	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Edgar Tista García

Asignatura: Estructura de Datos y Algoritmos I

Grupo: 1

No de Práctica(s): 10

Integrante(s): Díaz Hernández Marcos Bryan

No. de Equipo de cómputo empleado:

No. de Lista o Brigada: 9

Semestre: 2020-2

Fecha de entrega: 05 de abril de 2020

Observaciones:

CALIFICACIÓN: _____

Objetivo de la práctica

Aplicar las bases del lenguaje de programación Python en el ambiente de ipython notebook.

Introducción

El siguiente reporte ha sido elaborado para mostrar las características que he aprendido del lenguaje Python durante la elaboración de los ejercicios.

Ejercicios de la guía de laboratorio

- Ejercicio 1

Las distintas secciones del programa estaban enfocadas al uso de los ciclos de repetición, las instrucciones de condición, el uso de bibliotecas estándar, poder generar una gráfica por medio de una biblioteca especial. Lo que requería la actividad era el comentar las secciones de código que me parecieran más inusuales y compararlo con el lenguaje C.

La imagen 1 muestra la combinación de if-else en una misma instrucción de código lo que me pareció muy distinto a las reglas de c, debido a que el else normalmente no lleva una condición y que se debe separar del if.

```
def obtenerMayor_idiom(param1,param2):
    #La variable valor va a tener el valor de param2 si el if es verdadero
    #de lo contrario tendra el valor de param1
    valor = param2 if (param1 < param2) else param1 #dentro de c, nunca habia visto una instruccion de asignacion con if-else
    return valor
```

Imagen 1

La imagen 2 muestra como se hace una condicional con distintos casos lo que en se tendría que hacer con un switch, en python se puede realizar en un if, que hace más fácil la operación cuando todos los casos tienen que llevar a cabo la misma operación.

```
def numeros_idiom(num):
    #La tupla tiene las opciones válidas
    if num in (1,2,3,4): #en c tendria que ser un switch, ya que son opciones determinadas
        print("tu numero es {}".format(num))
    else:
        print ("{} no es una opcion".format(num))
```

Imagen 2

La imagen 3 muestra algo que me pareció curioso, porque el for tienes dos variables por las cuales recorrer, me confunde algunas veces como esta formulada la función ya que, en C, se tendrían que anidar dos ciclos para poder vincular dos variables.

```
#Creando un diccionario
elementos = { 'hidrogeno': 1, 'helio': 2, 'carbon': 6 }

for llave, valor in elementos.items(): #Utiliza un solo ciclo para dos variables, en c no es posible
    print(llave, " = ", valor)         #Imprime todos los valores de un diccionario
```

Imagen 3

La imagen 4 muestra el uso de una biblioteca especial para operaciones matemáticas, lo que me pareció interesante fue que es posible despegar un menú de ayuda que te indica como utilizar la función, además se puede definir el nombre o la operación que se va a utilizar. Cosa que en C es posible pero no hay un menú que te indique como se utiliza la función.

```
#También se pueden importar todas las funciones de la bibliotecas, de esta manera no se tiene que usar el prefijo
#de la biblioteca, que en el ejemplo anterior fue math
from math import * #Se indicae el orrigen y llo que se va a importar

x = cos(pi)  #No se utiliza el prefijo math

print(x)
```

Imagen 4

Análisis de los ejercicios de la práctica.

