



Recuperación 1

Programación de Redes

Profesor: Gabriel Barron Rodríguez

Alumno: Alan Francisco Emmanuel Aguilar Fuentes

Laboratorios Python 2

## Contenido Resúmenes de Laboratorios de Python ..... ¡Error! Marcador no definido.

1.1.1.11 RESUMEN DE SECCIÓN.....	1
1.2.1.17 RESUMEN DE LA SECCIÓN .....	2
1.3.1.11 RESUMEN DE LA SECCIÓN .....	3
1.4.1.18 RESUMEN DE LA SECCIÓN .....	4
2.1.1.4 RESUMEN DE LA SECCIÓN.....	5
2.2.1.15 RESUMEN DE LA SECCIÓN .....	5
2.3.1.17 RESUMEN DE LA SECCIÓN .....	6
2.4.1.5 RESUMEN DE LA SECCIÓN.....	7
2.5.1.5 RESUMEN DE LA SECCIÓN.....	8
2.6.1.12 RESUMEN DE LA SECCIÓN .....	9
2.7.1.18 RESUMEN DE LA SECCIÓN .....	10
2.8.1.5 RESUMEN DE LA SECCIÓN .....	11
3.1.1.8 RESUMEN DE SECCIÓN .....	12
3.2.1.13 RESUMEN DE SECCIÓN.....	12
3.3.1.9 RESUMEN DE SECCIÓN .....	13
3.4.1.11 RESUMEN DE SECCIÓN.....	14
3.5.1.22 RESUMEN DE SECCIÓN.....	15
3.5.1.23 RESUMEN DE SECCIÓN.....	15
3.6.1.9 RESUMEN DE SECCIÓN .....	16
4.1.1.15 RESUMEN DE SECCIÓN.....	17
4.2.1.12 RESUMEN DE SECCIÓN.....	18
4.3.1.18 RESUMEN DE SECCIÓN.....	19
4.4.1.9 RESUMEN DE SECCIÓN .....	19
4.5.1.23 RESUMEN DE SECCIÓN.....	20
4.6.1.14 RESUMEN DE SECCIÓN.....	20

Recibidos (1.561) - alanhenric... x U2\_R1\_prog\_notas/Unidad2\_F... x U2\_R1\_Laboratorio\_Terminad... x U2\_R1\_Jab\_Resumenes\_Alan F... x DH-PDR/TSU Unidad 2 Recu... x Nueva pestaña x Course: Programación de Red...

lms.netacad.com/course/view.php?id=2123636

Gmail YouTube Maps

### Fundamentos de Python 1 - EXAMEN DE SECCIÓN

PE1: Evaluación Parcial ☒

Certificarse en PCEP ☐

### Fundamentos de Python 2: INTERMEDIO

#### Fundamentos de Python 2 - RECURSOS DE ESTUDIO

PE2: Bienvenido a Fundamentos de Python 2 ☐

PE2: Módulo 1 - Módulos, paquetes y PIP ☒

FP2: Examen Módulo 1 ☒

PE2: Módulo 2 - Excepciones, Cadenas y Métodos de Listas ☒

FP2: Examen Módulo 2 ☒

PE2: Módulo 3 - Programación Orientada a Objetos y Procesamiento de Archivos en Python ☒

FP2: Examen Módulo 3 ☒

PE2: Módulo 4 - Misceláneo ☒

FP2: Examen Módulo 4 ☒

### Fundamentos de Python 2 - EXAMEN DE SECCIÓN

PE2: Evaluación Parcial ☐

### EXAMEN FINAL, COMENTARIOS DE FIN-DE-CURSO, Y CÓDIGO DE DESCUENTO PCAP-31-03

Examen final ☐

**Not available unless:** You achieve a required score in **PE2: Evaluación Parcial**

Encuesta de satisfacción ☐

**Not available unless:** You achieve a required score in **Examen final**

PCAP-31-03 Formulario de Solicitud para Código de Descuento

**Not available unless:**

- You achieve a required score in **Examen final**
- The activity **Encuesta de satisfacción** is marked complete

Al seguir utilizando nuestro sitio web, confirma el uso de cookies.

[Declaración de privacidad](#) [Cambiar configuración](#)

# Resúmenes de Laboratorios de Python

## 1.1.1.11 RESUMEN DE SECCIÓN.

Plataforma Educativa Institución: x + -

https://elearning... pacute-n-12

RESUMEN DE SECCIÓN

Ejercicio 1

Quieres invocar la función `make_money()` contenida en el módulo llamado `mint`. Tu código comienza con la siguiente línea:

```
import mint
```

¿Cuál es la forma adecuada de invocar a la función?

Revisar

```
mint.make_money()
```

Ejercicio 2

Quieres invocar la función `make_money()` contenida en el módulo llamado `mint`. Tu código comienza con la siguiente línea:

```
from mint import make_money
```

¿Cuál es la forma adecuada de invocar a la función?

Revisar

```
make_money()
```

Plataforma Educativa Institución: x + -

https://elearning... pacute-n-12

RESUMEN DE SECCIÓN

Ejercicio 3

Has escrito una función llamada `make_money` por tu cuenta. Necesitas importar una función con el mismo nombre del módulo `mint` y no deseas cambiar el nombre de ninguno de tus nombres previamente definidos. ¿Qué variante de la sentencia `import` puede ayudarte con el problema?

Revisar

```
# solución de muestra
from mint import make_money as make_more_money
```

Ejercicio 4

¿Qué forma de invocación de la función `make_money` es válida si tu código comienza con la siguiente línea?

Revisar

```
from mint import *
```

```
make_money()
```

Ejercicio 5

La forma más general de la sentencia anterior te permite importar **todas las entidades** ofrecidas por un módulo:

```
from my_module import *
```

```
result = my_function(my_data)
```

**Nota:** la variante de esta importación no se recomienda debido a las mismas razones que antes (la amenaza de un conflicto de nombres es aún más peligrosa aquí).

5. Puede cambiar el nombre de la entidad importada "sobre la marcha" utilizando la frase: `as` del `import`. Por ejemplo:

```
from module import my_function as fun, my_data as dat
```

```
result = fun(dat)
```

## 1.2.1.17 RESUMEN DE LA SECCIÓN

Plataforma Educativa Institución: x

https://elearning... | pacute-n-13

7 RESUMEN DE SECCIÓN

ALAN FRANCISCO EMMANUEL AGUILAR FUENTES

- Área personal
- Perfil
- Calificaciones

2. El módulo `math` contiene más de 50 funciones y constantes que realizan operaciones matemáticas (como `sin()`, `pow()`, `factorial()`) o aportando valores importantes (como  $\pi$  y la constante de Euler  $e$ ).

3. El módulo `random` agrupa más de 60 entidades diseñadas para ayudarte a usar números pseudoaleatorios. No olvides el prefijo "pseudo", ya que no existe un número aleatorio real cuando se trata de generarlos utilizando los algoritmos de la computadora.

4. El módulo `platform` contiene alrededor de 70 funciones que te permiten sumergirte en las capas subyacentes del sistema operativo y el hardware. Usarlos te permite aprender más sobre el entorno en el que se ejecuta tu código.

