|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Marco Antonio Martínez Quintana |
| *Asignatura:* | Estructuras de Datos y Algoritmos I |
| *Grupo:* | 17 |
| *No de Práctica(s):* | Practica No.12 |
| *Integrante:* | Arciga Guzmán Fernando |
| *No. de Equipo de cómputo empleado:* | X |
| *No. de Lista:* | 4 |
| *Semestre:* | 2º |
| *Fecha de entrega:* | 2 de mayo de 2020 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

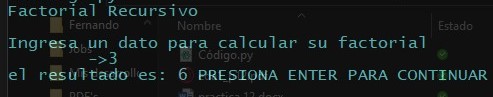
Objetivo:

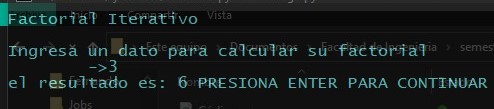
Conocer para aplicar la recursividad en la solución de problemas

Desarrollo:

La recursión en pocas palabras es una función que se llama a si misma hasta encontrar la solución de la llamada principal.

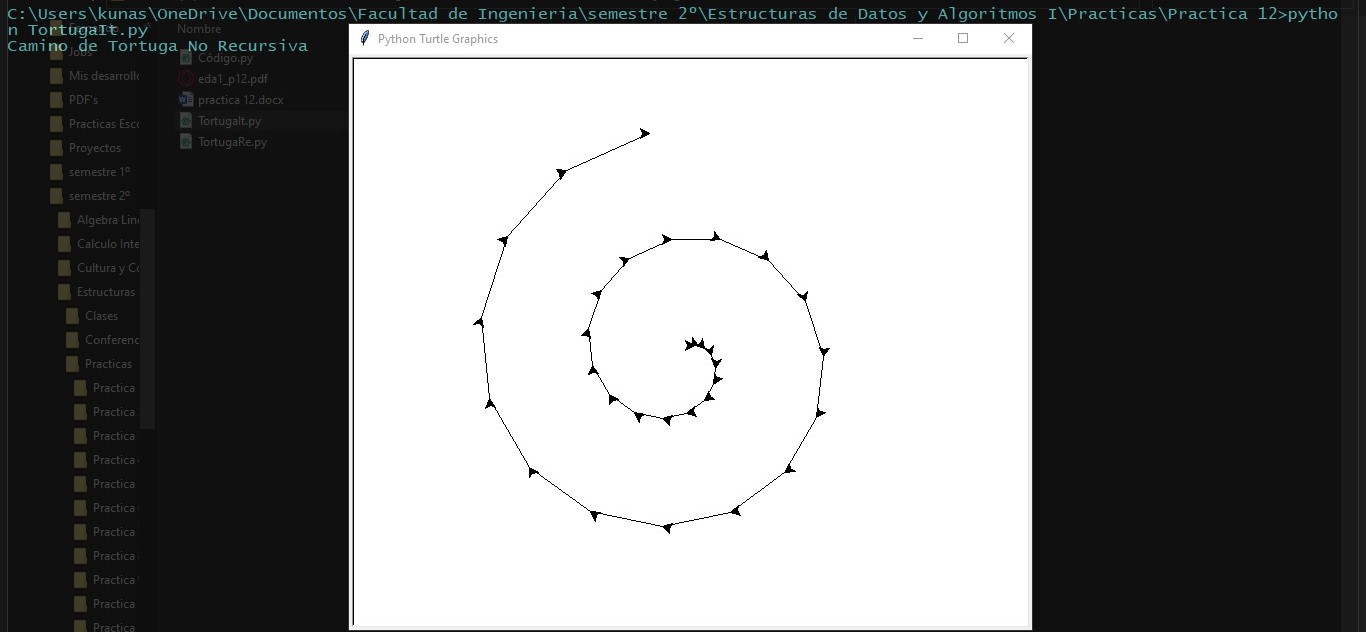
La primera parte del código de la practica es la recursión factorial, mientras el valor no sea 1 se puede restar de uno en uno para multiplicar con el anterior.