- Ejercicio 1

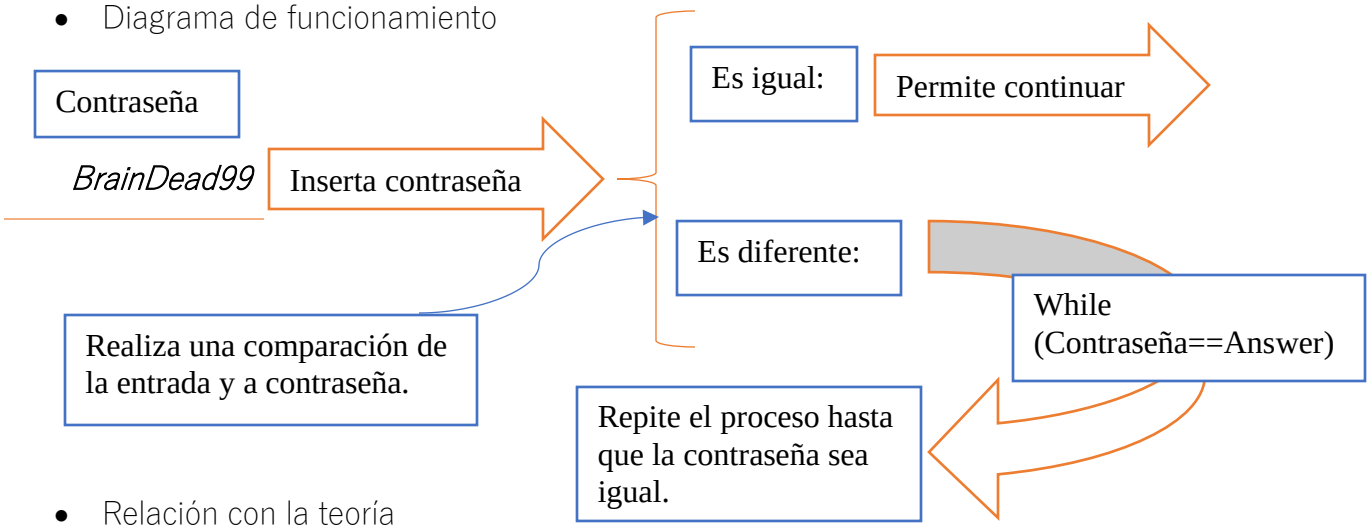
Los ejercicios de la practica se desarrollaron en la IDE de Spyder, que por la experiencia que tuve, tiene varias características que ayudan a poder saber el comportamiento de las variables, una herramienta muy útil.

El primer ejercicio, consistía en definir una contraseña que el usuario tenia que insertar para poder continuar con las instrucciones del programa, el proceso se repite hasta que la contraseña sea la que se definió en el programa.

- ¿Cómo lo resolví?

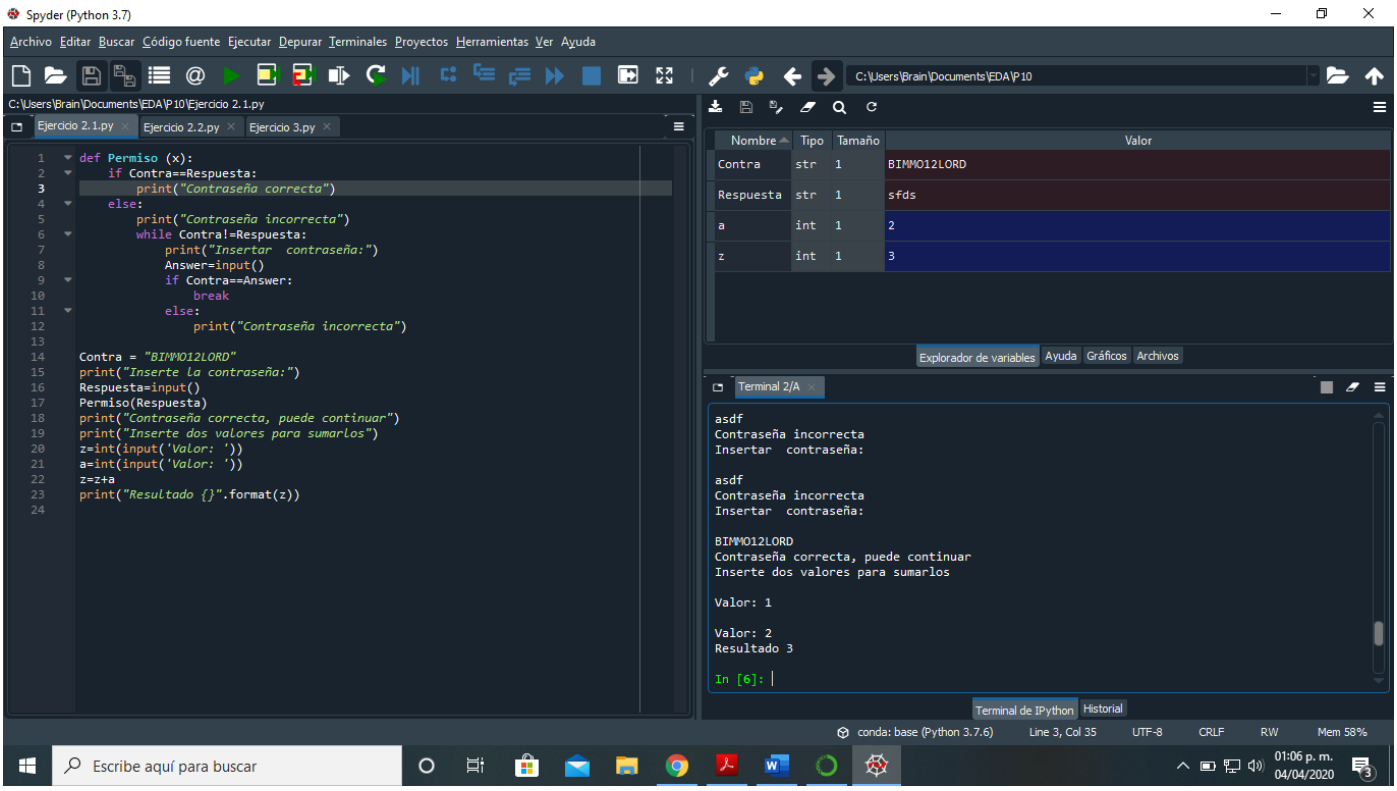
Lo primero fue definir una condición con la cual poder descartar si es igual o distinta la contraseña, por medio de un if-else, en el if iba si la contraseña era igual y daba permiso de continuar, el else contenía un ciclo while con el cual reiterar la comprobación de la contraseña, hasta que esta la entrada del usuario fuese la misma que la contraseña.

