Trabajo Practico N°1

Carrera: Licenciatura en sistema de infromacion.

Catedra: Ingenieria de Software 2.

Docentes:Colla, Pedro.

Sanchez, Hernan.

Alumna: Landini, Magalí.

Año 2023

# Guía de Trabajo Práctico #1 Gestión de la configuración

* Instale los siguientes paquetes de software en la versión apropiada para el sistema operativo que utilice.
  + Git.
  + Python 3 (instalar desde python.org)
  + Pip3 (instalar desde python.org)

Se instalo desdepython.org python 3.11.2

En esta versión para viene incluido pip3, por lo que desde la consola se instalo el mismo, a traves del comando python install pip3

* Obtenga una cuenta en [www.github.com](http://www.github.com/) y a la que llamará UADER\_IS2\_{su\_apellido}, a continuación genere una estructura de carpetas formada por:
  + src
  + doc
  + bin
  + Script

Se creo un repositorio denominado [UADER\_IS2\_LandiniMagal-](https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-), debido a que se usa la cuenta personal de git.

Comparto el link al repositorio

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-.git>

* Obtenga el programa primos.py (en *Source Python.gz*) y siga las siguientes consignas:
  + Colóquelo en el directorio src local en su máquina.
  + Ejecútelo con “*python3 primos.py*” y verifique que corre bien.
  + Sincronícelo con el repositorio github.
    - *git add .*
    - *git commit -n carga\_inicial*
    - *git push origin*
    - verifique la correcta actualización.
  + Simule el borrado “accidental” en su máquina y a continuación recupere el archivo desde el repositorio Github.
  + Coloque comentarios al programa, al finalizar pruebe que el mismo siga ejecutando correctamente. Al hacerlo sincronice con el repositorio GitHub.

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/blob/main/src/primos.py>

* Actualice el propósito del repositorio en GitHub por medio de un archivo *README.md* que coloque en la raíz del repositorio. Actualice y verifique. Utilice la notación simple para que resalten los títulos, secciones, referencias. Incorpore al menos tres niveles de títulos, dos listas ordenadas, una lista numerada, una figura y una referencia a una página Web fuera del repositorio.

En el siguiente link se encuentra el archivo README.md

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-.git>

# Programación Python

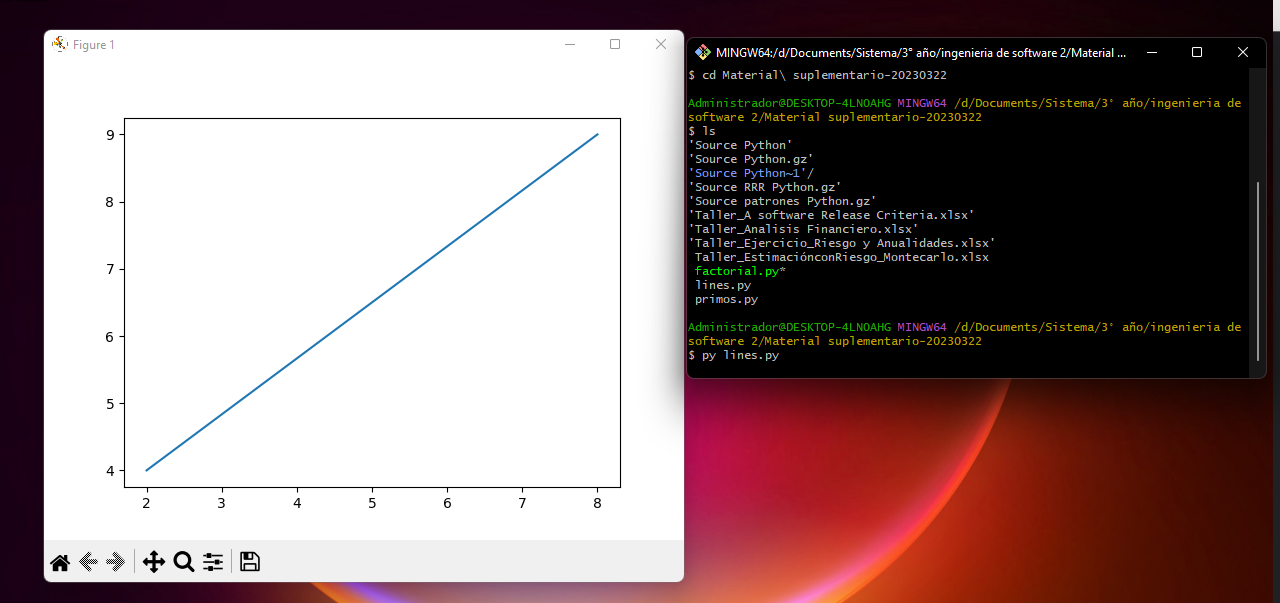
* Utilice el comando *pip* para instalar el paquete *Matplotlib* e intente ejecute el archivo