primero se itera y luego

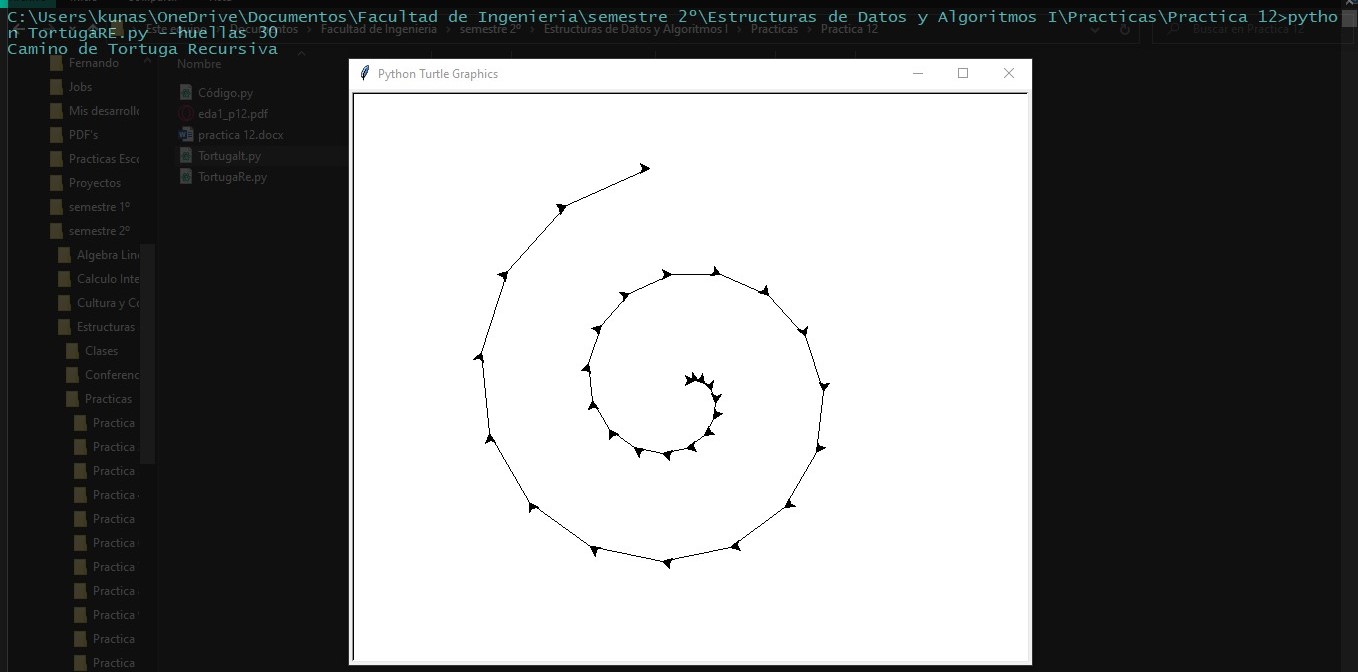
es recursivo para calcular 

En la segunda se utiliza la librería turtle de Python, conocida por su capacidad de crear gráficos, y en esta se crean dos variantes:

* Iterativa. Se crea un ciclo for para obtener el recorrido de la tortuga en la ventana.

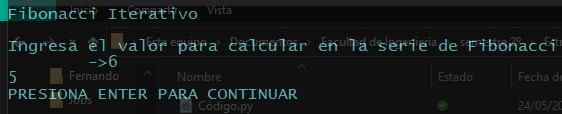


* Recursiva. Se utiliza una función que manda el movimiento de la tortuga siempre y cuando este no sea 0.

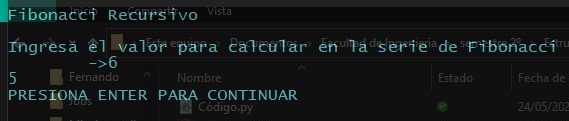


La tercera genera 3 versiones de la serie de Fibonacci:

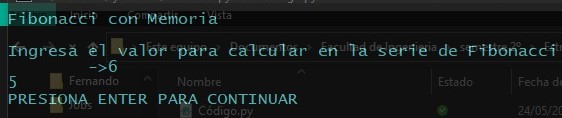
* En la primera se itera para llegar al resultado.



* En la segunda se utiliza la recursión para generar el dato buscado.



* En la tercera se utiliza memoria inicial y dinámica para guardar los datos.



Conclusiones:

La recursividad es fácil de encontrar en la vida diaria. Un ejemplo de recursividad es en la película Inception de 2010, un sueño (principal) no puede terminar hasta que se terminen los demás (secundarios), pero dentro del sueño principal se puede “llamar” nuevamente la función “soñar” y generar otro sueño.

Sin duda es algo recomendable para evitar que el código se detenga, al ser recursivo continua en ejecución todo el script y así se desarrolla de una manera mas orgánica, sin detenerse.

Referencias Bibliográficas:

Design and analysis of algorithms; Prabhakar Gupta y Manish Varshney; PHI Learning, 2012, segunda edición.

Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest y Clifford Stein; The MIT Press; 2009, tercera edicion.

Problem Solving with Algorithms and Data Structures using Python; Bradley N. Miller y David L. Ranum, Franklin, Beedle & Associates; 2011, segunda edition.

http://openbookproject.net/thinkcs/python/english3e/hello\_little\_turtles.html