- Diagrama de funcionamiento



- Relación con la teoría

Se utilizan las instrucciones de condición if y else para poder hacer una comprobación de los elementos, y poder dar una respuesta al usuario, esta condición se asocia con un ciclo while, para hacer una reiteración. Igual se utilizan las funciones para elaborar un programa más ordenado. Se utilizan los conceptos indicados en el archivo de la practica al igual que el contenido teórico de la práctica anterior.



- Ejercicio 2

El segundo ejercicio utiliza de base el primer ejercicio, lo que requiere el segundo es que la modificación del primero permita al usuario equivocarse un numero determinado de veces, y al termino de estas oportunidades el programa termina.

- Dificultades en el código

Fue un error, debido a que utilice una instrucción que es única para los diccionarios, y al intentar realizar el ciclo, marcaba un error entonces cambie a la función range() para poder realizar el bucle de forma correcta (Imagen 1.)

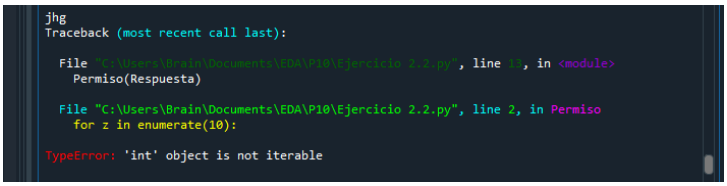
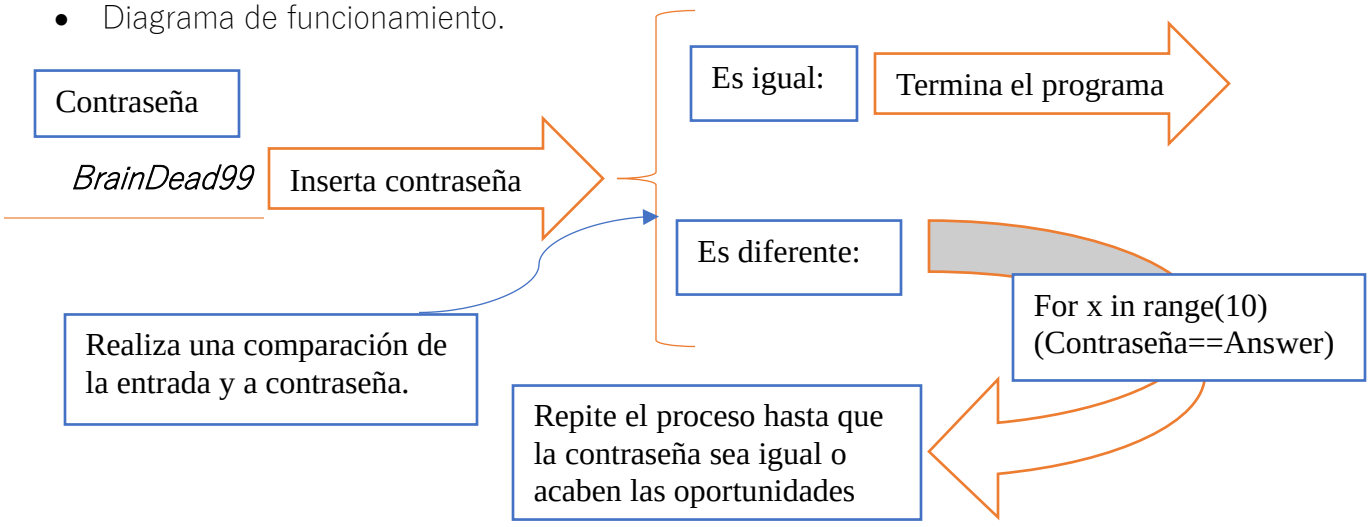


