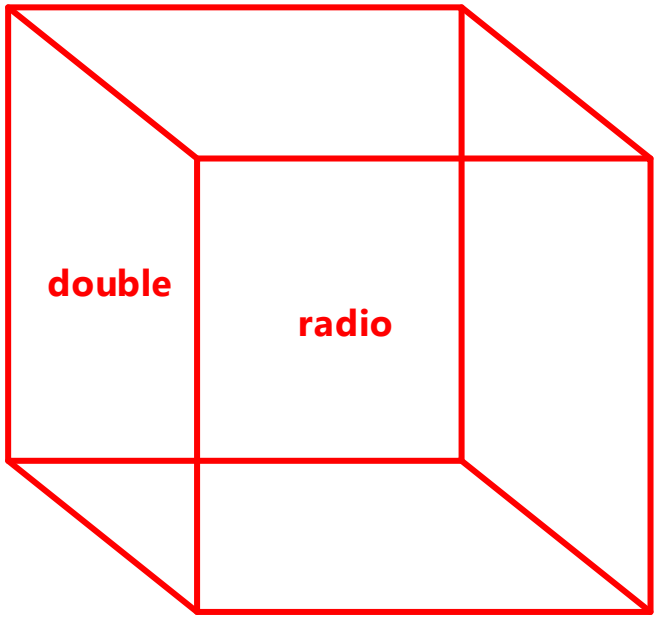
Tipos de datos simples en Java

universidad cenfotec
La U de la informática

A la hora de estructurar un **algoritmo** es posible que se requiera utilizar datos muy distintos, como números enteros, números decimales, letras, palabras, entre otros. Por ejemplo, para guardar la edad de una persona se requiere de un número entero, para su salario se requerirá de un número con decimales, para su grupo sanguíneo una letra y para su nombre una palabra.

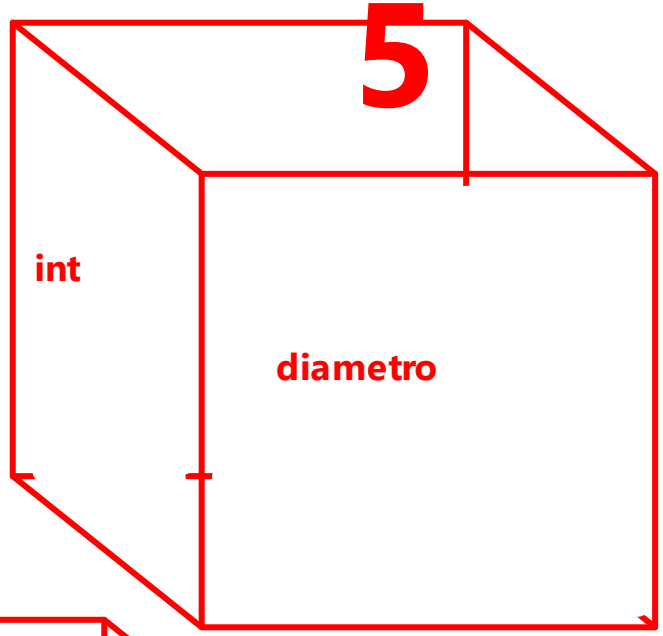
Cada uno de estos datos tiene una representación concreta dentro de Java, sobre los cuales se pueden ejecutar varias operaciones para interactuar y modificar sus valores.

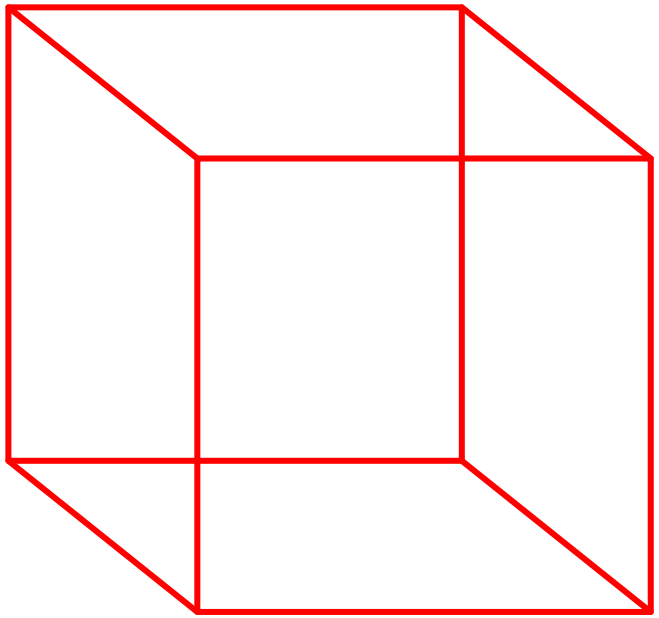
Estas representaciones llevan como nombre **tipos de datos**.

