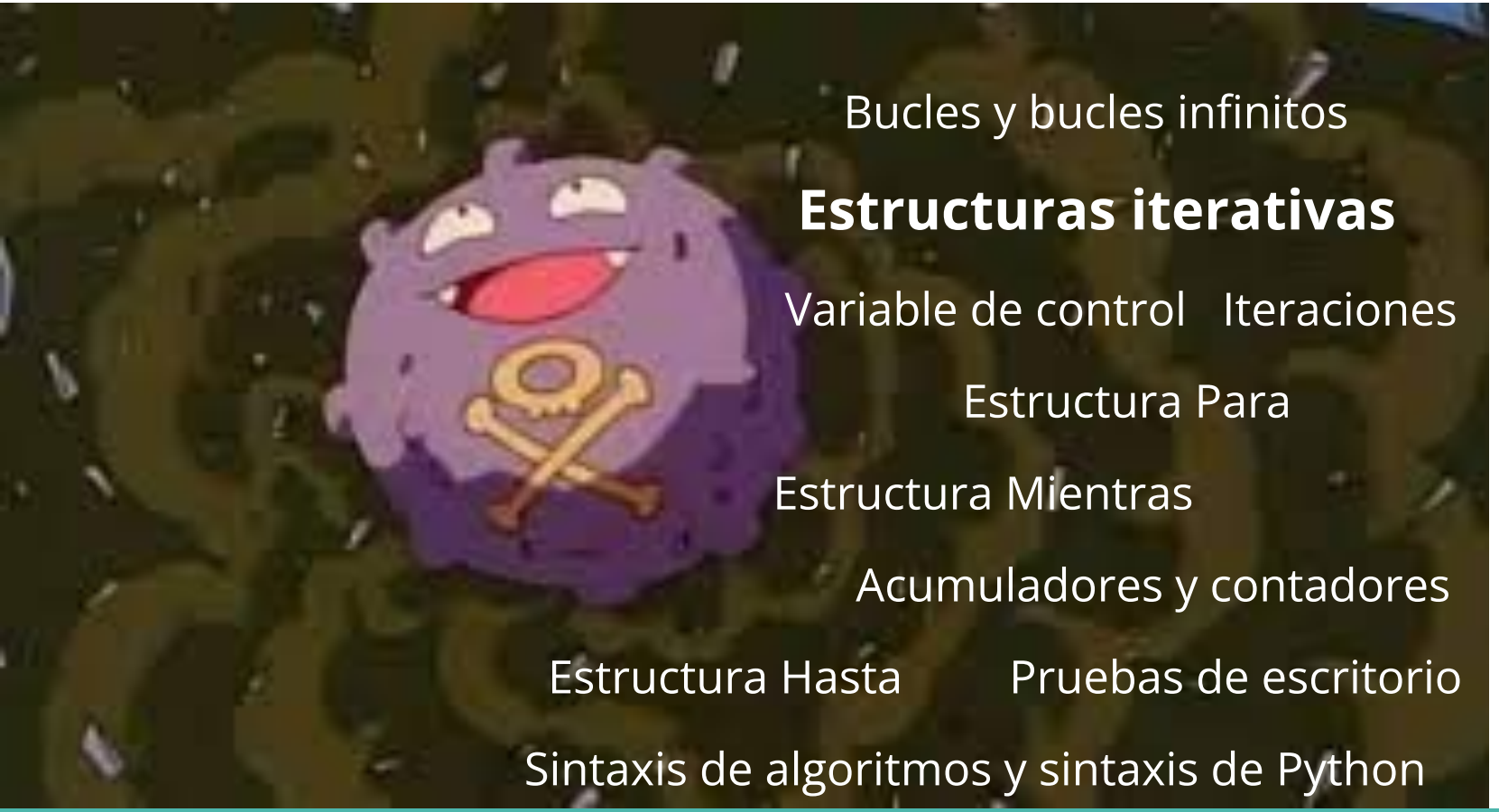
Fundamentos de Informática

Ing. Lic. Manuel Quintana

Wordcloud de hoy



Estructura iterativa

En la clase de hoy estudiaremos la estructura iterativa aprendiendo a resolver los ejercicios en forma algorítmica y luego traduciendolos al lenguaje de programación Python.

Hablamos de **estructuras iterativas**, porque el bloque de código a continuación se ejecutará repetidas veces siempre y cuando la condición aplicada a nuestra **variable de control** se cumpla.

Tal es el caso de las estructuras Mientras, Para y Hasta.

1. Estructura Para

Sintaxis de Algoritmos	Sintaxis de Python
Para <i> Desde <inicio> Hasta <tope> Con Paso <p> Hacer <Flujo que se itera> FinPara	for i in range(inicio, fin, paso): <flujo que se itera>

1. Estructura Para (continuación)

¿Cómo funciona esto?

