

</ Clase 01 fundamentos de la programación

/>

} /> [

Nombre: Alex Mellado
Gamboa

</ Table of contents

{01}

Introducción

{02}

Condicionales

{03}

Ejercicios a resolver



Introducción

01



</ ¿Hacer código o pensar lógico?

Es importante manejar la
lógica para crear código, ya
que en base a la lógica es
donde uno comprende más a la
hora de diseñar buen código



1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1

</ Importancia de los diagramas o papel />

Útiles para modelar un código a resolver o tener una idea base de cómo abordar un problema

Permiten visualizar de forma más detallada los datos que se deben tratar a código

Entonces... ¿Por dónde comenzar?

A modo de experiencia personal se ha optado por utilizar diagramas para diseñar el flujo del problema abordar, adicionalmente escribir a papel los detalles importantes

Condicionales

02

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

</ ¿Qué son los condicionales?

/> **

Condición

Estructura de control de decisiones a la hora de ejecutar un programa

Ejemplo: Un semáforo, controla el tráfico, ya siendo permitir o negar

} /> [

Ciclos

Ejecución de más de una vez una acción

Ejemplo: Sumatorias, series

Tipos de condicionales



If, else, elif

Estructura que obedece similar a un semáforo, es decir, existe una condición de corte, un caso “intermedio” y “otro caso”

Switch

Similar a una estructura del tipo if, else, elif, con la diferencia que se ve por casos

Tipos de ciclos



Ciclo while

Itera sobre un “contador” para satisfacer la condición, es más lento que el ciclo for

Itera sobre un conjunto booleano (True o false)

Ciclo for

Itera mucho más rápido que el ciclo while, no depende tanto de un “contador”

Itera sobre un conjunto de datos, listas incluso

Ejercicios a resolver de condiciones



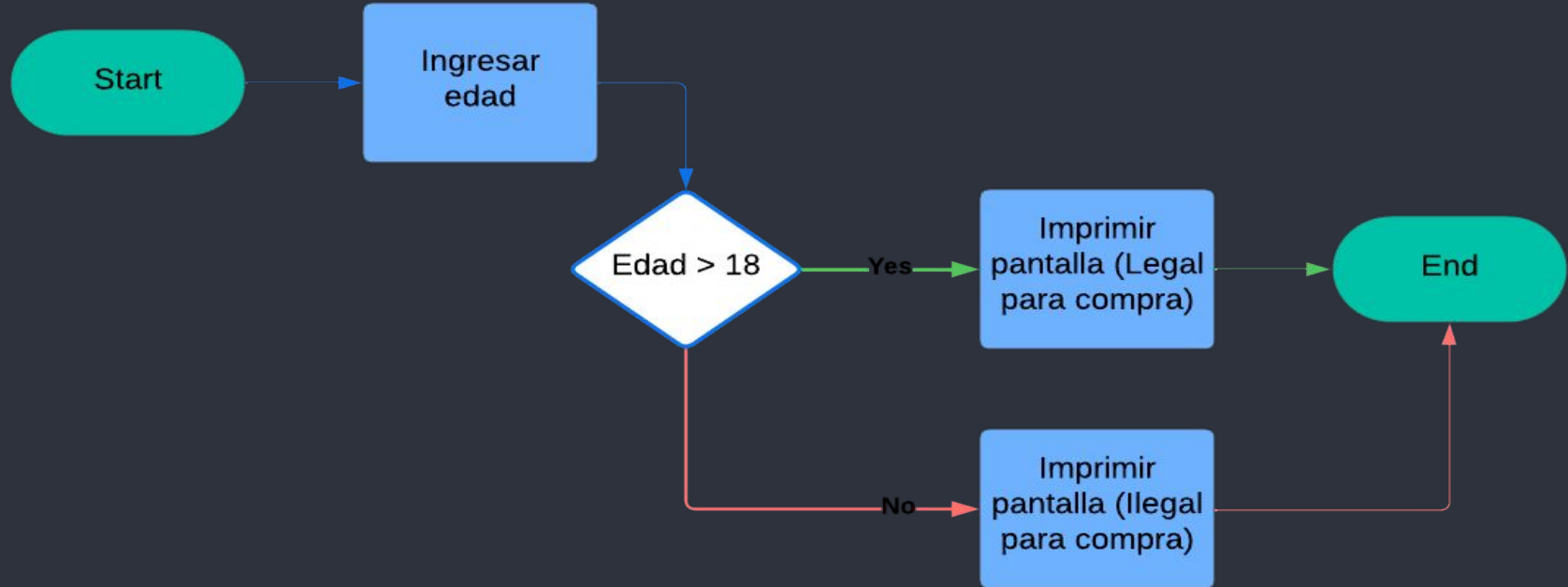
1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio 1

