



Estructura de Datos Avanzados

LABORATORIO 01

Realizado por:

- JEAN PIERRE CHAVEZ GUEVARA

Profesora:

- ROSA YULIANA GABRIELA PACCOTACYA YANQUE

21 de septiembre del 2023

1. El código que se usó para generar los puntos y poder calcular las distancias es el siguiente:

```
#include <iostream>
#include <random>
#include <cmath>
#include <fstream>
using namespace std;

double distancia_euclidiana(const double punto_1[], const double
punto_2[], int dim)
{
    double distancia = 0.0;
    for (int i = 0; i < dim; ++i)
    {
        double aux = punto_1[i] - punto_2[i];
        distancia += pow(aux,2);
    }
    return sqrt(distancia);
}

int main()
{
    int dimensiones[] = {2};
    for (int dim : dimensiones)
    {
        cout<<"Dimension: "<<dim<<endl;
        random_device rd;
        mt19937 gen(rd());
        uniform_real_distribution<double> dist(0.0, 1.0);

        double puntos[100][2];
        for (int i = 0; i < 100; ++i)
        {
            for (int j = 0; j < dim; ++j)
            {
                puntos[i][j] = dist(gen);
            }
        }

        ofstream archivo("distancias_2.txt");
        if (archivo.is_open()) {
            for (int i = 0; i < 100; ++i)
            {
                for (int j = i + 1; j < 100; ++j)
                {
                    double distancia = distancia_euclidiana(puntos[i],
puntos[j], dim);
                    archivo<<distancia<<" ";
                }
            }
        }
    }
}
```

```

    }
    }
    archivo.close();
} else
    cerr<<"Error al abrir el archivo para escritura."<<endl;
}
return 0;
}

```

Como adicional a lo que se pedía se agregó líneas de código para poder capturar los datos de las distancias en un archivo .txt y luego usarlas en otro código de Python para los gráficos

2. Luego para generar los gráficos de cada una de las dimensiones se usó el siguiente código en Python:

```

import matplotlib.pyplot as plt
import numpy as np

distancias_2 = np.loadtxt("distancias_2.txt")
plt.hist(distancias_2, bins=10, edgecolor='black', color='steelblue')
plt.xlabel('Distancia Euclidiana')
plt.ylabel('Frecuencia')
plt.title('Histograma de Distancias Euclidianas Dimensión 2')
plt.grid(True)
plt.show()

```

3. Ahora observamos cada una de las graficas resultantes con cada dimensión:

- a) 2 dimensiones

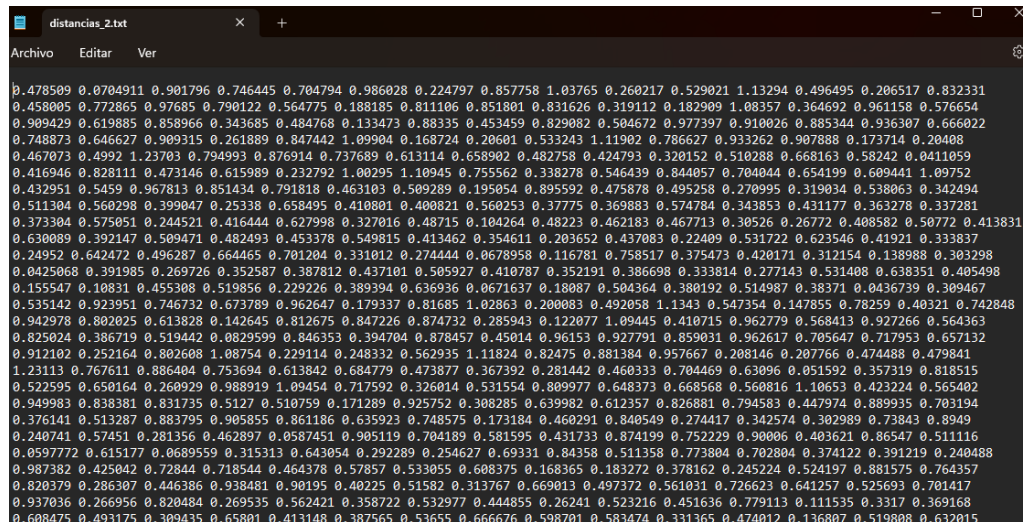
- Compilamos el código con 2 dimensiones para probar que funciona

```

[notice] To update, run: pip install --upgrade pip
PS D:\2023\Segundo semestre\EDA\LAB_01> & 'c:\Users\Jean Pierre\.vscode\extensions\ms-vscode.cpptools-1.17.5-win32-x64\bin\clang++.exe' -stdout=Microsoft-MIEngine-Out-qlzwldiz.lr0' '--stderr=Microsoft-MIEngine-Error-ovojdi3n.jzl' '--pid=Microsoft-MIEngine-Error-ovojdi3n.jzl'
Dimension: 2
PS D:\2023\Segundo semestre\EDA\LAB_01>

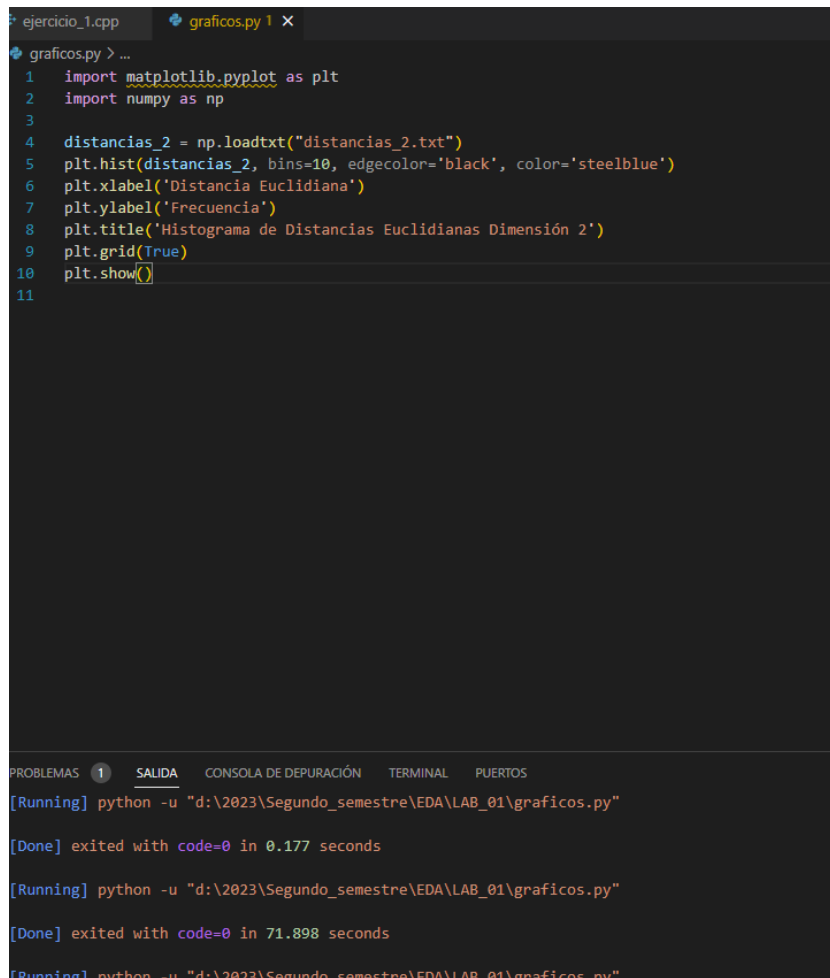
```

