



# UEA

UNIVERSIDAD  
ESTATAL AMAZÓNICA





**UEA**  
UNIVERSIDAD  
ESTATAL AMAZÓNICA

# **ASIGNATURA**

# **FUNDAMENTOS DE PROGRAMACIÓN**



*Transformamos el mundo desde la Amazonía*

**Ing. Edwin Gustavo Fernández Sánchez, Mgs.**

DOCENTE - PERSONAL ACADÉMICO NO TITULAR OCASIONAL

DIRECTOR DE GESTIÓN DE TECNOLOGÍAS INFORMACIÓN Y COMUNICACIÓN

**UNIVERSIDAD ESTATAL AMAZÓNICA**





**UEA**  
UNIVERSIDAD  
ESTATAL AMAZÓNICA

**SEMANA**

**5**

**DESARROLLO DE LA SEMANA 5: DEL LUN. 23 AL DOM. 29 DE JUNIO/2025**

**Resultado de aprendizaje: Aplicar métodos computacionales para resolver problemas de su campo de estudio utilizando un lenguaje de programación.**



## **CONTENIDOS**

### **UNIDAD II: Introducción a la Programación**

- **Tema 1: Conceptos básicos de programación**
  - **Subtema 1: Variables y tipos de datos: enteros, flotantes, cadenas de texto, booleanos.**



**UEA**  
UNIVERSIDAD  
ESTATAL AMAZÓNICA

## **Unidad 2**

# **Conceptos básicos de programación**

### **Tema 3**

# **Conceptos básicos de programación**



**UEA**  
UNIVERSIDAD  
ESTATAL AMAZÓNICA

# **Subtema 3.1: Variables y tipos de datos: enteros, flotantes, cadenas de texto, booleanos.**

**FUNDAMENTOS DE PROGRAMACIÓN  
(UEA-L-UFB-026)**



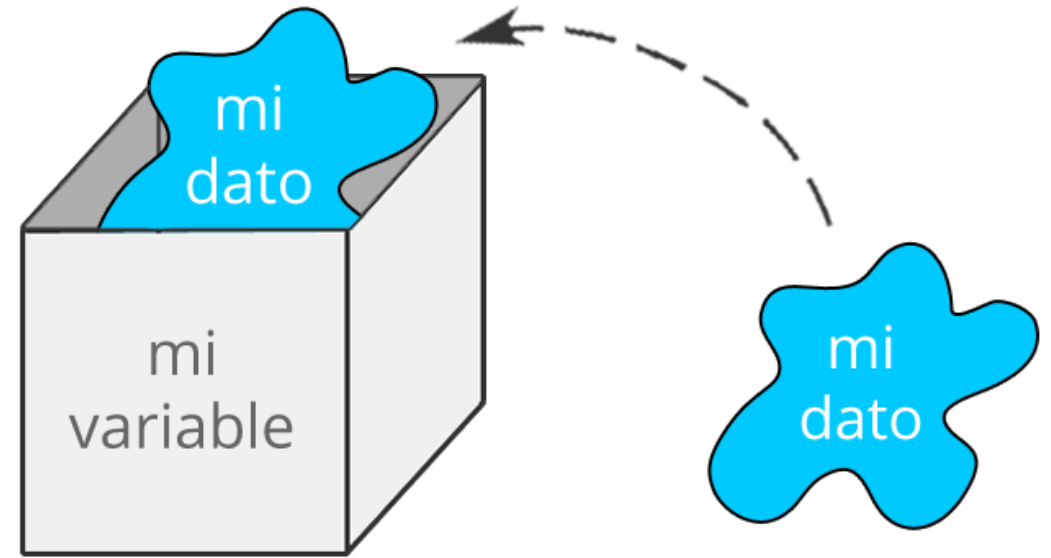
# Variables y tipos de datos

- En el contexto de la **programación**, una variable es un espacio de memoria que se utiliza para almacenar y representar valores. Cada variable tiene un nombre único que se utiliza para referirse a ella. Las variables nos permiten almacenar información como números, texto o valores lógicos, y luego utilizar esa información en nuestro código.
- Por otro lado, los tipos de datos definen el tipo de información que una variable puede contener. Algunos ejemplos comunes de tipos de datos son los **enteros**, que representan números enteros sin decimales; los **flotantes**, que representan números con decimales; las **cadenas de texto**, que representan palabras o frases; y los **booleanos**, que representan valores verdadero o falso.



# ¿Qué es una variable?

- Una variable en programación es un contenedor que se utiliza para almacenar y representar un valor.
- Es como una caja en la memoria de la computadora donde podemos guardar información.
- Cada variable tiene un nombre único que la identifica y nos permite acceder a su contenido.





# ¿Qué es una variable?

- Las variables nos permiten almacenar diferentes tipos de datos, como números, texto o valores lógicos, y utilizar esa información en nuestro código.
- Por ejemplo, podemos tener una variable llamada "edad" que almacena un número entero, o una variable llamada "nombre" que almacena una cadena de texto.



# ¿Qué es una variable?

- La principal ventaja de utilizar variables es la flexibilidad que nos brindan para manejar y manipular datos en nuestros programas. Podemos asignar un valor inicial a una variable y luego modificarlo o utilizarlo en operaciones y cálculos. Esto nos permite crear programas dinámicos y adaptativos.
- Es importante tener en cuenta que las variables deben ser declaradas antes de su uso, especificando su tipo de dato y asignándoles un nombre significativo. También es necesario asegurarse de utilizar el tipo de dato correcto para cada variable, para evitar errores y asegurar la integridad de nuestros datos.



