

FUNDAMENTOS DE PROGRAMACIÓN



FUNDAMENTOS DE PROGRAMACIÓN



¿QUÉ VAMOS A VER?

- Variables
- Tipos de datos comunes
- Expresiones aritméticas
- Operadores Relacionales
- Operadores Lógicos
- Estructuras de control condicional





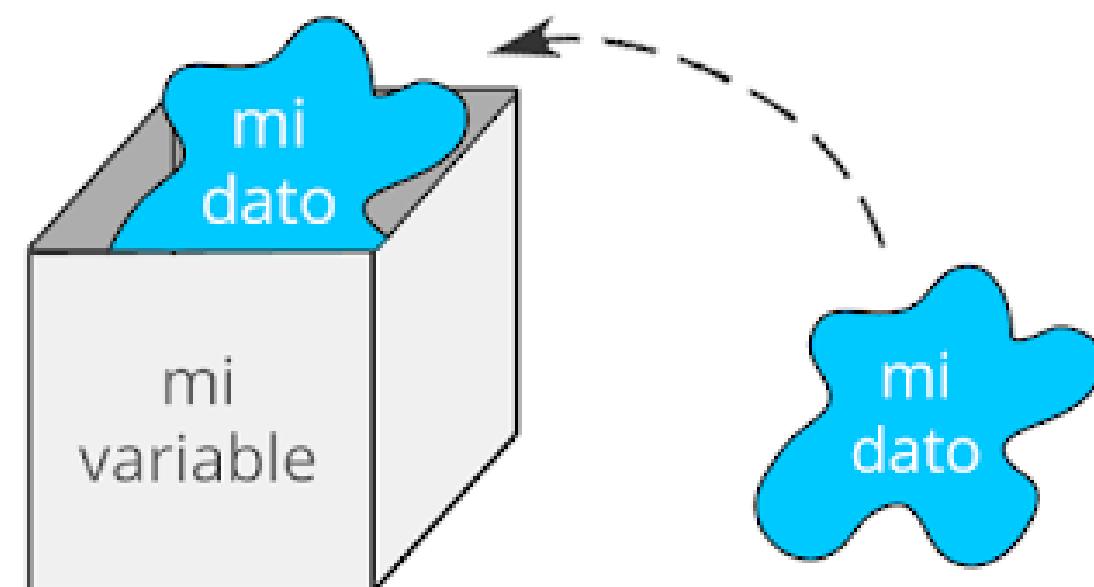
VARIABLES

¿QUÉ ES UNA VARIABLE?

- Es un espacio en memoria reservado para almacenar un tipo de dato.
- Tiene su propio nombre o etiqueta.
- El nombre de una variable debe estar relacionado con el dato que se está almacenando.

Ejemplo:

ciudad = “Talca”





TIPOS DE DATOS

- Es una restricción para las variables, permitiendo: **manipulación, interpretación y representación** de los datos.
- Los tipos de datos más comunes que nos vamos a encontrar son:
 - **Numérico:** Representan números enteros y decimales.
 - **Caracteres:** Representa un solo carácter.
 - **Alfanumérico:** Representa los caracteres del código ASCII (letras, números y símbolos). Tienen la cualidad de que sus datos pueden ser utilizados tanto como letras o números.
 - **Booleano:** Representa un valor binario, verdadero o falso, sí o no.



TIPOS DE DATOS

¿QUÉ ES UNA CONSTANTE?

- Es un valor que no puede ser modificado a lo largo de la ejecución del programa, su valor permanece inmutable.
- Vienen a representar dentro de nuestro código un valor que sabemos que jamás va a cambiar.
- Se puede declarar por defecto desde la ejecución del programa o ser ingresada en alguna etapa de manera manual.

Ejemplo:

$\pi = 3.14159265$

π



EXPRESIONES ARITMÉTICAS Y OPERADORES

- Es una expresión aritmética aquella donde los operadores que intervienen son numéricos, y por ende su resultado es un número.
 - Por otro lado, los operadores aritméticos son aquellos que permiten resolver operaciones matemáticas.
 - Los que se utilizan comúnmente son:
 - Suma +
 - Resta -
 - Multiplicación *
 - División /
 - Resto %
- $a + b$ $a - b$ $a * b$ a / b $a \% b$





OPERADORES RELACIONALES

- Corresponden al tipo de operadores que utilizan símbolos para comparar dos valores.
- Si el resultado de una comparación es correcto, dicha expresión va a retornar un valor booleano de verdadero, en caso contrario, retornaría falso.