- Vemos el archivo .txt con los datos generados



```
p.478509 0.0704911 0.901796 0.746445 0.704794 0.986028 0.224797 0.857758 1.03765 0.260217 0.529021 1.13294 0.496495 0.206517 0.832331
0.458005 0.772865 0.97685 0.790122 0.564775 0.188185 0.811106 0.851801 0.831626 0.319112 0.182909 1.08357 0.364692 0.961158 0.576654
0.909429 0.619885 0.858966 0.343685 0.484768 0.133473 0.88335 0.453459 0.829082 0.504672 0.977397 0.910026 0.885344 0.936307 0.666022
0.748873 0.646627 0.909315 0.261889 0.847442 1.09904 0.168724 0.20601 0.533243 1.11902 0.786627 0.933262 0.907888 0.173714 0.20408
0.467073 0.4992 1.23703 0.794993 0.876914 0.737689 0.613114 0.658902 0.482758 0.424793 0.320152 0.510288 0.668163 0.58242 0.0411059
0.416946 0.828111 0.473146 0.615989 0.232792 1.00295 1.10945 0.755562 0.338278 0.546439 0.844057 0.704044 0.654199 0.609441 1.09752
0.432951 0.5459 0.967813 0.851434 0.791818 0.463103 0.509289 0.195054 0.895592 0.475878 0.495258 0.270995 0.319034 0.538063 0.342494
0.511304 0.560298 0.399047 0.25338 0.658495 0.410801 0.400821 0.560253 0.37775 0.369883 0.574784 0.343853 0.431177 0.363278 0.337281
0.373304 0.575051 0.244521 0.416444 0.627998 0.327016 0.48715 0.104264 0.48223 0.462183 0.467713 0.30526 0.26772 0.408582 0.50772 0.413831
0.630089 0.392147 0.509471 0.482493 0.453378 0.549815 0.413462 0.354611 0.203652 0.437083 0.22409 0.531722 0.623546 0.41921 0.333837
0.24952 0.642472 0.496287 0.664465 0.701204 0.331012 0.274444 0.0678958 0.116781 0.758517 0.375473 0.420171 0.312154 0.138988 0.303298
0.0425068 0.391985 0.769726 0.352587 0.387812 0.437101 0.505927 0.410787 0.352191 0.386698 0.333814 0.277143 0.531408 0.638351 0.405498
0.155547 0.10831 0.455308 0.519856 0.229226 0.389304 0.636936 0.0671637 0.18087 0.504364 0.380192 0.514987 0.38371 0.0436739 0.309467
0.535142 0.923951 0.746732 0.673789 0.962647 0.179337 0.81685 1.02863 0.200083 0.492058 1.1343 0.547354 0.147855 0.78259 0.40321 0.742848
0.942978 0.802025 0.613828 0.142645 0.812675 0.847226 0.874732 0.285943 0.122077 1.09445 0.410715 0.962779 0.568413 0.927266 0.564363
0.825024 0.386719 0.519442 0.0829599 0.846353 0.394704 0.878457 0.45014 0.96153 0.927791 0.859031 0.962617 0.705647 0.717953 0.657132
0.912102 0.252164 0.802608 1.08754 0.229114 0.248332 0.562935 1.11824 0.82475 0.881384 0.957667 0.208146 0.207766 0.474488 0.479841
1.23113 0.767611 0.886404 0.753694 0.613842 0.684779 0.473877 0.367392 0.281442 0.460333 0.704469 0.63096 0.051592 0.357319 0.818515
0.522595 0.650164 0.260929 0.988919 1.09454 0.717592 0.326014 0.531554 0.809977 0.648373 0.668568 0.560816 1.10653 0.423224 0.565402
0.949983 0.838381 0.831735 0.5127 0.510759 0.171289 0.925752 0.308285 0.639982 0.612357 0.826881 0.794583 0.447974 0.889935 0.703194
0.376141 0.513287 0.883795 0.905855 0.861186 0.635923 0.748575 0.173184 0.460291 0.840549 0.274417 0.342574 0.302989 0.73843 0.8949
0.240741 0.57451 0.281356 0.462897 0.0587451 0.905119 0.704189 0.581595 0.431733 0.874199 0.752229 0.90006 0.403621 0.86547 0.511116
0.0597772 0.615177 0.0689559 0.315313 0.643054 0.292289 0.254627 0.69331 0.84358 0.511358 0.773804 0.702804 0.374122 0.391219 0.240488
0.987382 0.425042 0.72844 0.718544 0.464378 0.57857 0.533055 0.608375 0.168365 0.183272 0.378162 0.245224 0.524197 0.881575 0.764357
0.820379 0.286307 0.446386 0.938481 0.90195 0.40225 0.51582 0.313767 0.669013 0.497372 0.561031 0.726623 0.641257 0.525693 0.701417
0.937036 0.266956 0.820484 0.269535 0.562421 0.358722 0.532977 0.444855 0.26241 0.523216 0.451636 0.779113 0.111535 0.3317 0.369168
0.608475 0.493175 0.309435 0.65801 0.413148 0.387565 0.53655 0.666676 0.598701 0.583474 0.331365 0.474012 0.136807 0.519808 0.632015
```

