

Tarea_1

January 23, 2021

1 Tarea 1

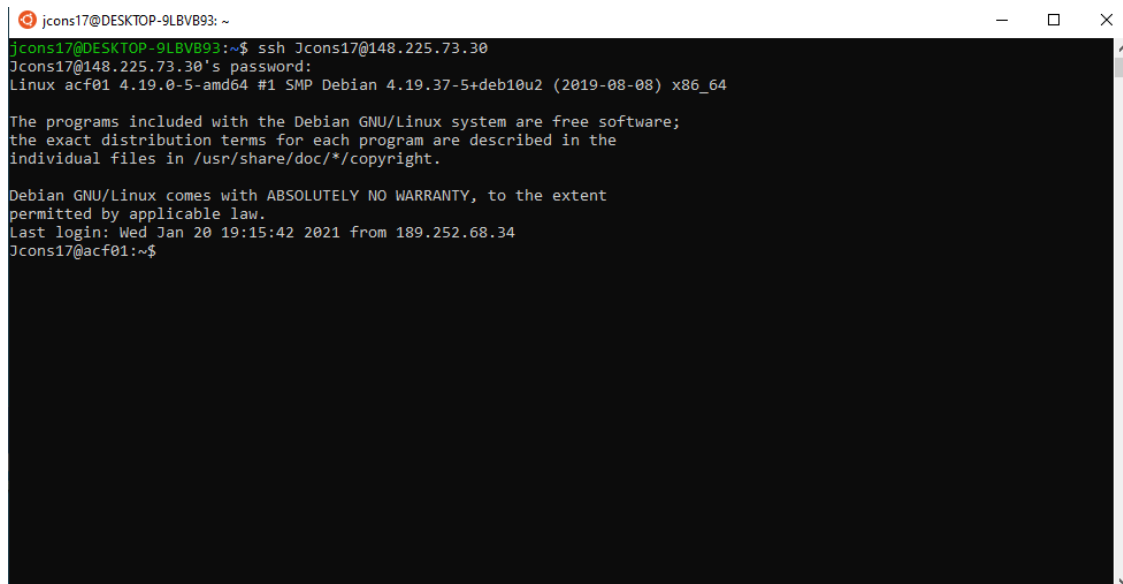
1.0.1 Desarrollo Experimental 2

1.0.2 Julio Cesar Cons Calderón

Objetivos Realizar rutinas (seguir protocolos) para comprobar el buen estado inicial del equipo de cómputo, recordar comandos básicos, revisión de edición, graficación y compilación en actividades específicas.

Actividad 0

- Conectarse al servidor del centro de computo



```
jcons17@DESKTOP-9LBVB93: ~  
jcons17@DESKTOP-9LBVB93:~$ ssh jcons17@148.225.73.30  
jcons17@148.225.73.30's password:  
Linux acf01 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Jan 20 19:15:42 2021 from 189.252.68.34  
jcons17@acf01:~$
```

- Conectarse al servidor del centro de computo
- Crear el directorio DE2
- Cambiarse al directorio DE2 y generar otro directorio Tarea1
- Regresar al directorio raiz
- Salirse de la coneccion al servidor

```
jcons17@DESKTOP-9LBVB93: /mnt/c/Users/abelc/Downloads/Desarrollo
jcons17@DESKTOP-9LBVB93:~$ ssh jcons17@148.225.73.30
jcons17@148.225.73.30's password:
Linux acf01 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan 22 18:27:15 2021 from 189.252.102.69
jcons17@acf01:~$ mkdir DE2
jcons17@acf01:~$ cd DE2
jcons17@acf01:~/DE2$ mkdir Tarea1
jcons17@acf01:~/DE2$ cd Tarea1
jcons17@acf01:~/DE2/Tarea1$ cd ../../
jcons17@acf01:~$ exit
cerrar sesión
Connection to 148.225.73.30 closed.
```

