Universidad Nacional Autónoma de México

Facultad de Ingeniería

División de Ingeniería Mecánica e Industrial

Laboratorio de Cómputo de Ingeniería Mecatrónica (1964)

Profesor: Miguel Serrano Reyes
Semestre 2021-2

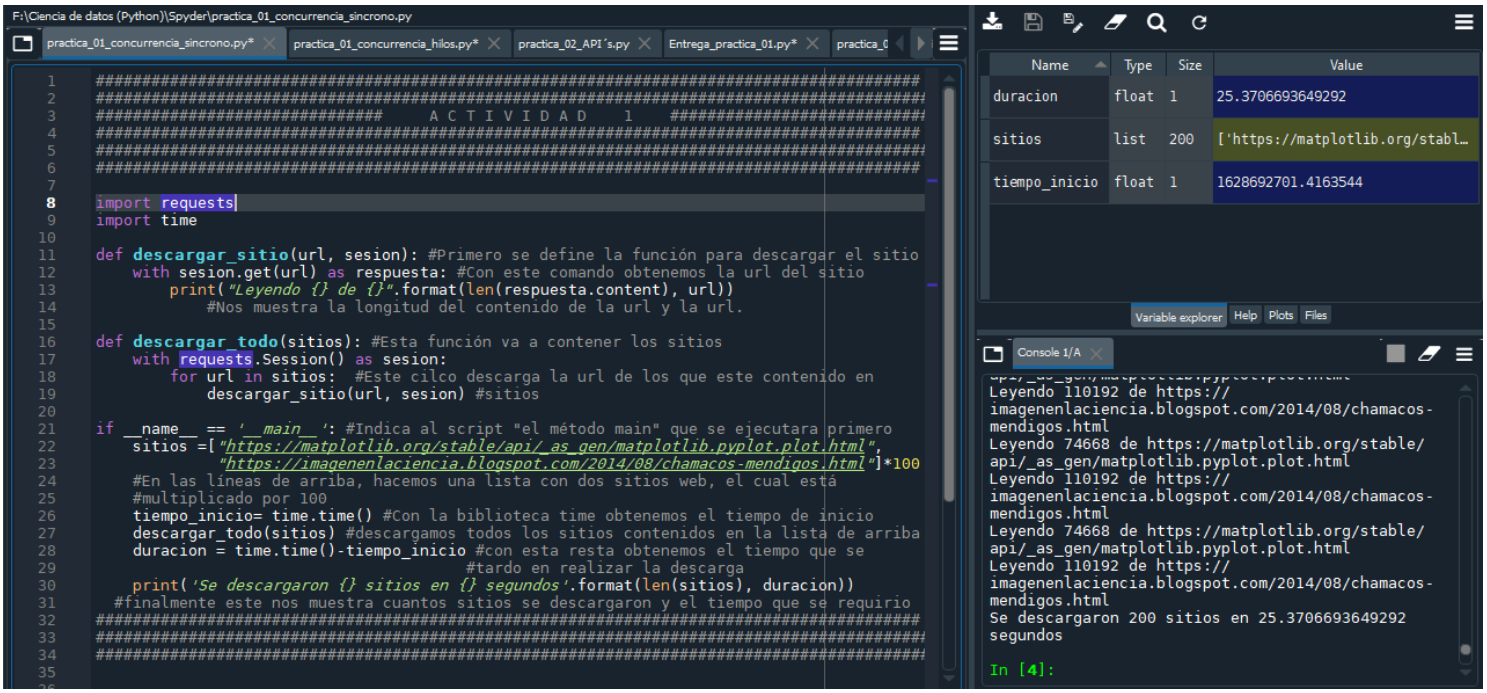
Práctica No. 1

Nombre de la práctica

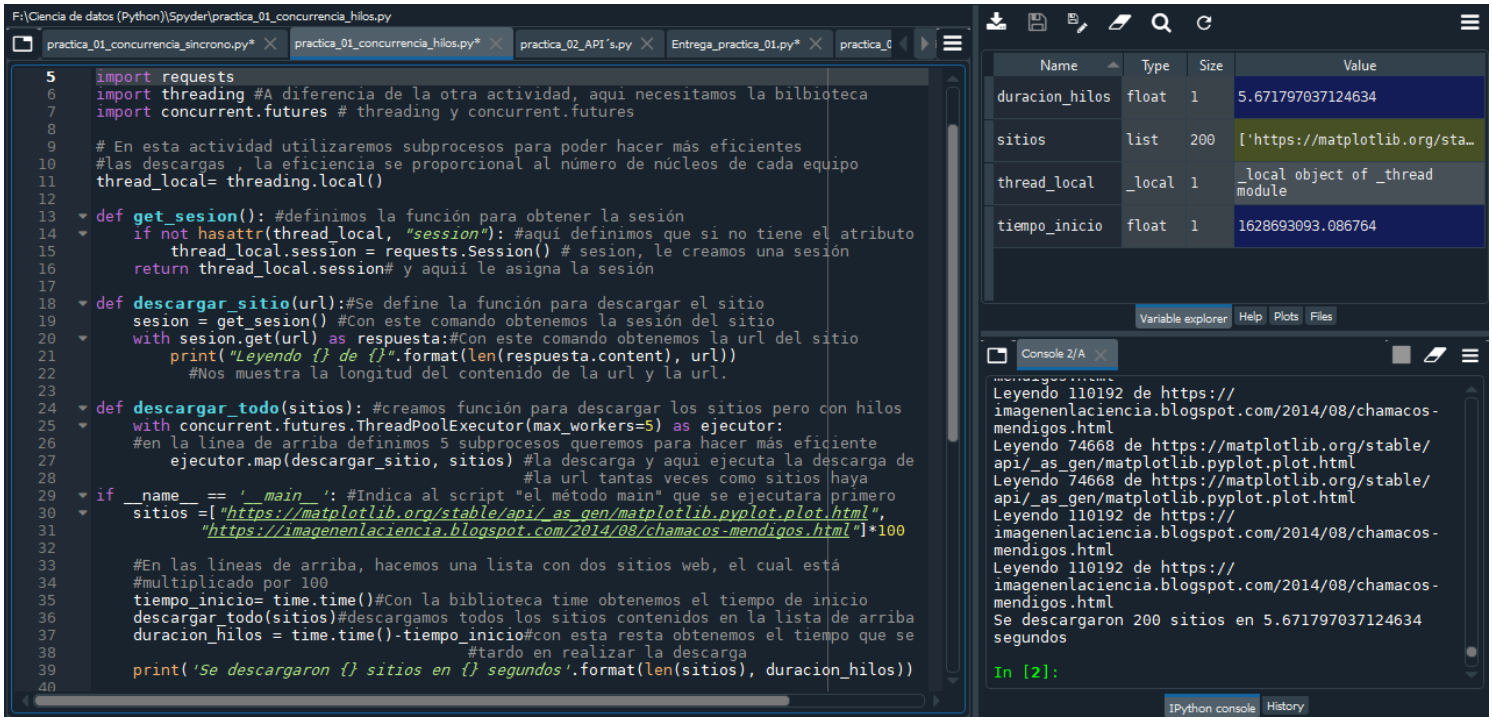
Programación Concurrente

Nombre del Estudiante:
Dimas Ramírez Luis Daniel

Actividad 1
Comentar a detalle todas las líneas del ejercicio
“practica_01_concurrencia_sincrono.py”



Actividad 2
Comentar a detalle todas las líneas del ejercicio
“practica_01_concurrencia_hilos.py”



Actividad 3

Ejecutar 10 veces el primer código y guardas los tiempos en una lista.
Ejecutar 10 veces el segundo código, guarda los tiempos en otra lista.
Comparar ambos tiempos con un boxplot.

```
        print("Leyendo {} de {}".format(len(respuesta2.content), url))

def descargar_todo2(sitios):
    with concurrent.futures.ThreadPoolExecutor(max_workers=5) as ejecutor:
        ejecutor.map(descargar_sitio2, sitios)

##############################################################################
####################            ACTIVIDAD 3          #########################
##############################################################################

duracion_sincrono = []
for i in range(10):
    tiempo_inicio1= time.time()
    descargar_todo1(sitios)
    duracion_sincrono.append(time.time()-tiempo_inicio1)

duracion_hilos = []
for m in range(10):
    tiempo_inicio2= time.time()
    descargar_todo2(sitios)
    duracion_hilos.append(time.time()-tiempo_inicio2)

print(duracion_hilos)
print(duracion_sincrono)

tiempos = [duracion_sincrono, duracion_hilos]
plt.boxplot(tiempos, patch_artist= True)
plt.xticks([1,2], ["Concurencia sincrona", "Concurrencia hilos"])
plt.title("Comparación entre concurrencias")
plt.ylabel("Tiempo de concurrencia")
plt.show()
##############################################################################
##############################################################################
##############################################################################
```