5. El **Índice de Módulos de Python** (<https://docs.python.org/3/py-modindex.html>) es un directorio de módulos impulsado por la comunidad disponible en el universo de Python. Si deseas encontrar un módulo que se adapte a tus necesidades, comienza tu búsqueda allí.

Ejercicio 1

¿Cuál es el valor esperado de la variable `result` después de que se ejecuta el siguiente código?

```
import math
result = math.e ** math.exp(1)
```

Revisar

True

Ejercicio 2

(Completa el enunciado) Establecer la semilla del generador con el mismo valor cada vez que se ejecuta tu programa garantiza que...

Revisar

... los valores pseudoaleatorios emitidos desde el módulo `random` serán exactamente los mismos.

Ejercicio 3

¿Cuál de las funciones del módulo `platform` utilizarías para determinar el nombre del CPU que corre dentro de tu computadora?

Revisar

7 RESUMEN DE SECCIÓN

Revisar

... los valores pseudoaleatorios emitidos desde el módulo `random` serán exactamente los mismos.

Ejercicio 3

¿Cuál de las funciones del módulo `platform` utilizarías para determinar el nombre del CPU que corre dentro de tu computadora?

Revisar

La función `processor()`

Ejercicio 4

¿Cuál es el resultado esperado del siguiente fragmento de código?

```
import platform
print(len(platform.python_version_tuple()))
```

Revisar

3

### 1.3.1.11 RESUMEN DE LA SECCIÓN

The screenshot shows a web browser window with the URL `https://elearning...`. The page title is "RESUMEN DE SECCIÓN". On the left, there is a sidebar with the user's name "ALAN FRANCISCO EMMANUEL AGUILAR FUENTES" and links to "Área personal", "Perfil", and "Calificaciones". The main content area contains three exercises:

- Ejercicio 1:** "Deseas evitar que el usuario de tu módulo ejecute tu código como un script ordinario. ¿Cómo lograrías tal efecto?"  
Code snippet:

```
import sys

if __name__ == "__main__":
    print("¡No hagas eso!")
    sys.exit()
```
- Ejercicio 2:** "Algunos paquetes adicionales y necesarios se almacenan dentro del directorio `D:\Python\Project\Modules`. Escribe un código asegurándote de que Python recorra el directorio para encontrar todos los módulos solicitados."  
Code snippet:

```
import sys

# ¡Toma en cuenta las diagonales invertidas dobles!
sys.path.append("D:\\Python\\Project\\Modules")
```
- Ejercicio 3:** (partially visible)

At the bottom, there is a cookie consent banner: "We use cookies to improve our service. By continuing to use the site, you agree to our [privacy policy](#) and [terms of use](#)."

This screenshot shows the same page as the previous one, but with Exercise 3 fully visible:

- Ejercicio 3:** "El directorio mencionado en el ejercicio anterior contiene un subárbol con la siguiente estructura:"  
Code snippet:

```
abc
├── def
│   └── mymodule.py
```

  
"Asumiendo que `D:\Python\Project\Modules` se ha adjuntado con éxito a la lista `sys.path`, escribe una directiva de importación que te permita usar todas las entidades de `mymodule`."  
Code snippet:

```
import abc.def.mymodule
```

The same cookie consent banner is visible at the bottom.

### 1.4.1.18 RESUMEN DE LA SECCIÓN

The screenshot shows a web browser window with the URL `https://elearning...`. The page title is "RESUMEN DE SECCIÓN". On the left, a sidebar menu for "ALAN FRANCISCO EMMANUEL AGUILAR FUENTES" includes links for "Área personal", "Perfil", and "Calificaciones". The main content area contains three exercises:

- Ejercicio 1**: ¿De donde proviene el nombre *The Cheese Shop*?  
Review button. Description: Es una referencia a un viejo sketch de *Monty Python* que lleva el mismo nombre.
- Ejercicio 2**: ¿Por qué deberías asegurarte de cuál *pip* o *pip3* es el correcto para ti?  
Review button. Description: Cuando Python 2 y Python 3 coexisten en el sistema operativo, es probable que *pip* identifique la instancia de *pip* que trabaja solo con paquetes de Python 2.
- Ejercicio 3**: ¿Cómo puedes determinar si tu *pip* funciona con Python 2 o Python 3?  
Review button. Description: `pip --version` te lo dirá.

Below the exercises, there is a section titled "4. La lista de las actividades principales de *pip* tiene el siguiente aspecto:" followed by a list of commands and their descriptions:

- `pip help operación_o_comando` → muestra una breve descripción de *pip*.
- `pip list` → muestra una lista de los paquetes instalados actualmente.
- `pip show nombre_del_paquete` → muestra información que incluyen las dependencias del paquete.
- `pip search cadena` → busca en los directorios de PyPI para encontrar paquetes cuyos nombres contengan cadena.
- `pip install nombre` → instala el paquete *nombre* en todo el sistema (espera problemas cuando no tengas privilegios de administrador).
- `pip install --user nombre` → instala *nombre* solo para ti; ningún otro usuario de la plataforma podrá utilizarlo.
- `pip install -U nombre` → actualiza un paquete previamente instalado.
- `pip uninstall nombre` → desinstala un paquete previamente instalado.

A cookie notice at the bottom states: "We use cookies to improve our service. By continuing to use the site, you agree to our [privacy](#) and [cookie policy](#)."

This screenshot is similar to the first one, showing the same sidebar and exercise list. However, the description for Ejercicio 4 is now visible:

**Ejercicio 4**: Desafortunadamente, no tienes privilegios de administrador. ¿Qué debes hacer para instalar un paquete en todo el sistema?  
Review button. Description: Tienes que consultar a tu administrador del sistema → ¡no intentes hackear tu sistema operativo!

The same list of *pip* commands and the cookie notice are also present at the bottom.

## 2.1.1.4 RESUMEN DE LA SECCIÓN

The screenshot shows a web browser window with the URL `https://elearning...`. The page title is "Edube Interactive : PE2 -- Modulo". The user is logged in as "ALAN FRANCISCO EMMANUEL AGUILAR FUENTES". The left sidebar contains links for "Área personal", "Perfil", and "Calificaciones". The main content area is titled "RESUMEN DE SECCIÓN" and contains two exercises:

**Ejercicio 1**  
¿Qué es BOM?  
**BOM** (Byte Order Mark). Una Marca de Orden de Bytes es una combinación especial de bits que anuncia la codificación utilizada por el contenido de un archivo (por ejemplo, UCS-4 o UTF-8).

**Ejercicio 2**  
¿Está Python 3 internacionalizado?  
Sí, está completamente internacionalizado; podemos usar caracteres UNICODE dentro de nuestro código, leerlos desde la entrada y enviarlos a la salida.