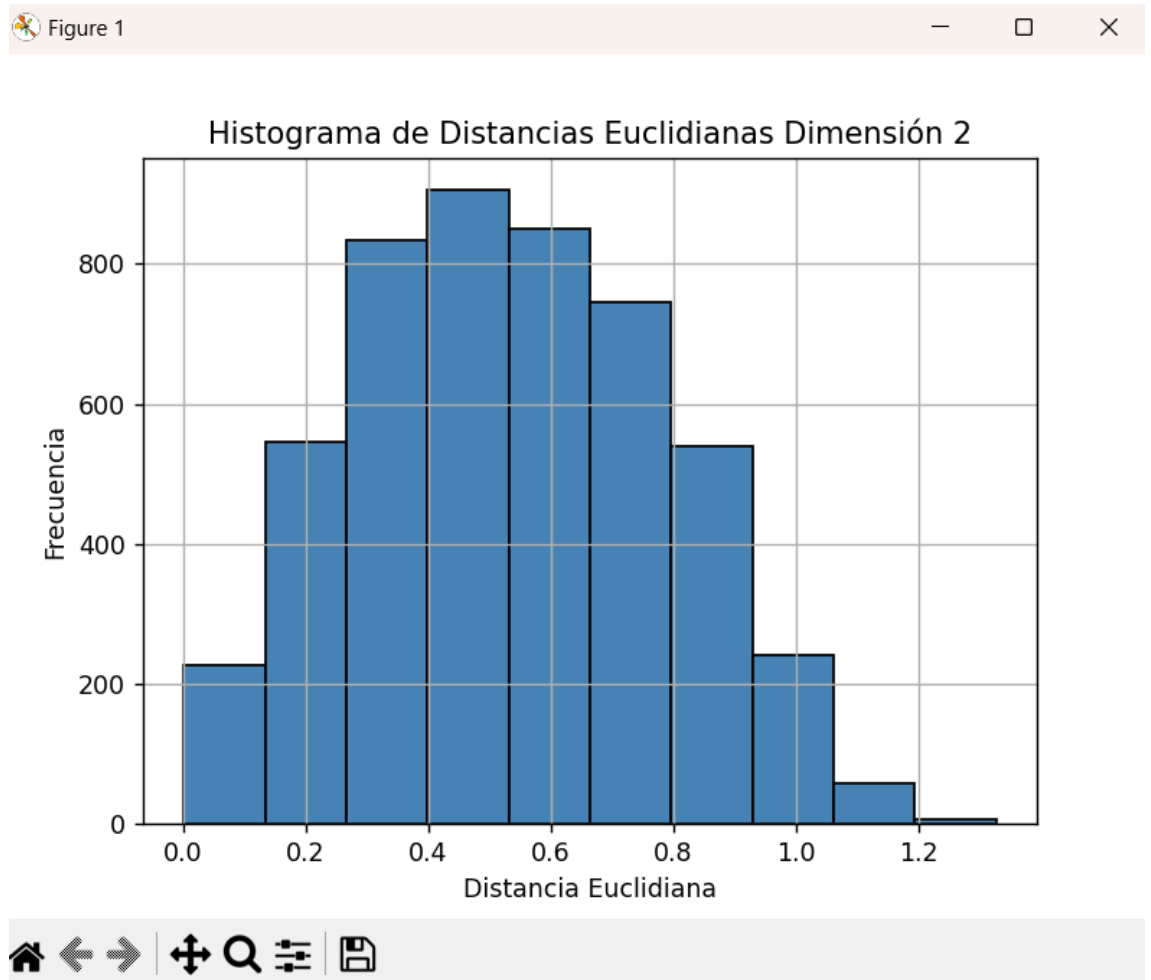
- Ahora compilamos el código Python para ver la grafica



```
ejercicio_1.cpp  graficos.py 1 X
graficos.py > ...
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 distancias_2 = np.loadtxt("distancias_2.txt")
5 plt.hist(distancias_2, bins=10, edgecolor='black', color='steelblue')
6 plt.xlabel('Distancia Euclidiana')
7 plt.ylabel('Frecuencia')
8 plt.title('Histograma de Distancias Euclidianas Dimensión 2')
9 plt.grid(True)
10 plt.show()
11

PROBLEMAS 1 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
[Running] python -u "d:\2023\Segundo_semestre\EDA\LAB_01\graficos.py"
[Done] exited with code=0 in 0.177 seconds
[Running] python -u "d:\2023\Segundo_semestre\EDA\LAB_01\graficos.py"
[Done] exited with code=0 in 71.898 seconds
[Running] python -u "d:\2023\Segundo_semestre\EDA\LAB_01\graficos.py"
```

