

Universidad Nacional de San Agustín
Facultad de Producción y Servicios

ESCUELA PROFESIONAL DE
CIENCIA DE LA COMPUTACIÓN
CURSO:
ESTRUCTURA DE DATOS
AVANZADOS
LAB - GRUPO A

- LABORATORIO 1 -
**LA MALDICIÓN DE LA
DIMENSIONALIDAD**

ALUMNO:

• NELZON JORGE APAZA APAZA
CUI: 20190652

DOCENTE:

• ROSA YULIANA GABRIELA
PACCOTACYA/YANQUE

Arequipa-Perú



LABORATORIO 1: LA MALDICIÓN DE LA DIMENSIONALIDAD

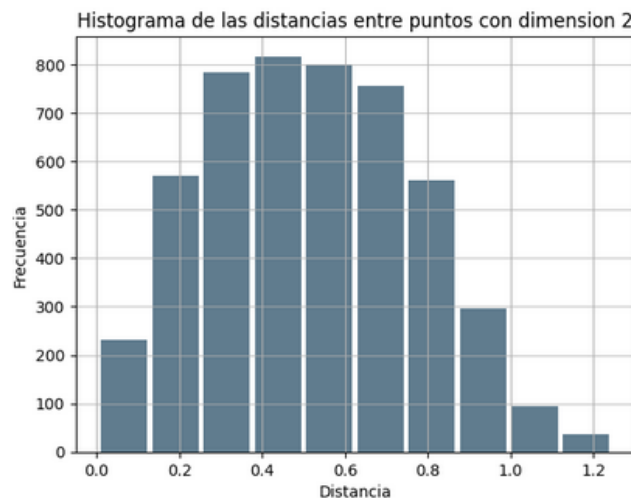
Objetivo

El objetivo de este laboratorio es analizar cómo el espacio cambia a medida que la dimensionalidad (cantidad de atributos o features) de los datos aumenta y cuán desafiante esto puede ser.

Descripción del laboratorio:

Realizaremos diversos experimentos en C++ donde trabajaremos con conjuntos de datos de diferentes dimensiones (10, 50, 100, 500, 1000, 2000, 5000). Para cada conjunto de datos de determinada dimensión d se debe:

- Generar 100 puntos aleatorios entre 0 y 1 de dimensión d .
- Calcular la distancia entre todos los pares de puntos (Distancia Euclidiana) (Hint 4950 distancias)
- Generar un histograma (pueden usar Python) de las distancias obtenidas para cada dimensión como el de la figura mostrada a continuación:



Desarrollo del Laboratorio:

Para cada dimensión d (10, 50, 100, 500, 1000, 2000, 5000) que ejecutaré en el programa, realizaré lo siguiente:

- Generaré 100 puntos aleatorios en un espacio de dimensión d (opción que ingresará el usuario, probaremos las 9 dimensiones propuestas), donde cada coordenada de un punto es un número aleatorio entre 0 y 1. Usaremos `std::uniform_real_distribution`.
- Calcularé la distancia euclidiana entre todos los pares de puntos que se debió generar en el punto A. Esto generará $100 * 99 / 2 = 4950$ distancias en total. Utilizaré la fórmula de distancia euclidiana en C++ para calcular las distancias.
- Exportaré los resultados de las distancias obtenidas para cada dimensión en un archivo TXT. Para que luego Python logre graficar los histogramas.

d. Usaremos las bibliotecas como NumPy y Matplotlib para crear un histograma para cada dimensión con los datos guardados en el txt.

Código implementado:

C++:

```
/*
Para cada dimensión d (10, 50, 100, 500, 1000, 2000, 5000) que ejecutaré en
el programa,
realizaré lo siguiente:

a. Generaré 100 puntos aleatorios en un espacio de dimensión d (opción que
ingresará el usuario, probaremos las 9 dimensiones propuestas),
donde cada coordenada de un punto es un número aleatorio entre 0 y 1.
Usaremos "std::uniform_real_distribution".

b. Calcularé la distancia euclidiana entre todos los pares de puntos que se
debió generar en el punto A. Esto generará  $100 * 99 / 2 = 4950$  distancias en
total.
Utilizaré la fórmula de distancia euclidiana en C++ para calcular las
distancias.

c. Exportaré los resultados de las distancias obtenidas para cada dimensión
en un
archivo TXT. Para que luego Python logre graficar los histogramas.
*/
/*
    Nelson Jorge Apaza Apaza
    EDA LABORATORIO
    La MALDICIÓN DE LA DIMENSIONALIDAD
*/
#include <iostream>
#include <random> //aquí tenemos a uniform_real_distribution
#include <cmath> //operación pow y sqrt
#include <fstream>

using namespace std;

int main() {
    // d: dimension
    int d;
    cout<<"Ingrese la dimension: "<<endl;
    cin>>d;
    int numPuntosAleatorios = 100; //100 numeros aleatorios

    // Aquí generamos el número aleatorio
```

```

// Tal y como indica el ejemplo de uniform_real_distribution
random_device rd; //rd se utilizará para obtener una semilla para el
motor de números aleatorios
mt19937 gen(rd()); //usamos la semilla
uniform_real_distribution<double> dis(0.0, 1.0); //desde 0 a 1

// Abre un archivo para escribir las distancias
std::ofstream archivo("distanciasEuclidianas.txt");

// Verifica que el archivo se haya abierto correctamente
if (!archivo) {
    std::cerr << "No se pudo abrir el archivo para escritura." <<
std::endl;
    return 1;
}

//Solo usaremos una matriz de tamaño numPuntos*dimension
double MatrizPuntos[numPuntosAleatorios][d];

// Generamos los puntos aleatorios
for (int i = 0; i < numPuntosAleatorios; ++i) {
    for (int j = 0; j < d; ++j) {
        MatrizPuntos[i][j] = dis(gen);
        /*
        Asigna un número aleatorio en el rango [0.0, 1.0]
        a la coordenada j del punto i en la matriz MatrizPuntos.
        */
    }
}

// Aquí calculamos la distancia eucladiana, imitando la formula
/*
* No se debe calcular la distancia entre un punto y sí mismo.
* No se debe calcular dos veces la misma distancia.
*/
for (int i = 0; i < numPuntosAleatorios; ++i) {
    for (int j = i + 1; j < numPuntosAleatorios; ++j) {
        double distEucladiana = 0.0; //Acumulador

        /*
        * Calculamos la diferencia al cuadrado entre las coordenadas k
de
        los puntos i y j.