OPERADOR	DESCRIPCIÓN	SÍMBOLO	EJEMPLO
Igual	Valida que dos datos sean iguales	<code>==</code>	<code>a == b</code>
Menor que	Valida si el primer dato es menor que el segundo	<code><</code>	<code>a < b</code>
Menor o igual que	Permite comparar si el primer dato es menor o igual que el segundo	<code><=</code>	<code>a <= b</code>
Mayor que	Valida si el primer dato es mayor que el segundo	<code>></code>	<code>a > b</code>
Mayor o igual que	Valida si el primer dato es mayor o igual que el segundo	<code>>=</code>	<code>a >= b</code>
Distinto	Valida que dos datos sean distintos	<code>!=</code>	<code>a != b</code>



OPERADORES LÓGICOS

- Se utilizan principalmente para comparar o combinar valores booleanos, además de que nos van a entregar un valor de la misma índole.
- Dentro de la programación son esenciales a la hora de ser utilizados en estructuras condicionales.
- Nos vamos a encontrar con tres de ellos:
 - **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.





OPERADORES LÓGICOS

**OPERADOR LÓGICO
AND**

X	Y	Resultado
Verdadero	Verdadero	Verdadero
Verdadero	Falso	???
Falso	Verdadero	???
Falso	Falso	???

**OPERADOR LÓGICO
OR**

X	Y	Resultado
Verdadero	Verdadero	Verdadero
Verdadero	Falso	???
Falso	Verdadero	???
Falso	Falso	???