Escriba un programa en python que permita decidir sobre la edad para la venta de alcohol, solo debe imprimir por pantalla el resultado

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Diagrama



1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Código

```
3
4
5
6  # Ejercicio nº1: Compra y venta de alcohol en Python (Ejercicio para comprender)
7
8  #Ingreso de edad por pantalla, un numero entero
9  #Recordar también la primitiva de int (entero)
10 edad = int(input("Ingrese una edad: "))
11
12 #Ejemplo de una condición simple
13 if (edad >= 18):
14     print("Edad legal para comprar alcohol")
15 elif (edad == 17):
16     print("Necesita mostrar el carnet para comprar")
17 else:
18     print("Es menor de edad, prohibido el consumo de bebidas alcohólicas")
19
```

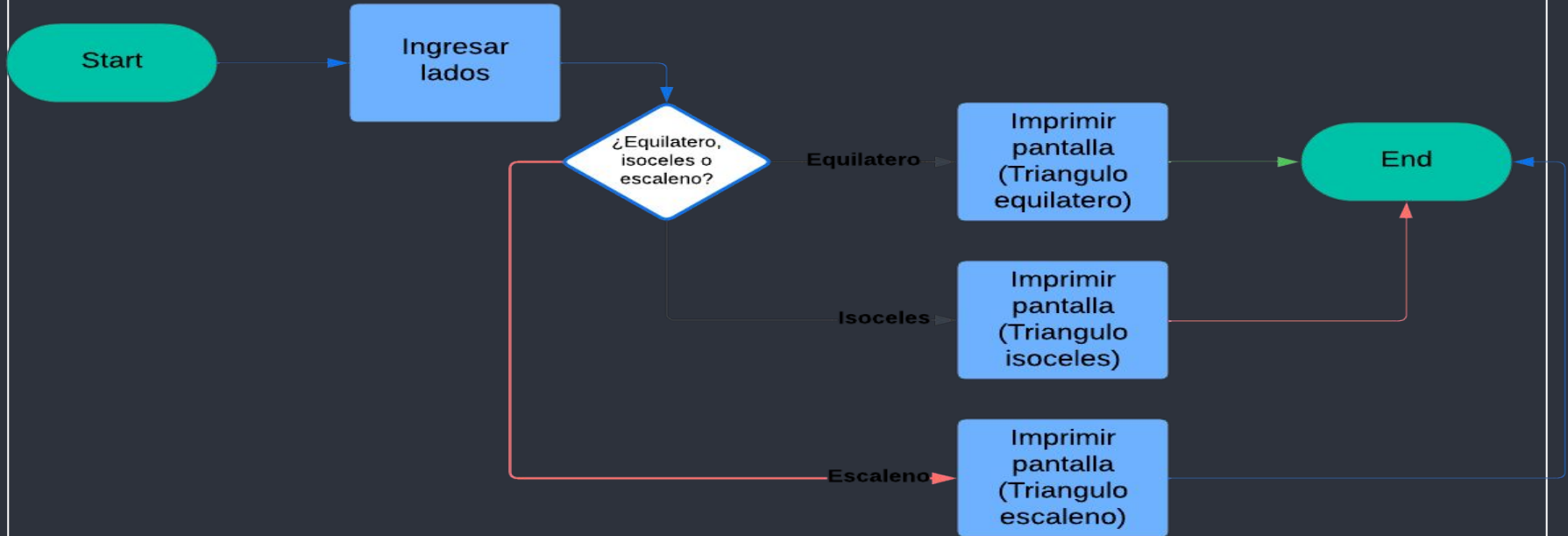
1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio nº2

Cree un programa que pueda mostrar qué tipo de triángulo a través de sus lados (equilatero, isosceles y escaleno)

