

Institución Universitaria Pascual Bravo

Herramientas de Programación I

Juan Camilo Cruz Peláez

CC : 1216718181

Grupo : 051

correo : juan.cruz181@pascualbravo.edu.co

Desarrollo de la tarea 1

2. Los dos **IDEs** de mi preferencia, de acuerdo con lo expuesto en clase, son:

1) Visual Studio

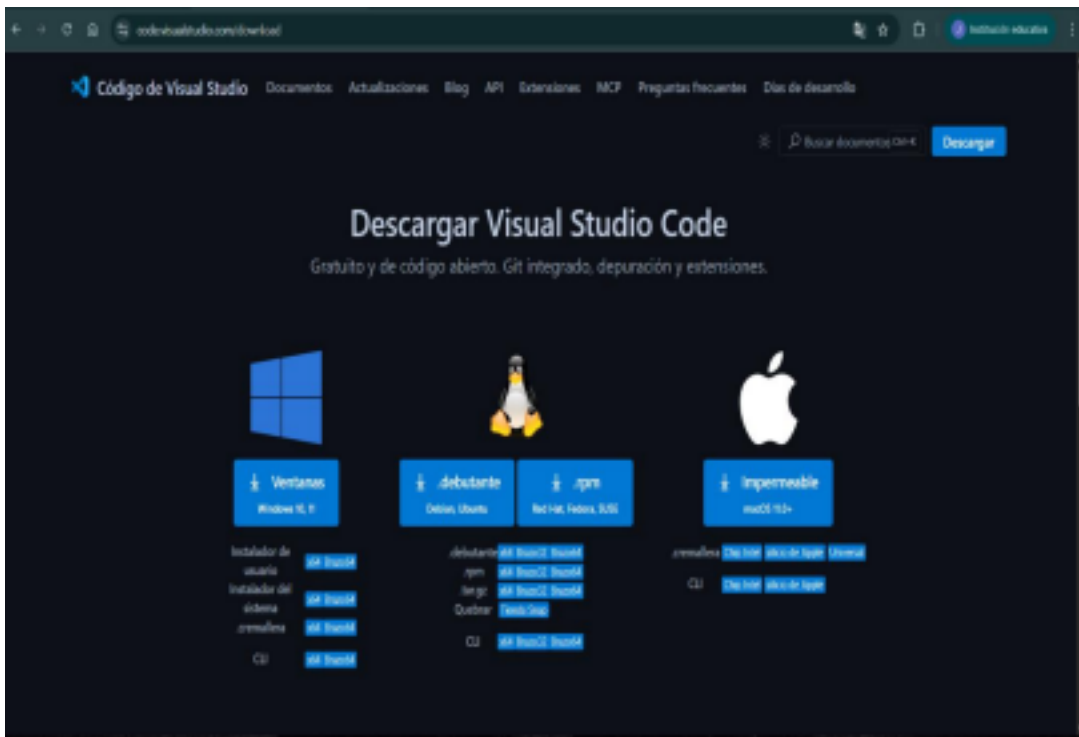
Visual Studio es un IDE desarrollado por Microsoft. Permite crear aplicaciones en C++, C#, Python y otros lenguajes, ofreciendo depuración, compilación y un entorno muy completo para proyectos grandes.

2) Visual Studio Code

Visual Studio Code, también de Microsoft, es un editor de código liviano y multiplataforma. Soporta múltiples lenguajes mediante extensiones y es ideal para proyectos más ágiles y variados.

En los siguientes pantallazos demuestro en dónde se encuentra la página de descarga de los IDEs escogidos (**Visual Studio** y **Visual Studio Code**).

Visual Studio Code.



En la página oficial de Microsoft se encuentra la sección de descargas, donde se ofrecen versiones para Windows, macOS y Linux.