## 2.2.1.15 RESUMEN DE LA SECCIÓN

The screenshot shows a web browser window with the URL `https://elearning...`. The page title is "Edube Interactive : PE2 -- Modulo". The user is logged in as "ALAN FRANCISCO EMMANUEL AGUILAR FUENTES". The left sidebar contains links for "Área personal", "Perfil", and "Calificaciones". The main content area is titled "RESUMEN DE SECCIÓN" and contains three exercises:

**Ejercicio 1**  
¿Cuál es la longitud de la siguiente cadena asumiendo que no hay espacios en blanco entre las comillas?  
`cadena = "cadena"`  
`len(cadena)`

**Ejercicio 2**  
¿Cuál es el resultado esperado del siguiente código?  
`s = "yearxyearx"`  
`the_list = list(s)`  
`print(the_list[3:6])`

**Ejercicio 3**  
¿Cuál es el resultado esperado del siguiente código?  
`s = "yearxyearx"`  
`the_list = list(s)`  
`print(the_list[3:6])`



Plataforma Educativa Institución: x + - Edube Interactive: PE2 -- Modulo: x + -

https://elearning... acute-n-17

RESUMEN DE SECCIÓN

Área personal

Perfil

Calificaciones

```

asterisk = "*"
plus = "+"
decoration = (asterisk + plus) * 4 + asterisk
print(decoration)

salida *****

```

4. El par de funciones `chr()` y `ord()` se pueden utilizar para crear un carácter utilizando su punto de código y para determinar un punto de código correspondiente a un carácter. Las dos expresiones siguientes son siempre verdaderas:

```

chr(ord(character)) == character
ord(chr(codepoint)) == codepoint

```

5. Algunas otras funciones que se pueden aplicar a cadenas son:

- `list()` → crea una lista que consta de todos los caracteres de la cadena.
- `max()` → encuentra el carácter con el punto de código máximo.
- `min()` → encuentra el carácter con el punto de código mínimo.

Ejercicio 3

¿Cuál es el resultado esperado del siguiente código?

```

for ch in "abc":
    print(chr(ord(ch) + 1), end="")

```

Reservar

Reservar

Reservar

## 2.3.1.17 RESUMEN DE LA SECCIÓN

Plataforma Educativa Institución: x + - Edube Interactive: PE2 -- Modulo: x + -

https://elearning... acute-n-18

RESUMEN DE SECCIÓN

Área personal

Perfil

Calificaciones

- `rstrip()` → elimina los caracteres en blanco al final de la cadena.
- `split()` → divide la cadena en una subcadena usando un delimitador dado.
- `strip()` → elimina los espacios en blanco iniciales y finales.
- `swapcase()` → intercambia las mayúsculas y minúsculas de las letras.
- `title()` → hace que la primera letra de cada palabra sea mayúscula.
- `upper()` → convierte todas las letras de la cadena en letras mayúsculas.

2. El contenido de las cadenas se puede determinar mediante los siguientes métodos (todos devuelven valores booleanos):

- `endswith()` → ¿La cadena termina con una subcadena determinada?
- `isalnum()` → ¿La cadena consta solo de letras y dígitos?
- `isalpha()` → ¿La cadena consta solo de letras?
- `islower()` → ¿La cadena consta solo de letras minúsculas?
- `isspace()` → ¿La cadena consta solo de espacios en blanco?
- `isupper()` → ¿La cadena consta solo de letras mayúsculas?
- `startswith()` → ¿La cadena consta solo de letras mayúsculas?

Ejercicio 1

¿Cuál es el resultado esperado del siguiente código?

```

for ch in "Abc123XYZ":
    if ch.isupper():
        print(ch.lower(), end="")
    elif ch.islower():
        print(ch.upper(), end="")
    else:
        print(ch, end="")

```

Reservar

ABC123xyz

Ejercicio 2

¿Cuál es el resultado esperado del siguiente código?

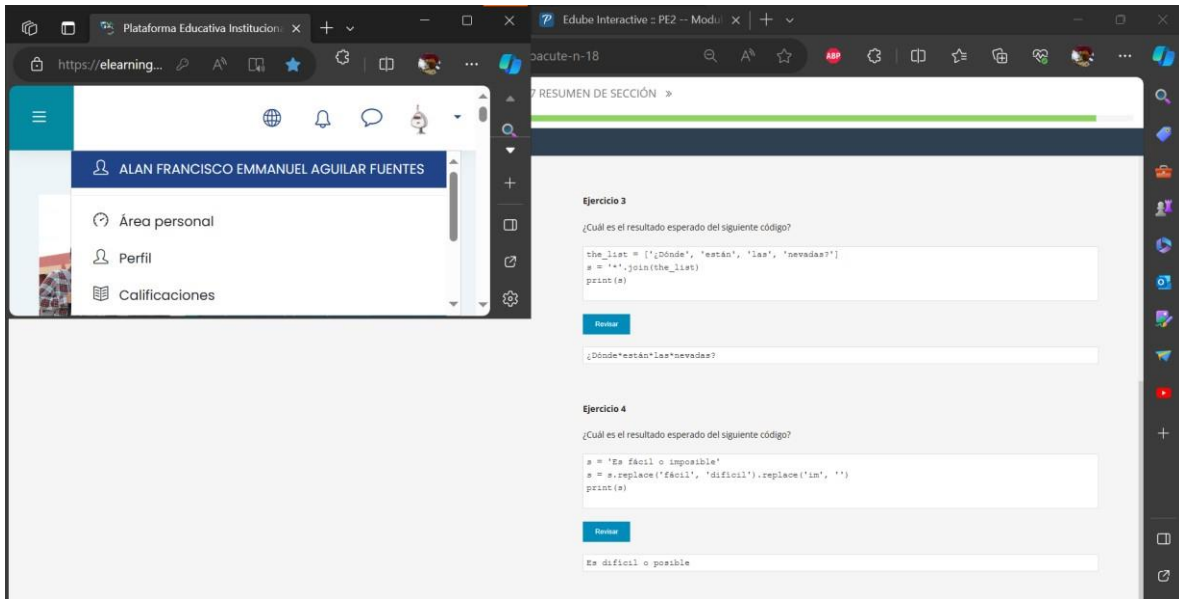
```

s1 = "¿Dónde están las nevadas de año?"
s2 = s1.split()
print(s2[-2])

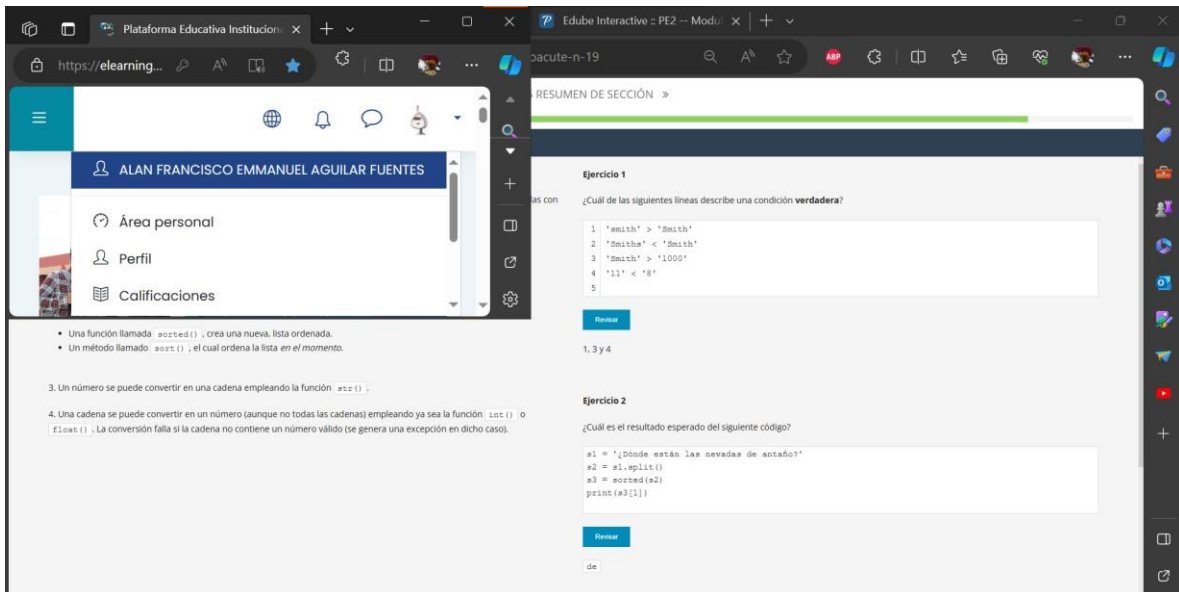
```

