



Activad UT2

Programación

Alfonso García Jorge

1ºDAWNA

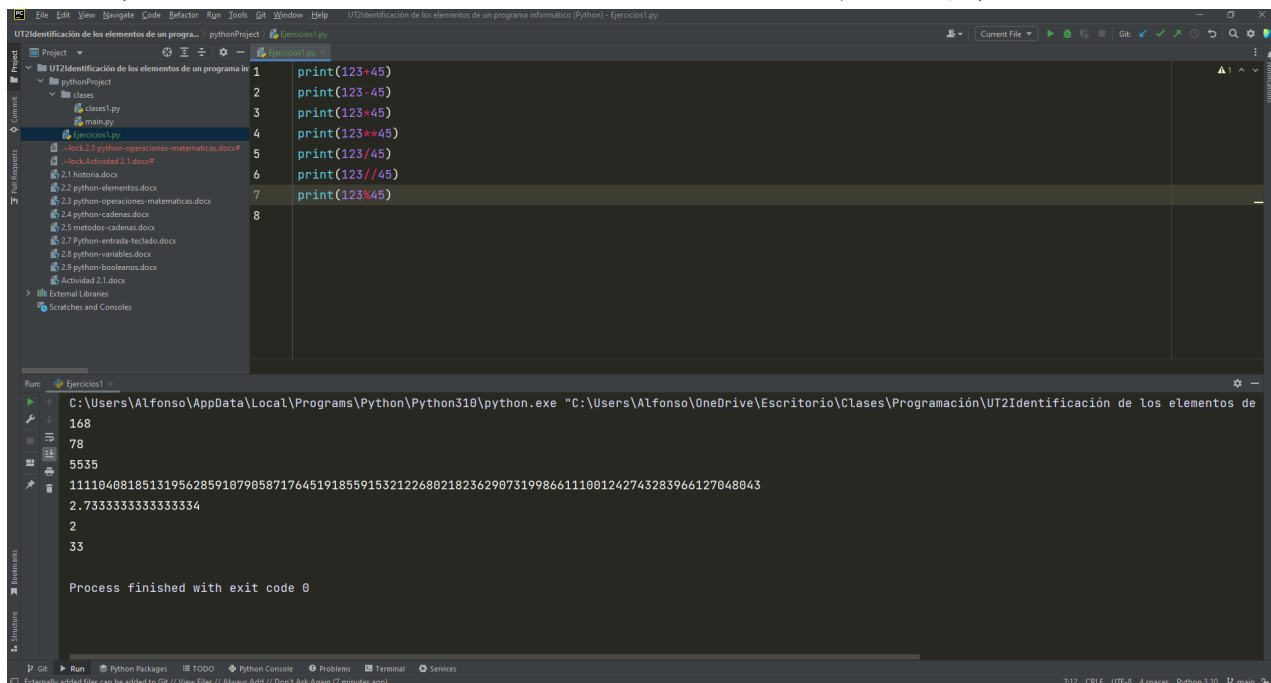
Actividad 2.1

Operaciones con números

1.- Ejecutar en PyCharm la sentencia `print(123+45)`, y comprobar que el resultado es entero (no tiene punto decimal).

Repetir en cada uno de los siguientes, reemplazando `+` (adición) por el operador indicado y ver el tipo de resultado que se obtiene (si tiene o no coma decimal).

- a) `-` (resta, es decir, calcular $123 - 45$),
- b) `*` (multiplicación, en matemáticas « 123×45 »),
- c) `**` (exponenciación, en matemáticas « 123^45 »)
- d) `/` (división),
- e) `//` (división con cociente entero),
- f) `%` (resto de la división con cociente entero, ;no confundir con porcentaje!).



The screenshot shows the PyCharm IDE with a Python file named `Ejercicios1.py` containing the following code:

```
1 print(123+45)
2 print(123-45)
3 print(123*45)
4 print(123**45)
5 print(123/45)
6 print(123//45)
7 print(123%45)
```

The Run console at the bottom displays the output of these operations:

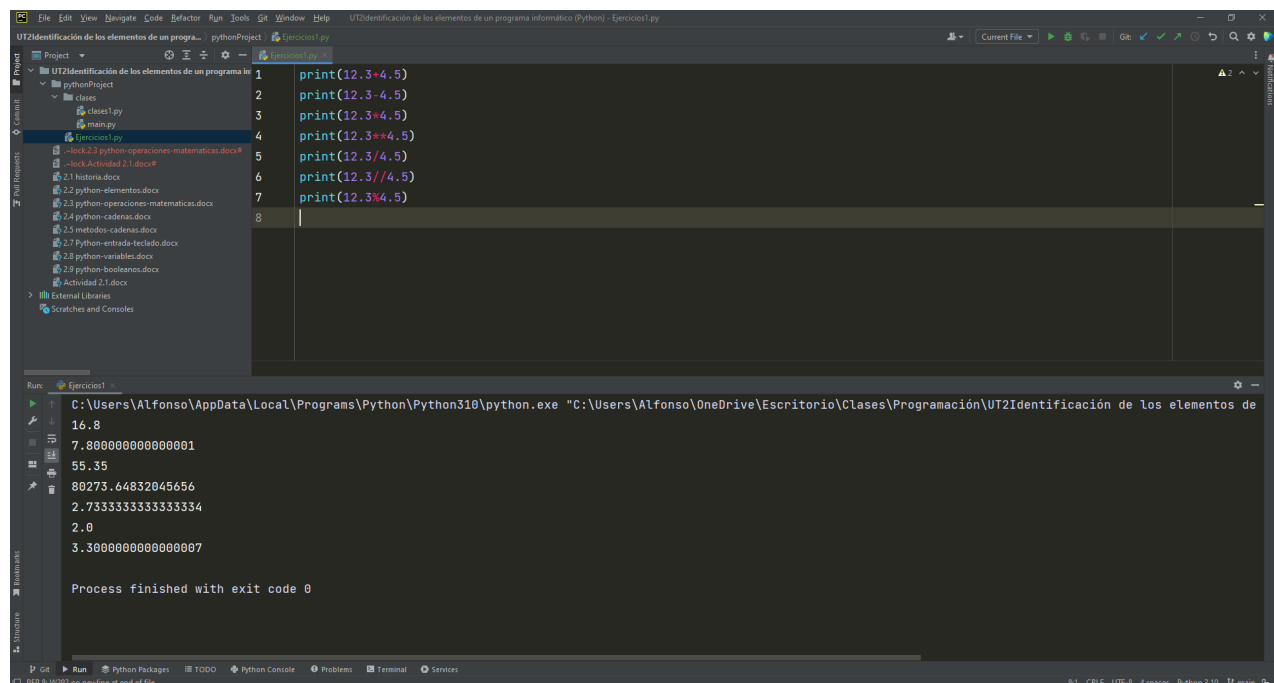
```
168
78
168
5535
11110408185131956285910790587176451918559153212268021823629073199866111001242743283966127048043
2.7333333333333334
2
33

Process finished with exit code 0
```

Observar que, excepto el caso de la división `123 / 45`, todos los resultados son enteros (no tienen coma decimal).

2.- Introducir en PyCharm los apartados del ejercicio anterior considerando 12.3 y 4.5 (en vez de, respectivamente, 123 y 45), y ver si los resultados tienen o no coma decimal.

Observar que el resultado de $12.3 // 4.5$ es 2.0 y no 2.



```
1 print(12.3+4.5)
2 print(12.3-4.5)
3 print(12.3*4.5)
4 print(12.3**4.5)
5 print(12.3/4.5)
6 print(12.3//4.5)
7 print(12.3%4.5)
8
```

Run console output:

```
16.8
7.800000000000001
55.35
80273.64832045656
2.7333333333333334
2.0
3.3000000000000007

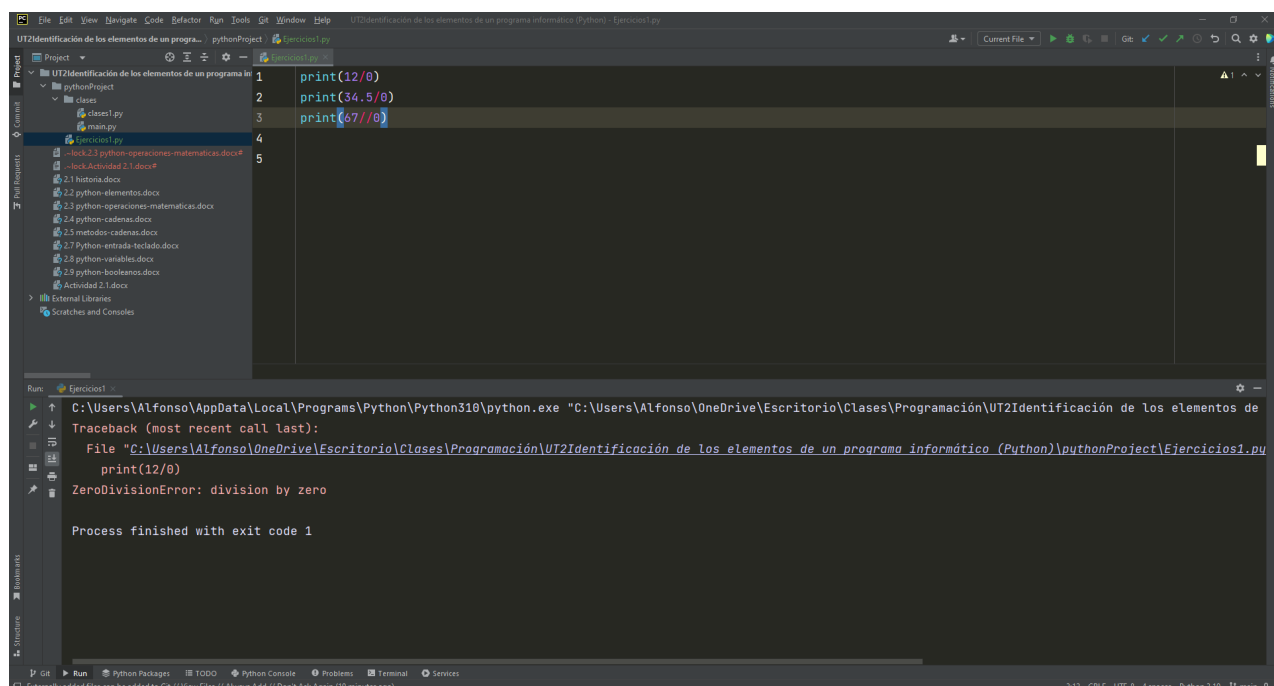
Process finished with exit code 0
```

3.- Introducir en PyCharm las siguientes operaciones. ¿Qué pasa si ponemos cualquiera de las siguientes?.

