

**TALENTO  
DIGITAL**  
INTELIGENCIA  
HUMANA

# Talento Digital para Chile:

## MÓDULO 1 PROGRAMACIÓN BÁSICA EN JAVA

UN PROYECTO DE:

DESARROLLADO POR:



## MÓDULO 1 - PROGRAMACIÓN BÁSICA EN JAVA

### 1.1 ALGORITMOS

#### Semana 1 - Día 2

---

#### Objetivo de la jornada

---

- Comprender el concepto de variables y tipos de datos para luego utilizarlos para resolver problemas planteados.
  - Entender el concepto de expresiones aritméticas y operadores para resolver problemas planteados según indicaciones entregadas.
  - Comprender y aplicar estructuras de control condicional para dar solución a un problema según requerimientos establecidos.
- 

#### Variables y tipos de dato

En programación, una variable es un espacio de memoria reservado para almacenar un valor que corresponde a un tipo de dato soportado por el lenguaje de programación. Una variable es representada y usada a través de una etiqueta (un nombre) que le asigna un programador o que ya viene predefinida.

Nombres de variables: Deben ser nombres que tengan relación con el dato que se almacena. Se puede utilizar caracteres y números pero no símbolos ni tildes. El único carácter especial que se permite es el guion bajo.

Un ejemplo de lo anterior, es tener una variable llamada ciudad en la que almacenaremos el valor Viña del Mar.

Ciudad = Viña del Mar

Los tipos de datos son una restricción almacenada en un espacio en memoria con restricciones impuesta para la interpretación, manipulación, o representación de datos. Tipos de datos comunes en lenguajes de programación son los tipos:

- **Tipo numérico:** Son aquellos datos que representan sólo números, ya sean enteros o con decimales.
- **Tipo caracteres:** Un carácter es una unidad de información mínima e indivisible de la escritura de una lengua. Un ejemplo de carácter es una letra, un número o un signo de puntuación. Este tipo de dato también abarca a los caracteres de control, como el salto de línea o el tabulador.
- **Tipo alfanumérico:** Este tipo corresponde a todos los caracteres del código ASCII, que incluye las letras desde la A a la Z, números del cero al nueve, y símbolos (incluye los tildes, la ñ y espacio). Si el dato 77 es almacenado en una variable de tipo numérico, es posible realizar operaciones matemáticas con él. En este caso  $77 + 3$  es 80, pues el computador los ha sumado. Por otro lado, si 77 es almacenado en una variable del tipo alfanumérico, entonces  $77 + 3$  es 773, pues el computador lo ha concatenado ya que se unen ambas cadenas de texto.
- **Tipo booleano:** Es aquel que puede representar valores de lógica binaria, dos valores que normalmente representan verdadero o falso.

Otro concepto básico en programación, además de las variables y los tipos de datos, es una constante. Esta corresponde a una zona de memoria que almacena un dato al igual que una variable, con la diferencia de que el dato almacenado no se puede cambiar.

## Expresiones aritméticas y operadores

En informática y lenguajes de programación, se entiende por expresión aritmética a aquella donde los operadores que intervienen en ella son numéricos, el resultado es un número y los operadores son aritméticos. Los operadores aritméticos más comúnmente utilizados son: El signo más (+) se emplea para sumar dos valores, el signo menos (-) para restar un valor de otro, el asterisco (\*) para multiplicar dos valores, la división (/) para dividir un valor por otro, y el signo % para obtener el resto de una división entera. Estos símbolos se conocen como operadores binarios, pues operan sobre dos valores o variables.

Los operadores aritméticos son aquellos que permiten resolver operaciones matemáticas básicas, incluso se pueden utilizar los paréntesis en las fórmulas matemáticas para problemas de precedencia.

Operador	Descripción	Símbolo	Ejemplo
Igual	Permite comparar si dos datos son iguales.	<code>==</code>	<code>a == b</code>
Menor que	Permite comparar si el primer dato es menor que el segundo	<code>&lt;</code>	<code>a &lt; b</code>
Menor o igual que	Permite comparar si el primera dato es menor o igual que el segundo	<code>&lt;=</code>	<code>a &lt;= b</code>
Mayor que	Permite comparar si el primer dato es mayor que el segundo	<code>&gt;</code>	<code>a &gt; b</code>
Mayor o igual que	Permite comparar si el primer dato es mayor o igual que el segundo	<code>&gt;=</code>	<code>a &gt;= b</code>
Distinto	Permite comparar si dos datos son distintos	<code>a != b</code>	<code>a != b</code>

Los operadores relacionales corresponden a los símbolos que se usan para comparar dos valores. Si el resultado de dicha comparación es correcta, la expresión evaluada es verdadera, en caso contrario es falsa.

Operador	Descripción	Símbolo	Ejemplo
Suma	Permite sumar dos datos.	<code>+</code>	<code>a + b</code>
Resta	Permite restar dos datos	<code>-</code>	<code>a-b</code>
Multiplicación	Permite multiplicar dos datos.	<code>*</code>	<code>a*b</code>
División	Permite dividir dos datos. Se debe considerar que la división por cero causa un error.	<code>/</code>	<code>a/b</code>
Módulo	Entrega el resto de una división entera	<code>%</code>	<code>a%b</code>



Por último, los operadores lógicos son aquellos en donde Los operadores lógicos nos proporcionan un resultado a partir de que se cumpla o no una cierta condición, producen un resultado booleano y sus operandos son también valores lógicos o asimilables a ellos (los valores numéricos son asimilados a cierto o falso según su valor sea cero o distinto de cero).