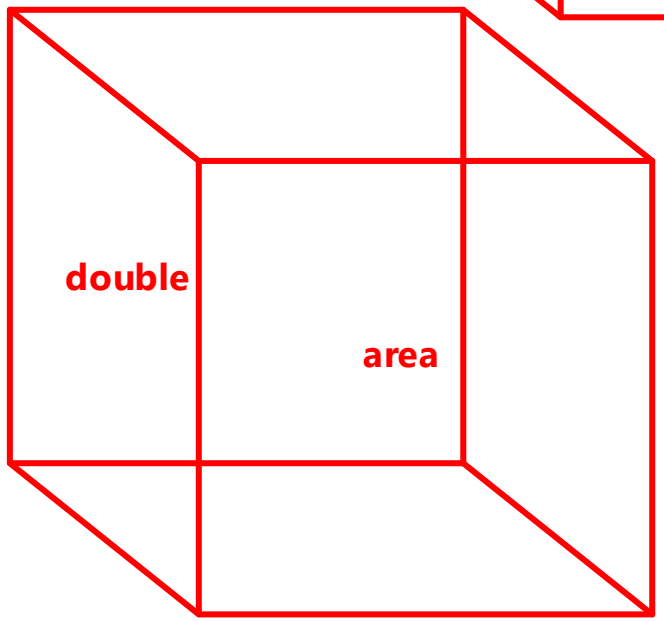


int (entero)

double
float







7

radio = diametro / 2
area = radio * radio * 3.141516

sout(area)

Se les solicita a ustedes que realicen una aplicación para calcular el área de un círculo, para lo cual reciben el valor del diametro del mismo.

diametro -> área del círculo

tomar el diametro y divirlo entre dos, con esto obtengo el radio
tomo el radio, lo elevo al cuadrado y lo múltiplo por PI para obtener el área

real son todos
los tipos con decimales

entero, no tienen decimales

entradas
diametro (real)

intermedia
radio (real)

salidas
area (real)

Tipos de datos en Java

Los diferentes **tipos de datos** en Java se pueden clasificar en:

int, long, double, short, char, boolean

- Tipos de datos **simples o primitivos**: como los enteros, reales, caracteres y de valor booleano.
- Tipos de datos **compuestos**: como los arreglos y matrices, los cuales serán estudiados posteriormente.
- Tipos de datos **definidos por el programador**: como las clases y sus relaciones, los cuales serán estudiados posteriormente.

