

# TRABAJO FINAL: INICIACIÓN A PYTHON

## TRABAJO FINAL

---

### 1.0. INSTRUCCIONES

Deberás entregar el ejercicio en un archivo comprimido. Este deberá incluir:

- Un archivo .py por ejercicio.
- o
- Un archivo .ipynb (jupyter notebook) que incluya la realización de todos los ejercicios.

*Importante:*

- *Si el trabajo se entrega con ejercicios sin realizar este quedará suspendido automáticamente.*
- *Recuerda que es obligatorio utilizar comentarios durante la realización de los ejercicios. Su falta de uso podrá ser motivo de penalización en la nota final del trabajo.*

## 1.1. EJERCICIO 1 (1.5 PUNTOS)

Crea un programa que solicite al usuario introducir un número entero positivo y lo almacene en una variable.

Si lo introducido por el usuario:

- a) *No es un número* se deberá mostrar por pantalla un mensaje de error que le informe de que lo introducido no es un número, similar a: "El dato introducido no es número. Por favor, vuelva a intentarlo". Tras ello, el programa deberá volver a solicitarle que introduzca un número entero positivo.
- b) *No es un número entero* se deberá mostrar por pantalla un mensaje de error que le informe de que lo introducido no es un número entero, similar a: "No ha introducido un número entero. Por favor, vuelva a intentarlo". Tras ello, el programa deberá volver a solicitarle que introduzca un número entero positivo.
- c) *No es un número entero positivo* se deberá mostrar por pantalla un mensaje de error que le informe de que lo introducido no es un número entero positivo, similar a: "No ha introducido un número entero positivo. Por favor, vuelva a intentarlo". Tras ello, el programa deberá volver a solicitarle que introduzca un número entero positivo.
- d) *Es un número entero positivo* deberás crear una función que lo reciba y compruebe si el número es primo o no. La función deberá devolver un mensaje que informe al usuario del resultado.

*Recuerda: Un número primo es un número entero divisible tanto por sí mismo como por el 1.*

## 1.2. EJERCICIO 2 (1.5 PUNTOS)

	Latín	Castellano	Francés	Inglés
Alumno 1	8	8	9	4
Alumno 2	7	6	7	2
Alumno 3	10	7	8	9
Alumno 4	4	4	3	2
Alumno 5	9	8	9	6

Con los datos mostrados en la tabla anterior crea un programa que:

- Muestre por pantalla cuántos alumnos suspendieron cada asignatura. Ejemplo: Inglés se ha suspendido por 3 alumnos
- Realiza la media de las notas de cada alumno y muéstralas por pantalla. Ejemplo: El alumno1 ha tenido una nota media de 7.25
- Muestra por pantalla los nombres de los alumnos que han aprobado el curso, sabiendo que para aprobar necesitan obtener una nota media igual o superior a 5. Ejemplo: Los alumnos que han aprobado el curso son: Alumno1, .....

### 1.3. EJERCICIO 3 (2 PUNTOS)

Crea un programa que simule una partida de dados entre dos amigos, siendo las normas del juego las siguientes:

Cada jugador lanzará dos dados de 6 simultáneamente y apuntará los resultados obtenidos. Si alguno de los números obtenidos por los jugadores coincide, el Jugador1 ganará la ronda. Por el contrario, si ninguno de los números coincide ganará el Jugador2.

*Ejemplo 1:*

Jugador1 ha obtenido: 6 y 4

Jugador2 ha obtenido: 4 y 5

Ganaría la ronda: Jugador1

*Ejemplo 4:*

Jugador1 ha obtenido: 1 y 4

Jugador2 ha obtenido: 2 y 4

Ganaría la ronda: Jugador1

*Ejemplo 2:*

Jugador1 ha obtenido: 4 y 4

Jugador2 ha obtenido: 3 y 5

Ganaría la ronda: Jugador1

*Ejemplo 5:*

Jugador1 ha obtenido: 1 y 3

Jugador2 ha obtenido: 2 y 4

Ganaría la ronda: Jugador2

*Ejemplo 3:*

Jugador1 ha obtenido: 1 y 4

Jugador2 ha obtenido: 5 y 5

Ganaría la ronda: Jugador1

*Ejemplo 6:*

Jugador1 ha obtenido: 2 y 4

Jugador2 ha obtenido: 3 y 2

Ganaría la ronda: Jugador1

El juego finalizará cuando alguno de los jugadores gane 3 rondas. El programa deberá enviar un mensaje de enhorabuena al jugador que haya ganado la partida y preguntar si se desea jugar otra. Si el usuario responde NO, el programa finalizará, en caso de que responda SI, el programa se repetirá y en caso de que escriba algo diferente, se le deberá indicar que no se ha entendido correctamente su respuesta y se le solicitará repetirla.