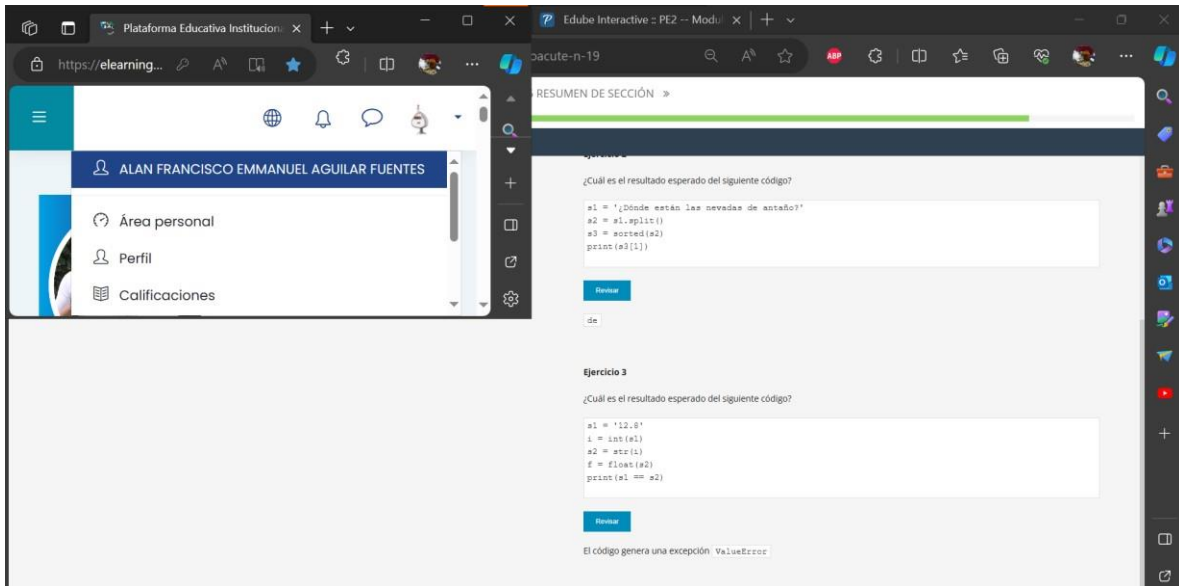
Reservar

de

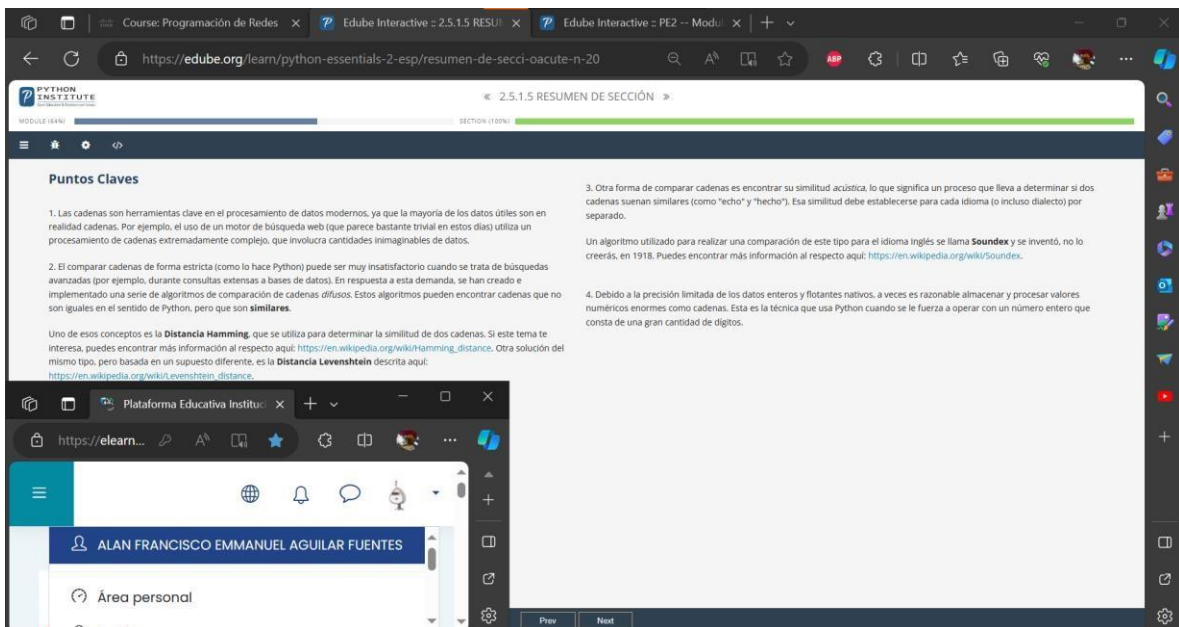


## 2.4.1.5 RESUMEN DE LA SECCIÓN





## 2.5.1.5 RESUMEN DE LA SECCIÓN



## 2.6.1.12 RESUMEN DE LA SECCIÓN

The screenshot shows the Edube Interactive interface. On the left, a sidebar displays the user's name 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' and navigation links for 'Área personal', 'Perfil', and 'Calificaciones'. The main content area is titled 'SUMEN DE SECCIÓN' and contains two exercises.

**Ejercicio 1**  
¿Cuál es el resultado esperado del siguiente código?

```
try:  
    print("Tratemos de hacer esto")  
    print("#"[2])  
    print("¡Tuvinos éxito!")  
except:  
    print("¡Hemos fallado!")  
    print("¡Hemos terminado!")
```

**Ejercicio 2**  
¿Cuál es el resultado esperado del siguiente código?

```
try:  
    print("alpha"[1/0])  
except ZeroDivisionError:  
    print("cero")  
except IndexError:  
    print("indice")  
except:  
    print("algo")
```

The screenshot shows the Edube Interactive interface, similar to the one above, but with a different set of exercises.

