

# PROGRAMACIÓN AVANZADA CON PYTHON

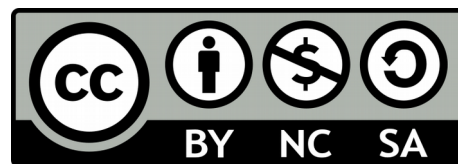
(CEFIRE CTEM)



Introducción  
Ejercicios voluntarios

Esta obra está sujeta a la licencia Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visitad

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.



Autora: María Paz Segura Valero (segura\_marval@gva.es)

## CONTENIDO

1. Introducción.....	2
2. Ejercicios voluntarios.....	2
2.1. Ejercicio 1: Multiplicando.....	2
2.2. Ejercicio 2: Validar opción de menú.....	3
2.3. Ejercicio 3: Jugadores on-line.....	4
2.4. Ejercicio 4: Vuelo.....	6

## 1. Introducción

En este documento puedes encontrar una serie de **ejercicios voluntarios** para que pongas en práctica lo aprendido durante esta unidad.

En el aula virtual dispondrás de las soluciones pero te recomiendo que intentes solucionarlos por ti mismo/a porque, aunque no sean obligatorios para superar el curso, sí pueden ayudarte a enfrentarte al *ejercicio obligatorio* del final.

Puedes utilizar el foro del curso para consultar tus dudas.

## 2. Ejercicios voluntarios

### 2.1. Ejercicio 1: Multiplicando

Escribe un programa que pida al usuario un número entero del 1 al 10 y muestre por pantalla su tabla de multiplicar.

Deberás convertir el texto introducido por el usuario al tipo entero (*int*) y validar que se encuentre en el rango correcto. Si no es así, mostraremos un error al usuario y acabará el programa.

```
lliurex@lliurex-VirtualBox:/media/sf_COMPARTIDA/MPAZ$ python3.8 ud2_ejer3_multiplicando.py

Dame un número entero del 1 al 10: 5

TABLA DE MULTIPLICAR DEL Nº 5 :
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40
9 * 5 = 45
10 * 5 = 50
```

Figura 1: Ejemplo de ejecución

En el aula virtual dispones de la solución del ejercicio en el fichero **ud1\_ejer1\_multiplicando.py**.

## 2.2. Ejercicio 2: Validar opción de menú

Escribe un programa en Python que contenga una función llamada **valida\_opcion()**. Esta función mostrará un menú por pantalla y validará que la opción elegida por el usuario es una de las correctas.

Para ello, habrá que tener en cuenta los siguientes puntos:

- La función no recibirá ningún parámetro.
- Estará mostrando el menú y pidiendo la opción al usuario continuamente hasta que se asegure de que la opción elegida está dentro de las correctas.
- Devolverá la opción elegida en la llamada.
- Documenta la función con un **docstring** donde expliques para qué sirve y cómo se utiliza.

El menú que debe mostrar es el que aparece a continuación:

```
=====
GAMIFICACIÓN EN EL AULA
=====
1 - Cargar datos del fichero
2 - Imprimir datos
3 - Jugar
4 - Guardar datos
5 - Cambiar contraseña
0 - SALIR
-----
Dame la opción:
```

En la imagen siguiente podemos ver algunos ejemplos de ejecución del programa:

```
=====
GAMIFICACIÓN EN EL AULA
=====
1 - Cargar datos del fichero
2 - Imprimir datos
3 - Jugar
4 - Guardar datos
5 - Cambiar contraseña
0 - SALIR
-----
Dame la opción: 8
Por favor, vuelve a intentarlo.
```

*Figura 2: Opción no válida*

```
=====
GAMIFICACIÓN EN EL AULA
=====
1 - Cargar datos del fichero
2 - Imprimir datos
3 - Jugar
4 - Guardar datos
5 - Cambiar contraseña
0 - SALIR
-----
Dame la opción: 2
La opción seleccionada es: 2
```

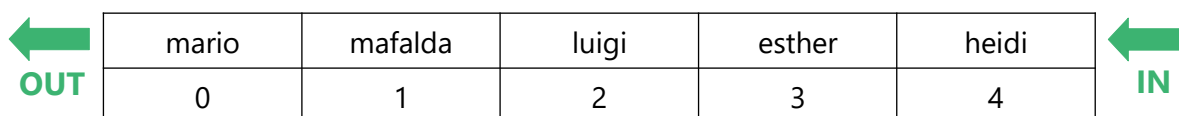
*Figura 3: Opción válida*

En el aula virtual dispones de la solución del ejercicio en el fichero **ud1\_ejer2\_valida\_opcion.py**.

## 2.3. Ejercicio 3: Jugadores on-line

Vamos a crear un programa que gestione la lista de jugadores on-line de un juego.

Para simplificar su gestión simularemos una cola FIFO (First In First Out), de tal forma que, cuando llegue un jugador lo añadiremos al final de la cola y cuando se vaya un jugador siempre se irá el de la primera posición.



