

25 de Mayo 385-Concepcion del Uruguay, CP: 3260. Dpto. Uruguay-
fcyt_cdelu@uader.edu.ar mailto:zubiaur79@yahoo.com.ar Tel-
03442-431442



TRABAJO PRACTICO

INGENIERÍA EN SOFTWARE

UADER – Facultad de Ciencia y Tecnología

Profesor: Pedro Colla

Carrera: Licenciatura en Sistemas de Información

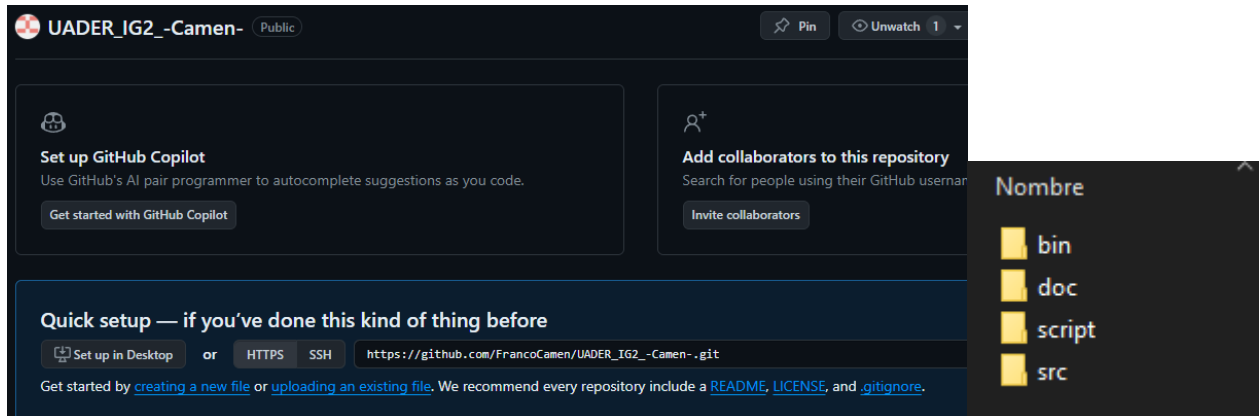
Alumno: Franco Leonardo Camen.

Fecha de entrega: 28/03/2024

Ciclo lectivo: 2024.

PRIMERO CONSIGNA

Obtenga una cuenta en www.github.com y a la que llamará UADER_IS2_{su_apellido}, a continuación, genere una estructura de carpetas formada por: bin, doc, script, src.



SEGUNDA CONSIGNA

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.

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python primos.py
Prime numbers between 1 and 100 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

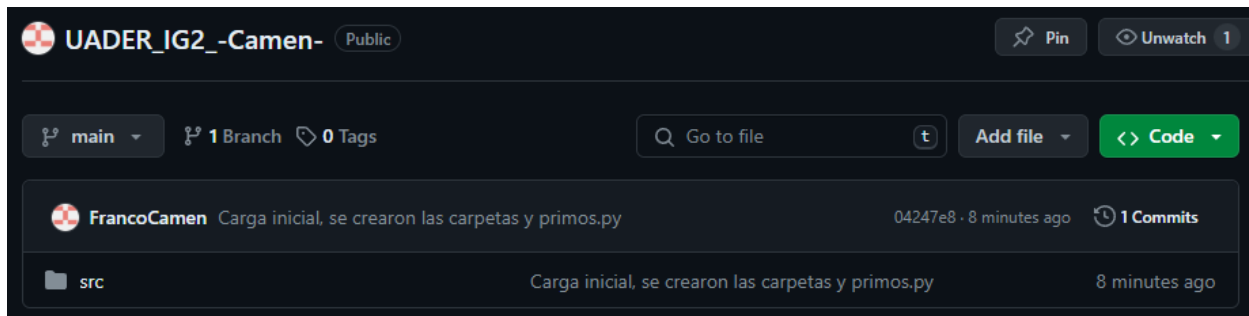
- Sincronícelo con el repositorio github.