```

```

        * Se suma a distEuclidiana y se acumulan estas diferencias al
        cuadrado (pow) para todas las dimensiones.
        */
        for (int k = 0; k < d; ++k) {
            distEuclidiana += pow(MatrizPuntos[i][k] -
MatrizPuntos[j][k], 2);
        }
        distEuclidiana = sqrt(distEuclidiana); //sacamos raiz
        //Resultado
        cout << "Distancia Euclidiana entre punto " << i << " y punto "
<< j << ": " << distEuclidiana << endl;
        //Guardamos distancias
        archivo << distEuclidiana << std::endl;
    }
}

// Cierra el archivo txt
archivo.close();

return 0;
}

```

Python:

```

"""
Usaremos las bibliotecas como NumPy y Matplotlib para crear un
histograma para cada dimensión con los datos guardados en el txt.
"""

import matplotlib.pyplot as plt #Para fácil uso

# Lista vacía para las distanciasEuclidianas
distanciasEuclidianas = []

with open("distanciasEuclidianas.txt", "r") as archivo:
    for linea in archivo: #recorrerá cada línea del archivo
        distancia = float(linea.strip()) #convertir de a float
        distanciasEuclidianas.append(distancia) #distancias agregadas

# Partes del Histograma
# Partes automáticas para el histograma según manual Matplotlib
plt.hist(distanciasEuclidianas, bins=20, edgecolor="k") #hist: histograma
#bins: cantidad de contenedores
# k: color de los bordes de las barras, k es negro
plt.xlabel("Distancia Euclidiana")

```

```
plt.ylabel("Frecuencia")
plt.title("Histograma de Distancias")
plt.grid(True) # Las líneas de la cuadrícula
plt.show() # Mostrar
```

Ejecución con ejemplo de dimensión 2 como prueba de que se obtienen de manera correcta los puntos aleatorios y las distancias.

Puntos aleatorios:

```
PS D:\UNSA\3ER AÑO\2do semestre\EDA> cd "d:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimens
ionesDistancias.cpp -o DimensionesDistancias } ; if ($?) { .\DimensionesDistancias }
Ingrese la dimension:
2
0.726249
0.915339
0.594324
0.515358
0.975149
0.661561
0.528652
0.788493
0.0741007
0.32985
0.583744
0.0309722
0.928593
0.0197524
0.779372
0.336045
```

Las distancias (parte de ellas):

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE COMMENTS > Code -
Distancia Euclidiana entre punto 8 y punto 57: 0.127043
Distancia Euclidiana entre punto 8 y punto 58: 0.116711
Distancia Euclidiana entre punto 8 y punto 59: 0.158988
Distancia Euclidiana entre punto 8 y punto 60: 0.613705
Distancia Euclidiana entre punto 8 y punto 61: 0.139516
Distancia Euclidiana entre punto 8 y punto 62: 0.6477
Distancia Euclidiana entre punto 8 y punto 63: 0.190758
Distancia Euclidiana entre punto 8 y punto 64: 0.577842
Distancia Euclidiana entre punto 8 y punto 65: 0.121251
Distancia Euclidiana entre punto 8 y punto 66: 0.23407
Distancia Euclidiana entre punto 8 y punto 67: 0.46109
Distancia Euclidiana entre punto 8 y punto 68: 0.464729
Distancia Euclidiana entre punto 8 y punto 69: 0.650794
Distancia Euclidiana entre punto 8 y punto 70: 0.557429
Distancia Euclidiana entre punto 8 y punto 71: 0.40421
Distancia Euclidiana entre punto 8 y punto 72: 0.0991759
Distancia Euclidiana entre punto 8 y punto 73: 0.498134
Distancia Euclidiana entre punto 8 y punto 74: 0.468851
Distancia Euclidiana entre punto 8 y punto 75: 0.582474
Distancia Euclidiana entre punto 8 y punto 76: 0.553687
Distancia Euclidiana entre punto 8 y punto 77: 0.381451
Distancia Euclidiana entre punto 8 y punto 78: 0.352479
Distancia Euclidiana entre punto 8 y punto 79: 0.333858
PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad>
```

Dimensión 10:

```
17  /*
18      Nelson Jorge Apaza Apaza
19      EDA LABORATORIO
20      La MALDICIÓN DE LA DIMENSIONALIDAD
21  */
22  #include <iostream>