**Ejercicio 2**  
¿Cuál es el resultado esperado del siguiente código?

```
try:  
    print("alpha"[1/0])  
except ZeroDivisionError:  
    print("cero")  
except IndexError:  
    print("indice")  
except:  
    print("algo")
```

Below the code, there is a text input field containing the word 'cero'.

En el mejor caso, se ejecuta uno de los bloques `except`; ninguno de los bloques se ejecuta cuando la excepción generada no coincide con ninguna de las excepciones especificadas.

3. No se puede agregar más de un bloque de `exceptión sin nombre` después de los bloques con nombre.

## 2.7.1.18 RESUMEN DE LA SECCIÓN

The screenshot shows a web browser with two tabs: 'Plataforma Educativa Institución...' and 'Edube Interactive: PE2 -- Mod...'. The address bar shows 'https://elearning.ut...'. The page has a sidebar with the user's name 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' and links to 'Área personal', 'Perfil', and 'Calificaciones'. The main content area is titled 'SUMEN DE SECCIÓN' and contains two exercises. Exercise 1 asks for the output of a Python code snippet. The code is:

```
try:
    print(1/0)
except ZeroDivisionError:
    print("Cero")
except ArithmeticError:
    print("Error")
except:
    print("algo")
```

Below the code is a text input field containing 'cero'. Exercise 2 asks for the output of another Python code snippet. The code is:

```
try:
    print(1/0)
except ArithmeticError:
    print("Error")
except ZeroDivisionError:
    print("Cero")
except:
    print("algo")
```

Below the code is a text input field containing 'cero'.

The screenshot shows the same web browser as the previous one, but the page has scrolled down to show more content. The sidebar remains the same. The main content area is still titled 'SUMEN DE SECCIÓN' and contains two exercises. Exercise 2 asks for the output of a Python code snippet. The code is:

```
try:
    print(1/0)
except ArithmeticError:
    print("Error")
except ZeroDivisionError:
    print("Cero")
except:
    print("algo")
```

Below the code is a text input field containing 'cero'. Exercise 3 asks for the output of another Python code snippet. The code is:

```
def foo(x):
    assert x
    return 1/x

try:
    print(foo(0))
except ZeroDivisionError:
```

Below the code is a text input field containing 'cero'.

The screenshot shows the Edube Interactive interface. On the left, a sidebar for user 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' contains links to 'Área personal', 'Perfil', and 'Calificaciones'. The main content area displays a Python exercise titled 'Ejercicio 3' with the question: '¿Cuál es la salida esperada del siguiente código?'. The code defines a function 'foo(x)' that returns '1/x' and uses a 'try' block to call 'foo(0)', catching 'ZeroDivisionError' and printing 'cero', and a 'finally' block that prints 'algo'. Below the code is a 'Reservar' button and an input field containing 'algo'. To the left of the exercise, a text block explains the difference between 'raise ExceptionName' and 'assert expression'.

3. La sentencia de Python `raise ExceptionName` puede generar una excepción bajo demanda. La misma sentencia pero sin `ExceptionName`, se puede usar **solamente** dentro del bloque `try` y genera la misma excepción que se está manejando actualmente.

4. La sentencia de Python `assert expression` evalúa la expresión y genera la excepción `AssertionError` cuando la expresión es igual a cero, una cadena vacía o `None`. Puedes usarla para proteger algunas partes críticas de tu código de datos devastadores.

**RESUMEN DE SECCIÓN**

```

def foo(x):
    assert x
    return 1/x

try:
    print(foo(0))
except ZeroDivisionError:
    print("cero")
except:
    print("algo")

```

Reservar

algo

## 2.8.1.5 RESUMEN DE LA SECCIÓN

The screenshot shows the Edube Interactive interface for the 'RESUMEN DE LA SECCIÓN'. The sidebar is the same as in the previous image. The main content area displays three exercises: 'Ejercicio 1' (KeyboardInterrupt), 'Ejercicio 2' (BaseException), and 'Ejercicio 3' (OverflowError). Each exercise has a 'Reservar' button and an input field for the answer.

**RESUMEN DE SECCIÓN**

**Ejercicio 1**  
¿Cuál de las excepciones se utilizará para proteger al código de ser interrumpido por el uso del teclado?  
Reservar  
KeyboardInterrupt

**Ejercicio 2**  
¿Cuál es el nombre de la más general de todas las excepciones de Python?  
Reservar  
BaseException

**Ejercicio 3**  
¿Cuál de las excepciones será generada a través de la siguiente evaluación fallida?  
Reservar  
OverflowError

• ImportError,  
• IndexError,  
• KeyboardInterrupt,  
• KeyError,  
• MemoryError,  
• OverflowError,

### 3.1.1.8 RESUMEN DE SECCIÓN.

Plataforma Educativa Institución: x + v

Edube Interactive : PE2 -- Modu... x

Edube Interactive : PE2 -- Modu... x + v

ute-n-24

RESUMEN DE SECCIÓN >

Área personal

Perfil

Calificaciones

Las relaciones entre clases se muestran como flechas dirigidas desde la subclase hacia su superclase.

3. Los objetos están equipados con:

- Un **nombre** que los identifica y nos permite distinguirlos.
- Un conjunto de **propiedades** (el conjunto puede estar vacío).
- Un conjunto de **métodos** (también puede estar vacío).

4. Para definir una clase de Python, se necesita usar la palabra clave reservada `class`. Por ejemplo:

```
class This_Is_A_Class:
    pass
```

5. Para crear un objeto de la clase previamente definida, se necesita usar la clase como si fuera una función. Por ejemplo:

```
this_is_an_object = This_Is_A_Class()
```

**Ejercicio 1**

Si asumimos que pitones, víboras y cobras son subclases de la misma superclase, ¿cómo la llamarías?

Reservar

Serpiente, reptil, vertebrado, animal: todas estas respuestas son aceptables.

**Ejercicio 2**

Intenta nombrar algunas subclases de la clase Pitón.

Reservar

Pitón india, Pitón de Roca Africana, Pitón Bola, Pitón Birmana: la lista es larga.

**Ejercicio 3**

¿Puedes usar la palabra "class" para darle nombre a alguna de tus clases?

Reservar

¡No, no puedes. `class` es una palabra clave reservada!

### 3.2.1.13 RESUMEN DE SECCIÓN.

Plataforma Educativa Institución: x + v

Edube Interactive : PE2 -- Modu... x

Edube Interactive : PE2 -- Modu... x + v

ute-n-25

RESUMEN DE SECCIÓN >

Área personal

Perfil

Calificaciones

la clase.

4. La parte de la clase en Python responsable de crear nuevos objetos se llama **constructor** y se implementa como un método de nombre `__init__`.

5. Cada declaración de método de clase debe contener al menos un parámetro (siempre el primero) generalmente denominado `self`, y es utilizado por los objetos para identificarse a sí mismos.

6. Si queremos ocultar alguno de los componentes de una clase del mundo exterior, debemos comenzar su nombre con `__`. Estos componentes se denominan **privados**.