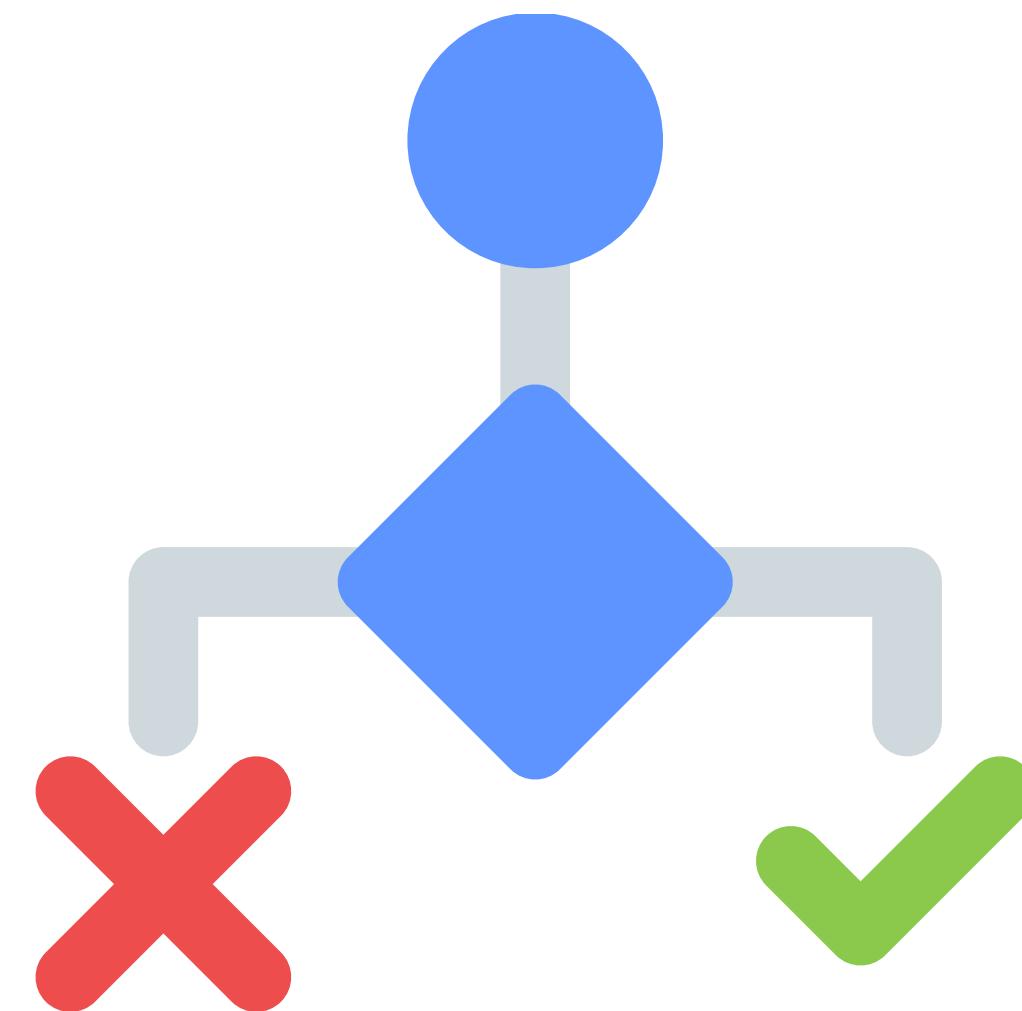




ESTRUCTURAS DE CONTROL CONDICIONAL

- Consiste en la ejecución de una o más instrucciones dependiendo de la evaluación de la condición que se nos presente.
- Si obtenemos un valor Verdadero, se ejecutará un bloque de código, en caso contrario, se ejecutará otro.

```
1 Algoritmo ejemplo
2   Si expresion_logica Entonces
3     acciones_por_verdadero
4   SiNo
5     acciones_por_falso
6   Fin Si
7 FinAlgoritmo
8
```

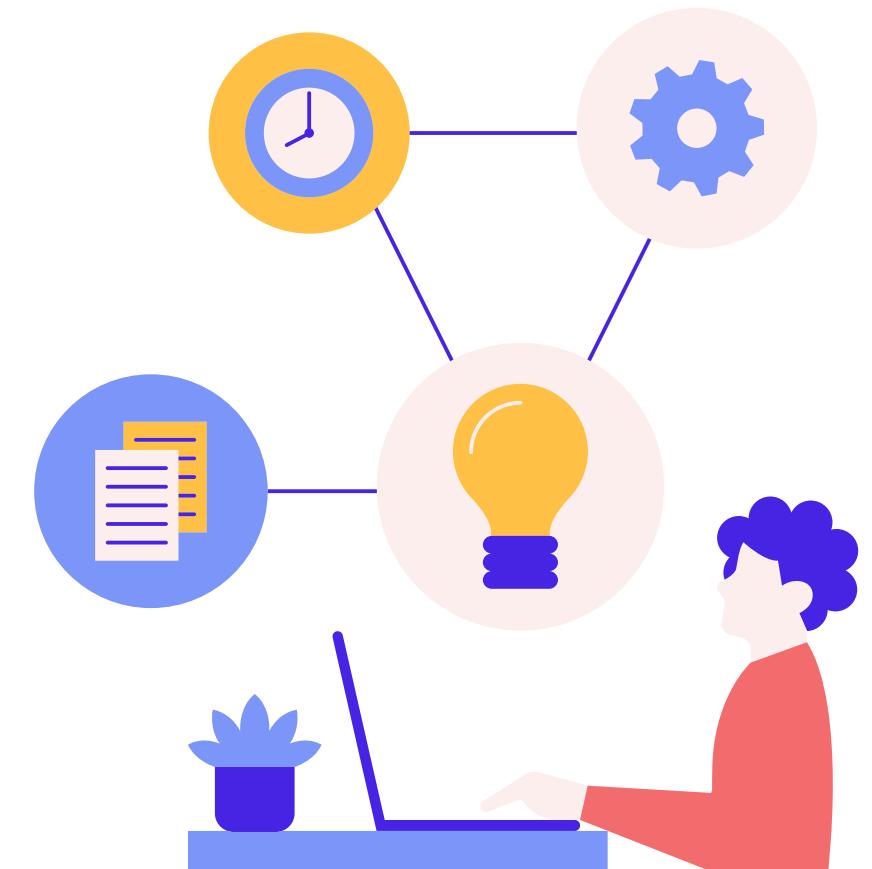




ESTRUCTURAS DE CONTROL CONDICIONAL

¿CÓMO FUNCIONA LA ESTRUCTURA?

1. Se evalúa la condición y si ésta se cumple, entra al bloque.
 - a. Se van a ejecutar las instrucciones que se encuentran en su interior.
 - b. Luego de ejecutadas, el programa sigue con su secuencia, ejecutando cualquier instrucción que continúe bajo la estructura de control.
2. Si la condición no se cumple, no entra al bloque y el programa va a continuar con las instrucciones que estén establecidas bajo la estructura de control.