- git add .
- git commit -m carga_inicial
- git push origin
- verifique la correcta actualización.

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .
warning: in the working copy of 'src/primos.py', LF will be replaced by CRLF the next time Git touches it

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Carga inicial, se crearon las carpetas y primos.py"
[main (root-commit) 04247e8] Carga inicial, se crearon las carpetas y primos.py
1 file changed, 16 insertions(+)
create mode 100644 src/primos.py
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git push origin
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 496 bytes | 496.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
* [new branch]      main -> main
```



- Simule el borrado “accidental” en su máquina y a continuación recupere el archivo desde el repositorio Github.

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ ls -a
./ ../ primos.py*

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git rm primos.py
rm 'src/primos.py'

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    primos.py

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Borrado del archivo primos.py"
[main d56bacf] Borrado del archivo primos.py
1 file changed, 16 deletions(-)
delete mode 100644 src/primos.py
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git log --oneline
d56bacf (HEAD -> main) Borrado del archivo primos.py
04247e8 Carga inicial, se crearon las carpetas y primos.py

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git restore --source=04247e8 --staged primos.py
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   primos.py

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    primos.py

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git restore primos.py
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ ls -a
./ ../ primos.py*
```

- Coloque comentarios al programa, al finalizar pruebe que el mismo siga ejecutando correctamente. Al hacerlo sincronice con el repositorio GitHub.

```
primos.py > ...
#!/usr/bin/python3
# Python program to display all the prime numbers within an interval

#Desde que numero cuenta los numeros primos
lower = 1
#Hasta que numero cuenta los numeros primos
upper = 100

print("Los numeros primos entre", lower, "y", upper, "son los sig.:")

#Recorre num desde lower hasta upper de uno en uno
for num in range(lower, upper + 1):

    #Chequea que en la primera ejecucion num sea mayor a 1
    #Todos los numeros primos son mayores que 1
    if num > 1:

        # Revisa si el numero es divisible por algun numero menor asi mismo
        for i in range(2, num):
            if (num % i) == 0: #Si es divisible no es primo
                break
            else: # Si no es divisible es primo
                print(num)

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python primos.py
Los numeros primos entre 1 y 100 son los sig.:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   primos.py

no changes added to commit (use "git add" and/or "git commit -a")

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Se agregaron comentarios en primos.py"
[main f10b7c4] Se agregaron comentarios en primos.py
1 file changed, 11 insertions(+), 4 deletions(-)

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 670 bytes | 670.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
9f9f51a..f10b7c4  main -> main
```

CONSIGNA 3

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.

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen- (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git add" to track)

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen- (main)
$ git add .

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen- (main)
$ git commit -m "Actualizacion del README.md"
[main 9e4959f] Actualizacion del README.md
1 file changed, 41 insertions(+)
create mode 100644 README.md

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen- (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.05 KiB | 1.05 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
f10b7c4..9e4959f  main -> main
```

```
# Gestion de la configuracion y programacion Python

## Propósito
El proposito de este repositorio es documentar el proceso del alumno en la gestion de versiones de multiples archivos python para arraigar conocimientos sobre el manejo del software de control de versiones git y programacion en python.

## Funcionamiento de programas .py

Podemos encontrar los siguientes programas .py dentro de la carpeta src:

1. Factorial
  - Define una funcion que calcula el factorial de un numero dado.
  - Si el numero dado es negativo imprime que su factorial no existe.
2. Primos
  - Define dos variables enteras, para definir el rango en el que buscara los numeros primos.
  - Mediante un ciclo for iterada este rango buscando los numeros primos.
  - Calcula si cada numero es divisible por uno menor que el.
3. Factorial_OOP
  - La logico del calculo factorial este implementada en un clase Factorial.
  - Implementa un metodo run(min, max) que calcula el factorial entre min y max, siendo estos numeros enteros.
4. Collatz
  - Define una función llamada iteracion_collatz que calcula el número de iteraciones necesarias para que la secuencia de Collatz converja a 1 para un número dado.
  - Utiliza una comprensión de lista para calcular el número de iteraciones para cada número en un rango de 1 a 10000.
  - Grafica los resultados utilizando plt.scatter.