- Consultar su correo institucional y guardar en un directorio local los archivos enviados por el profesor <
- Transferir el archivo del inciso anterior al directorio Tarea 1 generado en el servidor
- Salirse del protocolo de transferencia

```
jcons17@DESKTOP-9LBVB93: /mnt/c/Users/abelc/Downloads/Desarrollo
jcons17@DESKTOP-9LBVB93:/mnt/c/Users/abelc/Downloads/Desarrollo$ scp ei.dat ea.dat jcons17@148.225.73.30:~/DE2/Tarea1
jcons17@148.225.73.30's password:
ei.dat                                100% 423    11.8KB/s   00:00
ea.dat                                100% 1657   31.3KB/s   00:00
jcons17@DESKTOP-9LBVB93:/mnt/c/Users/abelc/Downloads/Desarrollo$
```

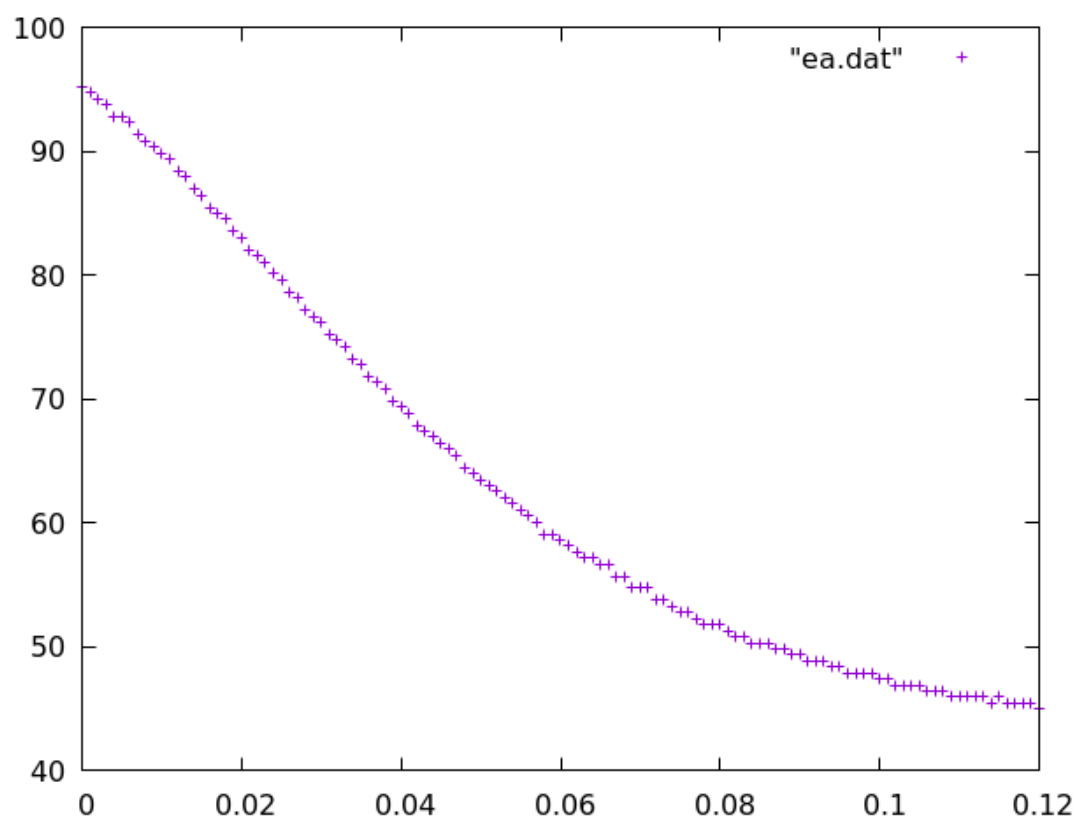
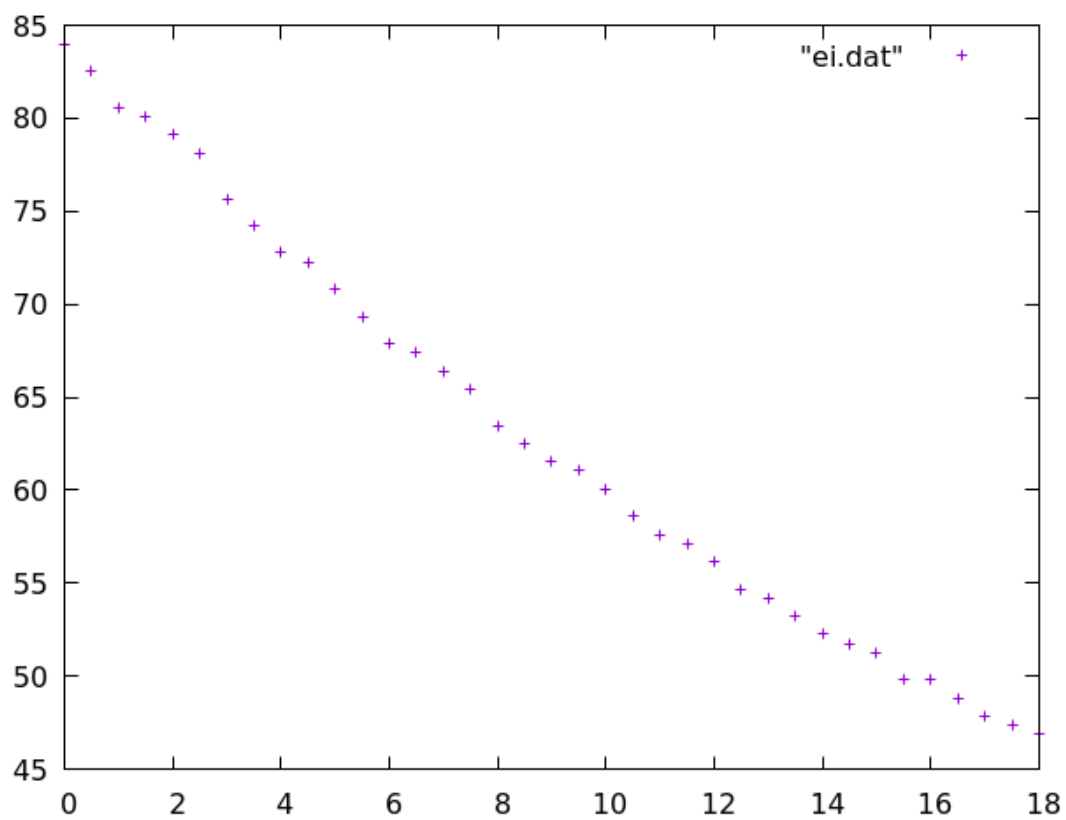
- Volver a conectarse al servidor e ir al directorio Tarea 1
- Confirmar que el directorio contiene el archivo transferido
- Visualizar el contenido del archivo

```
jcons17@DESKTOP-9LBVB93: /mnt/c/Users/abelc/Downloads/Desarrollo
jcons17@DESKTOP-9LBVB93:/mnt/c/Users/abelc/Downloads/Desarrollo$ ssh jcons17@148.225.73.30
jcons17@148.225.73.30's password:
Linux acf01 4.19.0-5-amd64 #1 SMP Debian 4.19.37-5+deb10u2 (2019-08-08) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan 22 18:34:06 2021 from 189.252.102.69
jcons17@acf01:~$ cd DE2/Tarea1/
jcons17@acf01:~/DE2/Tarea1$ ls
ea.dat ei.dat
jcons17@acf01:~/DE2/Tarea1$ cat ea.dat
0.0      95.218
1E-3     94.729
0.002    94.241
0.003    93.753
0.004    92.776
0.005    92.776
0.006    92.288
0.007    91.311
0.008    90.823
0.009    90.335
0.01     89.846
0.011    89.358
0.012    88.382
0.013    87.893
0.014    86.917
```