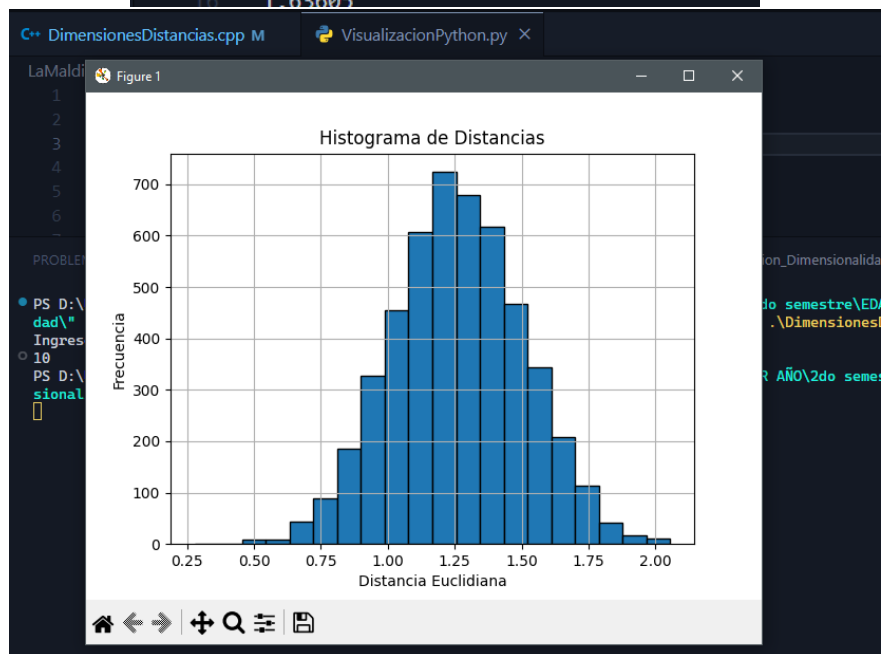
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE COMMENTS Code-Editor

PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad> cd "d:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad\" ; if ($?) { g++ DimensionesDistancias.cpp -o DimensionesDistancias } ; if ($?) { .\DimensionesDistancias }
Ingrese la dimension:
10
PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad>
```

distanciasEuclidianas.txt U C++ DimensionesDistancias

LaMaldicion_Dimensionalidad > distanciasEuclidianas.txt

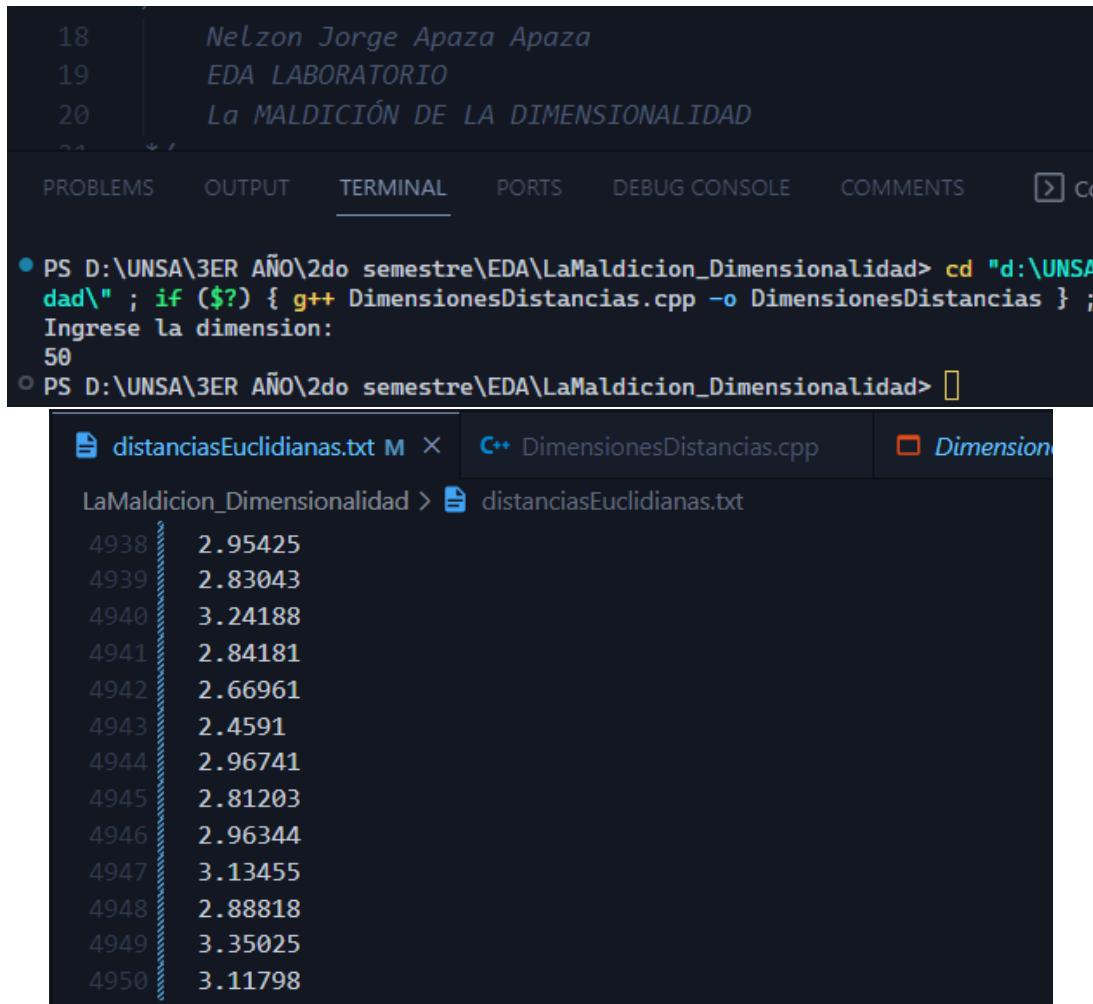
```
1 1.50865
2 1.38875
3 1.21606
4 1.06346
5 1.26884
6 0.779079
7 0.950414
8 1.36036
9 0.746455
10 1.30043
11 1.417
12 1.16466
13 1.58834
14 1.19692
15 1.09215
16 1.63605
```



Resultados:

Podemos observar que el histograma tiene un ajuste adecuado. Además, la distancia euclidiana más frecuente se encuentra entre los valores de 1.10 y 1.35.

Dimensión 50:



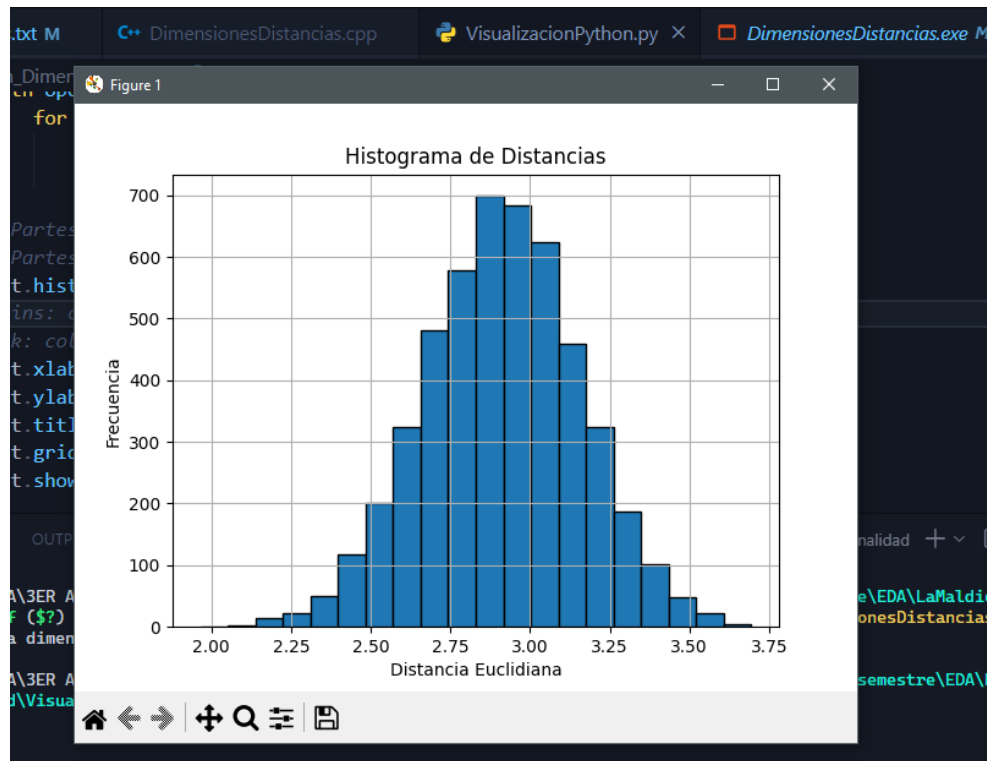
The screenshot shows a code editor with two windows. The top window, titled 'La Maldición DE LA DIMENSIONALIDAD', contains a C++ program. The bottom window, titled 'distanciasEuclidianas.txt', shows the output of the program for dimension 50.