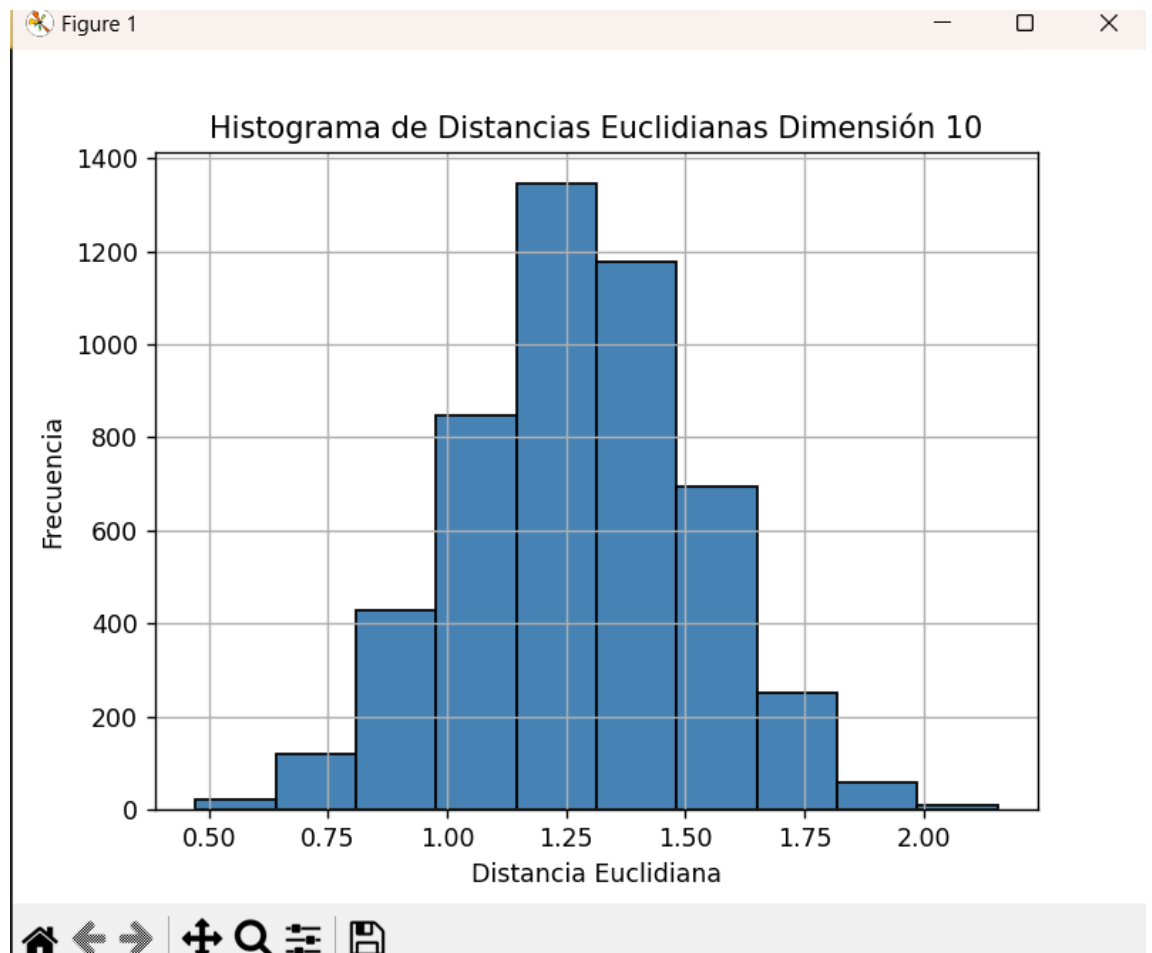
- Y observamos nuestra grafica generada:



- Ahora explicamos los valores de nuestros ejes en el X se muestra los rangos o valores que se obtuvo al calcular las distancias en este caso va desde 0 a 1.2, y en el eje Y se representa el número de veces que se encontró una distancia dentro del rango especificado.
- En este primer grafico se observa que la mayor cantidad de datos abarca desde los rangos 0.3 aproximadamente hasta el 0.6 luego empieza a bajar hasta tener los índices más bajos en 1.2

b) 10 dimensiones

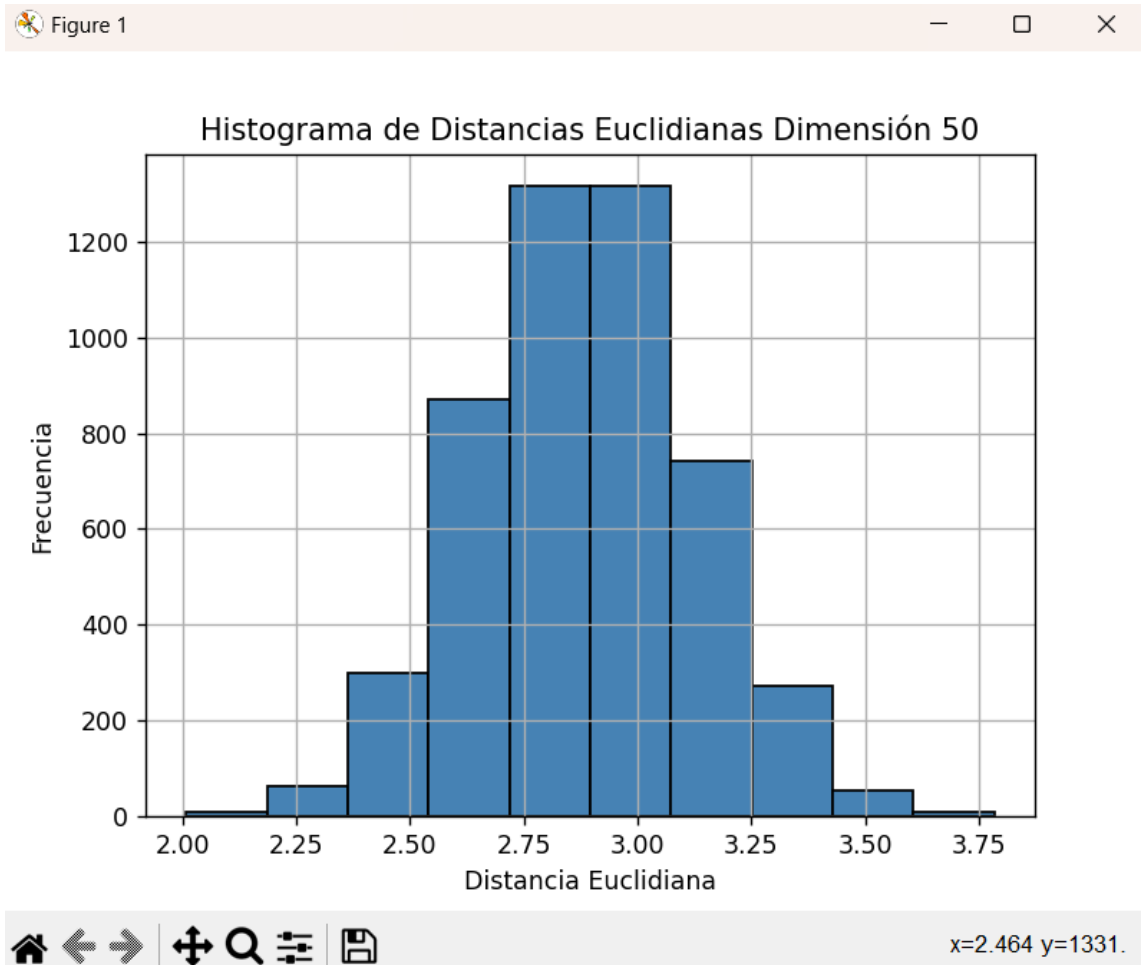
- Ahora mostramos de frente cada grafico ya que el proceso para generarlos es el mismo



- En este grafico vemos que los rangos de los valores aumentaron ahora llegando hasta 2.0 también vemos que los rangos de 1.2 a 1.4 tuvieron un aumento superior a los demás