- Gráficar los archivos



Actividad 1 Graficaremos la funcion:

$$f(x) = e^{\frac{-x^2}{2}}$$

Pasos a seguir

1. Importamos las librerias necesarias para la actividad. Seran tres librerias.

- NumPy es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas.
- Pandas es una biblioteca de software escrita como extensión de NumPy para manipulación y análisis de datos para el lenguaje de programación Python. En particular, ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.
- Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.

```
[1]: import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import math
```

2. Definimos una función en Python. Se usara el comando def que nos define la función y como unico argumento sera x.

```
[2]: def funcion(x):
    f=np.exp(-x**2/2)
    return f
```

3. Asignamos valor de inicio y final al intervalo en el cual se va a graficar la función.

```
[3]: x0=-3
xf=3
```

4. Creamos un array x que vaya desde el intervalo $[x_0, x_f]$ y tenga un step de 0.1.

5. Despues creamos otro array que contenga los valores de la función valuado en cada valor del elemento x .

6. Metemos los arrays en un marco de datos que podamos exportar a un archivo txt.

```
[4]: x=np.arange(x0,xf,step=0.1)
y=[funcion(z) for z in x]
df=pd.DataFrame({"x":x,"f(x)":y})
df.to_csv("grafica.txt",index=None)
```

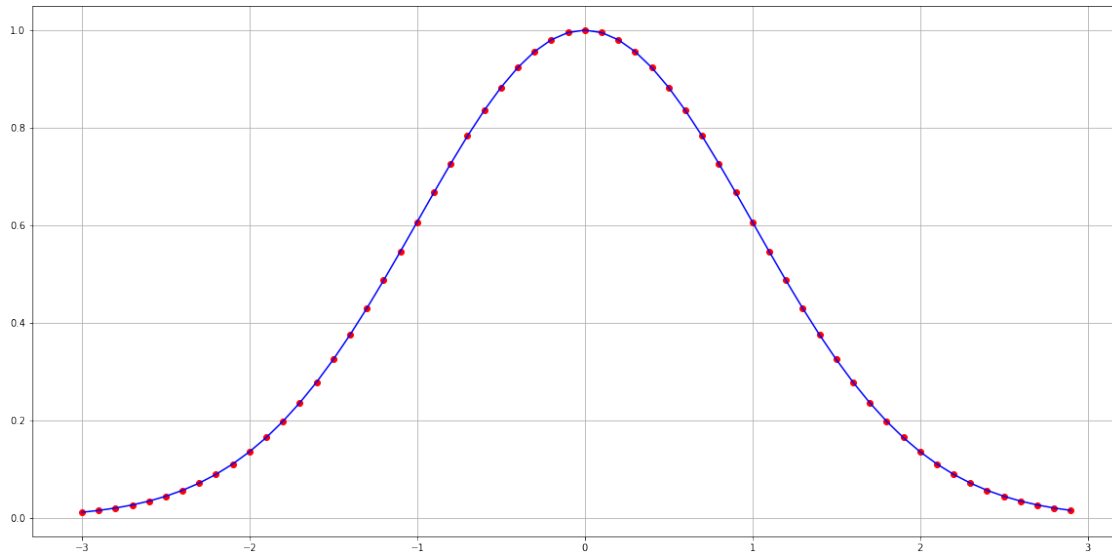
7. Leemos el archivo txt que habiamos exportado en el paso anterior y lo guardamos en un marco de datos llamado read.

```
[5]: read= pd.read_csv("grafica.txt")
```

8. Graficamos los datos guardados en el marco de datos read. Los graficamos en forma de puntos y continua.

```
[6]: plt.figure(figsize=(20,10))
plt.plot(read["x"],read["f(x)"], 'ro')
plt.plot(read["x"],read["f(x)"], 'b')

plt.grid()
plt.show()
```



Actividad 2 Elaborar un programa para colocar N partículas (puntos) en una recta de longitud L , de forma tal que la distancia de separación entre ellas sea uniforme.