```
18      Nelzon Jorge Apaza Apaza
19      EDA LABORATORIO
20      La MALDICIÓN DE LA DIMENSIONALIDAD

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE  COMMENTS  >

● PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad> cd "d:\UNSA
dad\" ; if ($?) { g++ DimensionesDistancias.cpp -o DimensionesDistancias } ;
Ingrese la dimension:
50
○ PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad>

LaMaldicion_Dimensionalidad > distanciasEuclidianas.txt
4938  2.95425
4939  2.83043
4940  3.24188
4941  2.84181
4942  2.66961
4943  2.4591
4944  2.96741
4945  2.81203
4946  2.96344
4947  3.13455
4948  2.88818
4949  3.35025
4950  3.11798
```

Resultados:

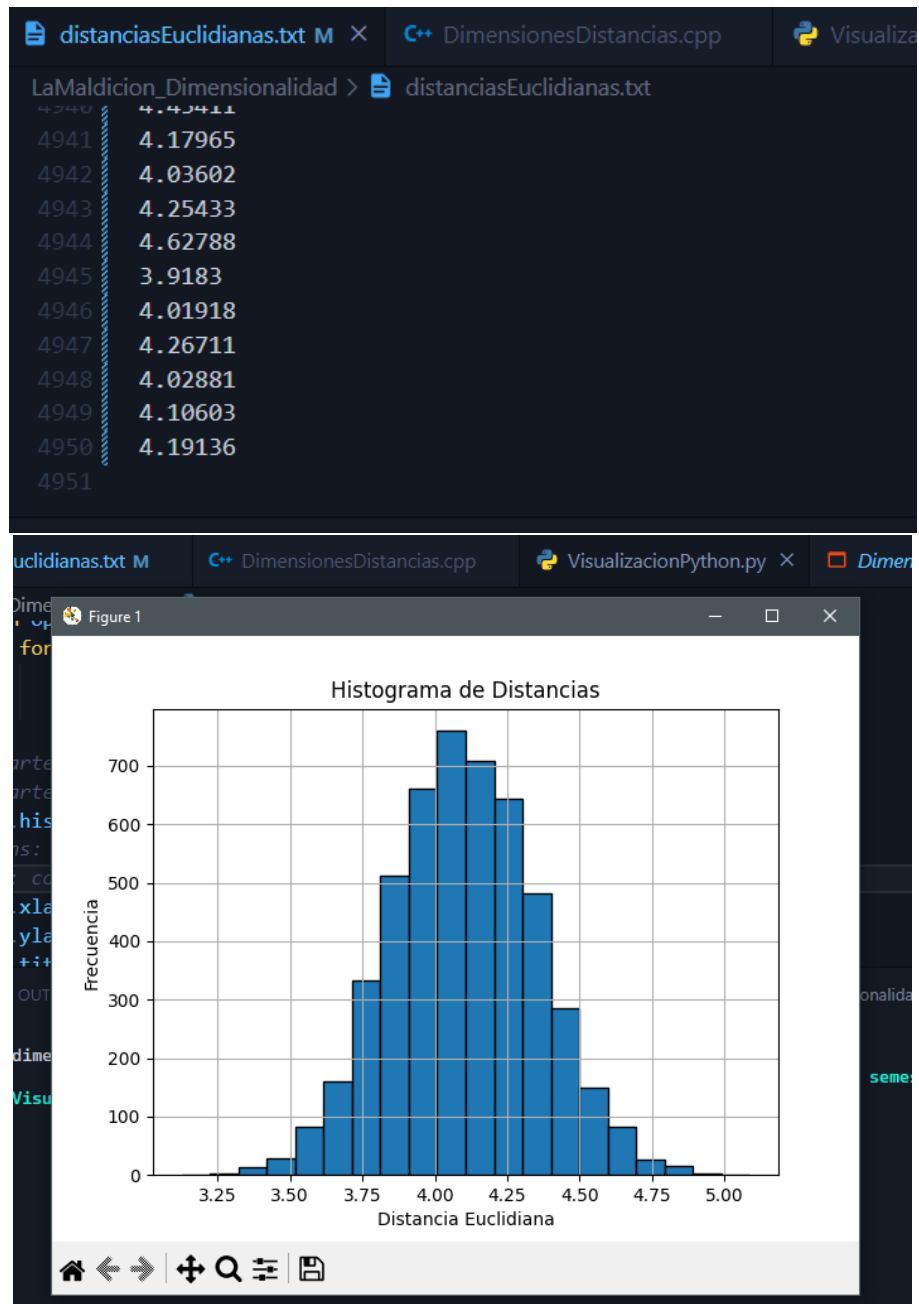
Podemos observar que el histograma es algo asimétrico hacia la izquierda, es decir tiene cierta inclinación a la izquierda con un ajuste adecuado. Además, la distancia euclidiana más frecuente se encuentra entre los valores de 2.75 y 3.20 y la dispersión entre las distancias aumentó.

Dimensión 100:

```
17  /*
18     Nelzon Jorge Apaza Apaza
19     EDA LABORATORIO
20     La MALDICIÓN DE LA DIMENSIONALIDAD
21  */
22  #include <iostream>
23  #include <random> //aquí tenemos a uniform real distribut...
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE COMMENTS

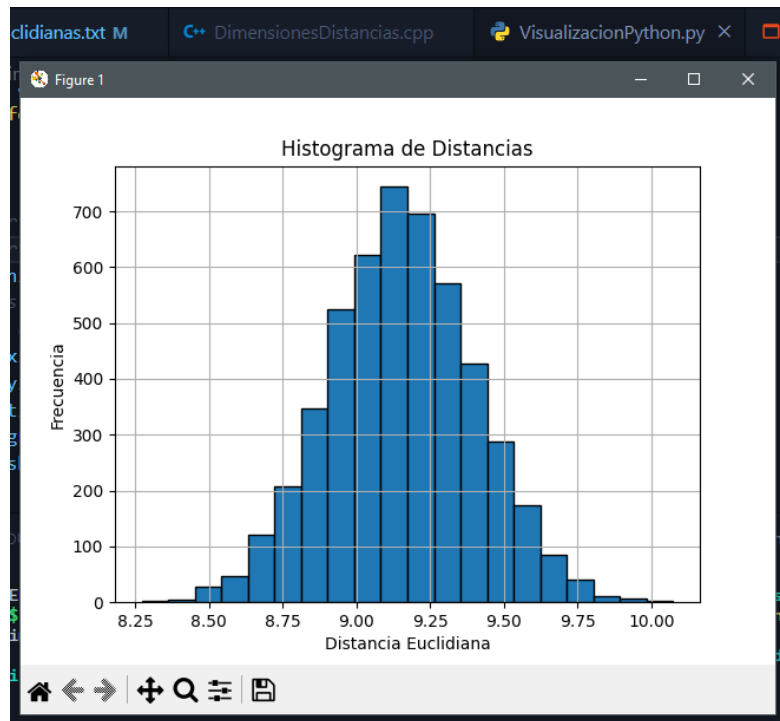
```
Ingrese la dimension:
100
PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad> |
```



Resultados:

Podemos observar que el histograma no es asimétrico hacia la izquierda ni a la derecha con un ajuste adecuado. Además, la distancia euclidiana más frecuente se encuentra entre los valores de 4.00 y 4.20 y la dispersión entre las distancias aumentó.

Dimensión 500:



Resultados:

Podemos observar que el histograma es algo asimétrico hacia la derecha, es decir tiene cierta inclinación a la derecha con un ajuste adecuado. Además, la distancia euclidiana más frecuente se encuentra entre los valores de 9.00 y 9.25 y la dispersión entre las distancias aumentó.

Dimensión 1000:

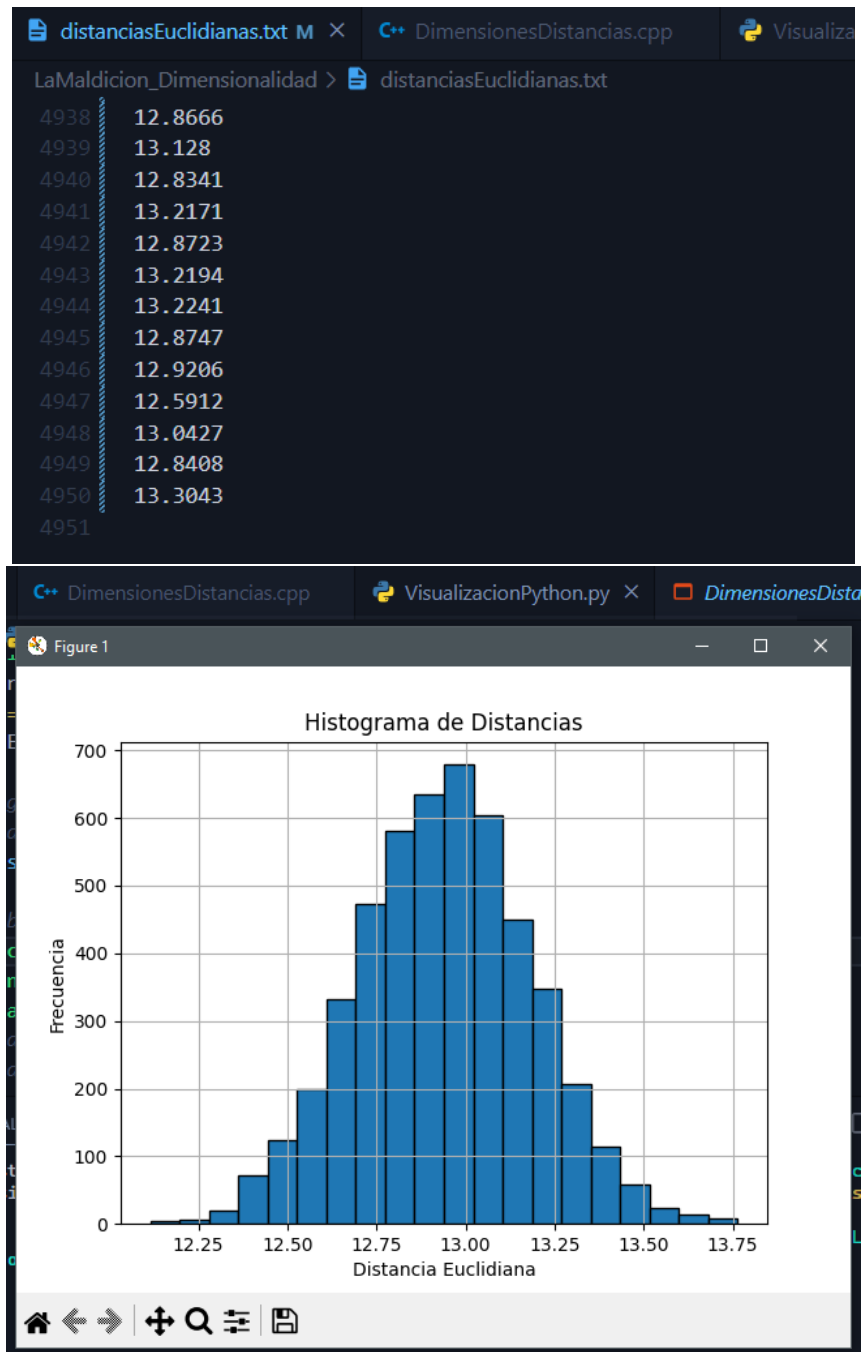
```

distanciasEuclidianas.txt M  C++ DimensionesDistancias.cpp  VisualizaciónPython.py X
LaMaldicion_Dimensionalidad > C++ DimensionesDistancias.cpp > ...
17  /*
18     Nelson Jorge Apaza Apaza
19     EDA LABORATORIO
20     La MALDICIÓN DE LA DIMENSIONALIDAD
21  */
22  #include <iostream>

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE  COMMENTS

PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad> cd "d:\
dad\" ; if ($?) { g++ DimensionesDistancias.cpp -o DimensionesDistancias
Ingrese la dimension:
1000
PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad>

```



Resultados:

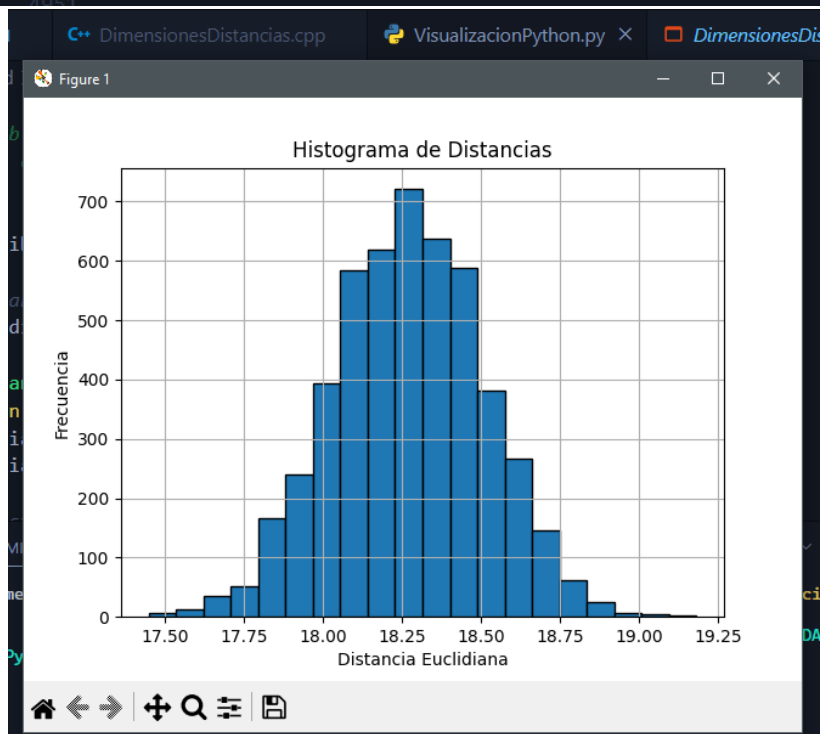
Podemos observar que el histograma es algo asimétrico hacia la derecha, es decir tiene cierta inclinación a la derecha con un ajuste adecuado. Además, la distancia euclidiana más frecuente se encuentra entre los valores de 12.80 y 13.10 aproximadamente y la dispersión entre las distancias aumentó.