1. $12 / 0$

2. $34.5 / 0$

3. $67 // 0$



```
1 print(12/0)
2 print(34.5/0)
3 print(67//0)
4
5
```

Run console output:

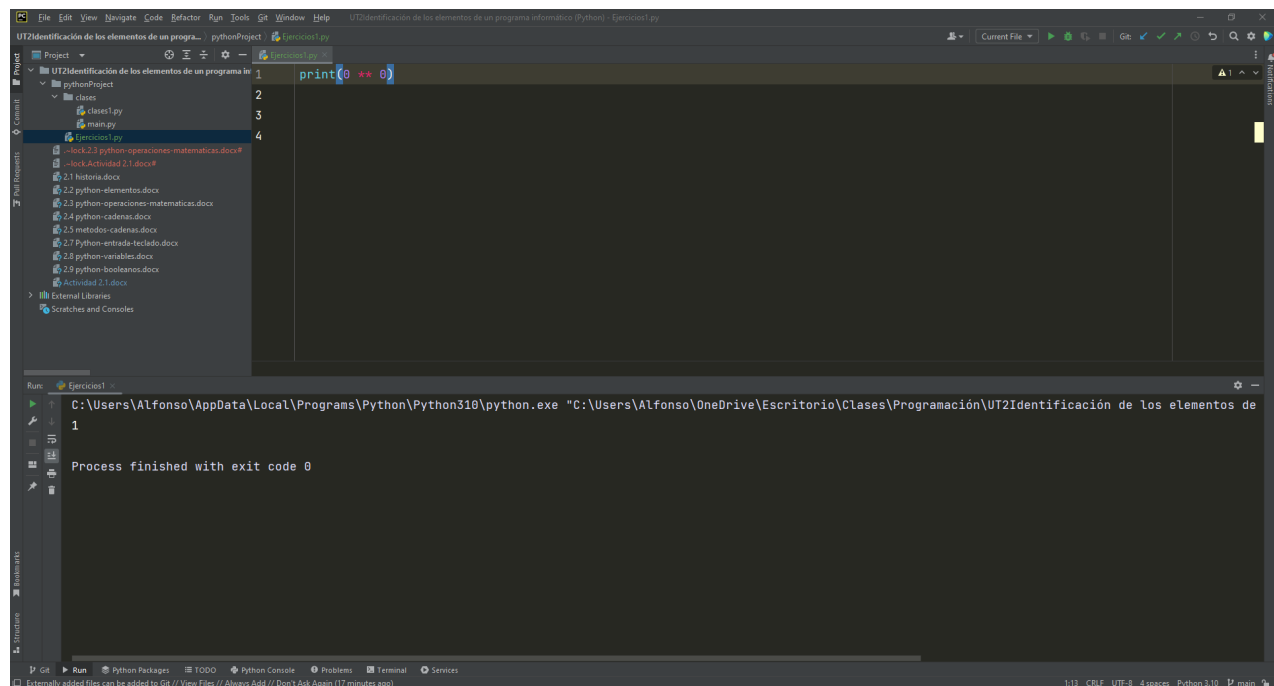
```
Traceback (most recent call last):
  File "C:\Users\Alfonso\OneDrive\Escritorio\Clases\Programación\UT2Identificación de los elementos de un programa informático (Python)\pythonProject\Ejercicios1.py", line 1, in <module>
    print(12/0)
ZeroDivisionError: division by zero

Process finished with exit code 1
```

Nos dá un error ya la división entre cero es una operación que matemáticamente no está definida.

4.- ¿Cuánto es 0^0 (cero a la cero) según las matemáticas? ¿Y según Python?. Ejecutar en PyCharm la operación anterior.

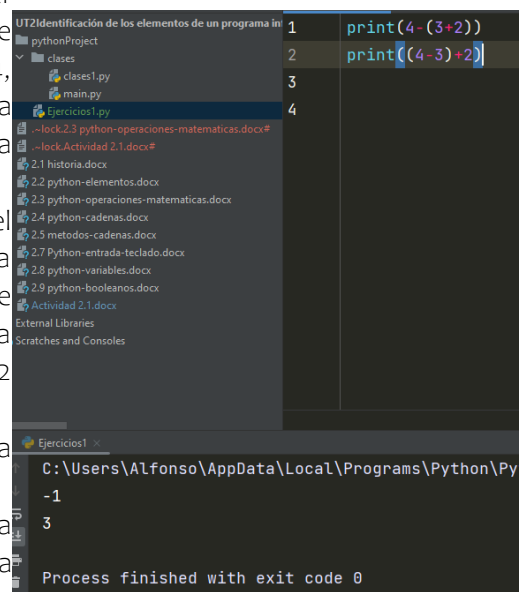
El resultado cero a la cero según las matemáticas es 1.



5.- Ejecutar en PyCharm las operaciones siguientes. Si tenemos más de una operación, podemos agrupar con paréntesis como hacemos en matemáticas. Ejecutar las siguientes instrucciones, comprobando que los resultados son los esperados.

1. `print(4-(3+2))` 2. `print((4-3)+2)`

- Cuando no usamos paréntesis, tenemos que tener cuidado con la precedencia (cuál se aplica primero) de los operadores. Por ejemplo, si ponemos $2 + 3 \times 4$, sabemos que tenemos que calcular primero 3×4 y a eso agregarle 2, o sea, « \times » tiene mayor precedencia que « $+$ ».
- No es claro cómo evaluar $2 - 3 + 4 - 5$. Parte del problema es que el signo « $-$ » se usa tanto para indicar la resta como para indicar el inverso aditivo de un número. También ayuda a la confusión el que la suma es conmutativa ($2 + 3 = 3 + 2$), pero la resta no ($2 - 3 \neq 3 - 2$).
- Algo similar pasa con la expresión $3/4 \times 5$: no queda claro si nos referimos a $(3/4) \times 5$ o a $3/(4 \times 5)$.
- Ante la ausencia de paréntesis, Python evalúa expresiones como $2 - 3 + 4 - 5$ y $3 / 4 * 5$ de izquierda a derecha, esto es, como si hiciera los reemplazos antes de la evaluación.



$$a - b \rightarrow a + (-b) \quad \text{y} \quad a/b \rightarrow a \times b^{-1}$$

6.- Introducir en PyCharm esta operación. $-3^{**}4$, ¿es equivalente en matemáticas a $(-3)^{-4}$ o a -3^{-4} ?
 ¿Cuáles son las precedencias de Python en este caso?

No es el mismo resultado ya que $-3^{**}4$ primero hace la raíz

The screenshot shows the PyCharm IDE with a file named 'Ejercicios1.py'. The code in the editor is:

```
1 print(-3**4)
2
3 print((-3)**4)
4
```

The Run window at the bottom shows the execution output:

```
-0.812345679812345678
0.812345679812345678
Process finished with exit code 0
```

En el curso trataremos de evitar construcciones como la de los ejercicios anteriores, agregando paréntesis aunque sean redundantes.

7.- Ejecutar en PyCharm las siguientes operaciones. Desde las matemáticas, ¿es $\sqrt{3}$ entero?, ¿y $\sqrt{4}$? ¿De qué tipo son $3^{**}(1/2)$ y $4^{**}(1/2)$? (Razonar las respuesta)

The screenshot shows the PyCharm IDE with a file named 'Ejercicios1.py'. The code in the editor is:

```
1 print(3**(1/2))
2
3 print(4**(1/2))
4
```

The Run window at the bottom shows the execution output:

```
1.7320508075688772
2.0
Process finished with exit code 0
```

$\sqrt{3} = 1.7320508075688772$

$\sqrt{4} = 2$

Al dividir python siempre lo hace en decimal por eso el resultado siempre será decimal dividiendo.

Puedes utilizar // si necesitas el resultado de forma entera.

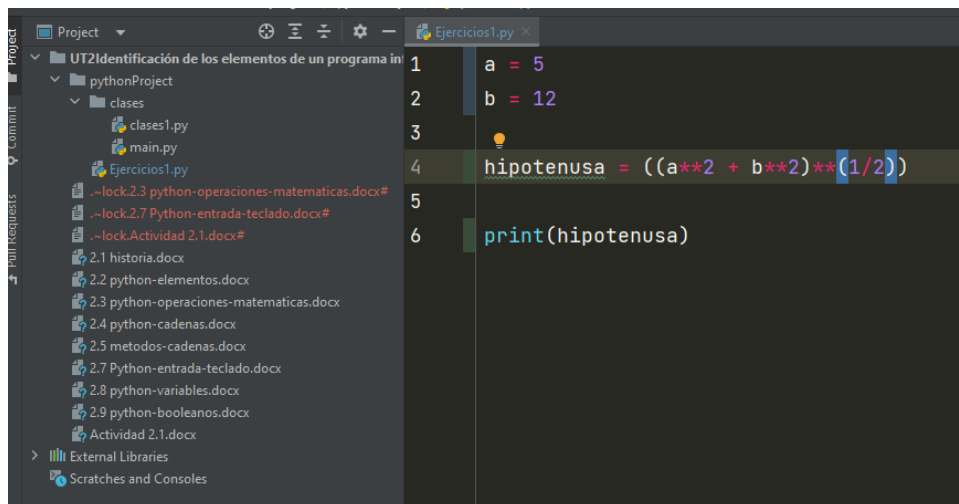
The screenshot shows the PyCharm IDE with a file named 'Ejercicios1.py'. The code in the editor is:

```
1 print(3**(1//2))
2
3 print(4**(1//2))
4
```

The Run window at the bottom shows the execution output:

```
1
2
Process finished with exit code 0
```

8.- En un triángulo rectángulo, un cateto mide 12 y el otro 5. Calcular cuánto mide la hipotenusa usando Python.