Imagen 1

- ¿Cómo lo resolví?

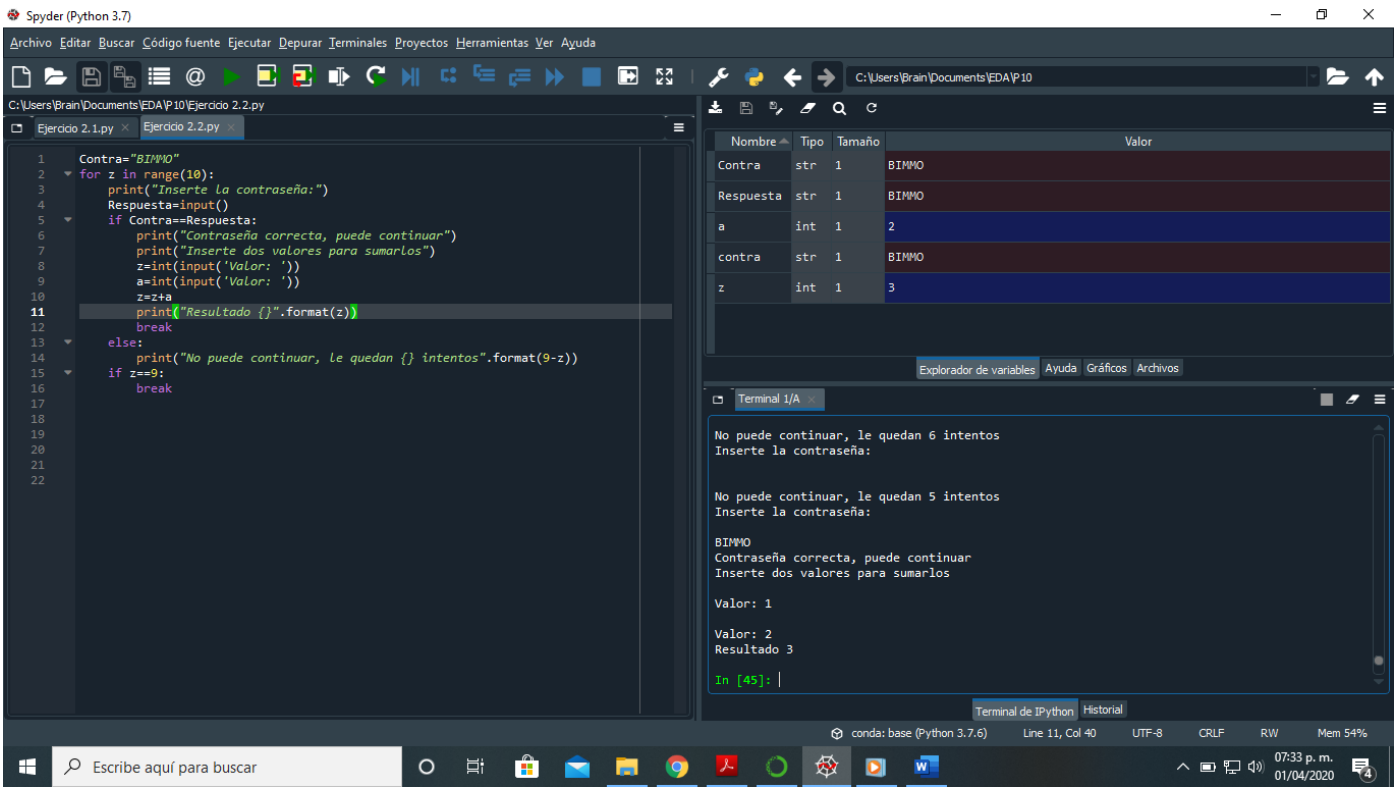
Lo principal fue realizar un ciclo for con condicionales if-else. Dentro del ciclo definí el número de intentos que el usuario puede realizar, con el if que condicionaba que la contraseña y la entrada del usuario fueran iguales, de lo contrario se repite el ciclo hasta que las oportunidades se acaben o la entrada del usuario sea igual a la contraseña.