Dimensión 2000:

```
distanciasEuclidianas.txt M x C++ DimensionesDistancias.cpp x VisualizacionPython.py
LaMaldicion_Dimensionalidad > C++ DimensionesDistancias.cpp > ...
17  /*
18     Nelson Jorge Apaza Apaza
19     EDA LABORATORIO
20     La MALDICIÓN DE LA DIMENSIONALIDAD
21  */
22  #include <iostream>
23  #include <random> //aquí tenemos a uniform_real_distribution

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE COMMENTS Code - LaMaldicion
dad\" ; if ($?) { g++ DimensionesDistancias.cpp -o DimensionesDistancias } ; if ($?) { .
Ingrese la dimension:
2000
PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad> []
```

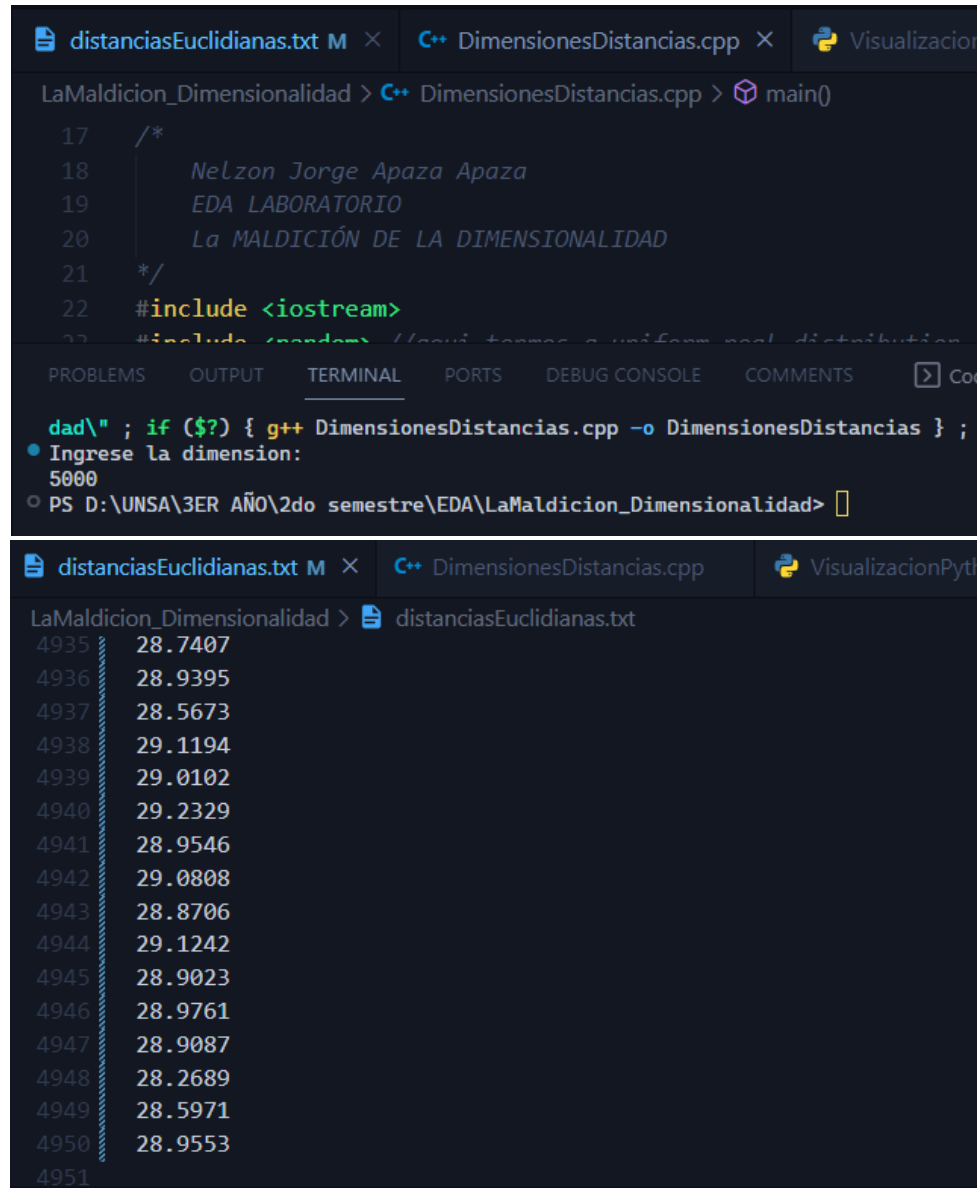
```
distanciasEuclidianas.txt M x C++ DimensionesDistancias.cpp x Visualizac
LaMaldicion_Dimensionalidad > distanciasEuclidianas.txt
4938 18.2551
4939 18.3844
4940 17.9332
4941 18.5
4942 17.851
4943 18.1165
4944 17.7915
4945 17.9588
4946 18.3853
4947 18.1928
4948 18.2454
4949 17.9456
4950 18.0113
4951
```



Resultados:

Podemos observar que el histograma es más asimétrico hacia la derecha, es decir tiene una inclinación a la derecha con un ajuste adecuado. Además, la distancia euclidiana más frecuente se encuentra entre los valores de 18.10 y 18.50 aproximadamente y la dispersión entre las distancias aumentó.