*code/charts/line.py*

Se instalo Matplotlib, ejecutando el comando pip3 install Matplotlib

Luego se ejecuto el archivo line.py obtenido desde el material complentario.

Se muestra una iamgen de lo obtenido



* Obtenga el programa fuente factorial.py y ejecute con *python3 factorial 10* confirme que funciona correctamente. Guarde en repositorio GitHub en una carpeta específica dentro del árbol “src” denominada “factorial”.

Se ejecuto el archivo de factorial y se subió a git.

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/commit/f4d1183cba27a5cdf0462a2582a1408f76a6cb52>

* + Realice una modificación al programa para que si se omite el número como argumento lo solicite. Pruebe. Sincronice en GitHub.

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/commit/72fa251e7bd46fed734eb666fe304dda73958169>

* + Modifique el argumento (y el ingreso manual) para aceptar números en el rango desde-hasta (ej. 4-8) y que calcule los factoriales entre ambos extremos. Pruebe. Sincronice en GitHub.

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/commit/40d9609b2c0937f8d69aa68db294046edbe7ee4e>

* + Modifique el argumento (y el ingreso manual) para que acepte rangos sin límite inferior “-hasta” calculando entre 1 y el número indicado (ejemplo “-10”), lo mismo para “desde-“ calculando entre el número indicado y 60. Tenga la precaución de transformar las cadenas de caracteres de la especificación de argumentos en valores enteros antes de intentar operaciones matemáticas. Pruebe. Sincronice en GitHub.

Link de la versión Factorial entre 1 y un número indicado

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/commit/a239dba59aea26112cb72ab9db362241f1f56378>

Link de la versión Factorial entre un número indicado y 60

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/commit/d5cbab8b156376f9fe94ded15308be42c7a74cf8>

* + Agregue comentarios al código generado. Pruebe. Sincronice con GitHub.

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/blob/main/src/Factorial/factorial.py>

En cada versión subida se fue generando comentarios en el código.

* Genere un proyecto copia del anterior denominado “factorial\_OOP” donde tomando como base el programa “factorial.py” genere un programa “factorial\_OOP.py” donde se construya la lógica de cálculo de factorial mediante una clase Factorial con un constructor y un método “run(min,max)” que calcule como resultado el factorial entre los números min y max. Pruebe. Sincronice en GitHub.

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/blob/main/src/Factorial/factorial_OOP.py>

* Desarrolle un programa en python para calcular el número de Collatz (conjetura 2n+1) para los números entre 1 y 10000, realice un gráfico donde en el eje de órdenadas muestre el número n de comienzo de la secuencia y en la absisas el número de iteraciones que tardó en converger a una secuencia repetitiva. Coloque en una carpeta en la jerarquía “src”. Pruebe. Sincronice en GitHub.

<https://github.com/MagaliLandini/UADER_IS2_LandiniMagal-/blob/main/src/Collatz/Collatz.py>