- Ejecutar su programa para N y L dadas por Usted.
- Mostrar gráficamente la distribución de partículas (puntos) obtenida.

Pasos a seguir

1. Dar un valor para los N puntos y para L de longitud.
2. Definir x como un array.
3. Crear un ciclo que se repeta N veces. En ese ciclo se le asignara un nuevo valor al array x . Ese nuevo valor ira creciendo $\frac{L}{N}$ veces cada ciclo, hasta completar los N ciclos.
4. Crearemos un array y que tenga la misma cardinalidad de x , solo que en cada entrada valdra 0.

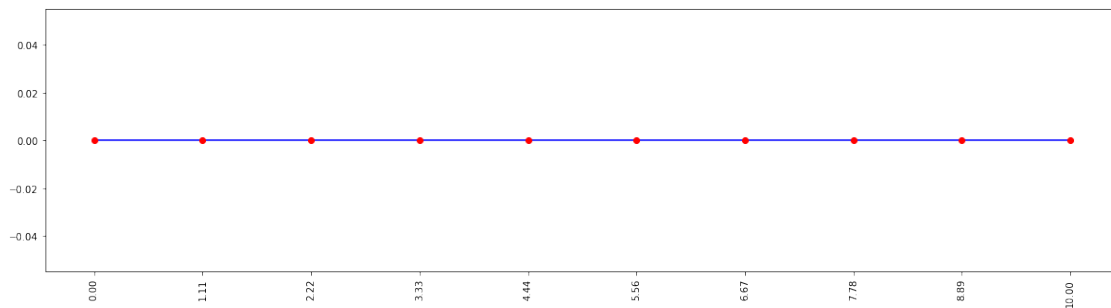
```
[7]: L=10
N=10
i=0
x=[]
while i<N:
```

```
x.append(i*L/(N-1))
i = i+1

y = [_*0 for _ in x]
```

5. Graficaremos los valores de x . Lo haremos primero de forma discreta y despues de forma continua para simular la linea recta.

```
[8]: plt.figure(figsize=(20,5))
plt.plot(x, y, 'b')
plt.plot(x, y, 'ro')
plt.xticks(x, rotation=90)
plt.show()
```



Actividad 3 Elaborar un programa para colocar partículas (puntos) en una superficie cuadrada de lado L , de forma tal que formen un arreglo cuadrangular. * Ejecutar su programa para N y L dadas por Usted. * Mostrar gráficamente la distribución de partículas obtenida

Pasos a seguir

1. Basicamente, el problema se basa en usar el mismo array pero aplicar el producto cartesiano a si mismo. Esto generará una pareja de pares que a cada valor del array x le asignara cada valor del mismo array.
2. Para generar un producto cartesiano usamos el comando set para generar un conjunto de pares. Despues creamos dos ciclos donde a la primer valor del par le asignamos cada elemento del array x . El segundo ciclo le asigna al segundo valor del par un elemento del array x .