### Dependencias

Lista de dependencias necesarias para ejecutar los programas python:

1. Python3
2. Matplotlib
3. Pip3

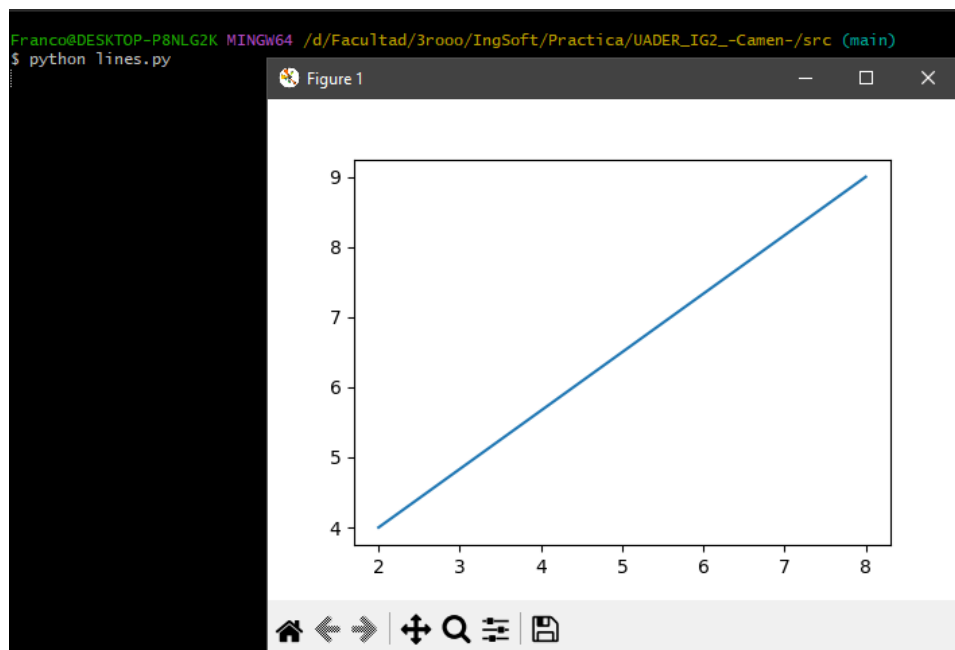
Links para descargar estas dependencias:
[python](https://www.python.org/downloads/)
[matplotlib](https://matplotlib.org/stable/users/installing/index.html)
```

 FrancoCamen	Actualizacion del README.md	9e4959f ·
 src	Se agregaron comentarios en primos.py	
 README.md	Actualizacion del README.md	

CONSIGNA 4

Utilice el comando pip para instalar el paquete Matplotlib e intente ejecute el archivo code/charts/line.py

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\franco\appdata\local\programs\python\python38\
packages (3.8.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\franco\appdata\local\programs\python\python38\
site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\franco\appdata\local\programs\python\python38\
packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\franco\appdata\local\programs\python\python38\
site-packages (from matplotlib) (4.50.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\franco\appdata\local\programs\python\python38\
site-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy<2, >=1.21 in c:\users\franco\appdata\local\programs\python\python38\
e-packages (from matplotlib) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\franco\appdata\local\programs\python\python38\
site-packages (from matplotlib) (24.0)
Requirement already satisfied: pillow>=8 in c:\users\franco\appdata\local\programs\python\python38\
packages (from matplotlib) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\franco\appdata\local\programs\python\python38\
site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\franco\appdata\local\programs\python\python38\
site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\users\franco\appdata\local\programs\python\python38\
packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```



CONSIGNA 5

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”.

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ ls -a
./ ../ factorial.py* lines.py primos.py*

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python factorial.py 10
Factorial 10 ! es 3628800
```



```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  factorial.py

nothing added to commit but untracked files present (use "git add" to track)

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .
warning: in the working copy of 'src/factorial.py', LF will be replaced by CRLF the next time Git touches it

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Se agrego factorial.py a la carpeta src y se comprobo su funcionamiento"
[main fb44a36] Se agrego factorial.py a la carpeta src y se comprobo su funcionamiento
1 file changed, 28 insertions(+)
create mode 100644 src/factorial.py
```

