

Informática - Práctica de laboratorio 07

Programación VI: Colecciones

1. Normas de Entrega

Todas las prácticas deberán entregarse siguiendo una convención de nombres y un formato específico. En el caso de las prácticas de Python, se solicitará un script de Python para cada ejercicio, generando así varios archivos de texto por práctica.

Todas las prácticas se desarrollarán utilizando el IDE [spyder](https://www.spyder-ide.org/). [Spyder](https://www.spyder-ide.org/) es un entorno científico de código abierto y gratuito diseñado para científicos, ingenieros y analistas de datos. Ofrece una combinación única de funcionalidades avanzadas para edición, análisis, depuración y perfilado, con capacidades excelentes para exploración de datos, ejecución interactiva, inspección profunda y visualización de paquetes científicos. Puedes acceder a él aquí: <https://www.spyder-ide.org/>.

1.1. Nombre del Archivo

Cada ejercicio de programación debe estar codificado en un archivo independiente, es decir, cada ejercicio será un archivo separado. A menos que se indique lo contrario, la convención de nombres será *Ejercicio_YY.py*, donde XX será el número de la práctica y YY el número del ejercicio en el documento de la práctica. Por ejemplo, el primer ejercicio de la práctica cero llevará el nombre *Ejercicio_01.py*.

Para entregar los ejercicios, debes acceder a la sección del curso en mi aula virtual (20XX_0_501103_91_G). Allí, en la columna izquierda, aparecerá la sección *Tareas*.



Figura 1: Primer paso para entregar las tareas.

En cada entrega aparecerá una tarea y podrás entregar los ejercicios. También tendrás la opción de escribir un mensaje junto con la entrega. Para cada práctica, se solicitarán diferentes ejercicios, y deberás subir un archivo para cada ejercicio (ver imagen 2).

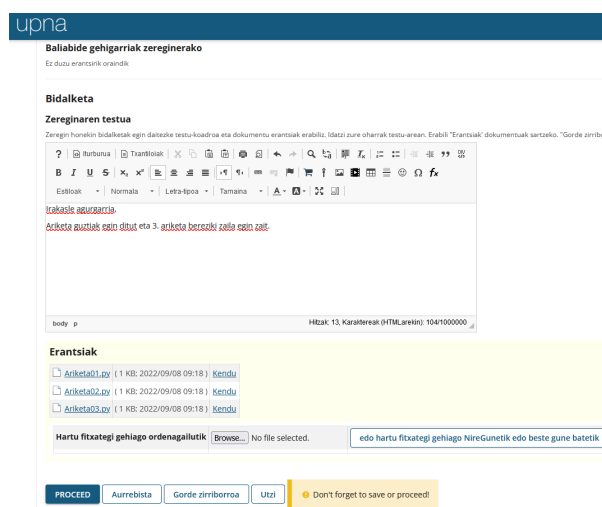


Figura 2: Primer paso para entregar las tareas.

2. Normas de Desarrollo

Con cada práctica tendrás una chuleta (LabXXPR_CheatSheet). En la chuleta encontrarás un resumen del material trabajado en clase. Por ejemplo, en Lab01PR_CheatSheet, se presentarán los elementos necesarios para realizar Lab01PR. Se recomienda imprimir la chuleta o, al menos, tenerla a la vista mientras desarrollas el programa. La CheatSheet contendrá tipos de datos, operadores, estructuras y funciones trabajadas en clase. Cualquier elemento de Python utilizado fuera de estos será evaluado con una nota de 0 en los evaluables.

Al desarrollar los programas, debes seguir las siguientes normas

1. Los nombres de las variables deben ser descriptivos
 - Todos en minúsculas
 - Los nombres compuestos por más de una palabra se separarán con '_'
2. Todos los programas deben tener una cabecera obligatoria
 - La primera línea de comentario debe ser `# python script`
 - En la segunda línea debe aparecer `# Autor: tu nombre`, donde `tu nombre` es el nombre del estudiante
 - En la tercera línea debe aparecer `# Descripción: descripción del programa`, donde `descripción del programa` es una frase o párrafo que describa el programa
3. No se deben usar funciones que no aparezcan en la CheatSheet.

3. Lista de Ejercicios

Ejercicio 1. Escribe una función que reciba un número representando el día de la semana (1 para lunes, 2 para martes, etc.) y luego muestre el nombre del día de la semana. Si el valor ingresado no está en el rango de 1 a 7, el mensaje mostrado deberá ser **Valor desconocido**. Para esta tarea, deberás utilizar el tipo de dato **dict** para relacionar los días de la semana con el número.