El programa debe funcionar de la siguiente forma:

1. Inicio: podemos tener una lista creada de jugadores on-line con varios de ellos. Si es así, la mostraremos al principio del programa para que el usuario sepa con qué jugadores cuenta ya la lista.
2. Cuerpo: mostraremos un menú como en el que se muestra a continuación.

```
=====
                JUGADORES ON-LINE
=====
1 - Llega un jugador nuevo
2 - Se va un jugador
3 - FIN
-----
Dame la opción:
```

Figura 4: Menú del programa

Según la opción seleccionada, el programa reaccionará como se indica en la siguiente tabla:

Opción	Respuesta
1	Se pide el nombre del nuevo jugador al usuario, se le da la bienvenida y se añade al final de la lista de jugadores.  Además, se muestra la lista actual de jugadores.
2	Se extrae el jugador del principio de la lista y se muestra un mensaje informativo. Por ejemplo: "El jugador Pepito se va".

	Además, se muestra la lista actual de jugadores.
3	Se muestra la lista actual de jugadores y acaba el programa.
<i>otra</i>	Se mostrará un mensaje de error y se seguirá pidiendo la opción.

3. Final: mostraremos un mensaje de despedida.

A continuación puedes ver un ejemplo completo de ejecución del programa:

```

LISTA DE JUGADORES: mario - mafalda - luigi - esther - heidi - songoku - beauty - beast -
=====
      JUGADORES ON-LINE
=====
1 - Llega un jugador nuevo
2 - Se va un jugador
3 - FIN
-----
Dame la opción: 1

¿Quién eres? pep
Bienvenid@ jugador pep
LISTA DE JUGADORES: mario - mafalda - luigi - esther - heidi - songoku - beauty - beast - pep -

=====
      JUGADORES ON-LINE
=====
1 - Llega un jugador nuevo
2 - Se va un jugador
3 - FIN
-----
Dame la opción: 1

¿Quién eres? mireia
Bienvenid@ jugador mireia
LISTA DE JUGADORES: mario - mafalda - luigi - esther - heidi - songoku - beauty - beast - pep - mireia -

```

*Figura 5: Añadimos dos jugadores nuevos*

```

=====
      JUGADORES ON-LINE
=====
1 - Llega un jugador nuevo
2 - Se va un jugador
3 - FIN
-----
Dame la opción: 2

Adiós al jugador mario

LISTA DE JUGADORES: mafalda - luigi - esther - heidi - songoku - beauty - beast - pep - mireia -

=====
      JUGADORES ON-LINE
=====
1 - Llega un jugador nuevo
2 - Se va un jugador
3 - FIN
-----
Dame la opción: 2

Adiós al jugador mafalda

LISTA DE JUGADORES: luigi - esther - heidi - songoku - beauty - beast - pep - mireia -

```

Figura 6: Eliminamos dos jugadores

```

=====
      JUGADORES ON-LINE
=====
1 - Llega un jugador nuevo
2 - Se va un jugador
3 - FIN
-----
Dame la opción: 3

LISTA DE JUGADORES: luigi - esther - heidi - songoku - beauty - beast - pep - mireia -
Hasta pronto

```

Figura 7: Salimos del programa

Puedes crear el programa desde cero o basarte en el esquema que tienes en el fichero **ud1\_ejer3\_jugadores\_online (ESQUEMA).py** del aula virtual. En este caso, deberás sustituir las instrucciones *pass* por las instrucciones adecuadas para que funcione el programa según el enunciado.

En el aula virtual dispones de la solución del ejercicio en el fichero **ud1\_ejer3\_jugadores\_online.py**.

## 2.4. Ejercicio 4: Vuelo

Crear un programa que utilice un diccionario para gestionar la información de un vuelo. Originalmente el diccionario contendrá la siguiente información:

Clave	Valor
origen	valencia

destino	menorca
día	15-08
clase	turista

El programa mostrará el siguiente menú:

```

=====
          GESTIÓN DE UN VUELO
=====
1 - Imprimir datos vuelo
2 - Imprimir claves
3 - Añadir <pasajeros>
4 - Imprimir un valor
5 - Borrar clave
0 - SALIR
-----
Dame la opción:

```

Según la opción seleccionada, el programa reaccionará como se indica en la siguiente tabla:

Opción	Respuesta
1	Se imprimen los datos del diccionario ( <i>claves</i> y <i>valores</i> ).
2	Se imprimen las <i>claves</i> del diccionario como una lista de valores. <u>Ejemplo:</u> <code>CLAVES: origen, destino, día, clase,</code>
3	Se pide el número de pasajeros del vuelo y se crea la clave "pasajeros" con el valor introducido por el usuario.
4	Se pregunta al usuario el nombre de una <i>clave</i> del diccionario. Si es una clave correcta, se muestra el valor asociado. Si no, se avisa del error.
5	Se pregunta al usuario el nombre de la <i>clave</i> que quiere borrar del diccionario. Si es una clave correcta, se borra el par ( <i>clave:valor</i> ) del diccionario. Si no, se avisa del error.
0	El programa acaba.



Después de realizar las tareas correspondientes a la opción seleccionada, se volverá a mostrar el menú al usuario.

Puedes crear el programa desde cero o basarte en el esquema que tienes en el fichero **ud1\_ejer4\_vuelo(ESQUEMA).py** del aula virtual. En este caso, deberás sustituir las instrucciones *pass* por las instrucciones adecuadas para que funcione el programa según el enunciado.

En el aula virtual dispones de la solución del ejercicio en el fichero **ud1\_ejer4\_vuelo.py**.