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

```
import sys
def factorial(num):
    if num < 0:
        print("Factorial de un numero negativo no existe")

    elif num == 0:
        return 1

    else:
        fact = 1
        while(num > 1):
            fact *= num
            num -= 1
        return fact

while True:
    if len(sys.argv) == 1:
        try:
            num = int(input("Por favor, ingrese un numero: "))
            print("Factorial", num, "! es", factorial(num))
            break
        except ValueError:
            print("¡Debe ingresar un numero entero!")
    else:
        num = int(sys.argv[1])
        print("Factorial", num, "! es", factorial(num))
        break
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python factorial.py
Por favor, ingrese un numero:
Debe ingresar un numero entero!
Por favor, ingrese un numero:
Debe ingresar un numero entero!
Por favor, ingrese un numero: 9
Factorial 9 ! es 362880
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   factorial.py

no changes added to commit (use "git add" and/or "git commit -a")

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .
warning: in the working copy of 'src/factorial.py', LF will be replaced by CRLF the next time Git
s it

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Se modifiko factorial.py"
[main 9d385b9] Se modifiko factorial.py
1 file changed, 15 insertions(+), 7 deletions(-)
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.28 KiB | 1.28 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
b71b084..9d385b9  main -> main
```

- 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.

```
import sys

def factorial(num):
    if num < 0:
        print("Factorial de un número negativo no existe")
    elif num == 0:
        return 1
    else:
        fact = 1
        while(num > 1):
            fact *= num
            num -= 1
        return fact

def calcular_factorial_entre_rango(desde, hasta):
    for num in range(desde, hasta + 1):
        print("Factorial de", num, "! es", factorial(num))

while True:
    if len(sys.argv) == 1:
        try:
            rango = input("Por favor, ingrese un rango de números en el formato desde-hasta (ej. 4-8): ")
            desde, hasta = map(int, rango.split("-"))
            calcular_factorial_entre_rango(desde, hasta)
            break
        except ValueError:
            print("¡Formato incorrecto! Debe ingresar un rango válido en el formato desde-hasta.")
    else:
        rango = sys.argv[1]
        desde, hasta = map(int, rango.split("-"))
        calcular_factorial_entre_rango(desde, hasta)
        break
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python factorial.py
Por favor, ingrese un rango de números en el formato desde-hasta (ej. 4-8): 6-9
Factorial de 6 ! es 720
Factorial de 7 ! es 5040
Factorial de 8 ! es 40320
Factorial de 9 ! es 362880
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   factorial.py

no changes added to commit (use "git add" and/or "git commit -a")

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .
warning: in the working copy of 'src/factorial.py', LF will be replaced by CRLF the next time Git touches it

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Se modifiko factorial.py para admitir un rango de numeros"
[main 2e16a2e] Se modifiko factorial.py para admitir un rango de numeros
1 file changed, 13 insertions(+), 8 deletions(-)

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 753 bytes | 753.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
   9d385b9..2e16a2e  main -> main
```