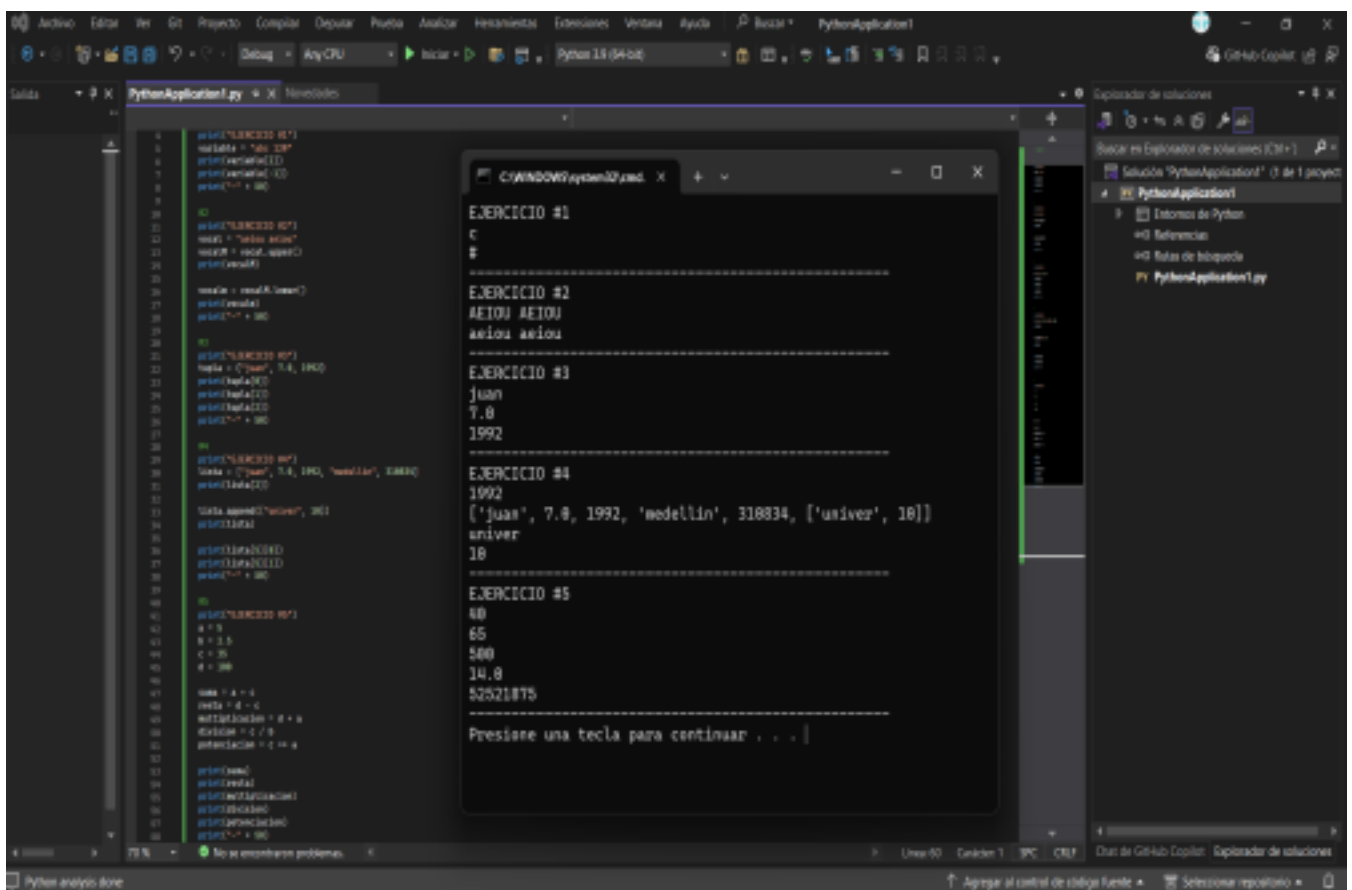
Visual Studio 2022



En la página oficial <https://visualstudio.microsoft.com/es/downloads/> se encuentra la sección de descargas de Visual Studio 2022. Desde allí se pueden descargar las ediciones **Comunidad**, **Profesional** y **Empresa**.

3. A continuación, se muestran los pantallazos de los programas ejecutados en ambos IDEs, donde se evidencia el código funcional y su respectiva salida en consola.

1) Ejercicios hechos en visual studio 2022.