- Diagrama de funcionamiento.



- Relación con la teoría

A diferencia del primer ejercicio, se utiliza un bucle con determinada duración que es el for, dentro de este se utilizan las condicionales if-else de nuevo para comprobar una condición que debe cumplirse para terminar el programa, o en caso contrario el usuario tendrá una oportunidad menos de poder continuar con las siguientes líneas del código.



- Ejercicio 3

El último ejercicio consistía en la creación de un programa que pudiera crear un promedio de tareas, exámenes, y prácticas, para después bajo ciertos criterios modificar la calificación final del alumno, y por ultimo mostrar las calificaciones finales. Las condiciones eran que únicamente el usuario realizaba introducción de las calificaciones obtenidas y se realizaba un ajuste en la calificación final en base al promedio de las tareas, el programa realiza todas las conversiones necesarias para poder dar una evaluación final.

- Dificultades en el código

Lo que se me complico fue el insertar valores en las listas, debido a que no coincidían el tipo de dato, y que no se podía concatenar enteros con listas, a parte de eso, la mayoría de las instrucciones fueron un poco intuitivas (Imagen 1.)

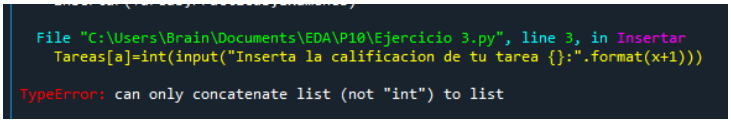


Imagen 1

Otra instrucción que se me complico realizar fue la suma de elementos, debido a que la instrucción normal de suma funcionaba en el resto del código, pero en esa línea la suma estaba errónea, lo que me hizo guardar el contenido de la lista en otra variable (Imagen 2.)

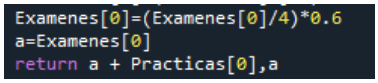


Imagen 2

- ¿Cómo lo resolví?

Lo primero que hice en base a la recomendación, fue crear las listas de Tareas, Practicas, y de Exámenes, posteriormente cree funciones para realizar el llenado y la modificación de las calificaciones del alumno. La primera función consistía en llenar las listas por medio de ciclos for separados para que hubiera un orden cuando se pidieran las calificaciones. La segunda función servía para crear los promedios de cada lista, para no utilizar mas variables guarde los promedios de cada lista en su posición cero respectiva, y en las listas Practicas y Exámenes coloque en la posición cero el valor porcentual para la calificación total, regresando el valor del primer promedio. La tercera función modificaba los valores del promedio en base al promedio de las tareas y regresaba el promedio final.

Para resolver el primer problema decidí utilizar un argumento de la misma función como variable del ciclo for, con esta modificación el programa no detectaba un error, debido a que reconocía como elemento una lista a la variable, y permitía almacenar los valores que el usuario insertaba (Imagen 3.)

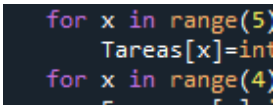


Imagen 3

El segundo problema como mencionaba lo resolví añadiendo otra variable donde se guardara el valor de la lista, y realizar la suma para el promedio, y al final de la función regresar el valor de la variable (Imagen 4.)

