

ALGORITMIA

Juan Carlos Molina Lozano
Docente

OBJETIVOS DE LA CLASE

1. Comprender el concepto y propósito de las funciones en programación.
2. Aprender la sintaxis básica para definir funciones en Python.
3. Comprender y aplicar la estructura de control switch-case en distintos lenguajes de programación para la toma de decisiones múltiples, identificando sus ventajas, desventajas y diferencias con if-else.
4. Explicar la teoría de switch-case, incluyendo su definición, propósito y características.
5. Comparar switch-case con if-else para comprender en qué situaciones es más conveniente.
6. Implementar ejemplos prácticos en distintos lenguajes de programación.
7. Resolver ejercicios para reforzar la comprensión y aplicación de switch-case.

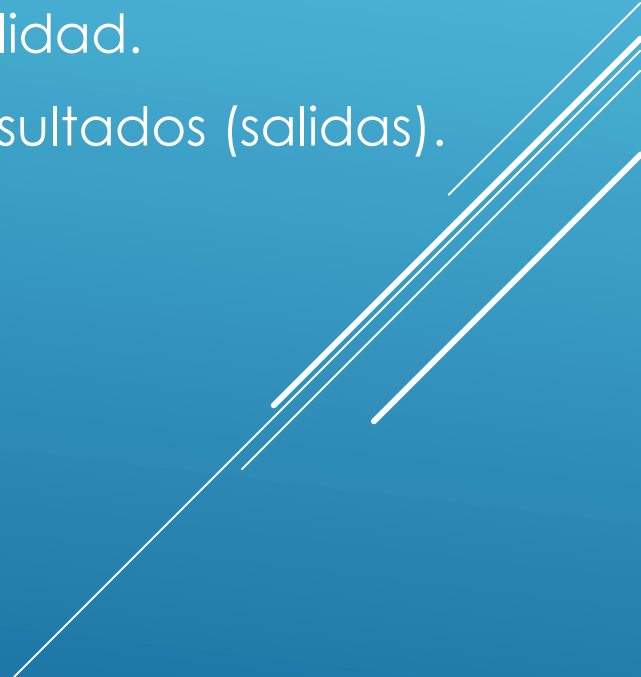
METODOLOGÍA

1. Se aplicará un enfoque constructivista donde los estudiantes aprenderán a través de la exploración y la resolución de problemas.
2. Se fomentará el aprendizaje activo mediante el desarrollo de ejercicios prácticos y proyectos sencillos.
3. Se utilizarán ejemplos del mundo real para contextualizar el uso de Python.
4. Se promoverá el trabajo colaborativo a través de discusiones y resolución de problemas en grupo.
5. Se incentivará la autoevaluación y el aprendizaje basado en errores para fortalecer la comprensión del lenguaje.

CONTENIDO

1. Introducción a funciones
 2. Concepto de switch-case
 3. Características principales
 4. Comparación entre switch-case y if-else.
 5. Ventajas y desventajas.
 6. Ejercicios prácticos.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

INTRODUCCIÓN A LAS FUNCIONES

- Una función es un bloque de código que se puede ejecutar cuando se le llama desde otro lugar en el programa.
 - Se utiliza para modularizar el código, reutilizarlo y mejorar la legibilidad.
 - Las funciones pueden recibir parámetros (entradas) y devolver resultados (salidas).
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

SINTAXIS DE LAS FUNCIONES EN PYTHON

Para definir una función en Python, usamos la palabra clave `def`, seguida por el nombre de la función, los parámetros entre paréntesis y, opcionalmente, el valor de retorno.

```
def nombre_de_la_funcion(parametros):  
    # Cuerpo de la función  
    return resultado
```

Ejemplo:

```
def saludo():  
    print("¡Hola, bienvenidos a la clase de algoritmia!")
```

Para llamar a la función:

```
saludo() # Imprime: ¡Hola, bienvenidos a la clase de algoritmia!
```