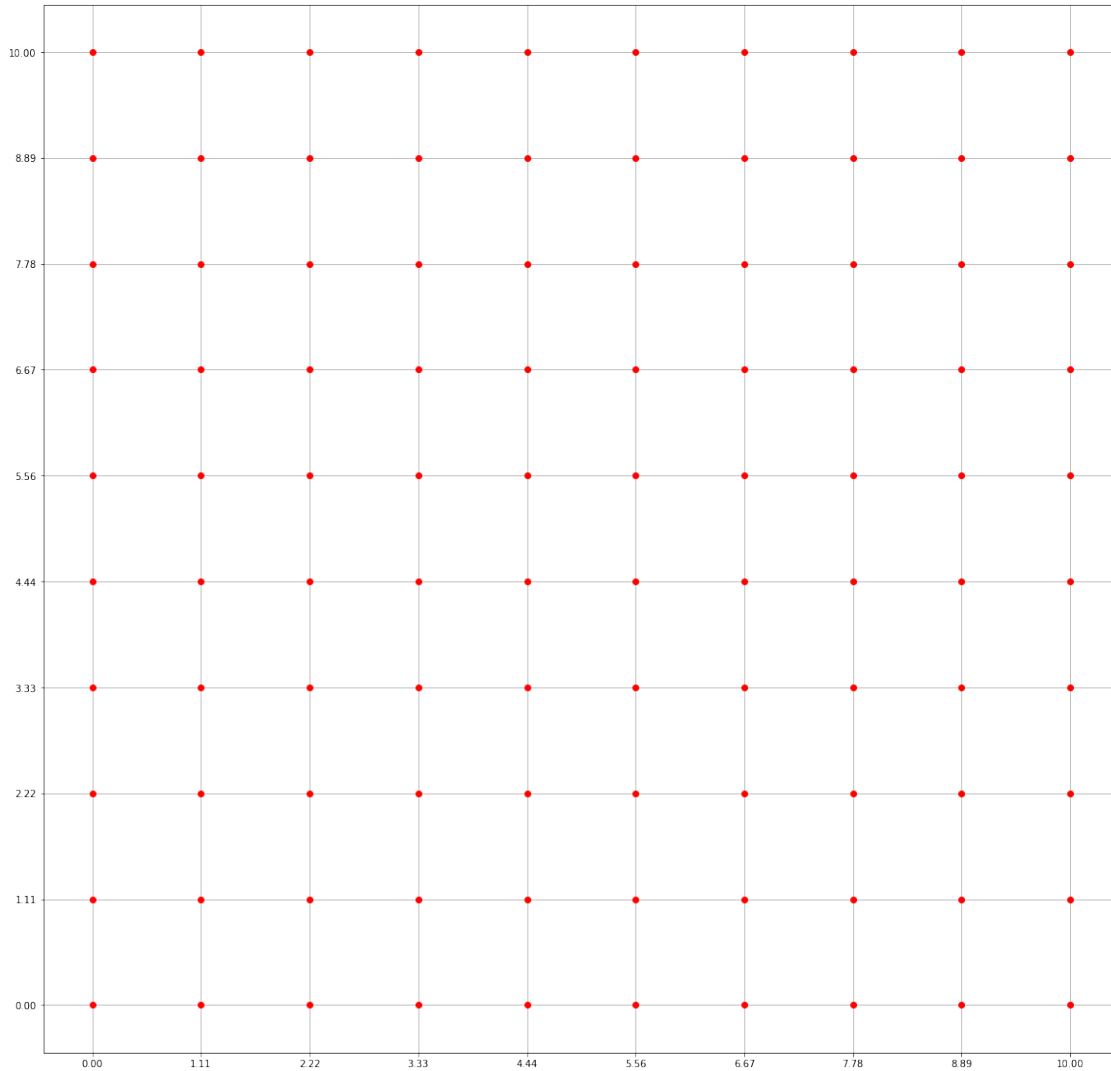
```
[9]: p=set([(z, y) for z in x for y in x])
```

3. Separamos el valor de los pares en dos arrays diferentes. Así, tendremos la facilidad de graficarlos.

```
[10]: x_val = [x[0] for x in p]
y_val = [x[1] for x in p]
```

4. Graficamos la cuadrícula.

```
[11]: plt.figure(figsize=(20,20))
plt.plot(x_val,y_val, 'ro')
plt.grid()
plt.xticks(x)
plt.yticks(x)
plt.show()
```



Observaciones finales: Esta actividad fue muy ilustrativa. Mis conocimientos usando la consola de Unix es limitado pero las pocas lineas de comando que usé me ayudo a recordar algunos elementos de Fortran. Para mi, el Lenguaje Python junto con el IDE Jupyter Notebook, forman una excelente dupla a la hora de resolver problemas que no necesiten lineas de codigo extensas.