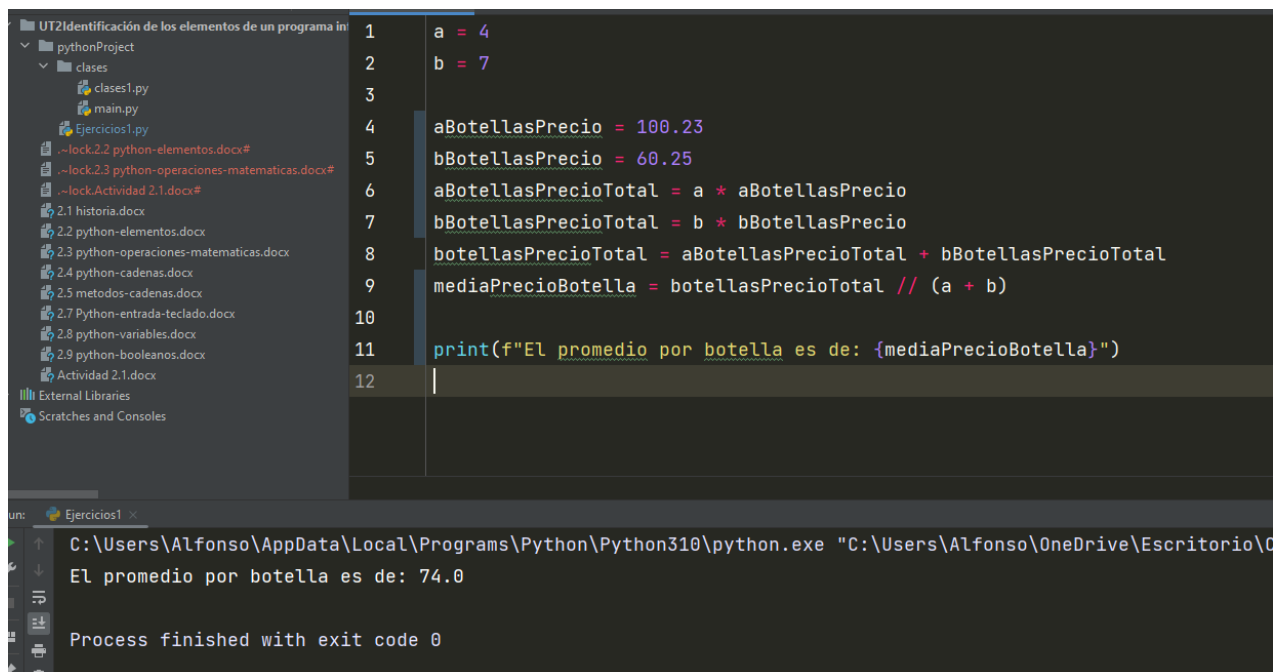


```
1 a = 5
2 b = 12
3
4 hipotenusa = ((a**2 + b**2)**(1/2))
5
6 print(hipotenusa)
```

9.- Compré 4 botellas a 100,23€ cada una y 7 a 60,25€ cada una.

Resolver con lápiz y papel (o mentalmente) y luego hacer las cuentas con Python:

- a) ¿Cuántas botellas compré? Pues compré $4 + 7$
- b) ¿Cuánto gasté en total? Gasté $100,23 \times 4$ (400,92) y $60,25 \times 7$ (421,75) En total sería 822,67
- c) En promedio, ¿cuál fue el costo por botella? La media por botella sería el total partido por el total de botellas que son 11. Sería 74,79 la media de la botella.



```
1 a = 4
2 b = 7
3
4 aBotellasPrecio = 100.23
5 bBotellasPrecio = 60.25
6 aBotellasPrecioTotal = a * aBotellasPrecio
7 bBotellasPrecioTotal = b * bBotellasPrecio
8 botellasPrecioTotal = aBotellasPrecioTotal + bBotellasPrecioTotal
9 mediaPrecioBotella = botellasPrecioTotal // (a + b)
10
11 print(f"El promedio por botella es de: {mediaPrecioBotella}")
12
```

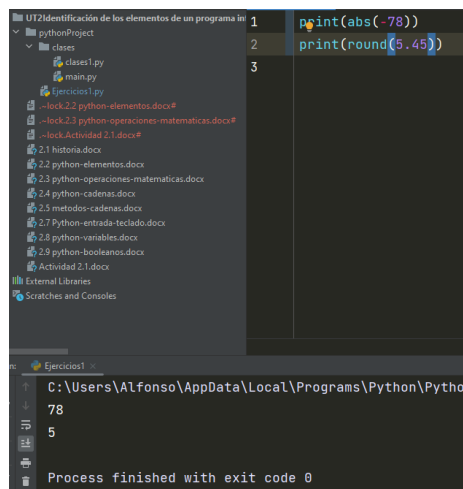
Run: Ejercicios1

```
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Escritorio\0
El promedio por botella es de: 74.0

Process finished with exit code 0
```

10.- Ejecutar en PyCharm y las siguientes funciones. Además de las operaciones entre números, podemos usar algunas funciones como el valor absoluto de x, $|x|$, que se escribe `abs(x)` en Python, o el redondeo de decimal a entero, `round(x)`, que nos da el entero más próximo a x.

a) Ejecutar en PyCharm `abs` y luego repetir para `round`.



```
1 print(abs(-78))
2 print(round(5.45))
3
```

78
5
Process finished with exit code 0

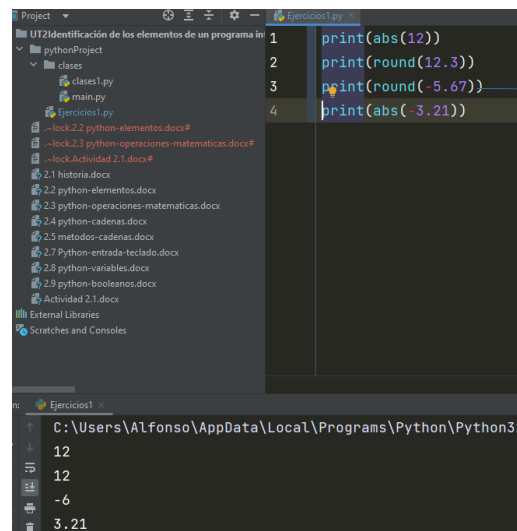
b) Resolver y analizar el resultado: (No hace falta comentar nada sino darse cuenta de cómo funcionan estas funciones en PyCharm)

1. `print(abs(12))` 2. `print(round(12.3))` 3. `print(round(-5.67))` 4. `print(abs(-3.21))`

c) Evaluar:

1. `abs` 2. `round`

d) y observar que al poner una función (como `abs`) sin argumento, Python responde diciendo que es una función (en este caso, propia).



```
1 print(abs(12))
2 print(round(12.3))
3 print(round(-5.67))
4 print(abs(-3.21))
```

12
12
-6
3.21

Nos redondea a 6 pero nos da el numero en negativo

e) Python diferencia entre mayúsculas y minúsculas. Probar con los siguientes, viendo que da error:

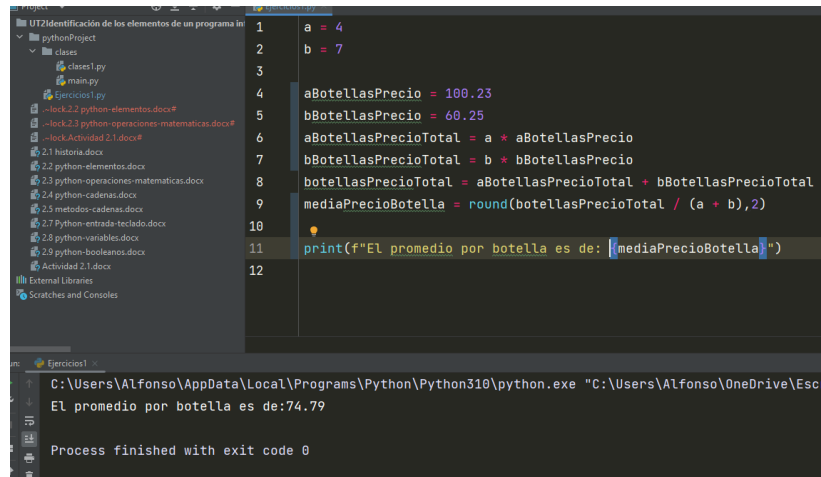
1. `print(Abs(2))` 2. `print(ABS(2))`

```
Traceback (most recent call last):
  File "C:\Users\Alfonso\OneDrive\Escritorio\Clases\Programación\UT2Identificación de los elementos de un programa informático (Python)\pythonProject\Ejercicios1.py", line 1, in <module>
    print(Abs(2))
NameError: name 'Abs' is not defined. Did you mean: 'abs'?
```

11.- round admite un argumento opcional indicando la cantidad de decimales deseados para el redondeo.

Evaluar en PyCharm:

1. print(round(123.4567, 3)) 2. print(round(765.4321, 2))
- Escribir el promedio calculado en el ejercicio 1.9.c) redondeado a céntimos.



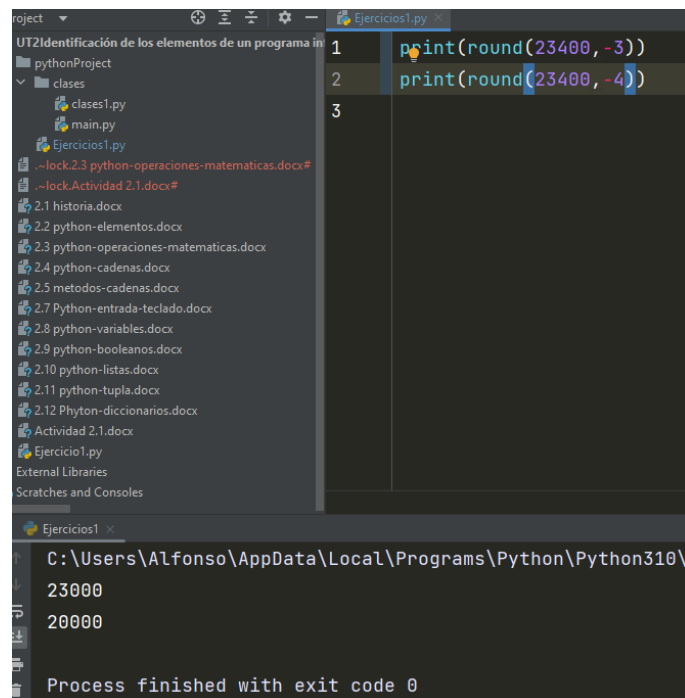
The screenshot shows the PyCharm IDE with a project named 'UT2Identificación de los elementos de un programa in'. The file explorer on the left shows a folder 'pythonProject' containing 'clases' and 'main.py'. The main editor shows a Python script 'Ejercicios1.py' with the following code:

```
1 a = 4
2 b = 7
3
4 aBotellasPrecio = 100.23
5 bBotellasPrecio = 60.25
6 aBotellasPrecioTotal = a * aBotellasPrecio
7 bBotellasPrecioTotal = b * bBotellasPrecio
8 botellasPrecioTotal = aBotellasPrecioTotal + bBotellasPrecioTotal
9 mediaPrecioBotella = round(botellasPrecioTotal / (a + b), 2)
10
11 print(f"El promedio por botella es de: {mediaPrecioBotella}")
12
```

The console output shows the execution of the script:

```
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Escriba
El promedio por botella es de:74.79
Process finished with exit code 0
```

- El precio de cierto auto o km es de 23400€: expresarlo redondeado a unidades de 1000 (mil) y de 10000 (diez mil) usando round.