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Diagrama



1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 0 1

Código

```
#Ejercicio n°2: Cree un programa en python en que se pueda mostrar que tipo de triangulo es

#Ingreso por pantalla de los tres lados de un triangulo
lado1 = int(input("Ingrese el primer lado: "))

#Ingreso del segundo lado
lado2 = int(input("Ingrese el segundo lado: "))

#Ingreso del tercer lado
lado3 = int(input("Ingrese el tercer lado: "))

#Como los numeros no pueden ser negativos
if(lado1 & lado2 & lado3 < 0):
    print("Los lados negativos no existen")

#Luego se genera la condicion para saber que tipo de triangulo es
if(lado1 == lado2 == lado3):
    print("Se tiene un triangulo equilatero")
elif(lado1 == lado2):
    print("Se tiene un triangulo isocetes")
else:
    print("Se tiene un triangulo escaleno")

#Se imprime por pantalla que tipo de triangulo es
```

Ejercicios con while

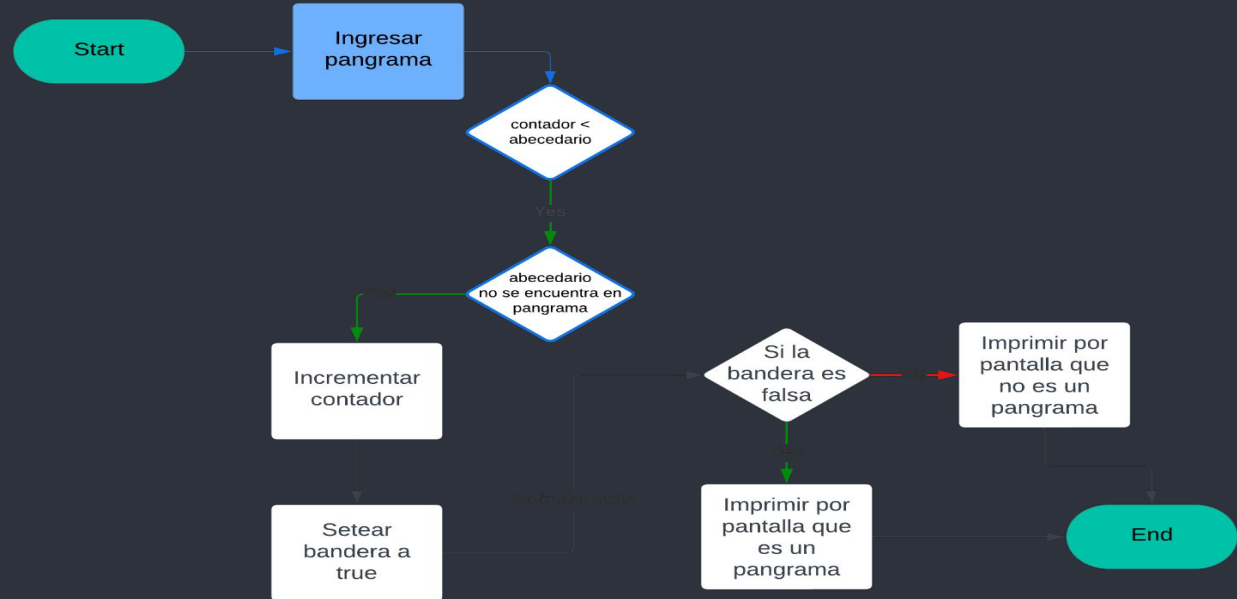
1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio nº3

Escriba código en python que permita reconocer si una frase es un pangrama, debe devolver un texto indicando si es o no un pangrama