## 1.4. EJERCICIO 4 (2 PUNTOS)

Crea una clase llamada PERSONA con las siguientes propiedades: nombre, apellido, nacionalidad y una función que se llame saludar que muestre un saludo por pantalla.

Después crearás una clase llamada ITALIANO, subclase de PERSONA, que tendrá como propiedad: idioma principal cuyo valor estará ya definido y será italiano. Además, la clase ITALIANO establecerá que el valor de la nacionalidad (propiedad que hereda de la clase PERSONA) estará ya definido y será siempre: italiana. Por último, la clase incorporará un método llamado saludar que mostrará por pantalla un saludo similar a: "Bonjorno! Mi nombre es: ..... , soy de nacionalidad italiana y mi idioma principal es italiano."

Tras la creación de las dos clases, deberás crear un programa que:

- Solicite al usuario los datos necesarios para crear un objeto de la clase Italiano. Recuerda que el idioma y la nacionalidad ya tienen valores definidos por lo que no se le deberían preguntar al usuario.
- El programa deberá contener un método que reciba los datos introducidos por el usuario para crear con ellos un objeto/instancia de la clase italiano.
- Por último, se deberá mostrar por pantalla el mensaje del saludo, llamando a la función saludar desde el objeto de la clase ITALIANO que se ha creado.

## 1.5. EJERCICIO 5 (2 PUNTOS)

Para este programa deberás definir una **tupla** compuesta por 15 números enteros positivos aleatorios entre 0 y 100, estos números serán los ganadores del último sorteo de la lotería.

Tras ello, el programa solicitará al usuario que introduzca un número entero positivo. El programa deberá comprobar que lo introducido por el usuario es un número entero positivo de igual forma que se realizó en el ejercicio 1, el programa no podrá continuar hasta que el dato introducido cumpla los requisitos, es decir, sea un número entero positivo.

Una vez que el dato introducido sea correcto, el programa mostrará por pantalla la lista de números ganadores del sorteo e informará del número ganador más pequeño y del número ganador más grande.

Después, el programa comprobará si el número indicado por el usuario, es decir, aquel con el que ha participado en el sorteo, se encuentra entre los ganadores (es decir, en la tupla). Si el número indicado por el usuario aparece en la lista de números ganadores el usuario habrá ganado 15€. Si el número se ha repetido varias veces, por cada repetición se le sumarán 5€ extra. El programa deberá informar al usuario del resultado y después finalizará. Ejemplo: Si el usuario introduce el 7 y este sólo aparece una vez el usuario habrá ganado 15€. Por el contrario, si el 7 apareciese dos veces, habría ganado 15€ por aparecer en la lista y 5€ extra por la repetición, total 20€.

*Ejemplo 1:* "¡Felicitaciones! Su número: .... se encuentra dentro de la lista de ganadores. Ha ganado un total de ....€"

*Ejemplo 2:* "¡Felicitaciones! Su número: .... se encuentra dentro de la lista de ganadores y además se ha repetido ... veces . Ha ganado un total de .... €"

Si el número no aparece en la lista de números ganadores se le deberá preguntar al usuario si dispone de otro número. Si el usuario responde SI entonces el programa se repetirá, si el usuario responde NO el programa finalizará, si el usuario responde algo diferente, se le pedirá que repita de nuevo su respuesta.

*Ejemplo 1:* Lo sentimos. Su número no ha resultado premiado. ¿Desea volver a intentarlo?

*Ejemplo 2:* "No hemos logrado entender su respuesta. Repítala, por favor."

## 1.6. EJERCICIO 6 (1 PUNTOS)

Para este programa deberás crear una agenda en la que un usuario pueda almacenar sus contactos.

La agenda será un diccionario cuya clave será el nombre de los contactos y cuyo valor serán sus teléfonos. En la agenda no se podrán guardar nombres repetidos. En caso de que el usuario intenté introducir un nombre que ya exista en la agenda se le deberá indicar que no es posible y que por favor, vuelva a intentarlo.

Tras introducir cada contacto se le deberá preguntar al usuario si desea continuar o salir del programa. Si la respuesta del usuario no es clara el programa deberá pedirle que repita su respuesta.

Cuando el usuario desee salir, se le mostrará la agenda y el programa finalizará.