The screenshot displays the Visual Studio 2022 interface with a Python file named 'PythonApplication1.py'. The code is organized into five exercises, each starting with a print statement and followed by calculations and prints. The output window on the right shows the execution results for each exercise, separated by dashed lines. The exercises involve basic arithmetic, string manipulation, and list operations.

```
1 print("EJERCICIO #1")
2 resultado = "40. 120"
3 print(resultado)
4 print(resultado * 10)
5 print("..." + resultado)
6
7
8
9
10
11 print("EJERCICIO #2")
12 resultado = "AEIOU aeioa"
13 resultado = resultado.upper()
14 print(resultado)
15 print("..." + resultado)
16
17
18
19
20
21 print("EJERCICIO #3")
22 nombre = "Juan", 5.4, 1992
23 print(resultado)
24 print(resultado)
25 print(resultado)
26 print("..." + resultado)
27
28
29
30
31
32 print("EJERCICIO #4")
33 datos = ("Juan", 5.4, 1992, "medellin", 318834)
34 print(resultado)
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Output:

```
EJERCICIO #1
40. 120
400. 1200
...40. 120

EJERCICIO #2
AEIOU AEIOU
AEIOU AEIOU

EJERCICIO #3
Juan
5.4
1992

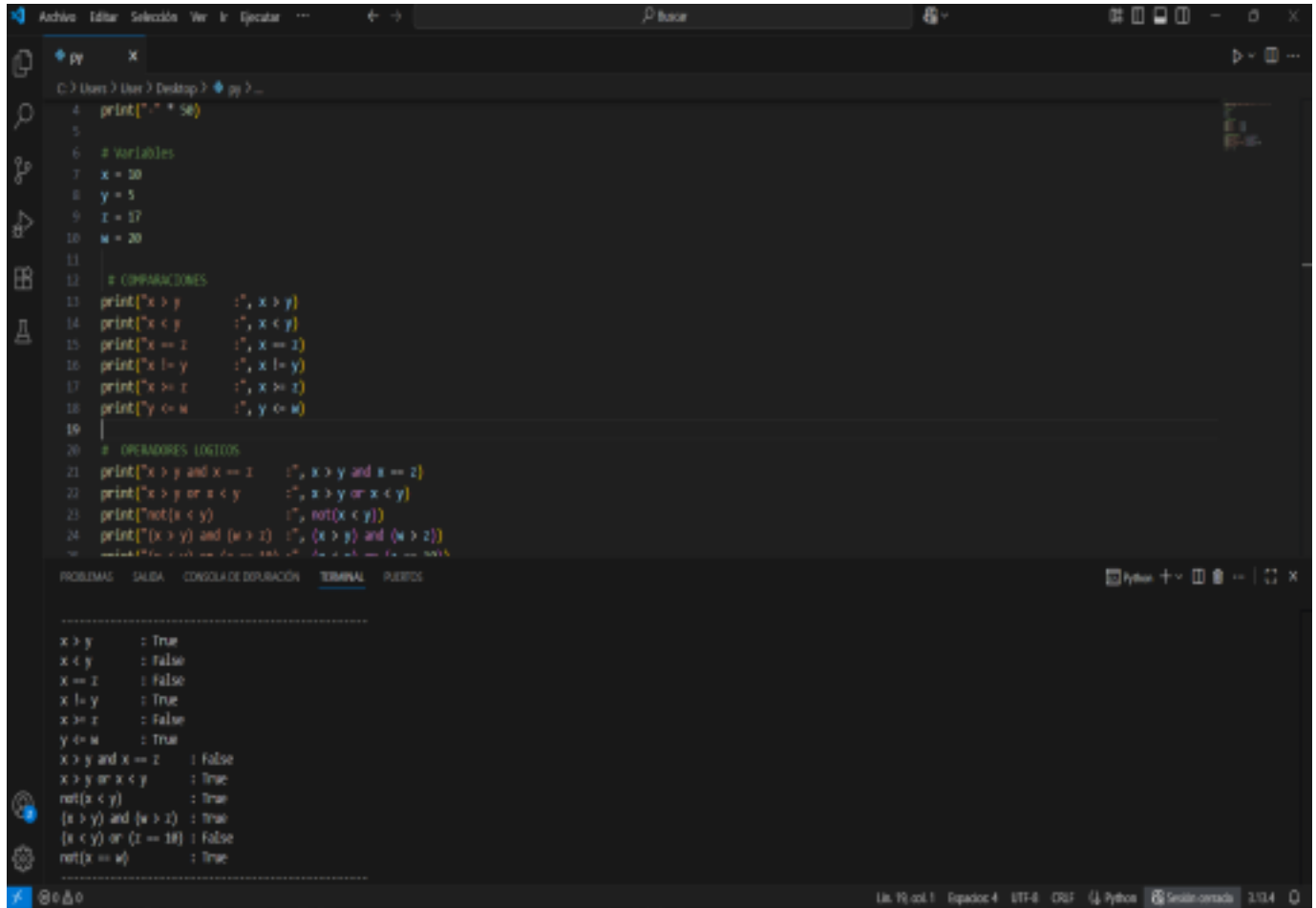
EJERCICIO #4
1992
['Juan', 5.4, 1992, 'medellin', 318834, ['univer', 10]]
univer
10