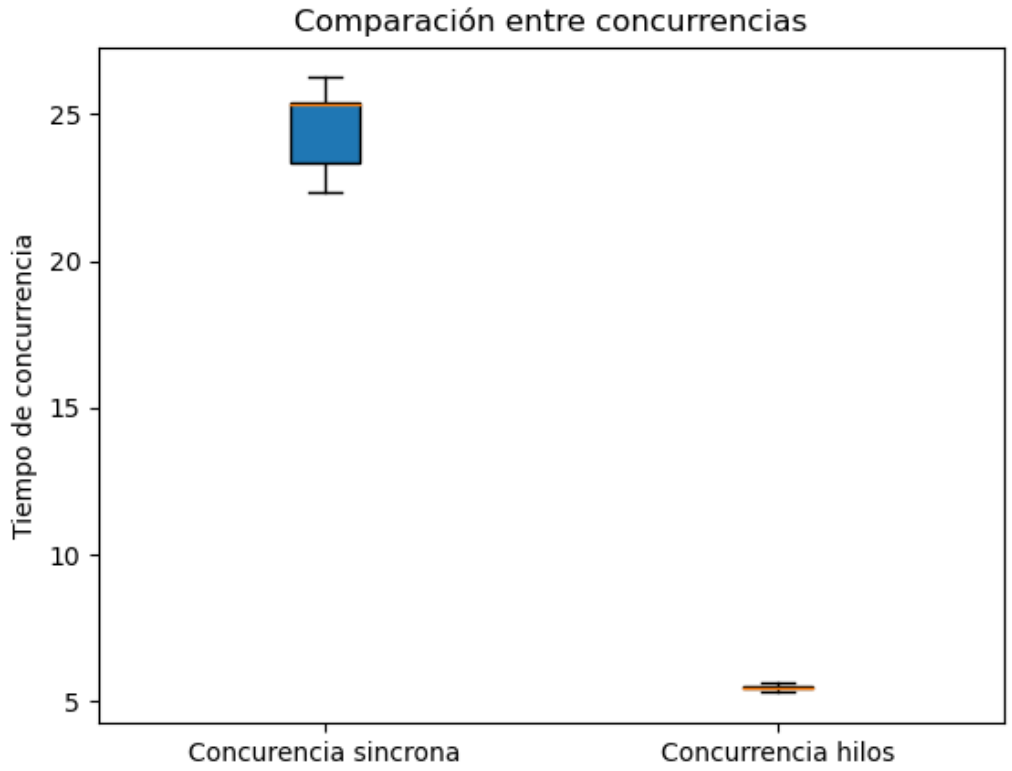
```
Built-in mutable sequence.

If no argument is given, the constructor creates a
new empty list. The argument must be an iterable
if specified.
```

```
sincrona", "Concurrencia hilos"])
    ...: plt.title("Comparación entre
concurrencias")
    ...: plt.ylabel("Tiempo de concurrencia")
    ...: plt.show()

In [23]: tiempos = [duracion_sincrono,
duracion_hilos]
    ...: plt.boxplot(tiempos, patch_artist= True)
    ...: plt.xticks([1,2], ["Concurencia
sincrona", "Concurrencia hilos"])
    ...: plt.title("Comparación entre
concurrencias")
    ...: plt.ylabel("Tiempo de concurrencia")
    ...: plt.show()

In [24]:
```



Evidentemente la concurrencia con hilos es mucho mejor. Tiene menor dispersión de datos y además el tiempo en el que se ejecuta es casi cinco veces menor. La media de tiempo para la concurrencia con hilos es de 5.47 segundos mientas que para la concurrencia síncrona es de 24.59