- 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.

```
import sys
def factorial(num):
    if num < 0:
        print("Factorial de un número negativo no existe")
    elif num == 0:
        return 1
    else:
        fact = 1
        while(num > 1):
            fact *= num
            num -= 1
        return fact

while True:
    if len(sys.argv) == 1:
        try:
            rango = input("Por favor, ingrese un rango (desde-hasta) para calcular los factoriales: ")
            if rango.startswith("-"):
                desde = 1
                hasta = int(rango.split("-")[1])
            elif rango.endswith("-"):
                desde = int(rango.split("-")[0])
                hasta = 15 #Se le define un limite fijo ya que no es necesario tanta iteracion
            elif "-" in rango:
                desde, hasta = map(int, rango.split("-"))
            else:
                raise ValueError
            break
        except ValueError:
            print("¡Debe ingresar un rango válido en el formato 'desde-hasta', '-hasta' o 'desde-!'")
    else:
        rango = sys.argv[1]
        if rango.startswith("-"):
            desde = 1
            hasta = int(rango.split("-")[1])
        elif rango.endswith("-"):
            desde = int(rango.split("-")[0])
            hasta = 60
        elif "-" in rango:
            desde, hasta = map(int, rango.split("-"))
        break

print("Los factoriales en el rango", desde, "-", hasta, "son:")
for num in range(desde, hasta + 1):
    print("Factorial", num, "! es", factorial(num))
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python factorial.py
Por favor, ingrese un rango (desde-hasta) para calcular los factoriales: -5
Los factoriales en el rango 1 - 5 son:
Factorial 1 ! es 1
Factorial 2 ! es 2
Factorial 3 ! es 6
Factorial 4 ! es 24
Factorial 5 ! es 120

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python factorial.py
Por favor, ingrese un rango (desde-hasta) para calcular los factoriales: 10-
Los factoriales en el rango 10 - 15 son:
Factorial 10 ! es 3628800
Factorial 11 ! es 39916800
Factorial 12 ! es 479001600
Factorial 13 ! es 6227020800
Factorial 14 ! es 87178291200
Factorial 15 ! es 1307674368000
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   factorial.py

no changes added to commit (use "git add" and/or "git commit -a")

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .
warning: in the working copy of 'src/factorial.py', LF will be replaced by CRLF the next time Git touches it

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Se modifiko factorial.py para admitir desde-hasta"
[main d4c7a0f] Se modifiko factorial.py para admitir desde-hasta
1 file changed, 25 insertions(+), 19 deletions(-)

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 951 bytes | 951.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
2e16a2e..d4c7a0f main -> main
```

CONSIGNA 6

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.

```
Franco@DESKTOP-P8NKG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python factorial_OOP.py
Ingrese un rango (desde-hasta) para calcular los factoriales: -5
Factorial de 1 es 1
Factorial de 2 es 2
Factorial de 3 es 6
Factorial de 4 es 24
Factorial de 5 es 120

Franco@DESKTOP-P8NKG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ python factorial_OOP.py
Ingrese un rango (desde-hasta) para calcular los factoriales: 10-
Factorial de 10 es 3628800
Factorial de 11 es 39916800
Factorial de 12 es 479001600
Factorial de 13 es 6227020800
Factorial de 14 es 87178291200
Factorial de 15 es 1307674368000
```

```
class Factorial:
    def __init__(self,min,max):
        self.min = min
        self.max = max

    def calcular_factorial(self,num):
        if num < 0:
            print("Factorial de un número negativo no existe")

        # Manejo de caso de 0
        elif num == 0:
            return 1

        # Manejo de caso numero enteros positivos
        else:
            fact = 1
            # Iteracion calculo factorial
            while(num > 1):
                fact *= num
                num -= 1
            return fact

    def run(self):
        # Itera por el intervalo
        for num in range(self.min,self.max+1):
            # Calcula usando calcular_factorial(num)
            factorial = self.calcular_factorial(num)
            # Si existe lo escribe
            if factorial is not None:
                print(f"Factorial de {num} es {factorial}")
```