**Ejercicio 1**

Suponiendo que hay una clase llamada `Snake`, escribe la primera línea de la declaración de clase `Python`, expresando el hecho de que la nueva clase es en realidad una subclase de `Snake`.

Reservar

```
class Python(Snake):
```

**Ejercicio 2**

Algo falta en la siguiente declaración, ¿qué es?

```
class Snake
    def __init__(self):
        self.nombre = "Serpiente"
```

Reservar

El constructor `__init__(self)` carece del parámetro obligatorio (deberíamos llamarlo `self` para cumplir con los estándares).



The screenshot shows a web browser with two tabs. The left tab is 'Plataforma Educativa Institución' with the URL 'https://elearning.ut...'. It displays a user profile for 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' with a sidebar menu containing 'Área personal', 'Perfil', and 'Calificaciones'. The right tab is 'Edube Interactive : PE2 -- Modu...' with the URL 'ute-n-25'. It shows a 'RESUMEN DE SECCIÓN' section with a Python class definition:

```
class Snake:
    def __init__(i):
        self.sound = 'Sssssss'
```

Below the code, there is a text box with the instruction: 'El constructor \_\_init\_\_() carece del parámetro obligatorio (deberíamos llamarlo =self para cumplir con los estándares)'. This is followed by 'Ejercicio 3' with the task: 'Modifica el código para garantizar que la propiedad: venomous sea privada.' The code box shows the modified class:

```
class Snake:
    def __init__(self):
        self.venomous = True
```

Below this, it says 'El código debería verse como sigue:' and shows the final corrected code:

```
class Snake:
    def __init__(self):
        self.__venomous = True
```

### 3.3.1.9 RESUMEN DE SECCIÓN.

The screenshot shows a web browser with two tabs. The left tab is 'Plataforma Educativa Institución' with the URL 'https://elearning.ut...'. It displays a user profile for 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' with a sidebar menu containing 'Área personal', 'Perfil', and 'Calificaciones'. The right tab is 'Edube Interactive : PE2 -- Modu...' with the URL 'ute-n-26'. It shows a 'RESUMEN DE SECCIÓN' section with a Python class definition:

```
class Python:
    population = 1
    victims = 0
    def __init__(self):
        self.length_ft = 3
        self.__venomous = False
```

Below the code, there is a text box with the instruction: '¿Cuáles de las propiedades de la clase: Python, son variables de instancia y cuáles son variables de clase? ¿Cuáles de ellos son privados?'. This is followed by 'Ejercicio 1' with the task: '¿Cuáles de las propiedades de la clase: Python, son variables de instancia y cuáles son variables de clase? ¿Cuáles de ellos son privados?'. The text box contains the following explanation:

3. Una **variable de clase** es una propiedad que existe exactamente en una copia y no necesita ningún objeto creado para ser accesible. Estas variables no se muestran como contenido de `__dict__`.

Todas las variables de clase de una clase se almacenan dentro de un diccionario dedicado llamado `__dict__`, contenido en cada clase por separado.

4. Una función llamada `hasattr()` se puede utilizar para determinar si algún objeto o clase contiene cierta propiedad especificada.

Por ejemplo:

```
1 class Sample:
2     game = 0 # Clase variable.
3     def __init__(self):
4         self.alpha = 1 # Variable de instancia.
5         self.__delta = 2 # Variable de instancia privada.
6
7
```

Below this, there is a text box with the instruction: 'Vas a negar la propiedad \_\_venomous del objeto: version\_2, ignorando el hecho de que la propiedad es privada. ¿Cómo vas a hacer esto?'. This is followed by 'Ejercicio 2' with the task: 'Vas a negar la propiedad \_\_venomous del objeto: version\_2, ignorando el hecho de que la propiedad es privada. ¿Cómo vas a hacer esto?'. The code box shows the code to be executed:

```
version_2 = Python()

version_2._Python__venomous = not version_2._Python__venomous
```



### 3.4.1.11 RESUMEN DE SECCIÓN.

The image displays two screenshots of the Edube Interactive platform interface, showing a user's profile and a section summary for a Python exercise.

**Left Screenshot:** The user's profile is visible on the left, showing the name "ALAN FRANCISCO EMMANUEL AGUILAR FUENTES". The main content area displays a section summary titled "RESUMEN DE SECCION". It includes a description of the exercise and a code snippet for a class named `Snake`.

**Right Screenshot:** The user's profile is visible on the left, showing the name "ALAN FRANCISCO EMMANUEL AGUILAR FUENTES". The main content area displays a section summary titled "RESUMEN DE SECCION". It includes a description of the exercise and a code snippet for a class named `Snake`.

**Ejercicio 1**

La declaración de la clase `Snake` se muestra a continuación. Enriquezca la clase con un método llamado `increment()`, el cual incrementa en 1 la propiedad `victima`.

```
class Snake:
    def __init__(self):
        self.victima = 0
```

**Review**

```
class Snake:
    def __init__(self):
        self.victima = 0

    def increment(self):
        self.victima += 1
```

**Ejercicio 2**

Redefina el constructor de la clase `Snake` para que tenga un parámetro que inicialice el campo `victima` con un valor pasado al objeto durante la construcción.

**Review**

```
class Snake:
    def __init__(self, victima):
        self.victima = victima
```

**Ejercicio 3**

¿Puedes predecir el resultado del siguiente código?

```
class Snake:
    pass

class Python(Snake):
    pass

print(Python.__name__, "es una", Snake.__name__)
print(Python.__bases__[0].__name__, "puede ser una", Python.__name__)
```

**Review**

Python es una Snake  
Snake puede ser una Python

### 3.5.1.22 RESUMEN DE SECCIÓN.

Edube Interactive : PE2 -- Modu... | Edube Interactive : PE2 -- Modu...

Resumen de Sección 1/2

5. Una función sin parámetros llamada `espejo()` retorna la referencia a la superclase más cercana de la clase. Por ejemplo:

```
1 class Mouse:
2     def __init__(self):
3         self.name = "Mouse"
4
5 class SubMouse(Mouse):
6     def __init__(self):
7         self.name = "SubMouse"
8
9 class SubSubMouse(SubMouse):
10     def __init__(self):
11         self.name = "SubSubMouse"
12
13 print(espejo(Mouse)) # Imprime "Mouse"
14 print(espejo(SubMouse)) # Imprime "SubMouse"
15 print(espejo(SubSubMouse)) # Imprime "SubSubMouse"
```

6. Los métodos, así como las variables de instancia y de clase definidas en una superclase son heredados automáticamente por sus subclases. Por ejemplo:

```
1 class Mouse:
2     Population = 0
3     def __init__(self, name):
4         self.name = name
5         self.name = name
6
7 class SubMouse(Mouse):
8     def __init__(self):
9         self.name = "SubMouse"
10
11 class SubSubMouse(SubMouse):
12     def __init__(self):
13         self.name = "SubSubMouse"
14
15 print(SubMouse.Population) # Imprime "0"
16 print(SubSubMouse.Population) # Imprime "0"
```

7. Para encontrar cualquier propiedad de objetos/clases, Python la busca dentro:

- Del objeto mismo.
- Todas las clases involucradas en la línea de herencia del objeto de abajo hacia arriba.
- Si existe más de una clase en una ruta de herencia en particular, Python las examina de izquierda a derecha.
- Si lo mencionado anteriormente falla, la excepción `AttributeError` es generada.