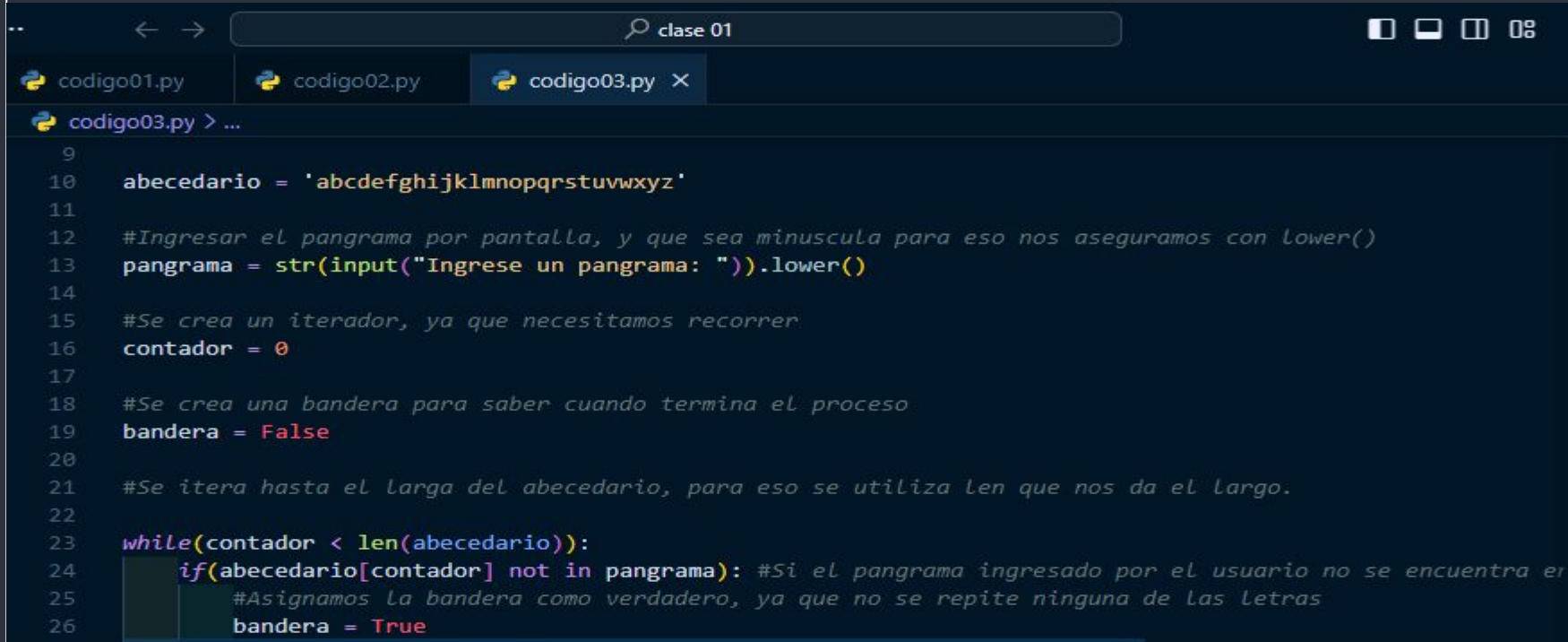
1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Diagrama



1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1

Código



The image shows a code editor interface with a dark theme. At the top, there's a search bar containing 'clase 01' and several window management icons on the right. Below the search bar, three tabs are visible: 'codigo01.py', 'codigo02.py', and 'codigo03.py' (which is active and has a close button). The main area displays the code for 'codigo03.py' with line numbers 9 through 26. The code is a Python script that checks if an input string is a pangram. It defines an alphabet string, takes user input, and iterates through the alphabet to see if each letter is present in the input. Comments in Spanish explain the steps: entering the pangram, creating an iterator, setting a flag, and iterating through the alphabet.

```
9
10 abecedario = 'abcdefghijklmnopqrstuvwxyz'
11
12 #Ingresar el pangrama por pantalla, y que sea minuscula para eso nos aseguramos con lower()
13 pangrama = str(input("Ingresa un pangrama: ")).lower()
14
15 #Se crea un iterador, ya que necesitamos recorrer
16 contador = 0
17
18 #Se crea una bandera para saber cuando termina el proceso
19 bandera = False
20
21 #Se itera hasta el largo del abecedario, para eso se utiliza len que nos da el largo.
22
23 while(contador < len(abecedario)):
24     if(abecedario[contador] not in pangrama): #Si el pangrama ingresado por el usuario no se encuentra en
25         #Asignamos la bandera como verdadero, ya que no se repite ninguna de las letras
26         bandera = True
```

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

```
        #Asignamos la bandera como verdadero, ya
        bandera = True
    #Incrementamos el contador
    contador = contador + 1

#Salimos del ciclo y hacemos las verificaciones

if(bandera == False):
    #Indicamos que si es un pangrama
    print("Si es un pangrama")
else:
    print("No es un pangrama")
```

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Ejercicio nº4

Escriba un código en python que realice la sumatoria de gauss con cualquier número ingresado por pantalla

Código

```
# Ejercicio nº4 sumatoria de Gauss en Python

# Primero necesitamos el número ingresado
numero = int(input('Ingrese un número: '))

# Necesitamos un contador o iterador para realizar la sumatoria
contador = 1
sumatoria = 0

#Se recorre desde el contador hasta el numero ingresado
while contador <= numero:
    # Se suma cada valor del contador a la sumatoria
    sumatoria += contador
    #Se incrementa el contador
    contador += 1

#Se muestra por pantalla la sumatoria
print('La sumatoria es:', sumatoria)
```