- Para **i** → Se comienza definiendo una **variable de control.**
- Desde **inicio** → Se inicializa la variable de control.
- Hasta **tope** → Se define cual es el **valor de control**.
- Con Paso **p** → Se define cual es el **incremento** de la variable de control.
- A continuación se ejecuta el flujo.
- FinPara → Al llegar a este punto, se **incrementa la variable de control** en **p** unidades. A continuación la ejecución del código vuelve a la primera línea (Para i Desde ...) y se compara el valor de la variable de control con el valor de control. Si el tope no se superó, se continúa iterando; Caso contrario, se deja de iterar.

Práctica - Ejercicio 1 (Para)

Ejercicio: Se solicita crear un programa que muestre los números del 1 al 10 en pantalla.

1. **Resolución algorítmica**:

   i: Variable de tipo número entero

   Para i Desde 1 Hasta 10 Con Paso 1 Hacer

      Mostrar(i)

   FinPara

1. **Resolución Pythonesca**:

   for i in range(1,11,1): **# ¿Qué sucede acá?**

      print(i)

2. Estructura Mientras

Sintaxis de Algoritmos	Sintaxis de Python

variable = valor_inicial
Mientras variable < tope Hacer
  <Flujo a iterar>
  variable = variable + p
FinMientras
	variable = valor_inicial
	while variable < tope:
	  <Flujo a iterar>
	  variable = variable + p

2. Estructura Mientras (continuación)

¿Cómo funciona esto?

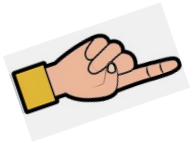
- variable = valor_inicial → Se inicializa la variable de control**.**
- Mientras (condicion) Hacer → Se valida la condición con la variable de control. Si se cumple, se ejecuta el flujo.
- Se ejecuta el flujo iterado
- Se actualiza la variable de control **¡Muy importante! ¿Por qué?**
- A continuación la ejecución del código vuelve a la primera línea (Mientras (condicion) Hacer) y se compara el valor de la variable de control con el valor de control. Si la condición se cumple, se continúa iterando; Caso contrario, se deja de iterar.

Práctica - Ejercicio 2 (Mientras)

Ejercicio: Se solicita crear un programa que muestre los números del 1 al 10 en pantalla pero **utilizando la estructura Mientras**.

Algoritmo	Python

i: Variable de tipo número entero
i = 1
Mientras i < 11 Hacer
  Mostrar(i)
  i = i + 1 # Incremento la variable de control
FinMientras
	i = 1
	while (i < 11):
	  print(i)
	  i = i + 1



Observación: Todo lo que se puede resolver con Para también se puede resolver con Mientras (la inversa no se cumple).

Contadores y acumuladores

Para cerrar la clase de hoy estudiaremos dos comportamientos que pueden tener las variables:

1. Contador: Una variable se comporta como **contador** si en cada iteración, al cumplirse una determinada condición, incrementa su valor en **una unidad**. Ejemplo: Contar la cantidad de veces que se ingresa el número 1 en una secuencia de números.
2. Acumulador: Una variable se comporta como **acumulador** si en cada iteración, al cumplirse una determinada condición, incrementa su valor en **un valor cualquiera.** Ejemplo: Sumar los números que un usuario ingresa por teclado.

Ej. de contador y acumulador

Se quiere contar la cantidad de veces que el usuario ingresa el número 1. Con 0 se corta la iteración.	**Se quiere mostrar el total de facturación del día. Se ingresan los montos de cada factura y con 0 se finaliza el ingreso de datos.**

numero: Variable de tipo número real
cont: Variable de tipo número entero
cont = 0 → cont es un “contador”
Mostrar(“Ingrese un numero (0 para finalizar):”)
Ingresar(numero)
Mientras (numero != 0) Hacer
  Si (numero == 1) Entonces
  cont = cont + 1
  FinSi
  Mostrar(“Ingrese un numero (0 para finalizar):”)
  Ingresar(numero)
FinMientras
Mostrar(“Cantidad de 1 ingresados: ”, cont)
	monto, suma: Variable de tipo número real
	suma = 0 → suma es un “acumulador”
	Mostrar(“Ingrese monto vendido (0 para finalizar):”)
	Ingresar(monto)
	
	Mientras (monto != 0) Hacer
	  suma = suma + monto
	  Mostrar(“Ingrese monto vendido (0 para finalizar):”)
	  Ingresar(monto)
	FinMientras
	
	Mostrar(“Total facturado en el dia: $”, suma)

Tabla de equivalencias Python - Algoritmos

Algoritmo	Python
Para <i> Desde <inicio> Hasta <tope> Con Paso <p> Hacer <Flujo que se itera> FinPara	for i in range(inicio, fin, paso): <flujo que se itera>

variable = valor_inicial
Mientras variable < tope Hacer
  <Flujo a iterar>
  variable = variable + p
FinMientras
	variable = valor_inicial
	while variable < tope:
	  <Flujo a iterar>
	  variable = variable + p