EJERCICIO #5
40
65
500
14.8
52521875

Presione una tecla para continuar . . .
```

En estos pantallazos se presentan los ejercicios desarrollados durante la clase utilizando el entorno de desarrollo Visual Studio 2022. Se incluyen los pantallazos del código y de su ejecución, evidenciando la práctica de los conceptos vistos en clase.

2) Ejercicios hechos en Visual Studio Code



The screenshot shows the Visual Studio 2022 interface with a Python file open. The code defines variables x=10, y=5, z=17, and w=20. It then tests various comparison and logical operators. The terminal output shows the results of these tests.

```
4 print(" " * 50)
5
6 # Variables
7 x = 10
8 y = 5
9 z = 17
10 w = 20
11
12 # COMPARACIONES
13 print("x > y      :", x > y)
14 print("x < y      :", x < y)
15 print("x == z     :", x == z)
16 print("x != y     :", x != y)
17 print("x >= z     :", x >= z)
18 print("y <= w     :", y <= w)
19
20 # OPERADORES LOGICOS
21 print("x > y and x == z   :", x > y and x == z)
22 print("x > y or x < y    :", x > y or x < y)
23 print("not(x < y)       :", not(x < y))
24 print("(x > y) and (w > z) :", (x > y) and (w > z))
25
```

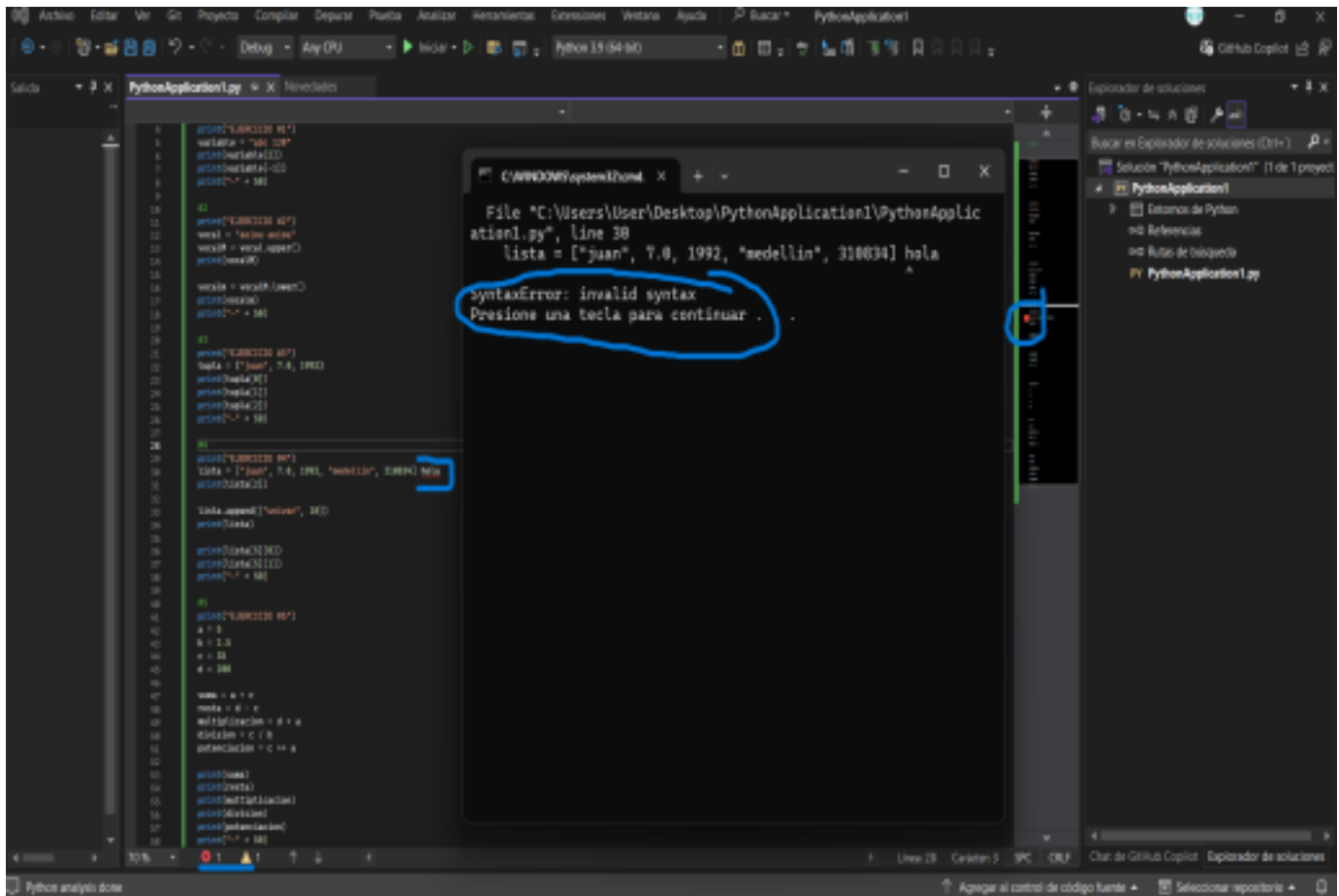
```
-----
x > y      : True
x < y      : False
x == z     : False
x != y     : True
x >= z     : False
y <= w     : True
x > y and x == z   : False
x > y or x < y    : True
not(x < y)       : True
(x > y) and (w > z) : True
(x < y) or (z == 18) : False
not(x == w)      : True
-----
```

En este pantallazo se desarrollaron ejercicios enfocados en el uso de **operadores lógicos y comparaciones** en Python. Se trabajaron ejemplos con los operadores and, or, not y con comparadores como >, <, >=, <=, ==, !=.

En los pantallazos se puede observar el código escrito y su salida en consola, lo cual permite comprender cómo estos operadores permiten tomar decisiones y evaluar condiciones dentro de un programa.

4.

Error intencional en Visual Studio 2022



En este ejercicio se introdujo de manera intencional un error de sintaxis dentro del código en Python. Al ejecutar el programa, Visual Studio muestra el error en la consola de salida (SyntaxError: invalid syntax) indicando la línea en la que ocurrió el problema.

Este ejemplo demuestra cómo el IDE identifica y señala los errores en el código, permitiendo al programador corregirlos más fácilmente.

Error intencional en Visual Studio Code