c) 50 dimensiones

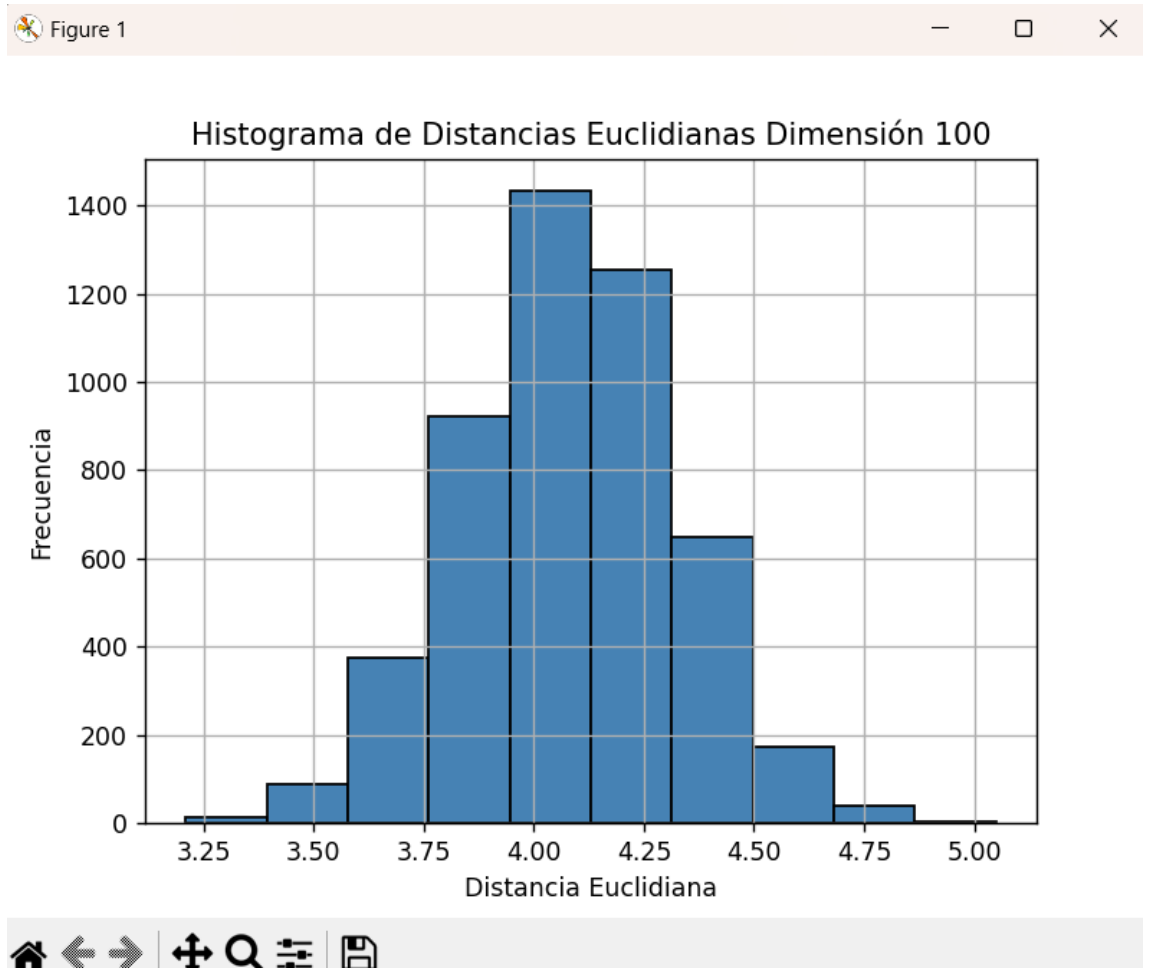
- Mostramos el grafico



- En este grafico vemos que los rangos de los valores aumentaron ahora llegando hasta 3.75 ahora hay que tener en cuenta que los rangos mínimos y máximos de este grafico muestran pocas repeticiones y que comparando con los otros dos gráficos se ve un claro aumento en los rangos del medio como por ejemplo el 2.75 y 3.00