Esto genera una serie de valores que, en los casos más sencillos, pueden ser parametrizados con los valores numéricos 0 y 1. La combinación de dos o más operadores lógicos conforma una función lógica.

Las expresiones conectadas con los operadores && y || se evalúan de izquierda a derecha, y la evaluación se detiene tan pronto como el resultado verdadero o falso es conocido (muchos programas tienen una lógica que se basa en esta propiedad).

- AND (el resultado es verdadero si ambas expresiones son verdaderas)
- OR (el resultado es verdadero si alguna expresión es verdadera)
- NOT (el resultado invierte la condición de la expresión)

Operador lógico AND		
X	Y	Resultado
true	true	true
true	false	false
false	true	false
false	false	false

Operador lógico NOT	
X	Resultado
true	false
false	true

Operador lógico OR		
X	Y	Resultado
true	true	true
true	false	true
false	true	true
false	false	false

## Estructuras de control condicional

Una estructura de control condicional consiste en la ejecución de una o más instrucciones dependiendo de la evaluación de una condición.

Si la condición es verdadera, se ejecuta el bloque de sentencias 1; de lo contrario, se ejecuta el bloque de sentencias 2-

Si (Condición) entonces

Bloque de sentencias 1 sino Bloque de sentencias 2 Fin

Si

Entendiendo como funciona la estructura SI

1. Se evalúa la condición y si ésta se cumple, entra al bloque
  - a. Se ejecutan las instrucciones que se encuentra en su interior
  - b. Luego de ejecutadas las instrucciones, el programa sigue con su secuencia, ejecutando la instrucción que se encuentre a continuación del fin del SI.
2. Si la condición no se cumple, no entra al bloque y el programa sigue con su secuencia, ejecutando la instrucción que se encuentre a continuación del fin del SI

Se pueden plantear múltiples condiciones simultáneamente: si se cumple la (Condición 1) se ejecuta (Bloque de sentencias 1).

En caso contrario se comprueba la (Condición 2);

si es cierta se ejecuta (Bloque de sentencias 2),

y así sucesivamente hasta n condiciones.

Si ninguna de ellas es cumple se

ejecuta (Bloque de sentencias else).

Si (Condición 1) entonces

(Bloque de sentencias 1) sino si (Condición 2) entonces

(Bloque de sentencias 2) sino si (Condición 3) entonces

(Bloque de sentencias 3) sino (Bloque de sentencias else)

Fin Si

Un caso práctico sería **“ingresar 2 números por teclado, si el segundo número es cero, entonces muestre el primer número en pantalla, en caso contrario, mostrar el segundo número.”**

Analizando el enunciado, se determina que el usuario del programa debe ingresar dos números y determinar si el segundo número es cero.

Así tenemos que las entradas son los dos números. Pero en cuanto a la salida, tenemos dos posibles, la primera cuando el segundo número ingresado es cero, y la segunda, cuando el segundo número ingresado no es cero.

Entonces, para entregar una solución se debe hacer uso de las estructuras de control, que permiten dirigir el flujo de acción dentro del programa, en base a la evaluación del cumplimiento de ciertas condiciones lógicas.

## INICIO

Leer num1, num2 Si (num2 == 0) entonces

Mostrar num1

sino

Mostrar num2

Fin Si FIN

Otra sentencia de control son las de tipo SWITCH. En este tipo de sentencias se especifica la variable a comparar y una lista de valores con lo que comparar. Aquel que sea el verdadero, se ejecutará. Adicionalmente, se puede dejar una respuesta por defecto, que tendría la misma función del “Sino”. La estructura se vería de la siguiente forma:

En-caso-de expresión

Caso expresión 1

Bloque de sentencias 1

Caso expresión 2

Bloque de sentencias 2

Por-defecto

Bloque de sentencias 3

Fin Caso



## Ejercicios individuales

1) Un entrenador de basquetbol escolar quiere organizar a sus jugadores según su estatura para aprovechar esta característica en las distintas posiciones del equipo. Si la estatura es menor a 1.60 mts, es “Base”; mayor a 1.60 mts es “Escolta”; mayor a 1.70 mts es “Alero” y mayor a 1.75 mts es “Pivot”. El ingreso se realiza indicando sólo la estatura del jugador, uno por uno.

2) Dado un número entre 1 y 7 escriba su correspondiente día de la semana.

## Ejercicios grupales

Según el gremio de Scouts de Chile, se deben organizar los niños según rango de edad, por lo que se le solicita a usted crear un módulo de un sistema que permita que al ingresar un nuevo niño a la institución, este sea derivado al grupo correspondiente. La forma de distribuir a los niños es la siguiente: si tiene entre 7-11 años y es hombre, le corresponde “Manada” y si es mujer le corresponde “Bandada”; si tiene entre 11-17 y es hombre, le corresponde “Tropa” y si es mujer le corresponde “Compañía”; si tiene más de 18 años le corresponde “Ruta”, sin importar si es hombre o mujer.

Recursos Bibliográficos: <http://robotica.uv.es/pub/Libro/PDFs/CAP13.pdf>