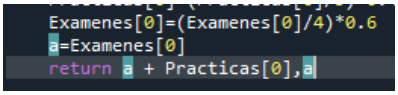
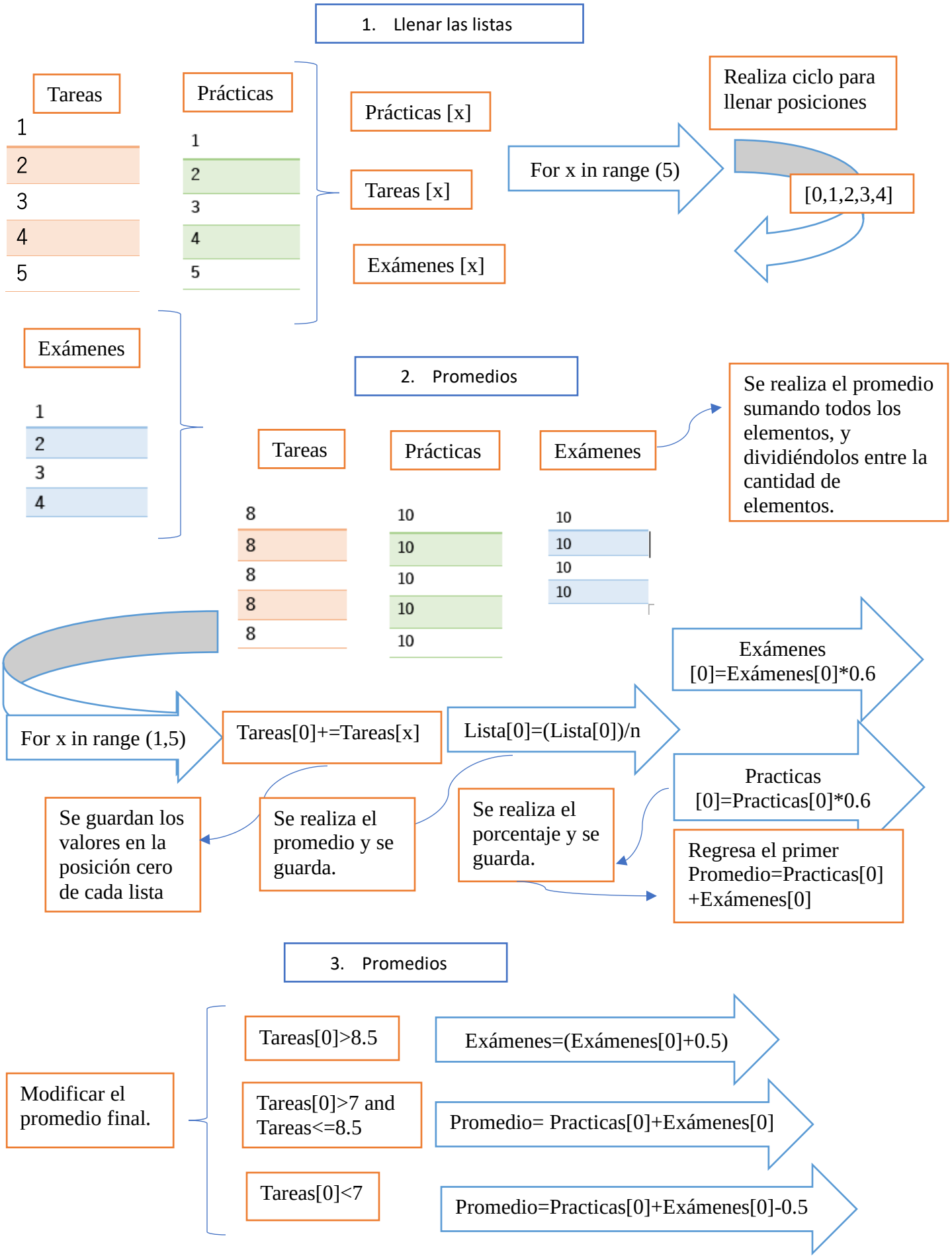
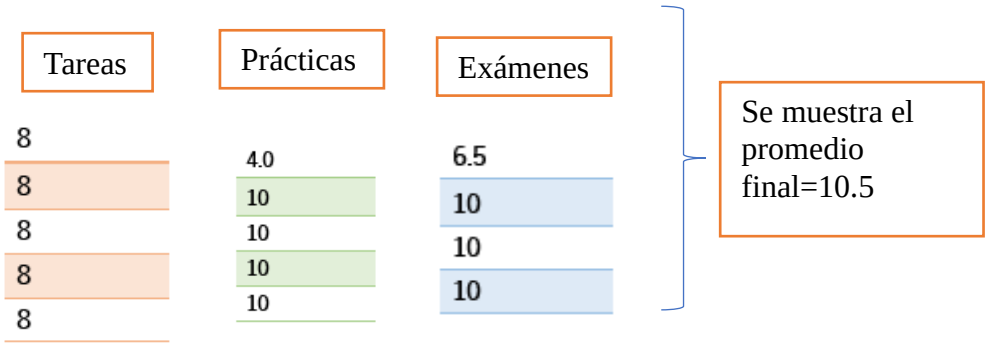