The screenshot shows the PyCharm IDE with a project named 'UT2Identificación de los elementos de un programa in'. The file explorer on the left shows a folder 'pythonProject' containing 'clases' and 'main.py'. The main editor shows a Python script 'Ejercicios1.py' with the following code:

```
1 print(round(23400, -3))
2 print(round(23400, -4))
3
```

The console output shows the execution of the script:

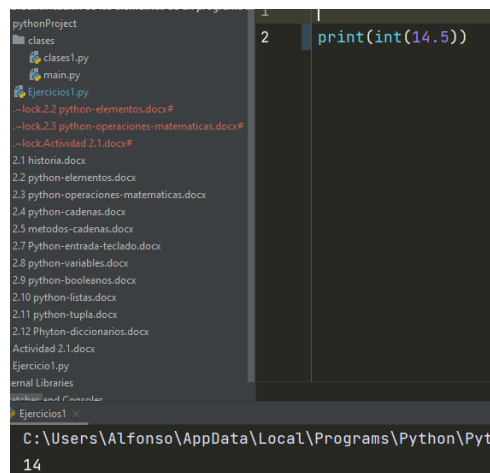
```
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Escriba
23000
20000
Process finished with exit code 0
```


12.- Cuando x es un número, type(x) nos dice si x es entero (int) o decimal (float) según Python.
Ver los resultados de las siguientes instrucciones ejecutándolas en PyCharm:

- a) print(type(12)) b) print(type(12.3))
c) print(round(12.3)) d) print(type(round(12.3)))
e) print(98 // 76)) f) print(type(98 // 76))
g) print(98.0 // 76.0) h) print(type(98.0 // 76.0))

- int viene de integer, o entero. float viene de floating point (punto flotante), el nombre de la codificación para números decimales se estudiarán más adelante.

Es posible pasar de uno a otro tipo de número usando int (para pasar de float a int) o float (para pasar de int a float). Claro que al pasar de decimal a entero perdemos los decimales después de la coma.

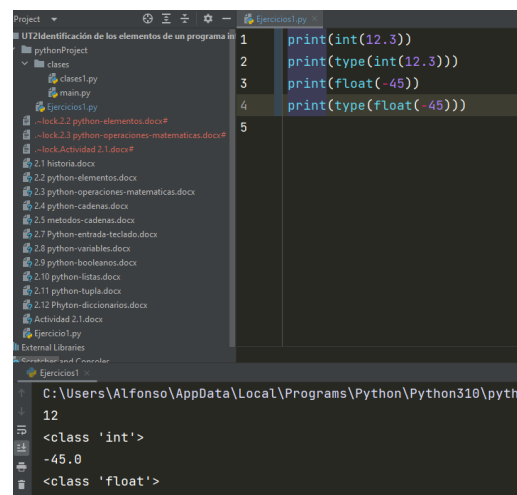


```
pythonProject
├── clases
│   ├── clases1.py
│   └── main.py
├── Ejercicios1.py
├── ~lock.2.2 python-elementos.docx#
├── ~lock.2.3 python-operaciones-matematicas.docx#
├── ~lock.Actividad 2.1.docx#
├── 2.1 historia.docx
├── 2.2 python-elementos.docx
├── 2.3 python-operaciones-matematicas.docx
├── 2.4 python-cadenas.docx
├── 2.5 metodos-cadenas.docx
├── 2.7 Python-entrada-teclado.docx
├── 2.8 python-variables.docx
├── 2.9 python-booleans.docx
├── 2.10 python-listas.docx
├── 2.11 python-tupla.docx
├── 2.12 Python-diccionarios.docx
├── Actividad 2.1.docx
├── Ejercicio1.py
└── External Libraries

Ejercicios1.py
14
```

13.- Analizar los resultados en PyCharm ejecutando las instrucciones de: (Comentar por qué se producen)

- a) print(int(12.3)) b) print(type(int(12.3)))
c) print(float(-45)) d) print(type(float(-45)))



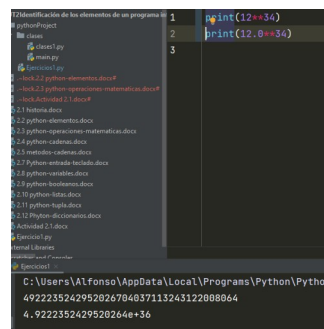
```
Project: UT2Identificación de los elementos de un programa in
pythonProject
├── clases
│   ├── clases1.py
│   └── main.py
├── Ejercicios1.py
├── ~lock.2.2 python-elementos.docx#
├── ~lock.2.3 python-operaciones-matematicas.docx#
├── ~lock.Actividad 2.1.docx#
├── 2.1 historia.docx
├── 2.2 python-elementos.docx
├── 2.3 python-operaciones-matematicas.docx
├── 2.4 python-cadenas.docx
├── 2.5 metodos-cadenas.docx
├── 2.7 Python-entrada-teclado.docx
├── 2.8 python-variables.docx
├── 2.9 python-booleans.docx
├── 2.10 python-listas.docx
├── 2.11 python-tupla.docx
├── 2.12 Python-diccionarios.docx
├── Actividad 2.1.docx
├── Ejercicio1.py
└── External Libraries

Ejercicios1.py
1 print(int(12.3))
2 print(type(int(12.3)))
3 print(float(-45))
4 print(type(float(-45)))
5
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python
12
<class 'int'>
-45.0
<class 'float'>
```

El a) esta convirtiendo el float en int, el b) nos dice el tipo de numero que es, el c) está convirtiendo el negativo en float y el d) esta diciendo el tipo de número que es.

14.- Python puede trabajar con enteros de cualquier tamaño, pero los tamaños de los decimales es limitado.

- a) Calcular `print(12**34)` y luego `print(12.0**34)`, y observar que Python representa al decimal con una variante de notación científica incorporando la letra e para indicar una potencia de 10:

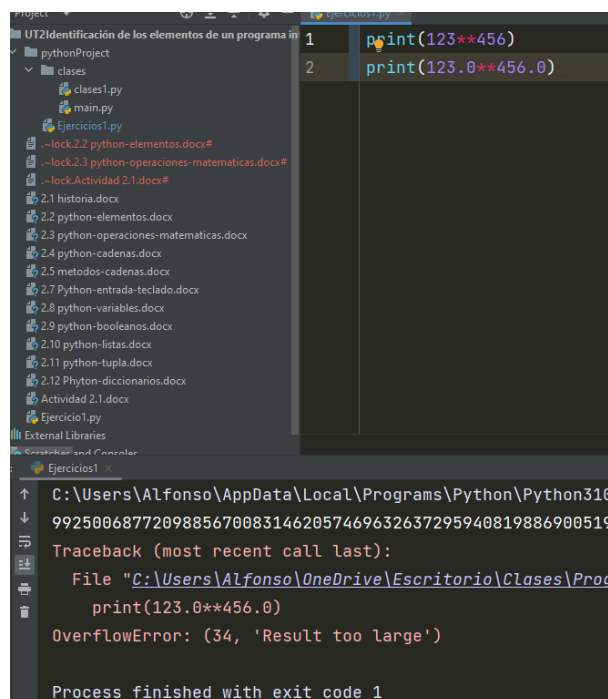


```
1 print(12**34)
2 print(12.0**34)
3
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310
492235242952027e+36
4.92235242952027e+36
```

- b) $4.92235242952027e+36 \leftrightarrow 4.92235242952027 \times 10^{36}$.

a. ¡No debe confundirse este e con el número e = 2.71828 ... !

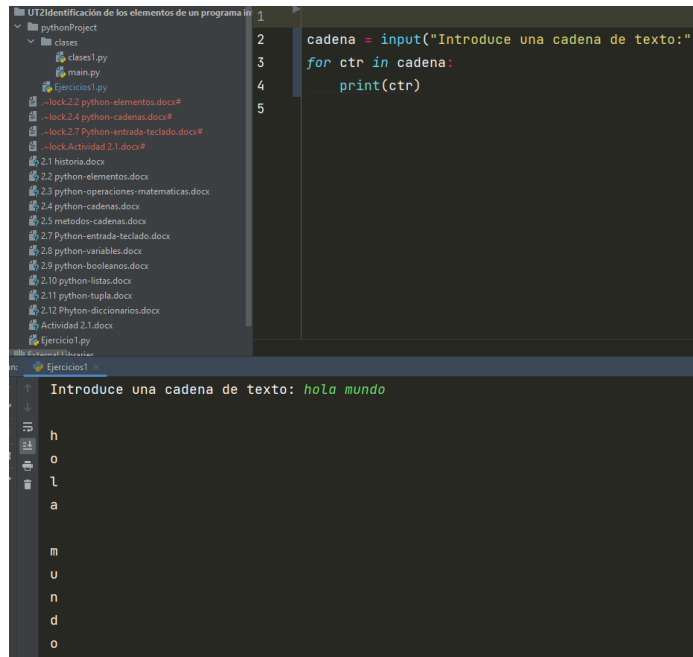
- c) Del mismo modo, poner `print(1 / 12**34)` y ver cómo se representa en Python. Nos pone e-37, que es elevado a -37
- d) Escribir en la notación de Python: 12.3×10^{45} y 12.3×10^{-45} (sin usar * ni ** sino e), observando cómo los representa Python.
- e) Poner `print(123**456)` viendo que obtenemos un número muy grande.
- f) Poner `print(123.0**456.0)` viendo que obtenemos un error (el mensaje de error indica Result too large o resultado demasiado grande.)



```
1 print(123**456)
2 print(123.0**456.0)
3
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310
99250068772098856700831462057469632637295940819886900519
99250068772098856700831462057469632637295940819886900519
Traceback (most recent call last):
  File "C:\Users\Alfonso\OneDrive\Escritorio\Clases\Prog
    print(123.0**456.0)
OverflowError: (34, 'Result too large')
Process finished with exit code 1
```

Operaciones con cadenas

15.- Escribir por pantalla cada carácter de una cadena introducida por teclado.