Dimensión 5000:

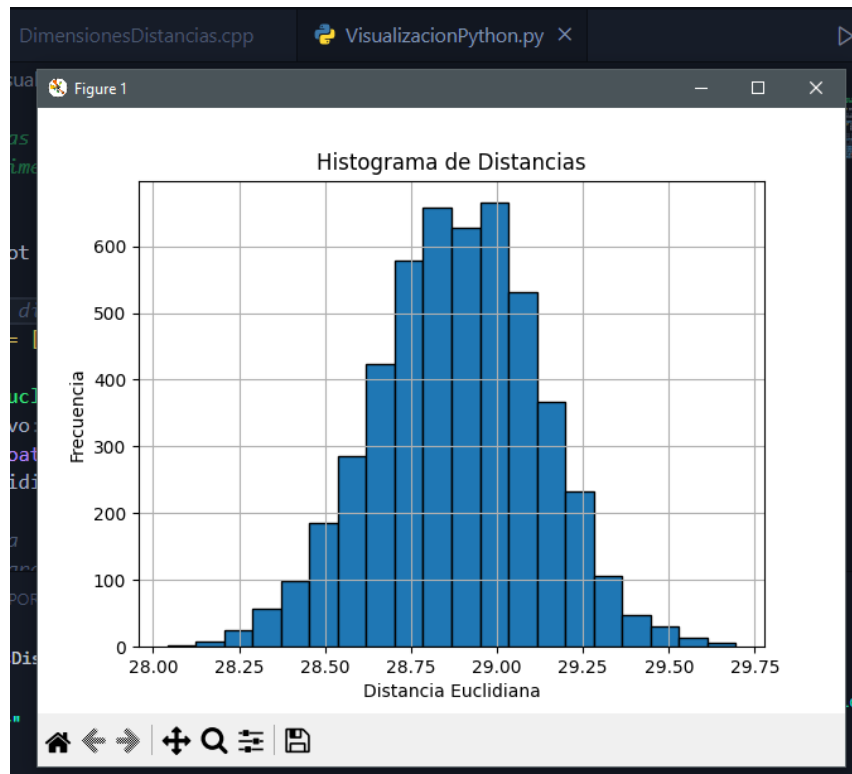


The image shows a C++ IDE with two windows. The top window, titled 'DimensionesDistancias.cpp', contains C++ code for a program. The code includes a header comment with the author's name 'Nelzon Jorge Apaza Apaza', the course 'EDA LABORATORIO', and the topic 'La MALDICIÓN DE LA DIMENSIONALIDAD'. It includes the `<iostream>` and `<random>` headers. The bottom window, titled 'distanciasEuclidianas.txt', shows the output of the program, which is a list of 17 Euclidean distance values. The values are: 28.7407, 28.9395, 28.5673, 29.1194, 29.0102, 29.2329, 28.9546, 29.0808, 28.8706, 29.1242, 28.9023, 28.9761, 28.9087, 28.2689, 28.5971, 28.9553, and 28.9553.

```
LaMaldicion_Dimensionalidad > C++ DimensionesDistancias.cpp > main()
17  /*
18     Nelzon Jorge Apaza Apaza
19     EDA LABORATORIO
20     La MALDICIÓN DE LA DIMENSIONALIDAD
21  */
22  #include <iostream>
23  #include <random> //equiv. to rand() in C

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE COMMENTS
dad\" ; if ($?) { g++ DimensionesDistancias.cpp -o DimensionesDistancias } ;
• Ingrese la dimension:
5000
○ PS D:\UNSA\3ER AÑO\2do semestre\EDA\LaMaldicion_Dimensionalidad>

LaMaldicion_Dimensionalidad > distanciasEuclidianas.txt
4935 28.7407
4936 28.9395
4937 28.5673
4938 29.1194
4939 29.0102
4940 29.2329
4941 28.9546
4942 29.0808
4943 28.8706
4944 29.1242
4945 28.9023
4946 28.9761
4947 28.9087
4948 28.2689
4949 28.5971
4950 28.9553
4951
```

Resultados:

Podemos observar que el histograma es algo asimétrico hacia la derecha, es decir tiene cierta inclinación a la derecha con un ajuste adecuado. Además, la distancia euclidiana más frecuente se encuentra entre los valores de 28.80 y 29.05 aproximadamente y la dispersión entre las distancias aumentó.

Conclusión:

Como se pudo ver en los gráficos y en los resultados observados en estos, a medida que aumenta la dimensionalidad, la dispersión de las distancias tiende a aumentar significativamente, es decir que la mayoría de los puntos tienden a estar lejos entre sí. Ya no hay muchas distancias cercanas entre los puntos.

La "Maldición de la Dimensionalidad" nos describe los desafíos y problemas que surgen al analizar datos en espacios de alta dimensionalidad. Si la dimensionalidad aumenta, los datos se vuelven más dispersos y esto puede dificultar la identificación de patrones entre los puntos.

Bibliografía:

- Cpp Reference / *std::uniform_real_distribution* / enlace web: https://en.cppreference.com/w/cpp/numeric/random/uniform_real_distribution
- Blog – Clases Particulares / *Cómo calcular la distancia entre dos puntos* / Sitio web: <https://www.tusclasesparticulares.com/blog/como-calcular-distancia-entre-dos-puntos>

- Blog – Geometría 2D / *Distancia entre dos puntos* / Sitio web: <https://www.problemasyecuaciones.com/geometria2D/distancia-puntos/distancia-puntos-formula-calcular-ejemplos-problemas-resueltos.html>
- Matplotlib - Python / Sitio web: <https://matplotlib.org/stable/tutorials/pyplot.html>
- Website – Aprendizaje Estadístico / *La Maldición de la Dimensionalidad* / Sitio web: https://rubenfcasal.github.io/aprendizaje_estadistico/dimen-curse.html

Repositorio GitHub: https://github.com/nelzonapa/EstructuraDatosAvanzados/tree/main/LaMaldicion_Dimensionalidad