```
1 # Variables
2 x = 10
3 y = 5
4 z = 12
5 w = 20
6
7 # COMPARACIONES
8 print("x > y" if x > y)
9 print("x < y" if x < y)
10 print("x == z" if x == z)
11 print("x != y" if x != y)
12 print("x >= z" if x >= z)
13 print("y <= w" if y <= w)
14
15 g1225
16
17 # OPERADORES LOGICOS
18 print("x > y and x == z" if x > y and x == z)
19 print("x > y or x < y" if x > y or x < y)
20 print("not(x < y)" if not(x < y))
21 print("(x > y) and (w > z)" if (x > y) and (w > z))
22 print("(x < y) or (z == 10)" if (x < y) or (z == 10))
23 print("not(x == w)" if not(x == w))
```

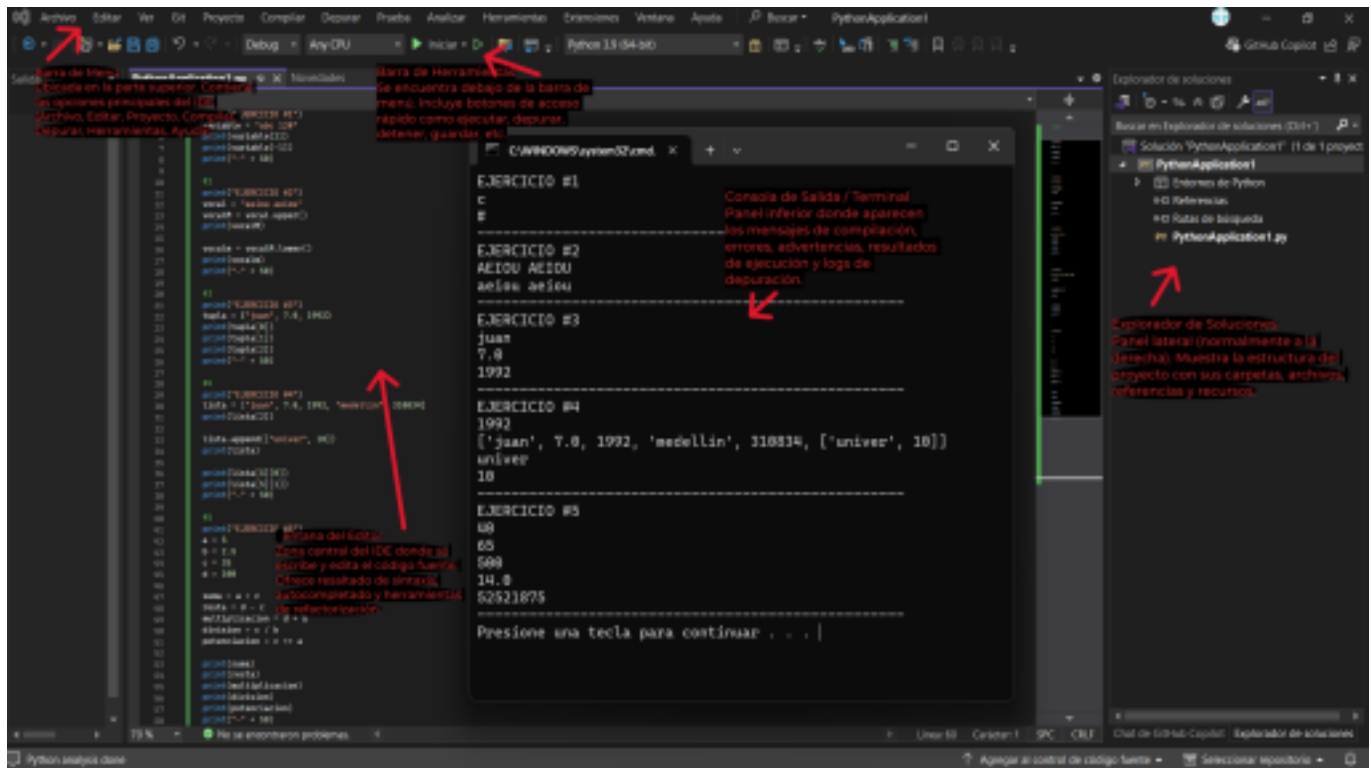
Traceback (most recent call last):
File "C:/Users/user/Desktop/py", line 20, in module
g1225
NameError: name 'g1225' is not defined

En este ejercicio se insertó intencionalmente un error escribiendo una variable inexistente (g1225).

Al ejecutar el programa, Visual Studio Code muestra en la terminal el mensaje de error: *NameError: name 'g1225' is not defined*.

Este resultado permite evidenciar cómo el IDE identifica que la variable no está definida y señala la línea del código donde ocurre el problema.

5. Pantallazos de los componentes principales de los dos IDEs: Visual Studio 2022



Visual Studio Code