# TIPOS DE DATOS

- Los tipos de datos son categorías que definen el rango y la naturaleza de los valores que una variable puede almacenar. Cada tipo de dato tiene un conjunto específico de valores y operaciones asociadas.
- En programación, algunos de los tipos de datos más comunes son:

## Enteros (int)

Representan números enteros sin decimales, positivos o negativos.

Por ejemplo, 5, -10, 100.



## Flotantes (float)

Representan números decimales o de punto flotante. Incluyen valores fraccionarios.

Por ejemplo, 3.14, -0.5, 2.71828.



## Cadenas de texto (string)

Representan secuencias de caracteres, como letras, números y símbolos. Se utilizan para almacenar texto.

Por ejemplo, "Hola", "Programación", "123".



## Booleanos (bool)

Representan valores lógicos que pueden ser verdadero (true) o falso (false). Se utilizan para evaluar condiciones y controlar el flujo del programa.



# TIPOS DE DATOS

- Estos son solo algunos ejemplos de tipos de datos, y en la programación existen otros tipos más complejos y especializados, como listas, arreglos y estructuras.
- Es importante elegir el tipo de dato adecuado para cada variable, considerando el tipo de información que se va a almacenar y las operaciones que se realizarán con ella. Utilizar el tipo de dato correcto es esencial para garantizar la integridad de los datos y evitar errores en el programa.



# TIPOS DE DATOS

Tipo	Descripción	Rango	Ejemplo
bool	Valor binario verdadero o falso.	true false	bool dato = false; dato = true;
char	Valor entero que representa un carácter de la tabla ASCII	-128 a 127 ó 0 a 255 compilado con /J	char letra = 'A'; letra = '\n'; letra = 65;
short	Valor entero de 2 bytes	-32,768 32,767	short x = 94; x = -54;
int	Valor entero de 4 bytes	-2,147,483,648 2,147,483,647	<u>int</u> x = 1598; x = -988574;
unsigned int	Valor entero positivo de 4 bytes	0 4,294,967,295	unsigned int x = 9887; x = 98745;
long long	Valor entero de 8 bytes	-9,223,372,036,854,775,808 9,223,372,036,854,775,807	long long x = 684574; x = -998564;
float	Valor decimal de 4 bytes	3.4E +/- 38 (7 dígitos)	float x = 45.6; x = -98.58;
double	Valor decimal de 8 bytes	1.7E +/- 308 (15 dígitos)	<u>double</u> x = 9878.568; x = -98745.668;
void	Tipo de dato nulo. Representa la ausencia de valor.		



# Declaración de variables

- La declaración de una variable es la acción de reservar un espacio en la memoria del programa para almacenar un valor específico. Para declarar una variable, se necesita especificar su tipo de dato y asignarle un **nombre único**.
- Especificar el tipo de dato: Se debe indicar el tipo de dato que la variable contendrá, como entero, flotante, cadena de texto, booleano, entre otros. Esto determina la naturaleza y los límites de los valores que la variable puede almacenar.
- Asignar un nombre a la variable: Se elige un **nombre descriptivo** para la variable, que permita identificar fácilmente su propósito en el programa. Es importante seguir las convenciones de nomenclatura, como utilizar letras, números y guiones bajos, y comenzar el nombre con una letra.



# Declaración de variables

- Opcionalmente, asignar un valor inicial: Se puede asignar un valor inicial a la variable en el momento de su declaración. Esto se hace utilizando el operador de asignación (=) seguido del valor deseado. Si no se asigna un valor inicial, la variable tendrá un valor predeterminado dependiendo de su tipo de dato.
- **Por ejemplo:** edad = 25; altura = 1.75; nombre = "Juan"; activo = true;
- Una vez que una variable ha sido declarada, se puede utilizar en el programa para almacenar y manipular datos. Es importante tener en cuenta que las variables deben ser declaradas antes de que se puedan utilizar en el programa.
- La declaración de variables es una parte fundamental en la programación, ya que permite asignar y gestionar datos de manera eficiente. Al seguir las reglas de declaración de variables, se promueve la claridad y la organización en el código, lo que facilita su comprensión y mantenimiento.



# Operaciones con variables

- Las variables en programación no solo almacenan datos, sino que también pueden participar en diversas operaciones matemáticas y manipulaciones. Estas operaciones se realizan utilizando operadores específicos que actúan sobre las variables y los valores que contienen.

## Operaciones aritméticas

Las variables numéricas, como las de tipo entero o flotante, pueden participar en operaciones aritméticas básicas como la suma (+), resta (-), multiplicación (\*) y división (/).

Estas operaciones se realizan entre variables o entre una variable y un valor constante.



## Concatenación de cadenas

Las variables de tipo cadena de texto pueden concatenarse utilizando el operador de concatenación (+).

Esto permite unir dos o más cadenas en una sola.



## Operaciones lógicas

Las variables booleanas pueden participar en operaciones lógicas como la negación (!), la conjunción (&&) y la disyunción (||).

Estas operaciones permiten evaluar condiciones y tomar decisiones en función del resultado.



# Operaciones con variables

- Es importante tener en cuenta que las operaciones con variables deben ser coherentes con sus tipos de datos. Por ejemplo, no se puede realizar una operación aritmética entre una variable de tipo entero y una de tipo cadena de texto.
- Las operaciones con variables son fundamentales en la programación, ya que permiten realizar cálculos, manipular datos y tomar decisiones en función de los valores almacenados. Al comprender y utilizar correctamente las operaciones con variables, se pueden crear programas más dinámicos y funcionales.



UEA  
UNIVERSIDAD  
ESTATAL AMAZÓNICA