

# Constantes, Variable y Expresiones

## Curso de Algoritmia y Programación

Luis Garreta

Pontificia Universidad Javeriana – Cali  
Facultad de Ingeniería - Carrera de Biología

14 de agosto de 2018

# Constantes

- ▶ Datos que no cambian su valor
- ▶ Ejemplos:
  - ▶ Un número (3 ó 5.34).
  - ▶ Una cadena ("Hola mundo")
  - ▶ Un valor de verdad (Falso o Verdadero)

# Variables

- ▶ Son datos que pueden cambiar su valor.
- ▶ Si cambian, generan un cambio en el valor de la expresión que los involucra.
- ▶ Se representan con nombres:
  - ▶ Letras y números
  - ▶ Sin espacios
  - ▶ Sin caracteres del español (tildes y ñes)
- ▶ Ejemplos:
  - ▶ `a = 5`
  - ▶ `numUsuario = 20`
  - ▶ `num_magico = 50`
  - ▶ `mensaje = "Hola Mundo"`
  - ▶ `condicionTerminar = Falso`

# Expresiones

- ▶ Sentencias que involucran el uso de constantes, variables y operadores.
- ▶ Se clasifican de acuerdo al tipo de datos y operadores que emplean, lo cual implica también el tipo de resultado que se obtiene en su desarrollo o evaluación.
- ▶ Ejemplos:
  - ▶  $a = 5$
  - ▶  $b = 6$
  - ▶  $c = a + b$

# Expresiones aritméticas

Una expresión aritmética es aquella que incluye variables numéricas (enteras o reales), constantes numéricas y operadores aritméticos.

► Ejemplo:  $(a + b) * 0.2$

# Operadores aritméticos

+	Operador de suma
-	Operador de resta
*	Operador de multiplicación
/	Operador de división

# Prioridad

- ▶ La prioridad corresponde a las reglas que indican cuál operación debe realizarse primero en una expresión aritmética
- ▶ Aunque estas reglas pueden variar de un lenguaje a otro, en general:
  - ▶ la mayor prioridad la tienen los operadores de multiplicación y división
  - ▶ después seguirían los de suma y resta.
- ▶ El uso de parentesis permite saltar las reglas de prioridad para obligar a que se desarrolle una operación primero

▶ **Ejemplo:**

- ▶  $4 + 8 * 2 - 6 / 2$
- ▶  $(4 + 8) * (2 - 6) / 2$

# Tipos de Datos en Operaciones

- ▶ En los lenguajes de programación, se tiene muy en cuenta los tipos de datos empleados en las operaciones, aunque puede variar, en términos generales
- ▶ Si se operan números enteros (sin parte decimal) el resultado será un número entero; pero si se operan números reales (que incluyen parte decimal) o una combinación entre reales y enteros, el resultado será un numero real.
- ▶ **Ejemplo:**
  - ▶  $10 / 4$  dará como resultado **2**, ya que los dos números son enteros
  - ▶  $10.0 / 4.0$  dará como resultado **2.5**, ya que los números son reales.



# Expresiones Lógicas

Las expresiones lógicas pueden emplear datos de diferentes tipos (lógicos o numéricos), emplean operadores lógicos o relacionales y obtienen como resultado un valor lógico (**verdadero** o **falso**)

► Ejemplo:

►  $10 > 5$

► `num_usuario == num_magico`

# Operadores relacionales

Estos operadores son los que permiten establecer algún tipo de relación entre dos operandos, las relaciones comunes son las siguientes:

Relación	Operador
Igual	==
Diferente	!=
Mayor	>
Menor	<
Mayor o igual	>=
Menor o igual	<=

# Ejemplo Operadores Relacionales

$3 > 4$	tendrá como resultado Falso
$5 < 20$	tendrá como resultado verdadero
$10 \geq 10$	Tendrá resultado Verdadero
$3 < 4 < 10$	no es una expresión válida porque en algún momento quedaría algo como verdadero $\& \times 10$ , lo cual no puede resolverse.

# Operadores lógicos

- ▶ Los operadores lógicos permiten enlazar expresiones lógicas a través de las conexiones lógicas que se emplean comunmente.

Ejemplo:

```
(num_intentos ==0) && (num_usuario != num_magico)
```

# Operadores Lógicos Comunes

Los operadores lógicos comúnmente empleados son:

Conexión lógica	Operador	Símbolo en C#
Conjunción	AND	&&
Disyunción	OR	
Negación	NOT	!

# Tablas de Verdad Operadores Lógicos

<b>AND</b>	Verdadero	Falso
Verdadero	Verdadero	Falso
Falso	Falso	Falso

<b>OR</b>	Verdadero	Falso
Verdadero	Verdadero	Verdadero
Falso	Verdadero	Falso

<b>NOT</b>	
Verdadero	Falso
Falso	Verdadero

# Ejemplo Operadores Lógicos

Si se quiere verificar que la variable nota se encuentre en el rango [0.0, 5.0] la expresión correcta sería:

$( \text{Nota} \geq 0.0 ) \ \&\& \ ( \text{nota} \leq 5.0 )$

- Si la variable Nota obtiene el valor de 2.6 tendríamos:

$( 2.6 \geq 0.0 ) \ \&\& \ ( 2.6 \leq 5.0 )$

Verdadero && verdadero

Verdadero

- Si la variable Nota obtiene el valor de 6.2 tendríamos:


$( 6.2 \geq 0.0 ) \ \&\& \ ( 6.2 \leq 5.0 )$

Verdadero && Falso

Falso

# Manejo de Prioridades Operadores Lógicos

/		*	
+		-	
>	<	>=	<=
==		!=	
&&			



**Ejercicio:** Qué resultado queda para el siguiente caso:

$$3 + 4 < 10 \ \&\& \ 56 - 3 > 100 \ || \ 4 * 6 - 2 <= 22$$



# Funciones matemáticas

- ▶ Algunas operaciones matemáticas comunes, no tienen representación en los lenguajes de programación como un operador.
- ▶ En lugar de ello debe emplearse un método que ya está incorporado en el lenguaje.
- ▶ En el caso de C#, muchas de estas operaciones se incorporan en la clase **Math**, entre las más comunes se encuentran las siguientes:

Nombre del método	Función
Cos(double x)	Retorna el coseno del ángulo especificado.
Exp (double x)	Retorna e elevado a la potencia especificada.
Log (double x)	Devuelve el logaritmo natural (en base e) de un número especificado.
Log10(double x)	Devuelve el logaritmo en base 10 de un número especificado.
Pow(double x, double y)	Retorna un número (x) elevado a la potencia especificada (y)
Round (decimal x)	Redondea un número decimal al entero más próximo.
Sin (double x)	Retorna el seno del ángulo especificado.
Sqrt(double x)	Retorna la raíz cuadrada del número indicado.
Tan(double x)	Retorna la tangente del ángulo especificado.

# Ejemplo Funciones Matemáticas

- ▶ Esta expresión dejaría en x el valor de 8 (raíz cuadrada de 64)

```
X = Math.sqrt(64);
```

- ▶ Estos métodos pueden ser involucrados dentro de una expresión:

```
58 - Math.sqrt(64) < 100
```

En este caso, se resuelve primero el método, luego la resta y finalmente la comparación; lo que daría como resultado Verdadero