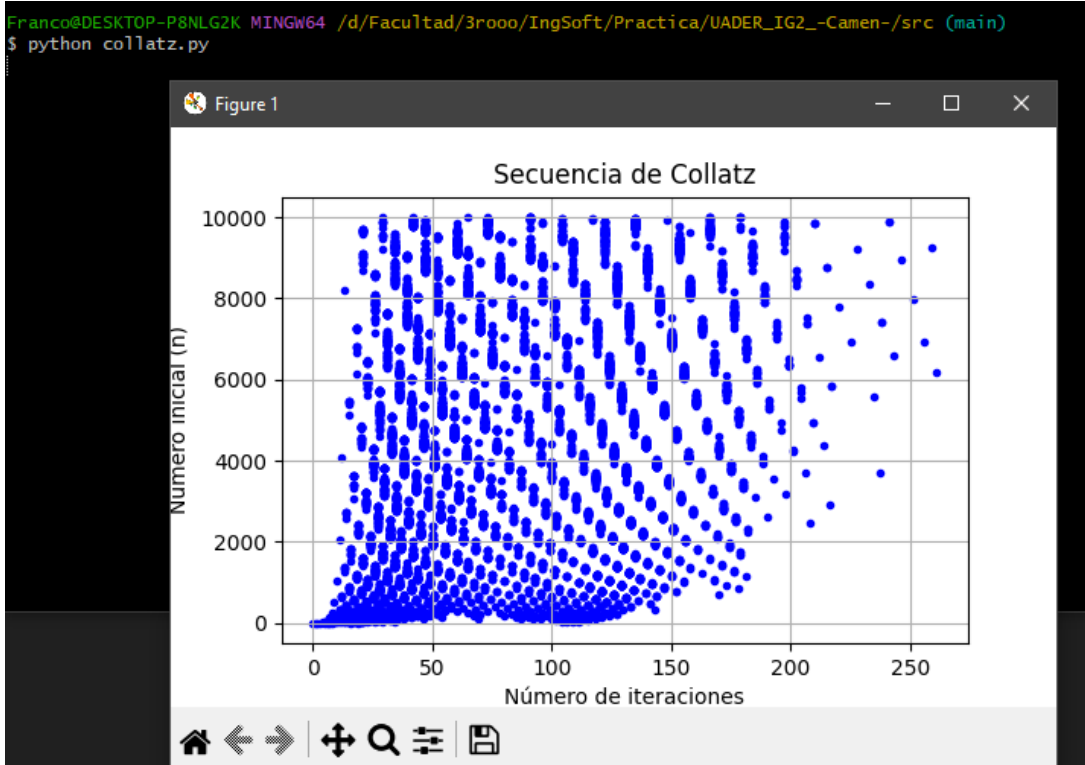
```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .
warning: in the working copy of 'src/factorial_00P.py', LF will be replaced by CRLF the next time Git touches it

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Se creo factorial_00P"
[main 77e6ea5] Se creo factorial_00P
1 file changed, 81 insertions(+)
create mode 100644 src/factorial_00P.py

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3rooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.25 KiB | 1.25 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
dc6f4d4..77e6ea5 main -> main
```

CONSIGNA 7

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 ordenadas 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.




```
import matplotlib.pyplot as plt

# Define una función llamada collatz_secuencia que toma un
# número num como entrada y devuelve la secuencia de Collatz para ese número.

# Define una función llamada iteracion_collatz que calcula el número de iteraciones
# necesarias para que la secuencia de Collatz converja a 1
def iteracion_collatz(num):
    iteraciones = 0
    while num != 1:
        if num % 2 == 0:
            num = num // 2
        else:
            num = 3 * num + 1
        iteraciones += 1
    return iteraciones

# Calcular la secuencia de Collatz y el número de iteraciones para cada número en el rango
num_valores = range(1, 10001)
iteraciones_valores = [iteracion_collatz(num) for num in num_valores]

# Graficar los resultados
plt.figure(figsize=(10, 6))
plt.scatter(iteraciones_valores, num_valores, marker=".", color="blue")
plt.title("Secuencia de Collatz")
plt.xlabel("Número de iteraciones")
plt.ylabel("Número inicial (n)")
plt.grid(True)
plt.show()
```

```
Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git add .

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git commit -m "Se agrego collatz.py"
[main 97163e1] Se agrego collatz.py
1 file changed, 31 insertions(+)
create mode 100644 src/collatz.py

Franco@DESKTOP-P8NLG2K MINGW64 /d/Facultad/3roooo/IngSoft/Practica/UADER_IG2_-Camen-/src (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 802 bytes | 802.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/FrancoCamen/UADER_IG2_-Camen-.git
77e6ea5..97163e1 main -> main
```