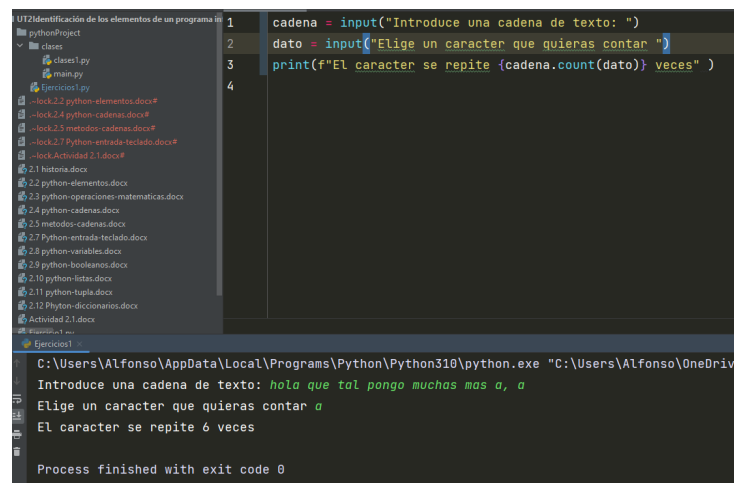
```
1
2 cadena = input("Introduce una cadena de texto:")
3 for ctr in cadena:
4     print(ctr)
5
```

Introduce una cadena de texto: hola mundo

h
o
l
a

m
u
n
d
o

16.- Pide una cadena y un carácter por teclado y muestra cuantas veces aparece el carácter en la cadena.



```
1 cadena = input("Introduce una cadena de texto: ")
2 dato = input("Elige un caracter que quieras contar ")
3 print(f"El caracter se repite {cadena.count(dato)} veces" )
4
```

C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Documents\Ejercicios1\16.py"

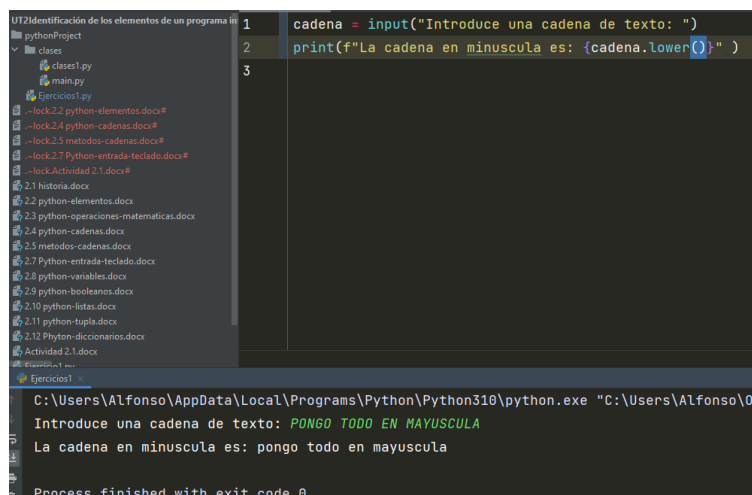
Introduce una cadena de texto: hola que tal pongo muchas mas a, a

Elige un caracter que quieras contar a

El caracter se repite 6 veces

Process finished with exit code 0

17.- Realizar un programa que dada una cadena de caracteres, genere otra cadena con todos sus caracteres en minúscula.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"La cadena en minúscula es: {cadena.lower()}")
3
```

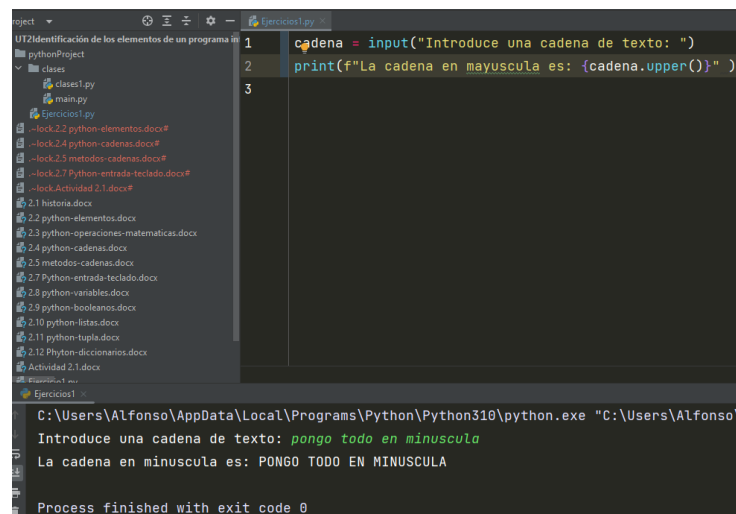
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Documents\Ejercicios1\17.py"

Introduce una cadena de texto: PONGO TODO EN MAYUSCULA

La cadena en minúscula es: pongo todo en mayuscula

Process finished with exit code 0

- 18.- Realizar un programa que dada una cadena de caracteres, genere otra cadena con todos sus caracteres en mayúscula.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"La cadena en mayuscula es: {cadena.upper()}")
3
```

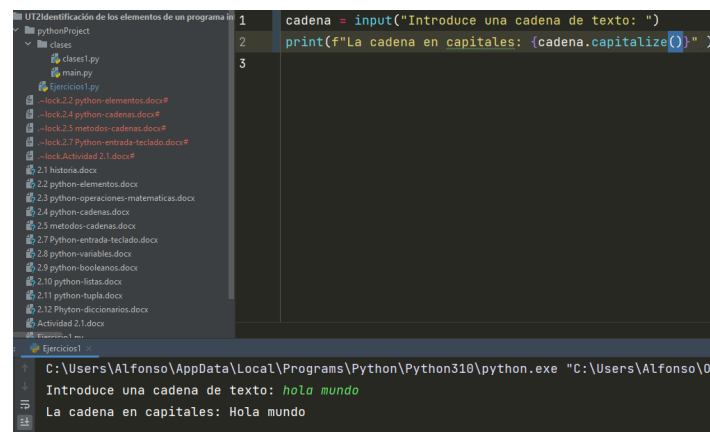
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\On..."

Introduce una cadena de texto: *pongo todo en minuscula*

La cadena en minuscula es: PONGO TODO EN MINUSCULA

Process finished with exit code 0

- 19.- Realizar un programa que dada una cadena de caracteres, genere otra cadena con todos sus caracteres en capital.



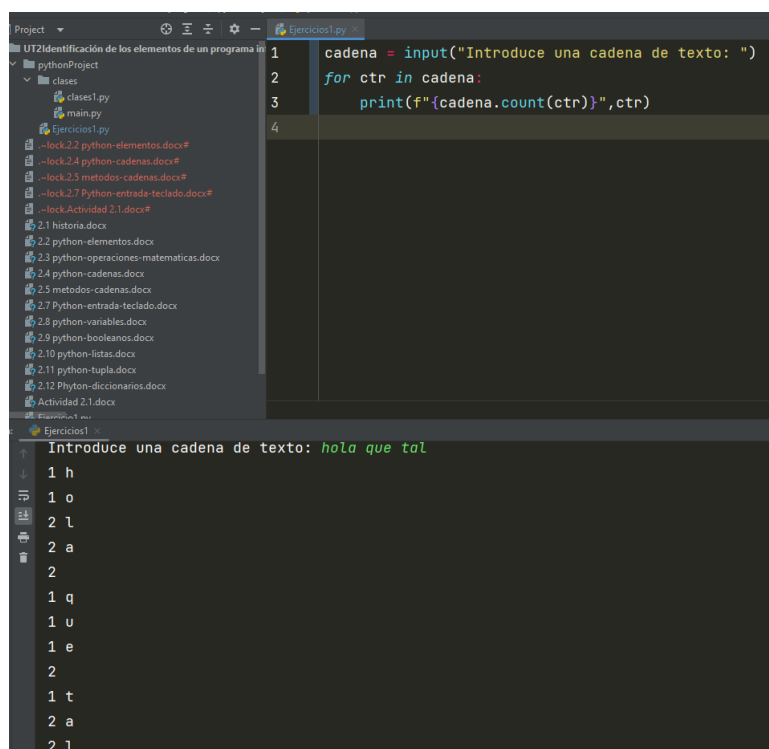
```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"La cadena en capitales: {cadena.capitalize()}")
3
```

C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\On..."

Introduce una cadena de texto: *hola mundo*

La cadena en capitales: Hola mundo

- 20.- Realizar un programa que cuente cuantas veces aparece un carácter en dicha cadena pasada por teclado.

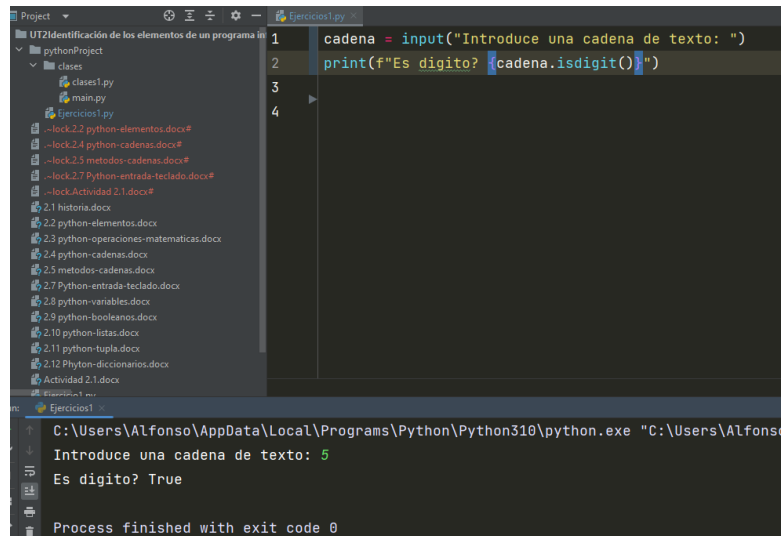


```
1 cadena = input("Introduce una cadena de texto: ")
2 for ctr in cadena:
3     print(f"{cadena.count(ctr)}",ctr)
4
```

Introduce una cadena de texto: *hola que tal*

1	h
1	o
2	l
2	a
1	q
1	u
1	e
1	t
2	a
2	l

- 21.- Realizar un programa que compruebe si todos los caracteres de una cadena pasada por teclado son dígitos.



```
Project
└── UT2Identificación de los elementos de un programa
    └── pythonProject
        ├── clases
        │   ├── clases1.py
        │   ├── main.py
        │   └── Ejercicios1.py
        ├── lock.2.2 python-elementos.docx#
        ├── lock.2.4 python-cadenas.docx#
        ├── lock.2.5 metodos-cadenas.docx#
        ├── lock.2.7 Python-entrada-teclado.docx#
        ├── lock.Actividad 2.1.docx#
        ├── 2.1 historia.docx
        ├── 2.2 python-elementos.docx
        ├── 2.3 python-operaciones-matematicas.docx
        ├── 2.4 python-cadenas.docx
        ├── 2.5 metodos-cadenas.docx
        ├── 2.7 Python-entrada-teclado.docx
        ├── 2.8 python-variables.docx
        ├── 2.9 python-booleanos.docx
        ├── 2.10 python-listas.docx
        ├── 2.11 python-tupla.docx
        ├── 2.12 Python-diccionarios.docx
        └── Actividad 2.1.docx

Ejercicios1.py
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Es digito? {cadena.isdigit()}")
3
4
```