8. Si alguna de las subclases define un método, variable de clase o variable de instancia del mismo nombre que existe en la superclase, el nuevo nombre *anula* cualquiera de las instancias anteriores del nombre. Por ejemplo:

```
1 class Mouse:
2     def __init__(self):
3         self.name = "Mouse"
```

### 3.5.1.23 RESUMEN DE SECCIÓN.

Edube Interactive : PE2 -- Modu... | Edube Interactive : PE2 -- Modu...

Resumen de Sección 2/2

Ejercicio 1

¿Cuál es el resultado esperado del siguiente código?

```
print(rocky)
print(jana)
```

Responde

Collie dice: ¡Bueno! ¡No hagas, no desistas!  
Obermann dice: ¡Bueno! ¡Quédese donde está, intruso!

Ejercicio 2

¿Cuál es el resultado esperado del siguiente código?

```
print(SubMouse(MouseDog, Dog, SubMouse(MouseDog, SubMouseDog))
print(SubMouse(MouseDog, SubMouseDog, SubMouse(MouseDog, SubMouseDog))
```

Responde

True False  
False True

Ejercicio 3

¿Cuál es el resultado esperado del siguiente código?

```
print(jana la jana, rocky la jana)
print(rocky, jana)
```

Responde

True False  
2

The screenshot shows the Edube Interactive platform interface. On the left, a sidebar displays the user's profile (ALAN FRANCISCO EMMANUEL AGUILAR FUENTES) and a list of course sections. The main content area shows the 'RESUMEN DE SECCIÓN 212' for the course 'Fundamentos de POO: Herencia'. Below the title, there are four exercises (Ejercicio 2, Ejercicio 3, Ejercicio 4, and Ejercicio 5) with their respective questions and code snippets. Each exercise has a 'Reservar' button and a 'Ver' button.

**Ejercicio 2**  
¿Cuál es el resultado esperado del siguiente código?  

```
print(isinstance(RoverDog, Dog))
print(isinstance(RoverDog, GuardDog))
print(isinstance(RoverDog, RoverDog))
```

 True False  
 False True

**Ejercicio 3**  
¿Cuál es el resultado esperado del siguiente código?  

```
print(isinstance(RoverDog, RoverDog))
print(isinstance(RoverDog, RoverDog))
```

 True False  
 2

**Ejercicio 4**  
Define una subclase de `BaseDog` llamada `GuardDog`, y encapsula con un método `__str__()` que anule un método heredado del mismo nombre `__str__()` que debe retornar la cadena "GuardDog me gusta las montañas".  

```
class GuardDog(BaseDog):
    def __str__(self):
        return "GuardDog me gusta las montañas"
```

### 3.6.1.9 RESUMEN DE SECCIÓN.

The screenshot shows the Edube Interactive platform interface. On the left, a sidebar displays the user's profile (ALAN FRANCISCO EMMANUEL AGUILAR FUENTES) and a list of course sections. The main content area shows the 'RESUMEN DE SECCIÓN 217' for the course 'Fundamentos de POO: Herencia'. Below the title, there are two exercises (Ejercicio 1 and Ejercicio 2) with their respective questions and code snippets. Each exercise has a 'Reservar' button and a 'Ver' button.

**Ejercicio 1**  
¿Cuál es el resultado esperado del siguiente código?  

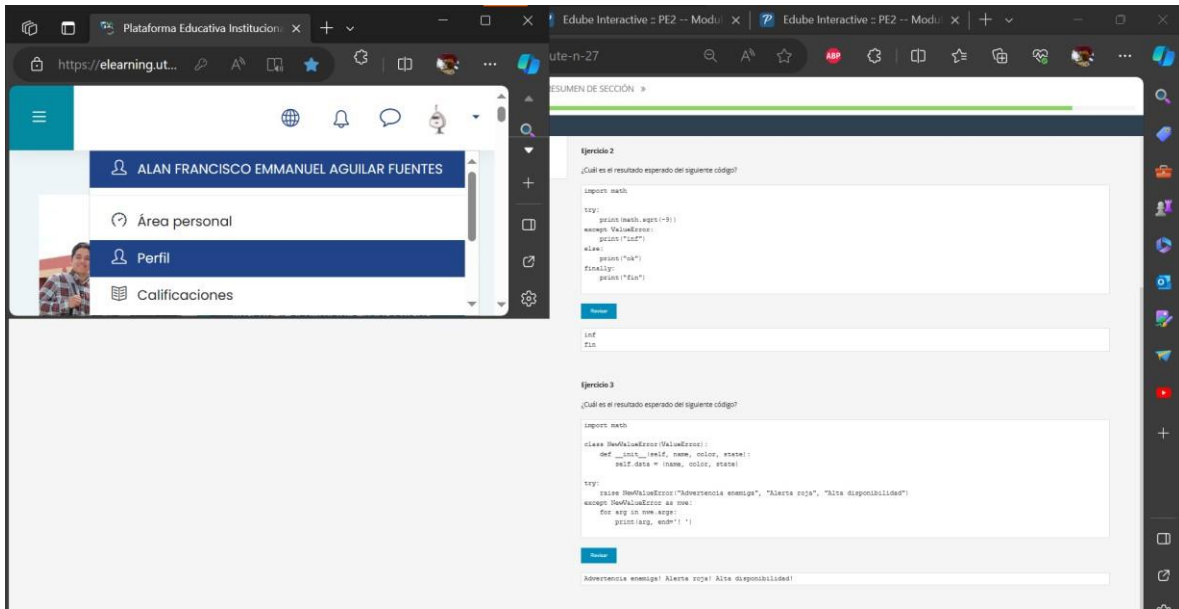
```
import math
try:
    print(math.sqrt(9))
except ValueError:
    print("Error")
else:
    print("Ok")
```

