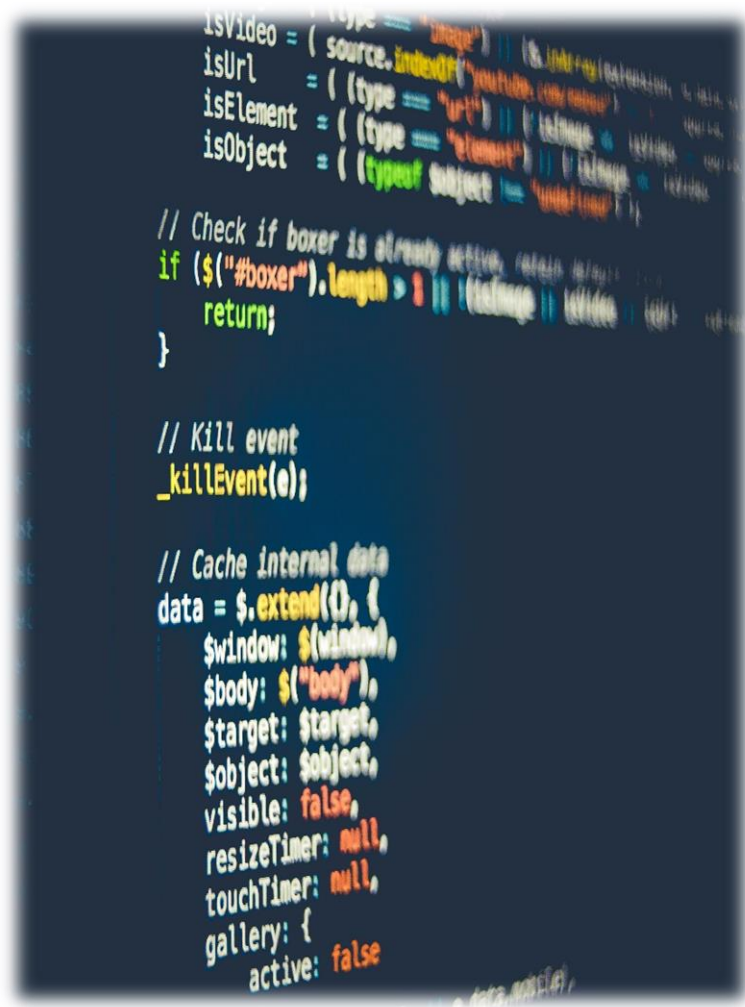


## TRABAJO DE INVESTIGACION – SEMANA 3



## TEMA: IDENTIFICADORES, CONSTANTES, OPERADORES Y EXPRESIONES, ABSTRACCIÓN DE DATOS Y VARIABLES TEMPORALES

**ESTUDIANTE:** ARIEL ALEJANDRO CALDERÓN

**CURSO:** SOFTWARE

NOVIEMBRE 2023

## CONSTRUCCIÓN DE ALGORITMOS Y PROGRAMAS

### Identificadores:

Los identificadores son nombres que se utilizan para identificar variables, funciones, clases u otros elementos en un programa. Deben seguir ciertas reglas, como comenzar con una letra, no contener espacios y ser significativos para facilitar la comprensión del código.

Los identificadores **son esenciales para referenciar** y manipular datos en un programa.

# Ejemplo de  
identificador para una  
variable

```
nombre_usuario =
```

# Ejemplo de constantes

```
PI = 3.14159
```

```
SALUDO = "Hola, mundo"
```

### Constantes:

Las constantes **son valores que no cambian durante la ejecución de un programa**. Pueden ser numéricas, como 3.14, o cadenas de texto, como "Hola, mundo". El uso de constantes mejora la legibilidad del código y facilita la actualización de valores en un solo lugar.

### Operadores y Expresiones:

Los operadores **son símbolos que realizan operaciones** sobre uno o más operandos. Las expresiones son combinaciones de valores y operadores que se evalúan para producir un resultado. Ejemplos de operadores incluyen aritméticos (+, -, \*, /), lógicos (&&, ||) y de comparación (==, !=).

# Ejemplo de operadores y expresiones

- `a = 5 b = 2`
- `suma = a + b`
- `resta = a - b`
- `multiplicacion = a * b`
- `division = a / b`
- `resultado_logico = (a > b) and (a != 0)`

# Ejemplo de abstracción de datos con una clase

```
class Persona:
    def __init__(self, nombre, edad):
        self.nombre = nombre
        self.edad = edad
    def obtener_info(self):
        return f"Nombre:{self.nombre},
        Edad: {self.edad}"
# Uso de la clase
persona1 = Persona("Alice", 25)
print(persona1.obtener_info())
```

### Abstracción de Datos:

La abstracción de datos implica ocultar los detalles internos y **mostrar solo las características esenciales de un objeto**. En programación, las estructuras de datos y las clases permiten la abstracción de datos al encapsular información y proporcionar interfaces para interactuar con ella.

### Variables Temporales:

Las variables temporales son aquellas que se utilizan **para almacenar datos de manera temporal durante la ejecución de un programa**. Son útiles para realizar cálculos intermedios o almacenar valores temporales que se requieren en un momento específico. Las variables temporales ayudan a gestionar la manipulación de datos y facilitan la comprensión del flujo de un programa.

## TEMA: IDENTIFICADORES, CONSTANTES, OPERADORES Y EXPRESIONES, ABSTRACCIÓN DE DATOS Y VARIABLES TEMPORALES

En resumen, en el contexto de la programación, los identificadores y constantes son esenciales para dar nombre y significado a los elementos del programa. Los operadores y expresiones permiten realizar operaciones y cálculos. La abstracción de datos ayuda a manejar la complejidad al encapsular información, y las variables temporales son útiles para almacenar datos de manera temporal durante la ejecución del programa. Todos estos conceptos se entrelazan para construir programas efectivos y comprensibles.

### **Bibliografía:**

[Identificadores en la programación con C++ - https://www2.eii.uva.es/](https://www2.eii.uva.es/)

[Constantes y variables - https://www.ibm.com](https://www.ibm.com)

[Operadores y expresiones - https://www.luisllamas.es](https://www.luisllamas.es)

[Abstracción de datos - https://www.cursosdesarrolloweb.es](https://www.cursosdesarrolloweb.es)