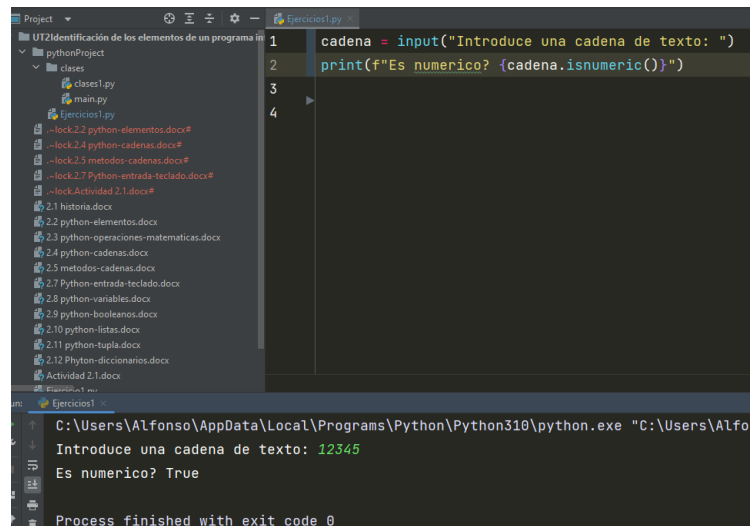
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\Ejercicios1.py"

Introduce una cadena de texto: 5

Es digito? True

Process finished with exit code 0

- 22.- Realizar un programa que compruebe si todos los caracteres de una cadena pasada por teclado son números.



```
Project
└── UT2Identificación de los elementos de un programa
    └── pythonProject
        ├── clases
        │   ├── clases1.py
        │   ├── main.py
        │   └── Ejercicios1.py
        ├── lock.2.2 python-elementos.docx#
        ├── lock.2.4 python-cadenas.docx#
        ├── lock.2.5 metodos-cadenas.docx#
        ├── lock.2.7 Python-entrada-teclado.docx#
        ├── lock.Actividad 2.1.docx#
        ├── 2.1 historia.docx
        ├── 2.2 python-elementos.docx
        ├── 2.3 python-operaciones-matematicas.docx
        ├── 2.4 python-cadenas.docx
        ├── 2.5 metodos-cadenas.docx
        ├── 2.7 Python-entrada-teclado.docx
        ├── 2.8 python-variables.docx
        ├── 2.9 python-booleanos.docx
        ├── 2.10 python-listas.docx
        ├── 2.11 python-tupla.docx
        ├── 2.12 Python-diccionarios.docx
        └── Actividad 2.1.docx

Ejercicios1.py
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Es numerico? {cadena.isnumeric()}")
3
4
```

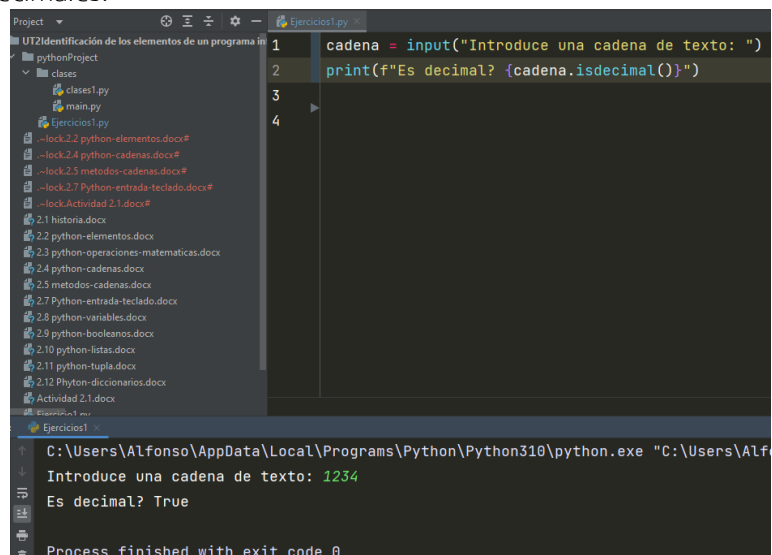
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\Ejercicios1.py"

Introduce una cadena de texto: 12345

Es numerico? True

Process finished with exit code 0

- 23.- Realizar un programa que compruebe si todos los caracteres de una cadena pasada por teclado son números decimales.



```
Project
└── UT2Identificación de los elementos de un programa
    └── pythonProject
        ├── clases
        │   ├── clases1.py
        │   ├── main.py
        │   └── Ejercicios1.py
        ├── lock.2.2 python-elementos.docx#
        ├── lock.2.4 python-cadenas.docx#
        ├── lock.2.5 metodos-cadenas.docx#
        ├── lock.2.7 Python-entrada-teclado.docx#
        ├── lock.Actividad 2.1.docx#
        ├── 2.1 historia.docx
        ├── 2.2 python-elementos.docx
        ├── 2.3 python-operaciones-matematicas.docx
        ├── 2.4 python-cadenas.docx
        ├── 2.5 metodos-cadenas.docx
        ├── 2.7 Python-entrada-teclado.docx
        ├── 2.8 python-variables.docx
        ├── 2.9 python-booleanos.docx
        ├── 2.10 python-listas.docx
        ├── 2.11 python-tupla.docx
        ├── 2.12 Python-diccionarios.docx
        └── Actividad 2.1.docx

Ejercicios1.py
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Es decimal? {cadena.isdecimal()}")
3
4
```

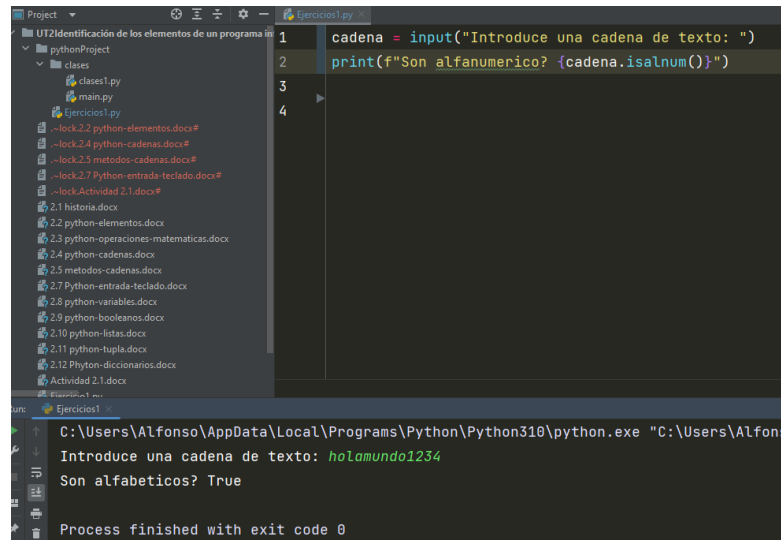
C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\Ejercicios1.py"

Introduce una cadena de texto: 1234

Es decimal? True

Process finished with exit code 0

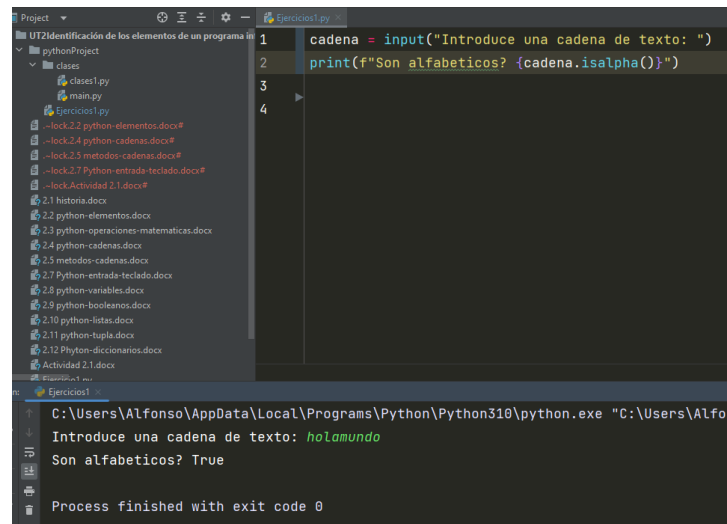
- 24.- Realizar un programa que compruebe si todos los caracteres son alfanuméricos de una cadena pasada por teclado.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Son alfanumerico? {cadena.isalnum()}")
3
4
```

C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\..."
Introduce una cadena de texto: holamundo1234
Son alfabeticos? True
Process finished with exit code 0

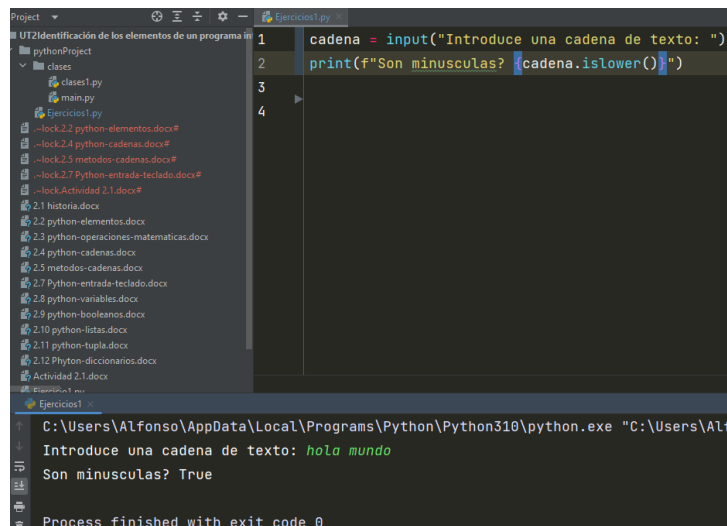
- 25.- Realizar un programa que compruebe si todos los caracteres son alfabéticos de una cadena pasada por teclado.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Son alfabeticos? {cadena.isalpha()}")
3
4
```

C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\..."
Introduce una cadena de texto: holamundo
Son alfabeticos? True
Process finished with exit code 0