 3.0  
 Ok

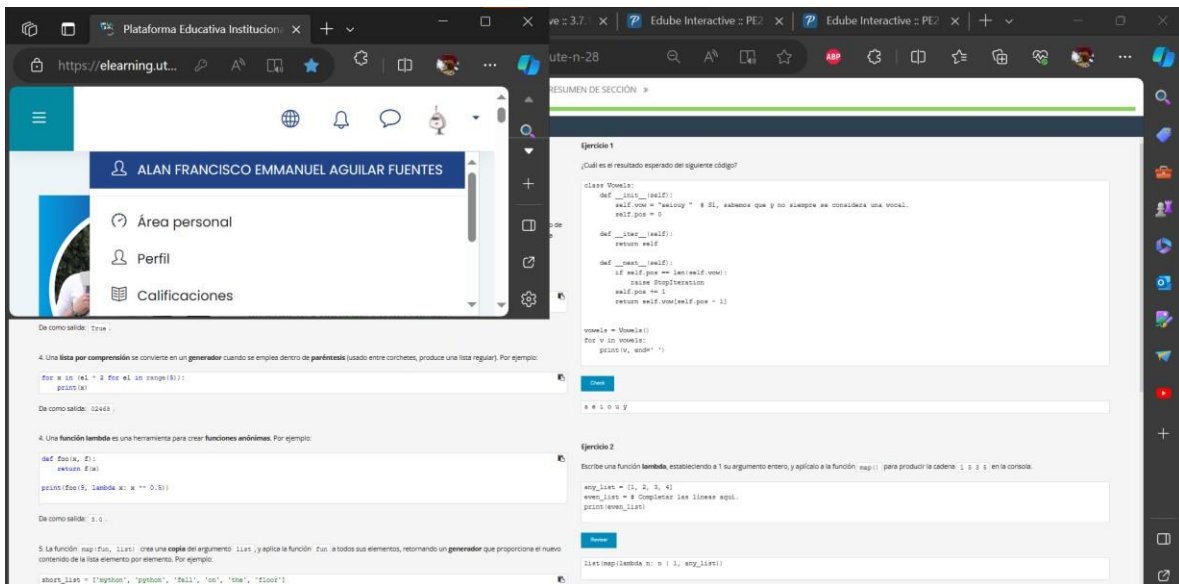
**Ejercicio 2**  
¿Cuál es el resultado esperado del siguiente código?  

```
import math
try:
    print(math.sqrt(9))
except ValueError:
    print("Error")
else:
    print("Ok")
finally:
    print("Fin")
```

 3.0  
 Fin



#### 4.1.1.15 RESUMEN DE SECCIÓN.



The screenshot shows a web browser window with the URL `https://elearning.ut.../ute-n-28`. The page displays a user profile for 'ALAN FRANCISCO EMMANUEL AGUILAR FUENTES' with navigation links for 'Área personal', 'Perfil', and 'Calificaciones'. The main content area is titled 'RESUMEN DE SECCIÓN' and contains 'Ejercicio 3'. The exercise asks for the expected result of a Python code snippet. The code defines a function `replace_spaces` that replaces spaces in a string with underscores. The expected output is `And Your Four Somewhat Completely Different`. Below the code, there is a note about Python 3.6's recommendation to use `def` for function definitions instead of `lambda`. The bottom of the page contains a footer with the text: 'Es importante comprender que a la realidad a menudo le gusta dibujar sus propios escenarios, que no necesariamente siguen las convenciones o recomendaciones formales. Si decides seguirlos o no, dependerá de muchas cosas: tus preferencias, otras convenciones adoptadas, las pautas internas de la empresa, la compatibilidad con el código existente, etc. Tómalo en cuenta esto.'

#### 4.2.1.12 RESUMEN DE SECCIÓN.

The screenshot shows a web browser window with the URL `https://elearning.ut.../ute-n-29`. The page displays the same user profile as the previous screenshot. The main content area is titled 'RESUMEN DE SECCIÓN' and contains three exercises. 'Ejercicio 1' asks for the code to create a new text file. 'Ejercicio 2' asks for the significance of the value represented by `errno.ENOENT`. 'Ejercicio 3' asks for the output of a Python code snippet that uses `open()` to create a file and `write()` to write text to it. The code snippet is:
 

```
try:
    stream = open("file1", "w")
    print("hello")
    stream.close()
except IOError as error:
    if error.errno == errno.ENOENT:
        print("No existe")
    else:
        print("Desconocido")
```

 The expected output is `hello`. The bottom of the page contains a footer with the text: 'Es importante comprender que a la realidad a menudo le gusta dibujar sus propios escenarios, que no necesariamente siguen las convenciones o recomendaciones formales. Si decides seguirlos o no, dependerá de muchas cosas: tus preferencias, otras convenciones adoptadas, las pautas internas de la empresa, la compatibilidad con el código existente, etc. Tómalo en cuenta esto.'

#### 4.3.1.18 RESUMEN DE SECCIÓN.

Platforma Educativa Institución: x + v

ute-n-30

RESUMEN DE SECCIÓN

Área personal

Perfil

Calificaciones

Ejercicio 1

¿Qué se espera del método `readLine()` cuando el stream está asociado con un archivo vacío?

Responder

Una lista vacía (una lista de longitud cero).

Ejercicio 2

¿Qué se pretende hacer con el siguiente código?

```
for (line in open("E1A", "r")) {
    for (char in line) {
        if (char.toLowerCase() != 'a') {
            print(char, end=" ")
        }
    }
}
```

Responder

Copia el contenido del archivo directamente a la consola, ignorando las vocales.

Ejercicio 3

Vas a procesar un mapa de bits almacenado en un archivo llamado `image.png` y quieres leer su contenido como un todo en una variable bytearray llamada `image`. Agrega una línea al siguiente código para lograr este objetivo.

```
try {
    stream = open("image.png", "rb")
    # Interpreta una línea aquí.
    stream.close()
    except IOError:
    print("Fallido")
    else:
    print("Exito")
```

Responder

```
image = bytearray(stream.read())
```

#### 4.4.1.9 RESUMEN DE SECCIÓN.

Platforma Educativa Institución: x + v

ute-n-31

RESUMEN DE SECCIÓN

Área personal

Perfil

Calificaciones

Ejercicio 1

¿Cuál es el resultado de siguiente fragmento si se ejecuta en Unix?

```
import os
print(os.name)
```

Responder

posix

Ejercicio 2

¿Cuál es el resultado de siguiente fragmento de código?

```
import os
os.mkdir("The13a")
print(os.listdir())
```

Responder

```
["The13a"]
```

#### 4.5.1.23 RESUMEN DE SECCIÓN.

Plataforma Educativa Institución: x + v

ute-n-32

RESUMEN DE SECCIÓN

Ejercicio 1

¿Cuál es el resultado del siguiente fragmento de código?

```
from datetime import datetime
t = datetime(14, 20)
print(t.strftime('%d-%M-%Y'))
```

Responde

14-00-00

Ejercicio 2

¿Cuál es el resultado del siguiente fragmento de código?

```
from datetime import datetime
dt1 = datetime(2020, 3, 20, 14, 0)
dt2 = datetime(2020, 3, 20, 14, 0)
print(dt1 - dt2)
```

Responde

1 day, 0:00:00

#### 4.6.1.14 RESUMEN DE SECCIÓN

Plataforma Educativa Institución: x + v

ute-n-33

RESUMEN DE SECCIÓN

Ejercicio 1

¿Cuál es el resultado del siguiente fragmento de código?

```
import datetime
print(datetime.datetime.strptime('2020, 3', '%Y, %m'))
```

Responde

2020-03-01 00:00:00

Ejercicio 2

¿Cuál es el resultado del siguiente fragmento de código?

```
import datetime
d = datetime.datetime.strptime('2020, 3, 20', '%Y, %m, %d')
print(datetime.datetime.strptime('2020, 3, 20', '%Y, %m, %d'))
```

Responde

2020-03-20 00:00:00