Código 1: Programa principal del ejercicio.

```
1 # define la funcion
2 ...
3
4 # ejecucion de la funcion
5 dia_de_la_semana(1)
6 dia_de_la_semana(4)
7 dia_de_la_semana(-5)
```



Ejercicio 2. Escribe una función que reciba un número representando el mes (1 para enero, 2 para febrero, etc.) y el año, y luego muestre la cantidad de días que tiene ese mes. Ten en cuenta los años bisiestos para febrero (29 días). Para determinar si un año es bisiesto, puedes utilizar la función **es_bisiesto** de la práctica 02. Si el valor ingresado no está en el rango de 1 a 12, el mensaje mostrado deberá ser **Valor desconocido**. Para esta tarea, deberás utilizar el tipo de dato **dict** para relacionar los meses con el número de días.

Código 2: Programa principal del ejercicio.

```
1 # define la funcion
2 ...
3
4 # ejecucion de la funcion
5 dias_del_mes(1,2020)
6 dias_del_mes(4,2021)
7 dias_del_mes(-5,-2023)
```



Ejercicio 3. Escribe una función que calcule la longitud de cualquier colección. La función debe aceptar colecciones de tipo **list**, **set** y **dict**.



Ejercicio 4. Crea una función llamada **entero** que tome una cadena de caracteres que representa un número entero positivo como argumento y devuelva el número entero (**int**) correspondiente. Es decir, deberás implementar tu propia versión de la función **int()**. No se debe usar la función **int()** para esta tarea.

□

Ejercicio 5. Escribe una función que lea una secuencia de números enteros positivos y la devuelva como un conjunto (**set**). La secuencia de números debe ser leída desde una única entrada.

Ejemplo: Si la secuencia ingresada es 1 5 3 23 3, el conjunto devuelto debe ser {1,5,3,23}. (El resultado no tiene que seguir necesariamente ese orden)

□

Ejercicio 6. Escribe una función que lea una secuencia de números enteros positivos y la devuelva como una lista (**list**). La secuencia de números debe ser leída desde una única entrada.

Ejemplo: Si la secuencia ingresada es 1 5 3 23 3, la función debe devolver la lista [1,5,3,23,3].

□

Ejercicio 7. Escribe una función que lea una secuencia de pares de nombre de estudiante y nota, y devuelva un diccionario (**dict**) que relacione cada nombre con su nota. La secuencia debe ser leída desde una única entrada.

Ejemplo: Si la secuencia ingresada es Unai Perez:5,Ugaitz Amozarrain:10,Josu Irisarri:8,Harkaitz Goyena:3.2, el diccionario que debe mostrarse en pantalla es {Unai Perez":5,Ügaitz Amozarrain":10,"Josu Irisarri":8,"Harkaitz Goyena":3.2}. (El resultado no tiene que seguir necesariamente ese orden)

□

Ejercicio 8. Escribe una función que lea dos conjuntos (**set**) desde la entrada y devuelva un tercer conjunto con los números que estén en uno de los dos conjuntos, pero no en ambos. Usa la función del ejercicio 5 para solicitar los dos conjuntos. *Ejemplo:* Si los conjuntos son {1,2,3,4,5} y {0,2,3,5,7}, el tercer conjunto devuelto debe ser {0,1,4,7}.

□

Ejercicio 9. Escribe un programa que lea dos frases y muestre las palabras que estén en una u otra, pero no en ambas.

Los caracteres especiales “.,-¿?!;” deben eliminarse para obtener las palabras individuales.

Ejemplo: Si las frases son Hola mundo! aquí estoy y Aquí llega el sol, las palabras que deben mostrarse son hola, mundo, estoy, llega, y sol.

□

Ejercicio 10. Utiliza la función del ejercicio 7 para crear un diccionario que relacione personas con sus edades. Después de presentar diferentes personas al usuario, el usuario selecciona una y el programa debe mostrar su edad.

□

Ejercicio 11. Escribe una función que calcule el promedio de las notas de una clase. Usa la función del ejercicio 7 para crear un diccionario que relacione a los estudiantes con sus notas.

□

Ejercicio 12. Escribe una función llamada `eliminar`. Esta función tomará un conjunto o una lista y un elemento que se desea eliminar. La función devolverá el conjunto o la lista sin el elemento especificado. Es decir, debes implementar la funcionalidad del método `discard()`. En el caso de listas, si el elemento a eliminar está repetido, todos los elementos con ese valor serán eliminados. Si el tipo de dato de entrada no es un `set` o una `list`, el mensaje de error debe ser `Tipo de dato no aceptado`. No se puede usar el método `discard()/remove()` para esta tarea.

□

Ejercicio 13. Escribe una función llamada `unir`. La función tomará dos listas como argumentos y debe devolver una lista con los elementos de ambas listas, sin repeticiones.

□

Ejercicio 14. En un diccionario que relaciona productos con sus precios, escribe una función que identifique el producto más caro. La función debe devolver el producto o productos más caros en el diccionario.

Nota: Si hay dos productos con el precio más alto, la función debe devolver ambos productos.

□

Ejercicio 15. Escribe una función que identifique los números pares en un conjunto de números. La función debe devolver un conjunto con los números pares.

□