d) 100 dimensiones

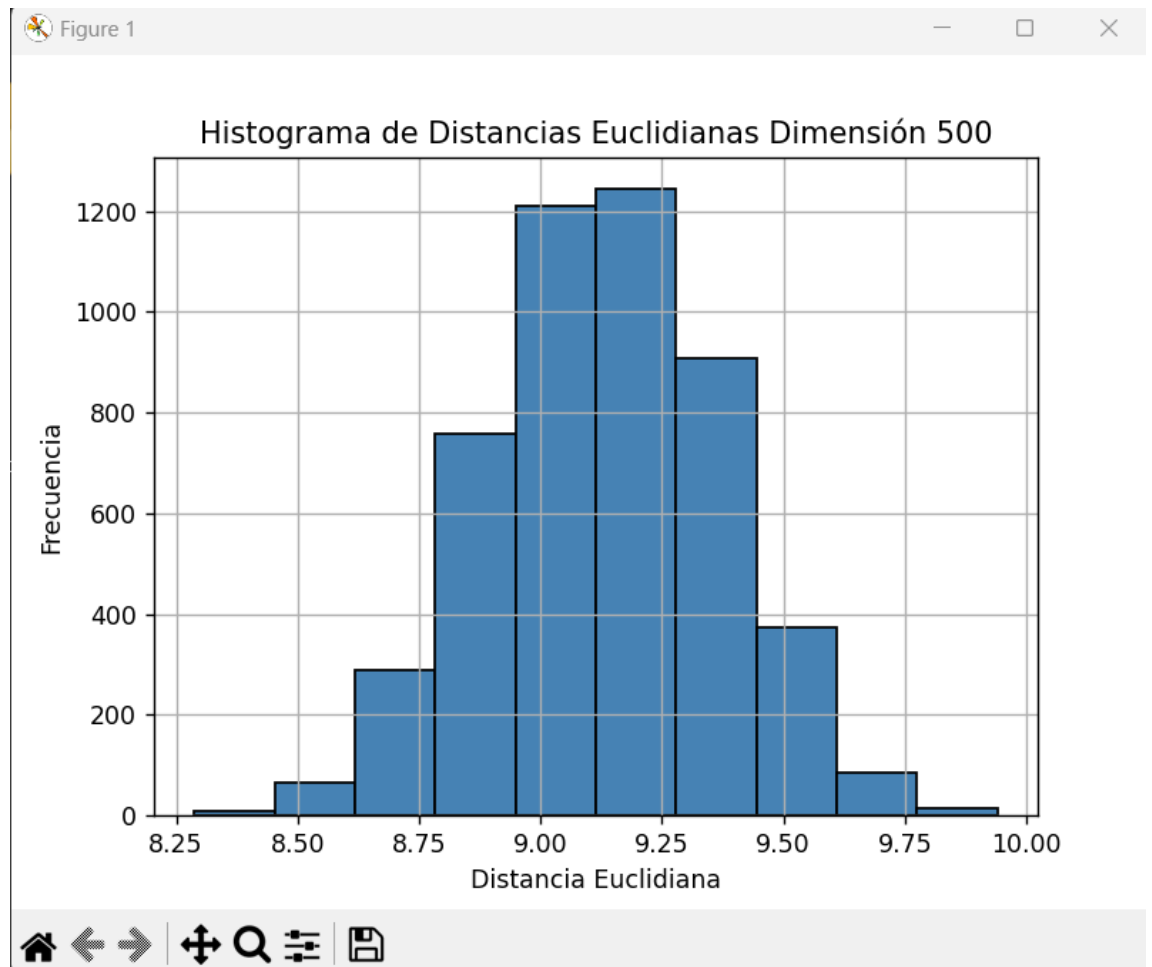
- Mostramos el grafico



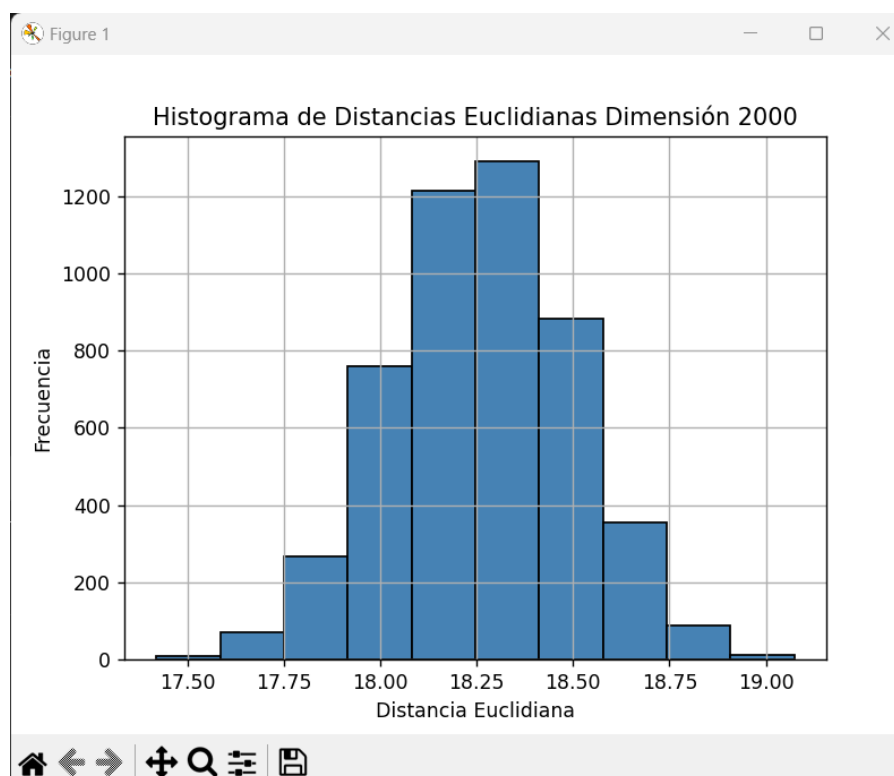
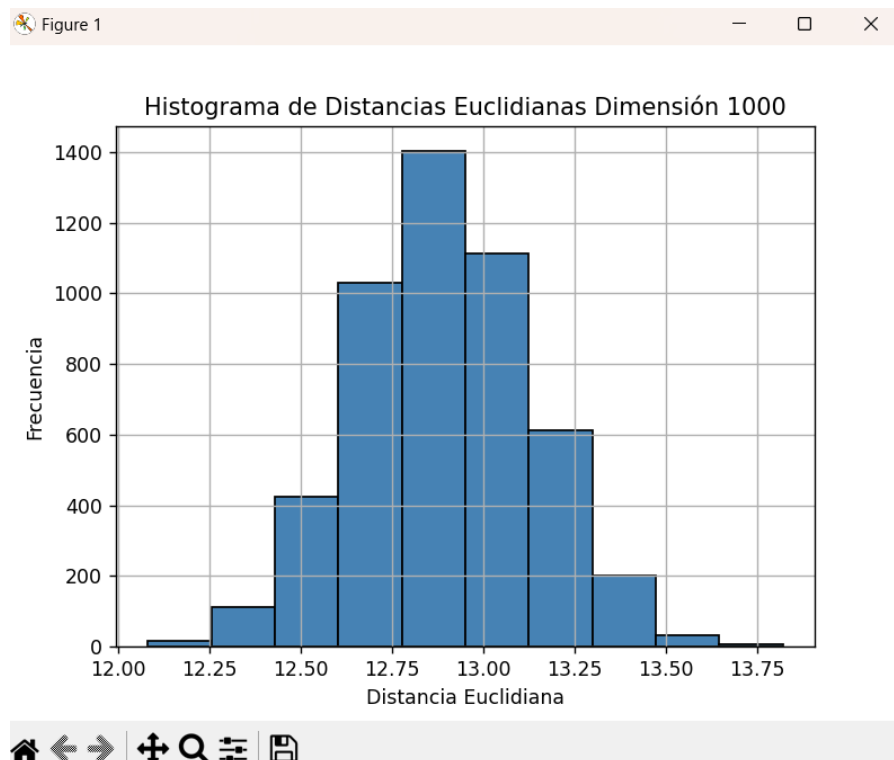
- En este grafico vemos que los rangos de los valores aumentaron ahora llegando hasta 5.0 vemos otra vez un crecimiento en los rangos del medio al parecer este proceso se repetirá a mayor escala en los siguientes graficos