String
Integer, Long, Double, Short, Character, Boolean

bit
byte
megabyte
gigabit
gigabyte

binary digit
bi it
bit

modem
modulador demodulador
modem

0
1

0 1 2 3 4 5 6 7 8 9

210

9 19

10 20

1 byte = 8 bits
B = 8b

000
001
010
011
100
101
110
111

3bit
0-7

3bit
2**3-1
8-1
7

00
01
02
03
04
05
06
07
08
09
10
11
12
..
19
20

7*10**1 + 5* 10**0

70 + 5
75

32

2**31-1

2**32-1

3bit -1
2bit
2**2-1
-3 .. 3

1 * 2**2 + 1 * 2**1 + 1* 2 **0

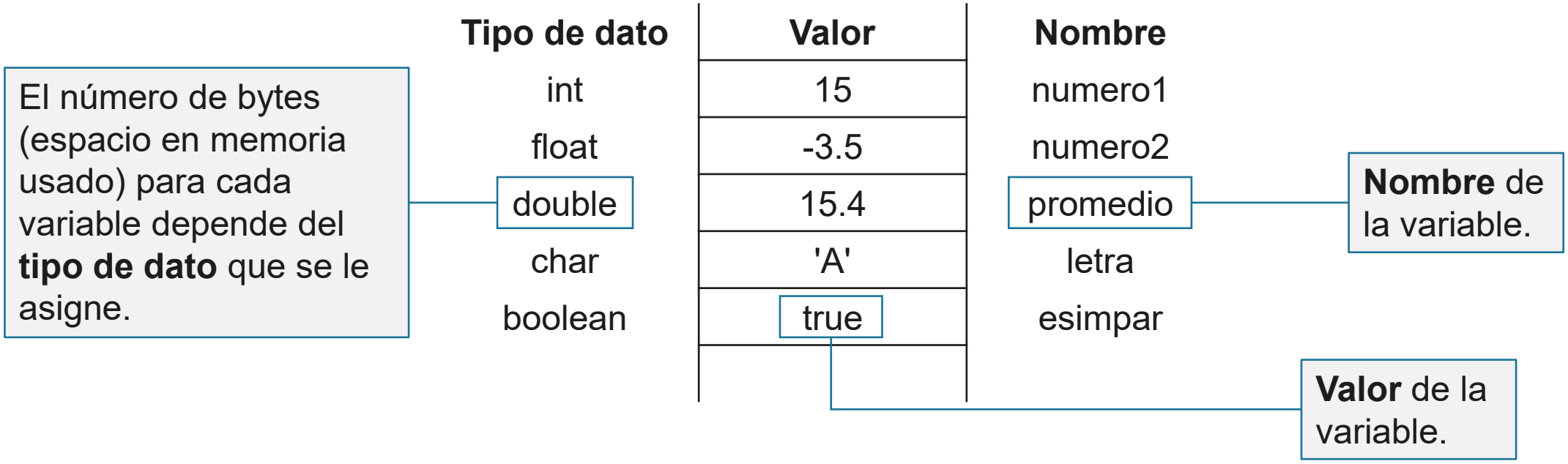
4+ 2+ 1

7

----------1,5----------

Variables, tipos de datos y valores

Una vez se define el tipo de dato necesario en un algoritmo, este se puede representar en un programa utilizando variables. Una **variable** es un espacio en memoria identificada por un nombre, que contiene un dato de cierto tipo (entero, real, carácter, etc.) y el contenido de la variable en un momento dado del tiempo es su **valor**.



Variables y tipos de datos simples en Java

universidad cenfotec
La U de la informática

En la siguiente tabla, en la primera columna se muestran algunos de los tipos de datos simples en Java, su descripción y ejemplos de declaración de variables con los respectivos tipos de datos:

Tipo	Descripción	Ejemplos

`int, long, short`	Números enteros.	`int numero1 = 155200;`
		`short numero2 = 120;`
		`long numero3 = 3538000;`

`float, double`	Números con decimales o de punto flotante.	`float numero4 = 23.43f;` `double numero5 = -223000.57;`
`char`	Caracteres, letras, dígitos o signos de puntuación de una única posición.	`char primeraLetra = 'A';` `char ultimaLetra = 'Z';`
`boolean`	Valores lógicos, verdadero o falso.	`boolean buenEstudiante;` `buenEstudiante = true;`

Declaración de una variable

Solo se hace una vez

Para poder utilizar una variable es necesario **declararla**. Lo cual consiste en indicarle a Java el tipo de dato y el nombre con el que se identificará la variable.

```
int numero;
```

esto es como el declare de Flowgorithm

Esto separa un espacio en memoria y le permite a Java utilizar la variable en futuras operaciones. Al declararla, se le puede dar un valor inicial, esto se conoce como **inicialización**.

```
int numero = 12;
```

Una vez declarada una variable, darle un valor en cualquier momento se conoce como **asignación** y si le asigna un nuevo valor este se sobre escribe y el valor anterior se pierde.

```
numero = 24;
```

Ejemplo de declaración de variables

universidad cenfotec
La U de la informática

```
public class Variables {

    public static void main(String[] args) {
        int numero1;
        float numero2;
        double promedio;
        char letra;
        boolean esImpar;
    }

}
```

Cuando se **declara** una variable se crea con un nombre y un tipo de dato específico. En memoria se reserva el espacio correspondiente a la variable según el tipo de dato.