Imagen 4

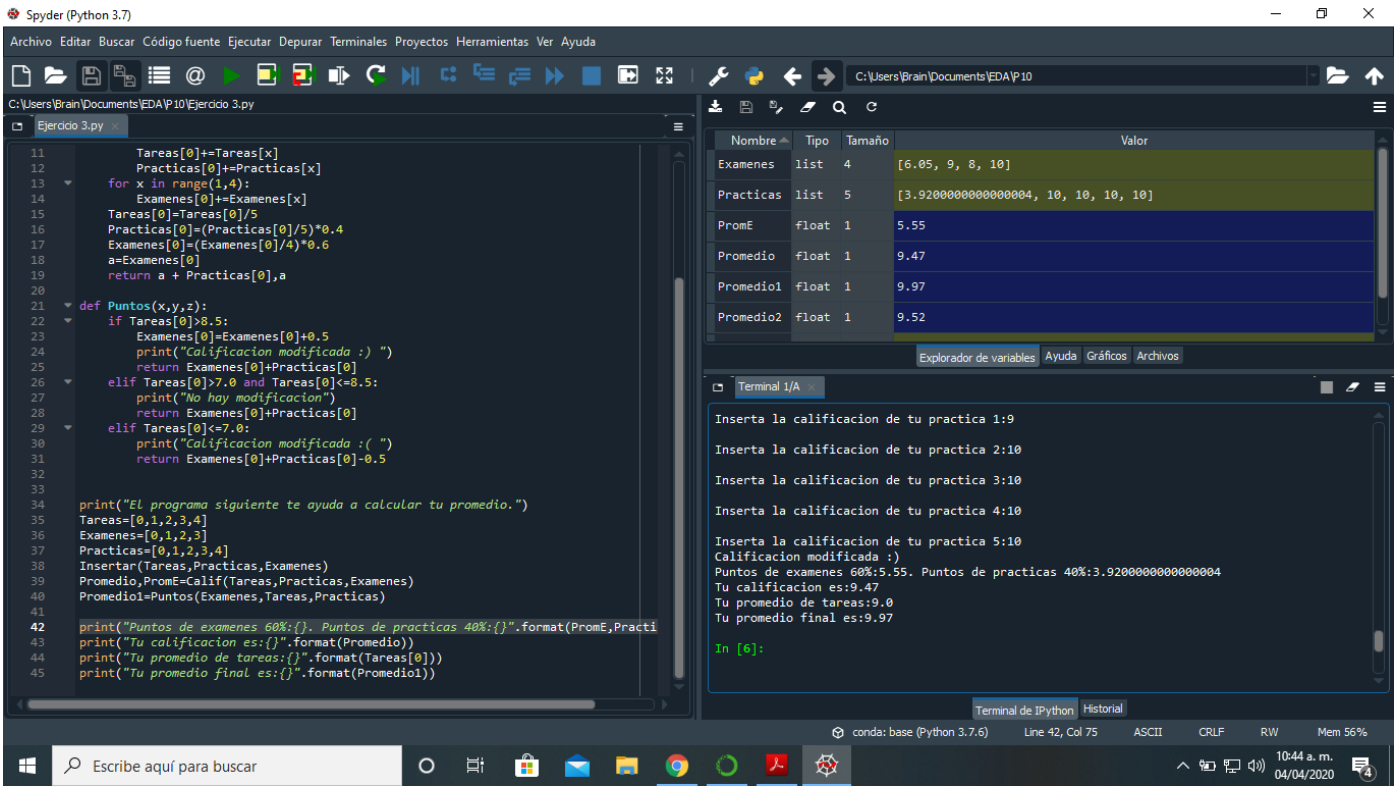
- Diagrama de funcionamiento





- Relación con la teoría

Relaciona la parte teórica correspondiente entre los bucles, instrucciones de selección, operadores, listas, funciones, el ejercicio por completo es una implementación de las bases teóricas de la práctica y del lenguaje de programación.



Conclusiones

La práctica en general requirió de atención a las partes teóricas, ya que la complejidad de los problemas radicaba en como hacerlos en nuevo lenguaje de programación, por las partes que son exclusivas de Python, logrando que aprenda las partes o comando fundamentales para poder realizar cualquier programa, al igual que la elaboración de los ejercicios me ha servido para darme cuenta de como realizar un programa eficientemente y contemplando las partes teóricas que conlleva el realizar un ejercicio. Por último, las IDE's son de mucha ayuda para poder comenzar a programar ya que cada una ofrece un apoyo distinto que sirve al comienzo de la programación, y me he llevado un conocimiento considerable.