e) 500 dimensiones

- Mostramos el grafico



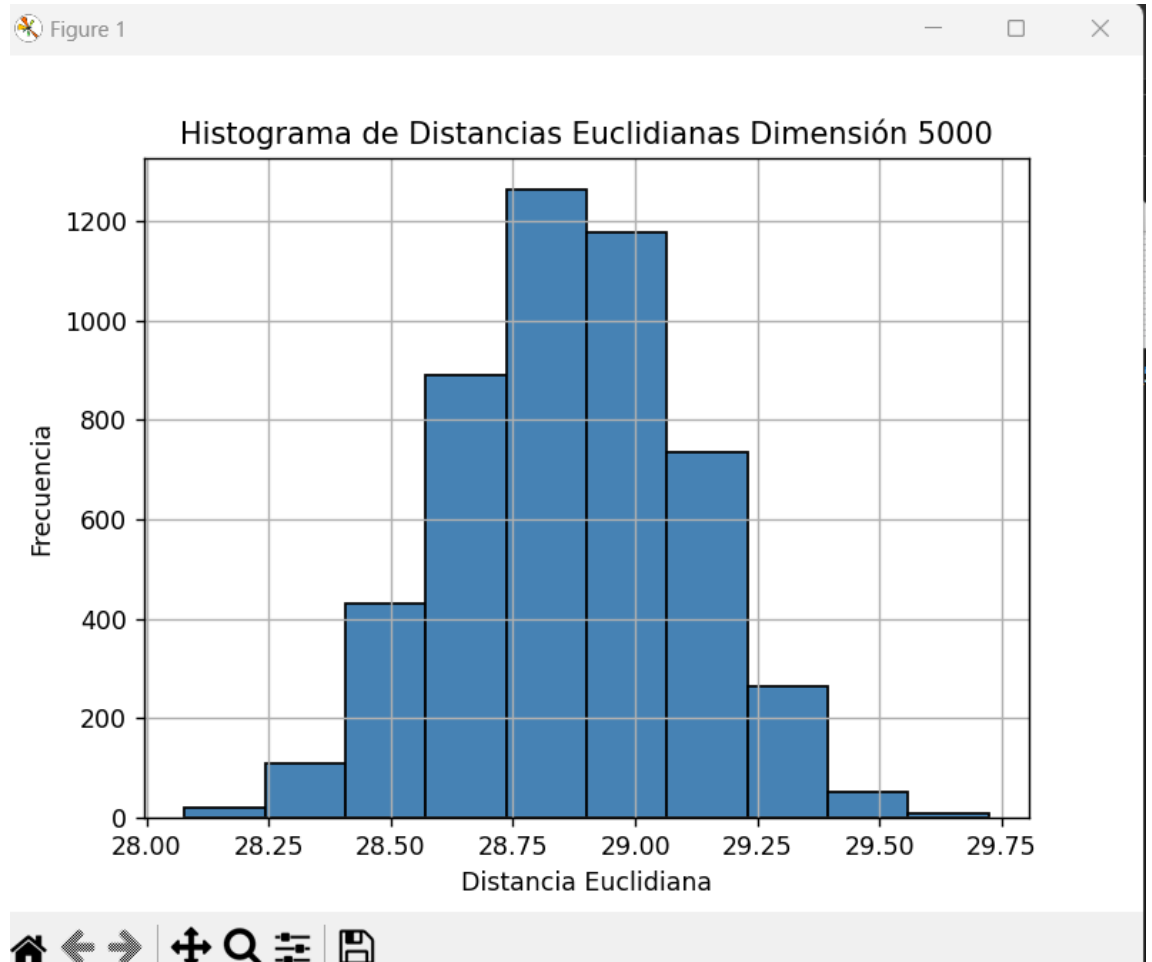
- Como en los puntos anteriores vemos como cada vez quedan más reducidos los rangos de los extremos y como llegan a tal punto que ya son poco visibles y siguen aumentando cada vez mas los del medio
- f) 1000 y 2000 dimensiones
- Mostramos los gráficos



- En ambos gráficos vemos el mismo patrón que se lleva observando ya desde los anteriores ejemplos varían los rangos y también varían nuestros datos intermedios a mayor escala que los demás.

g) 5000 dimensiones

- Mostramos el grafico



- Ya para finalizar vamos a sacar las conclusiones de todo el trabajo al parece mientras mas aumentan nuestras dimensiones se ve como los rangos van aumentando poco a poco lo que provoca un claro alejamiento de los puntos entre si esto provoca que las distancias entre los puntos sea cada vez menos distintiva al final esto provocara que los datos sean cada vez más difíciles de analizar. , también afecta trabajar con un conjunto de datos pequeños ya que siempre es de 0 hasta 1 si se podría variar eso tal vez mejoraría los resultados.

LINK GITHUB:

https://github.com/JpChavez16/LAB_01_EDA.git