En el ejemplo anterior la primera variable tiene como nombre **`numero1`** de tipo **`int`**, la segunda variable tiene como nombre **`numero2`** que es de tipo **`float`**, la tercera variable tiene como nombre **`promedio`** y es de tipo **`double`**, la cuarta variable es **`letra`** de tipo **`char`** y la quinta variable tiene como nombre **`esImpar`** y es de tipo **`boolean`**.

Ejemplo de declaración de variables

universidad cenfotec
La U de la informática

```java
public class Variables {

    public static void main(String[] args) {
        int numero1 = 15;
        float numero2 = -3.5f;
        double promedio = 15.4;
        char letra = 'A';
        boolean esImpar = true;
    }

}
```

En este ejemplo, además de declarar las variables se le **asigna** un valor a cada una. Es muy importante que para cada variable el valor corresponda con el tipo de dato de la misma.

Por ejemplo, a **`numero1`** se le debe asignar un valor **`int`** (número entero), en este caso 15. Luego a **`numero2`** se le asigna un valor **`float`** (número de punto flotante), en este caso -3.5. Lo mismo sucede con **`promedio`**, que se le asigna un valor **`double`** (número real) y con la variable **`letra`** de tipo **`char`** (un caracter), que se le asigna el valor 'A'. Por último, a la variable **`esImpar`** se le asigna un valor de tipo **`boolean`** (verdadero o falso), en este caso true (verdadero).

Ejemplo de declaración de variables

universidad cenfotec
La U de la informática

```
public class Variables {

    public static void main(String[] args) {
        int numero = 15;
        double promedio = 15.4;

        numero = 120;
        promedio = 500 / 3;
    }

}
```

Una vez que usted **declara** e **inicializa** una variable, es posible modificar su contenido **asignando** un nuevo valor a la variable.

Se puede asignar cualquier valor que concuerde con el tipo de dato de la variable, como en este caso, un número concreto, como en el caso de la variable **`numero`**, o el resultado de una operación matemática, como en el caso de la variable **`promedio`**.

Al asignar un valor no debe escribir el tipo de dato de la variable. Ya que esto se escribe únicamente cuando se está declarando la misma.

Tipo de dato `String`

universidad cenfotec
La U de la informática

```
public class Variables {

    public static void main(String[] args) {
        String nombre = "Alan Turing";
    }

}
```

A diferencia de los tipos de datos primitivos vistos anteriormente, **`String`** es un tipo de dato compuesto, el cual permite guardar una cadena de caracteres (**`char`**), es decir un conjunto de letras dentro de una sola variable.

Este tipo de dato es utilizado para almacenar nombres, frases, códigos, entre otros. En este ejemplo, se crea una variable llamada **`nombre`** de tipo de dato **`String`**, a la cual se le asigna el valor *Alan Turing*. Es muy importante que cuando se trabaja con **`String`** utilizar comillas dobles, a diferencia de del tipo de dato **`char`** que utiliza comillas simples.

Declaración de constantes

universidad cenfotec
La U de la informática

Una constante es un dato que se necesita para resolver el problema, pero que **no cambia su valor**. El valor está dado desde antes del problema y seguirá siendo el mismo valor cuando termine el algoritmo.

```
static final double G = 9.8;
static final double PI = 3.14159;
static final double I = 0.15;
```

Esto *separa un espacio en memoria con un valor que no cambia*. Las constantes pueden ser de todos los tipos de datos simples de Java. En el ejemplo se usan: la gravedad, el número pi y una constante llamada I, usada en cálculos que usen un interés constante.

Sintaxis del lenguaje

universidad cenfotec
La U de la informática

De la misma forma que existen reglas sintácticas y gramaticales en las lenguas naturales de comunicación entre seres humanos, los **lenguajes de programación** tienen pautas a seguir al momento de codificar.

Por ejemplo, en español una oración con algún error puede sonar extraña o con un significado distorsionado, en un lenguaje de programación un descuido en la sintaxis puede causar que el computador no logre ejecutar las instrucciones dadas.

Un ser humano tiene la capacidad de recuperar significado de una oración extraña, el proceso de compilación de una computadora no, debido a esto es importante que su código esté escrito de forma correcta.