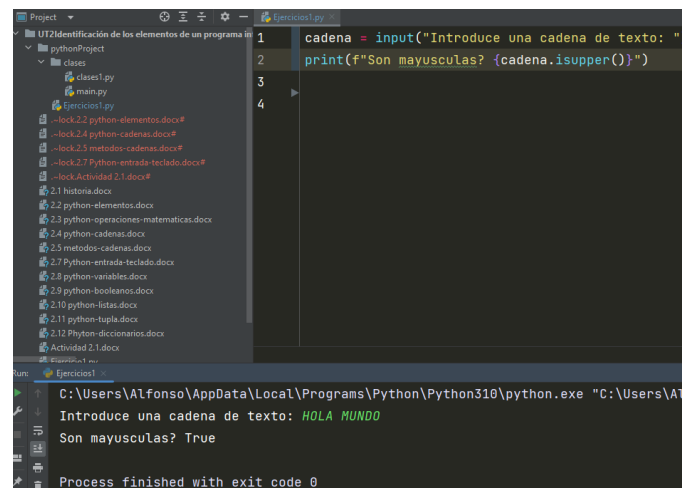
- 26.- Realizar un programa que compruebe si todas las letras son minúsculas de una cadena pasada por teclado.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Son minusculas? {cadena.islower()}")
3
4
```

C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\..."
Introduce una cadena de texto: hola mundo
Son minusculas? True
Process finished with exit code 0

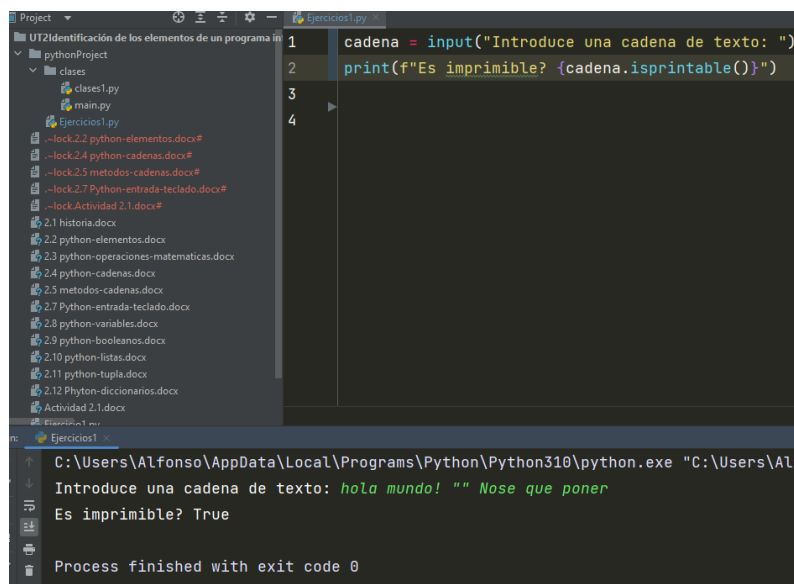
27.- Realizar un programa que compruebe si todas las letras son Mayúsculas de una cadena pasada por teclado.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Son mayusculas? {cadena.isupper()}")
3
4
```

Process finished with exit code 0

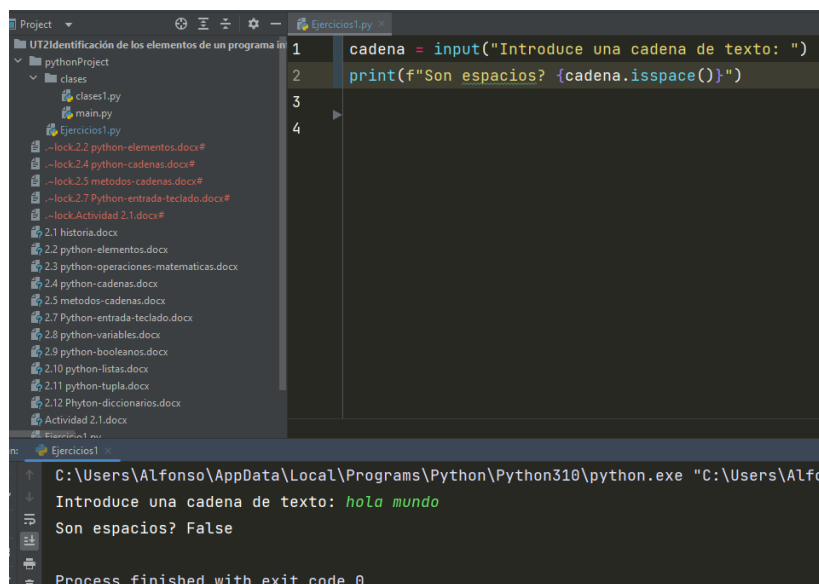
28.- Realizar un programa que compruebe si la cadena contiene todos caracteres imprimibles de una cadena pasada por teclado.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Es imprimible? {cadena.isprintable()}")
3
4
```

Process finished with exit code 0

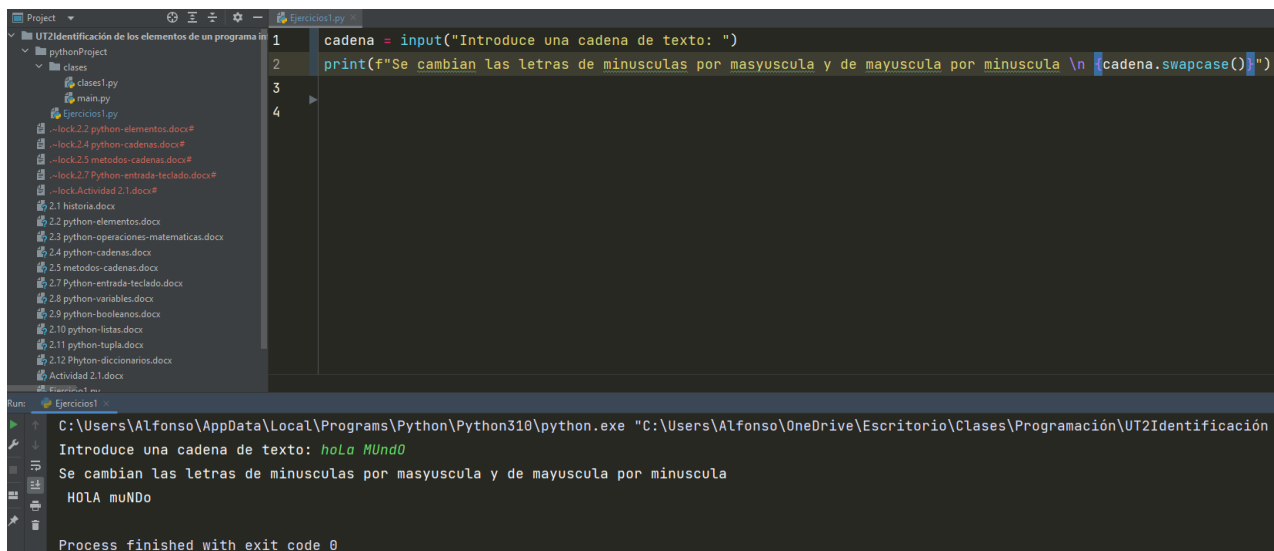
29.- Realizar un programa que compruebe si la cadena contiene solo espacios de una cadena pasada por teclado.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Son espacios? {cadena.isspace()}")
3
4
```

Process finished with exit code 0

- 30.- Realizar un programa que cambie las mayúsculas por minúsculas y viceversa de una cadena pasada por teclado.



```
1 cadena = input("Introduce una cadena de texto: ")
2 print(f"Se cambian las letras de minúsculas por mayúscula y de mayúscula por minúscula \n {cadena.swapcase()}")
3
4
```

Run: C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Escritorio\Clases\Programación\UT2Identificación

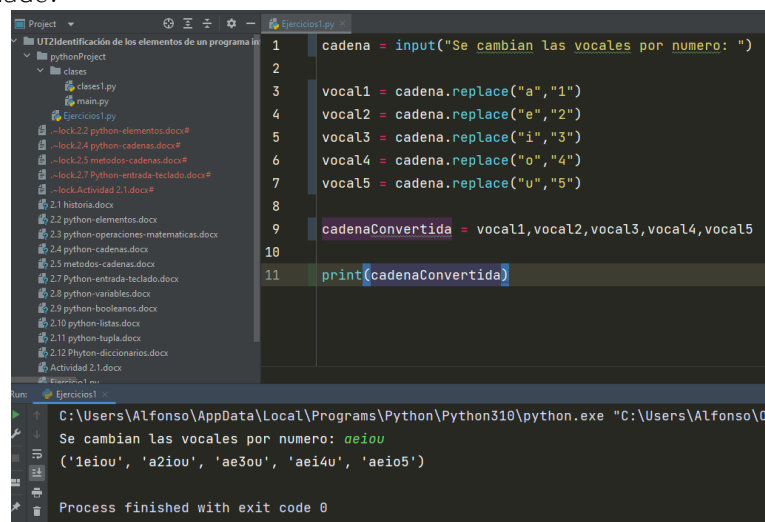
Introduce una cadena de texto: **hoLa MuNdO**

Se cambian las letras de minúsculas por mayúscula y de mayúscula por minúscula

H0LA mUNdO

Process finished with exit code 0

- 31.- Realizar un programa que cambie las vocales por números: a=1, e=2, i=3, o=4 y u=5, de una cadena pasada por teclado.



```
1 cadena = input("Se cambian las vocales por numero: ")
2
3 vocal1 = cadena.replace("a", "1")
4 vocal2 = cadena.replace("e", "2")
5 vocal3 = cadena.replace("i", "3")
6 vocal4 = cadena.replace("o", "4")
7 vocal5 = cadena.replace("u", "5")
8
9 cadenaConvertida = vocal1,vocal2,vocal3,vocal4,vocal5
10
11 print(cadenaConvertida)
```

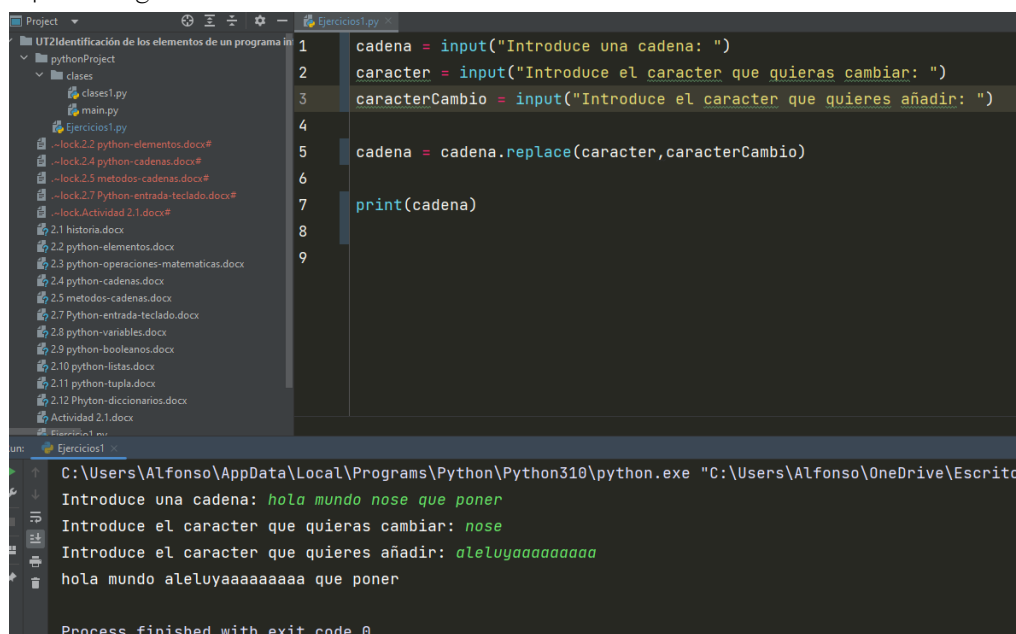
Run: C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Escritorio\Clases\Programación\UT2Identificación

Se cambian las vocales por numero: **aeiou**

('1eiou', 'a2iou', 'ae3ou', 'aei4u', 'aeio5')