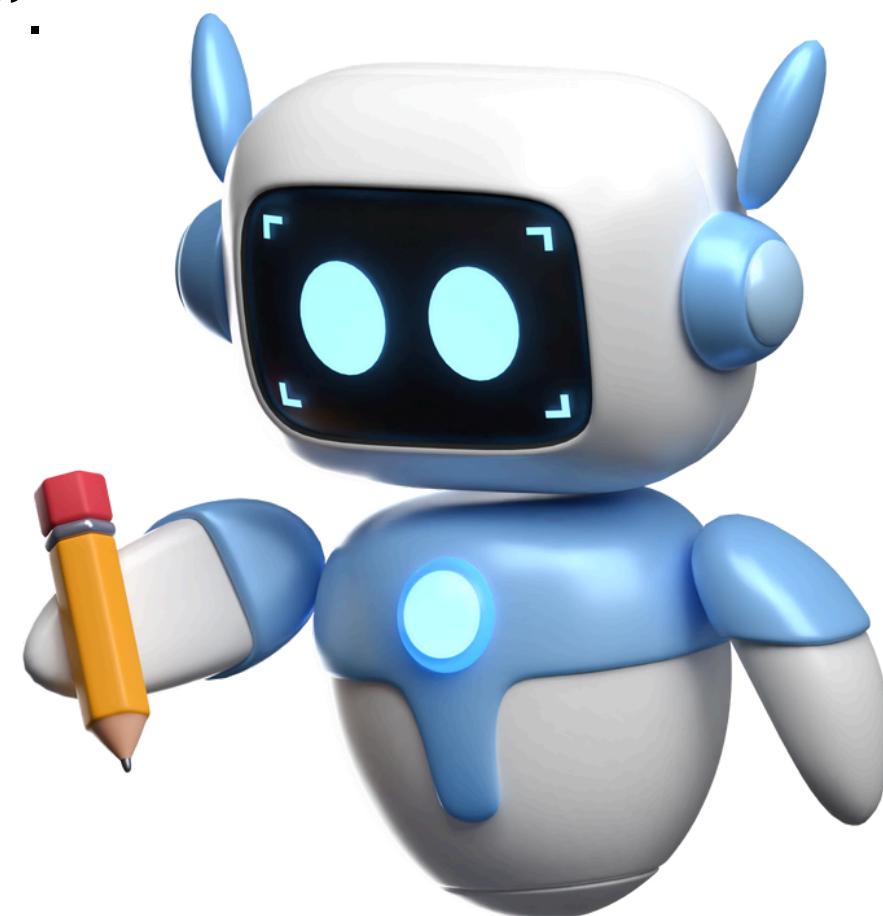


EJEMPLO PRÁCTICO

Ingresa 2 números por teclado, si el segundo número es cero, entonces muestre el primer número en pantalla, en caso contrario, mostrar el segundo número.

Datos a tener en cuenta:

- Para mostrar por pantalla un dato o un texto, se utiliza la expresión “**Escribir**”.
- Para ingresar valores por teclado, se utiliza la expresión “**Leer**”.
- Nunca debemos olvidar **Iniciar** y **Finalizar** nuestro algoritmo.





SWITCH (Según)

Esta estructura de control especifica claramente la variable que vamos a utilizar y una lista de valores con los que vamos a comprar.

Cuando exista la coincidencia de que nuestra variable sea Verdadero al compararla con uno de los valores, se ejecutará el bloque de código que tengamos dentro.

De todas formas, es posible dejar una respuesta “**Por defecto**” en caso de que no exista ninguna opción que se pueda comprar a nuestra variable.

```
1 Algoritmo ejemplo
2 Segun variable_numerica Hacer
3   opcion_1:
4     secuencia_de_acciones_1
5   opcion_2:
6     secuencia_de_acciones_2
7   opcion_3:
8     secuencia_de_acciones_3
9   De Otro Modo:
10    secuencia_de_acciones_dom
11 Fin Segun
12 FinAlgoritmo
```





CASO PRÁCTICO

Crea un programa que presente un menú con cinco diferentes opciones de comida (Pescado, Ensalada, Pizza, etc) y permita al usuario seleccionar una.

- El usuario debe ingresar el número de la opción que va a elegir.
- Según la opción seleccionada, deben mostrar un mensaje con el nombre del plato elegido y su precio.





AWAKELAB

#programmingbootcamp

nodovirtual.awakelab.cl

 jELOU futurejob by  Adalid Chile