La importancia de definir estándares

universidad cenfotec
La U de la informática

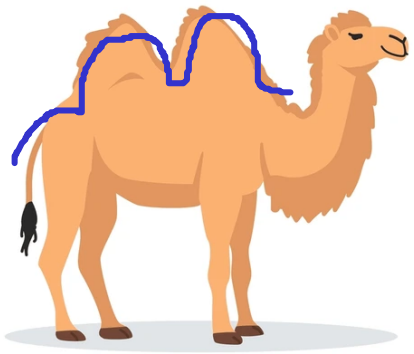
A través de los años, desarrolladores con gran experiencia han creado **estándares de programación** para promover la creación de código de calidad, para que aquellas personas que lean el programa logren entender fácilmente la solución dada.

Por ejemplo, existen muchas nomenclaturas establecidas para darle nombre a las variables, la cuales tienen el objetivo de facilitar la lectura y comprensión dentro del programa. Dos de esas nomenclaturas que se utilizarán a lo largo del curso son:

lowerCamelCase

- **camelCase**: la primera letra de una variable irá en **minúscula** y si está compuesto por más palabras, estas irán en **mayúscula**. Por ejemplo: *primerNumero, sumaTotal, salarioNeto*.
- **PascalCase**: similar a camelCase, pero en este caso la primera letra de todas las palabras se escribe en **mayúscula**. Por ejemplo: *PromedioPonderado, ListaEstudiantes, ResultadoFinal.*

UpperCamelCase



Camel

primerNombreEstudiante

lowerCamelCase

numeroCedula

EstacionServicio

UpperCamelCase == PascalCase

```
File  Edit  Search  Run  Compile  Debug  Tools  Options  Window  Help
                              SOL4.PAS                          1=[↕]
var F,L:real;
    i,j,n:integer;
    x:array[1..10] of real;
    y:array[1..10] of real;

begin
     write('n=');readln(n);
     FOR i:=1 TO n DO
     begin
          write('x[',i,']=');readln(x[i]);
          write('y[',i,']=');readln(y[i]);
     end;
     begin
          write('x[',n+1,']=');readln(x[n+1]);
     end;
     y[n+1]:=0;
     F:=0;
     FOR j:=1 TO n DO begin
     L:=1;
     FOR i:=1 TO n DO
     begin
          IF i<>j THEN
```

Estándares para nombre de las variables

universidad cenfotec
La U de la informática

Se recomienda que las variables cumplan con el siguiente estándar:

- Tenga un nombre significativo que represente su funcionalidad dentro del programa.
- Utilice una de las nomenclaturas existentes a la hora de definir el nombre de la variable.
- Si se requiere utilizar números, se escriben al final del nombre de la variable.

porcentajeAsistencia **cantidadTotalEntregas** **notasEstudiante2**

Sintaxis para nombre de las variables

universidad cenfotec
La U de la informática

Existen **restricciones** a la hora de definir el nombre de las variables que, en caso de omitirlas, Java las detectará como un error en el código. Por ejemplo:

- Se debe iniciar el nombre de la variable con una letra. No se debe iniciar con un símbolo o número, *2estudiante*, *#Mes*, *%adultosMayores* son ejemplos de identificadores **inadecuados**.
- Java al utilizar palabras provenientes del inglés en su sintaxis, no permite el uso de tildes ni virgulillas (como en la letra Ñ) como *opción3*, *primerAño*, *idCamión*.

Resumen

universidad cenfotec
La U de la informática

A la hora de trabajar con variables en Java es importante recordar lo siguiente:

- Definir **el tipo de dato** que se adecúe al valor que va a almacenar en su variable, por ejemplo, un número entero, un número decimal, una letra o una palabra.
- Los tres procesos relacionados con variables, **declaración** cuando crea una variable y le da un nombre, **inicialización** cuando le da un valor inicial y **asignación** cuando en su programa asigna un nuevo valor a una variable.
- Utilizar las **nomenclaturas** y **estándares** mencionados a la hora de nombrar sus variables para asegurarse que sea fácil comprender la funcionalidad en el código.


universidad
cenfotec_
La U de la informática