Process finished with exit code 0

- 32.- Pide una cadena y dos caracteres por teclado, sustituye la aparición del primer carácter en la cadena por el segundo carácter.



```
1 cadena = input("Introduce una cadena: ")
2 caracter = input("Introduce el caracter que quieras cambiar: ")
3 caracterCambio = input("Introduce el caracter que quieres añadir: ")
4
5 cadena = cadena.replace(caracter, caracterCambio)
6
7 print(cadena)
8
9
```

Run: C:\Users\Alfonso\AppData\Local\Programs\Python\Python310\python.exe "C:\Users\Alfonso\OneDrive\Escritorio\Clases\Programación\UT2Identificación

Introduce una cadena: **hola mundo nose que poner**

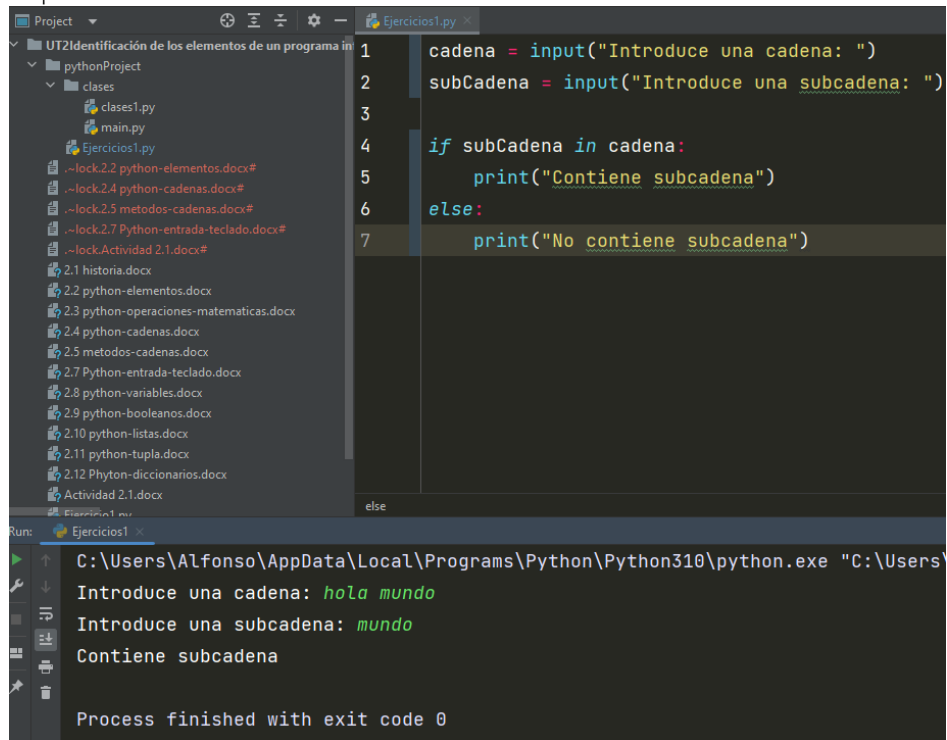
Introduce el caracter que quieras cambiar: **nose**

Introduce el caracter que quieres añadir: **aleluyaaaaaaaa**

hola mundo aleluyaaaaaaaa que poner

Process finished with exit code 0

- 33.- Realizar un programa que compruebe si una cadena contiene una subcadena. Las dos cadenas se introducen por teclado.

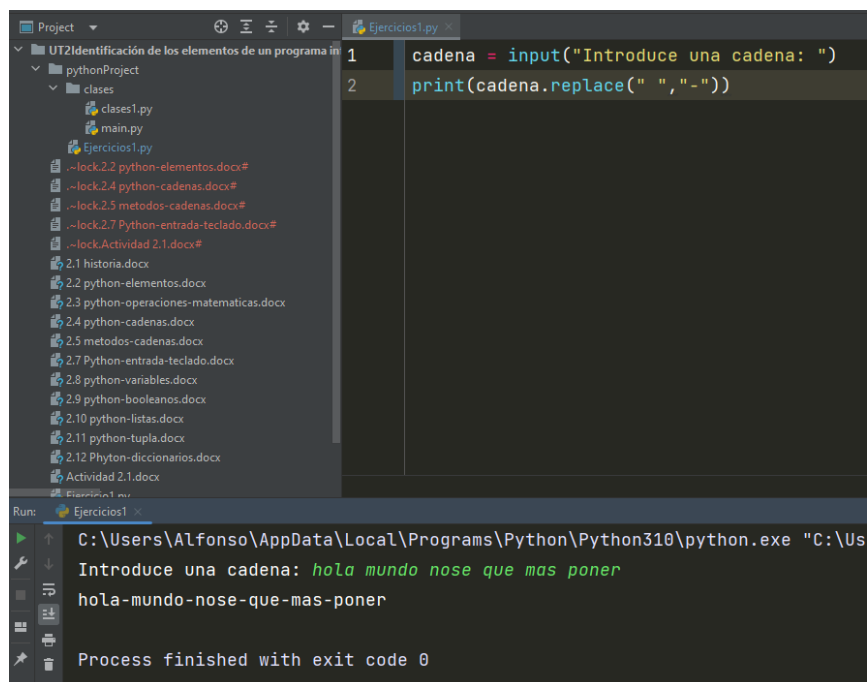


The screenshot shows a Python IDE with a project named 'UT2Identificación de los elementos de un programa in'. The file explorer on the left shows a folder 'pythonProject' containing 'clases' and 'main.py'. The main editor shows a file named 'Ejercicios1.py' with the following code:

```
1 cadena = input("Introduce una cadena: ")
2 subCadena = input("Introduce una subcadena: ")
3
4 if subCadena in cadena:
5     print("Contiene subcadena")
6 else:
7     print("No contiene subcadena")
```

The Run console shows the execution of the program. The user enters 'hola mundo' for the first prompt and 'mundo' for the second. The output is 'Contiene subcadena'. The process finished with exit code 0.

- 34.- Realizar un programa que introduzca un guion por un espacio en blanco de una frase pasada por teclado.

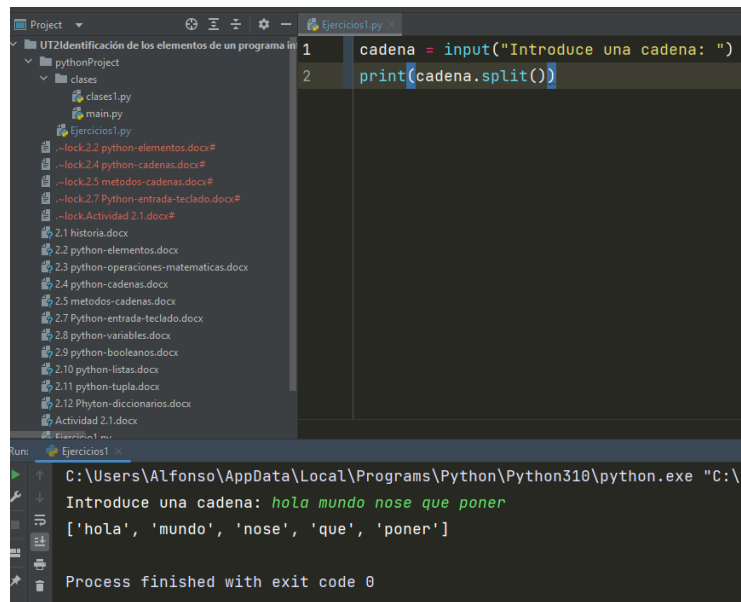


The screenshot shows a Python IDE with a project named 'UT2Identificación de los elementos de un programa in'. The file explorer on the left shows a folder 'pythonProject' containing 'clases' and 'main.py'. The main editor shows a file named 'Ejercicios1.py' with the following code:

```
1 cadena = input("Introduce una cadena: ")
2 print(cadena.replace(" ", "-"))
```

The Run console shows the execution of the program. The user enters 'hola mundo nose que mas poner' for the first prompt. The output is 'hola-mundo-nose-que-mas-poner'. The process finished with exit code 0.

- 35.- Realizar un programa que separe en una lista una frase pasada por teclado por los espacios en blanco.

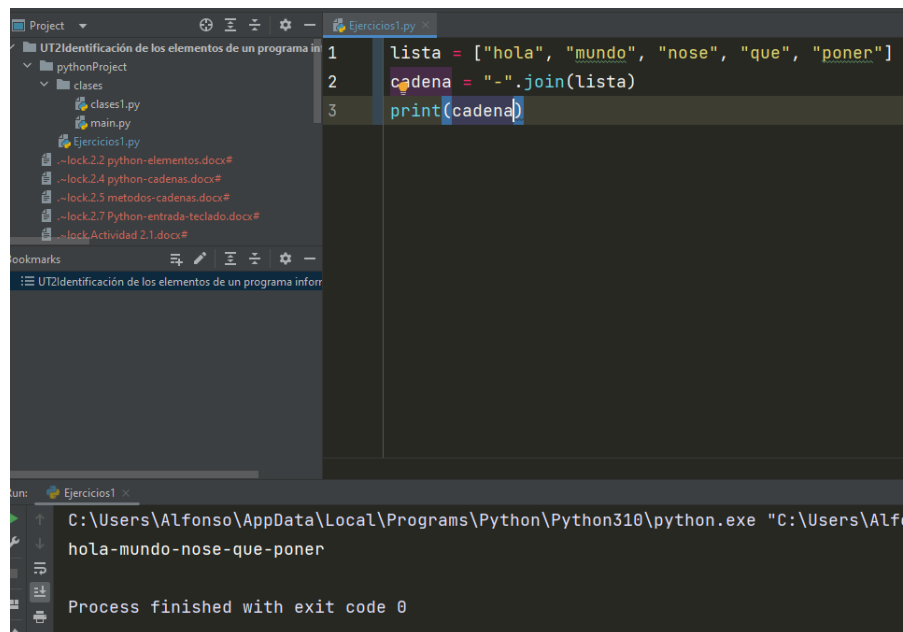


The screenshot shows a Python IDE with a project named 'UT2Identificación de los elementos de un programa informático'. The file explorer on the left shows a folder 'pythonProject' containing files like 'clases1.py', 'main.py', and 'Ejercicios1.py'. The main editor shows the following code in 'Ejercicios1.py':

```
1 cadena = input("Introduce una cadena: ")
2 print(cadena.split())
```

The console output shows the execution of the program. The user input is 'hola mundo nose que poner', and the output is a list of words: ['hola', 'mundo', 'nose', 'que', 'poner']. The process finished with exit code 0.

- 36.- Unir la lista del ejercicio anterior con un guion alto



The screenshot shows the same Python IDE as before, but with the following code in 'Ejercicios1.py':

```
1 lista = ["hola", "mundo", "nose", "que", "poner"]
2 cadena = "-".join(lista)
3 print(cadena)
```

The console output shows the execution of the program. The user input is 'hola-mundo-nose-que-poner'. The process finished with exit code 0.