SINTAXIS DE LAS FUNCIONES EN PYTHON

Funciones con Parámetros y Retorno de Valores

```
def suma(a, b):  
    return a + b
```

Llamada de la función:

```
resultado = suma(5, 3) # resultado = 8  
print(resultado)
```

SINTAXIS DE LAS FUNCIONES EN PYTHON

Función que verifica si un número es par o impar:

```
def par_impar(numero):  
    if numero % 2 == 0:  
        return "Par"  
    else:  
        return "Impar"
```

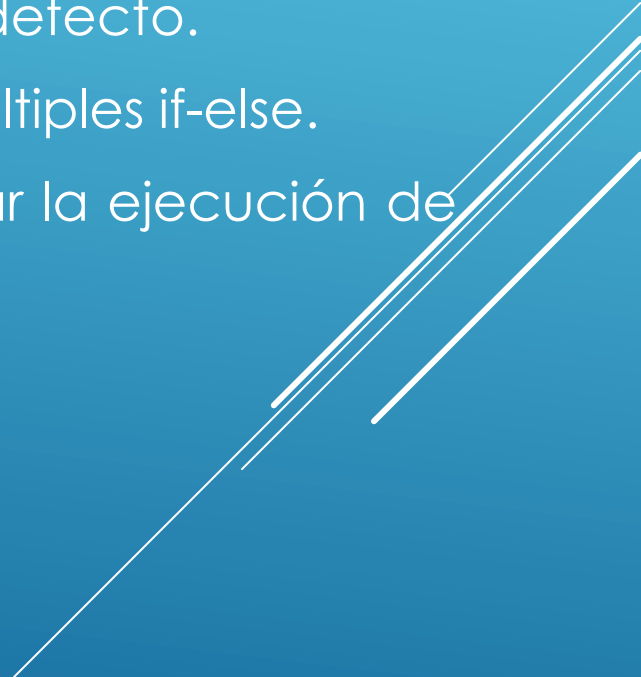
Llamada de la función:

```
print(par_impar(4)) # Imprime: Par  
print(par_impar(7)) # Imprime: Impar
```

¿QUÉ ES SWITCH-CASE?

- El switch-case es una estructura de control utilizada en programación para tomar decisiones basadas en el valor de una variable o expresión. Es una alternativa al uso de múltiples if-else, lo que hace que el código sea más legible y eficiente en ciertos casos.

CARACTERÍSTICAS DEL SWITCH-CASE

- Evalúa una expresión y compara su valor con diferentes case.
 - Ejecuta el bloque de código correspondiente al primer case coincidente.
 - Si no hay coincidencia, se puede usar default como opción por defecto.
 - Puede mejorar la legibilidad del código en comparación con múltiples if-else.
 - En algunos lenguajes, como C y Java, requiere break para evitar la ejecución de múltiples case consecutivos.
- 
- Several white lines of varying lengths and angles are drawn in the bottom right corner of the slide, creating a modern, abstract graphic element.

VENTAJAS Y DESVENTAJAS

- ✓ Mejora la legibilidad y organización del código.
- ✓ Puede ser más eficiente que múltiples if-else, ya que algunas implementaciones optimizan su ejecución.
- ✓ Útil cuando se tienen valores específicos a comparar.
- x Solo funciona con valores discretos (números, caracteres, constantes); no evalúa rangos de valores.
- x En algunos lenguajes, si se olvida el break, se pueden ejecutar varios case de forma no deseada.
- x Puede ser menos flexible que if-else, especialmente cuando las condiciones son complejas.

EJERCICIOS

1. Determinar si un número es positivo, negativo
2. Número par o impar: Escribe un programa que pida un número y determine si es par o impar.
3. Determinar el mayor de dos números: Pide al usuario dos números y muestra cuál es el mayor.
4. Solicitar dos números al usuario y mostrar su suma.
5. Escribir un programa que reciba tres números y determine cuál es el mayor de los tres.

NOTA: Todos utilizando def (funciones)

PREGUNTAS

