# Objetivos

Desarrollo Taller 1 AI: Decisión Bayesiana

Juan Sebastián Bravo Santacruz

### Visualizar e interpretar datos de acuerdo con la estadística descriptiva de los mismos.

### Diseñar e implementar clasificadores Bayesianos.

# Conjunto de datos data\_2D

El conjunto de datos data\_2D (ver archivo data.npy) contiene datos pertenecientes a dos clases {a, b}.

## Grafique los datos utilizando un color distintivo para cada clase:

Inicialmente se realizó la importación de los datos del archivo data.npy y se separaron los conjuntos data\_2D y data\_3D en 2 distintas variables:

Text

Description automatically generated

Una vez separados los conjuntos de datos, se extrajeron los valores de las clases a y b y se realizó la gráfica de estas dos clases usando el módulo matplotlib.

Text

Description automatically generated

En este caso, se eligió el color rojo para la clase a y el color azul para la clase b para poder distinguirlos adecuadamente en la gráfica.

La gráfica obtenida se presenta en la siguiente imagen:

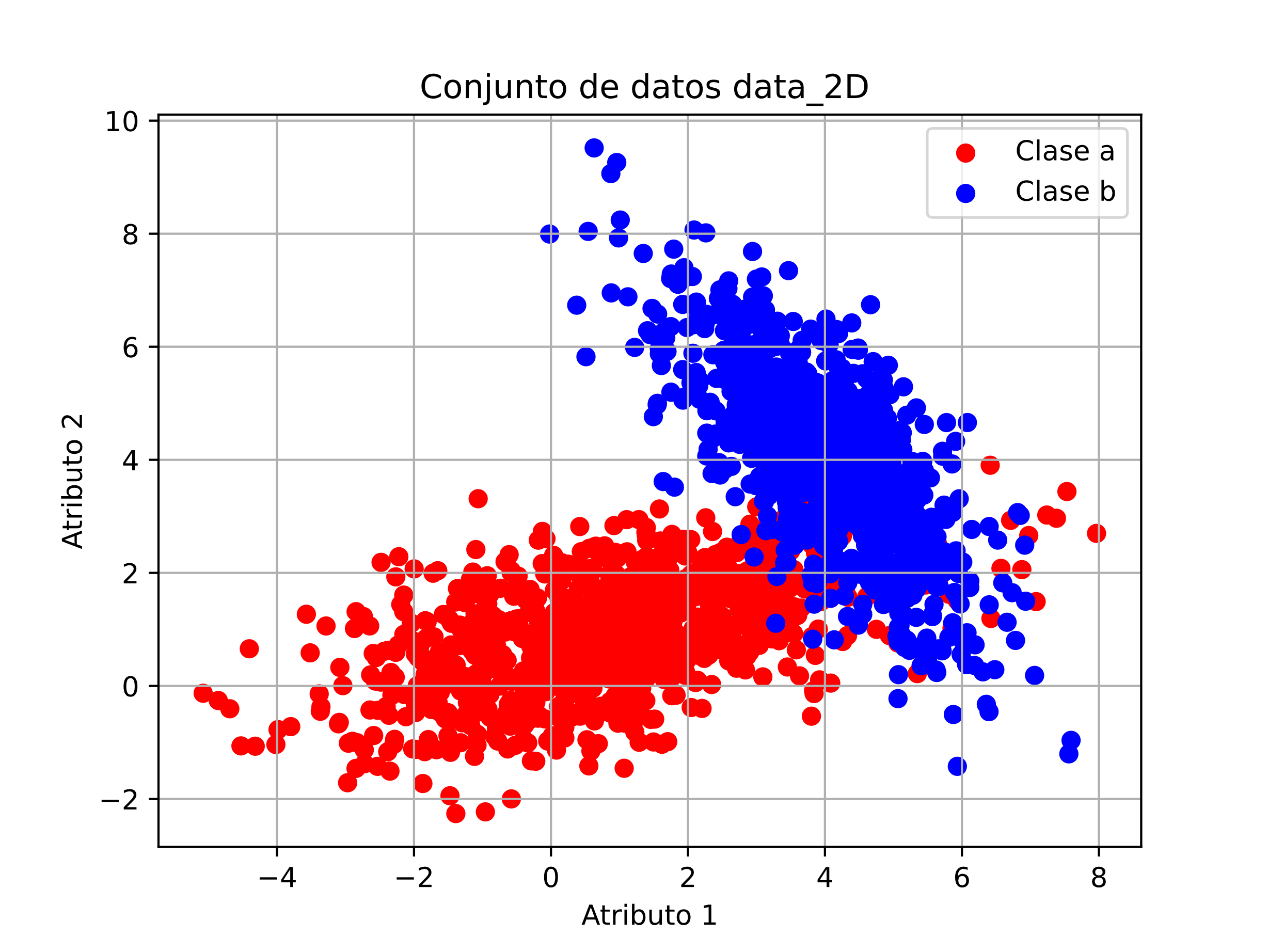
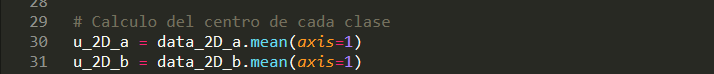


Figura 1. Conjunto de datos data\_2D.

## Determine el centro de cada clase :

Para encontrar el centro de cada una de las clases, se utilizó la función mean() del módulo numpy:



En este caso, al tener 2 atributos, se obtiene un vector de 2 valores de media, uno por cada atributo:

A screenshot of a computer

Description automatically generated

Figura 2. Vector medias para clase a.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 3. Vector medias para clase b.

## Determine las matrices de covarianza de cada clase {Ka, Kb}. ¿Qué se puede concluir?:

Para encontrar las matrices de covarianza de cada una de las clases, se utilizó la función cov() del módulo numpy:



En este caso, al tener 2 atributos, se obtiene una matriz cuadrada de 2x2:

A screenshot of a computer

Description automatically generated with medium confidence

Figura 4. Matriz de covarianza clase a.

De la matriz de covarianza de la clase a, se puede concluir que los atributos no son independientes debido a que no es una matriz diagonal, y que tienen una relación directa (a medida que el atributo 1 aumenta, el atributo 2 aumenta), adicionalmente, se puede concluir que los datos varían más en el atributo 1 que en el atributo 2.

Lo anteriormente dicho, se puede comprobar al ver la gráfica de datos realizados en el inciso A. del ejercicio.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 5. Matriz de covarianza clase b.

De la matriz de covarianza de la clase b, se puede concluir que los atributos no son independientes debido a que no es una matriz diagonal, y que tienen una relación inversa (a medida que el atributo 1 aumenta, el atributo 2 disminuye), adicionalmente, se puede concluir que los datos varían más en el atributo 2 que en el atributo 1.

Lo anteriormente dicho, se puede comprobar al ver la gráfica de datos realizados en el inciso A. del ejercicio.

## Determine y visualice el histograma de los datos.

Se utilizó el módulo matplotlib para encontrar el histograma de los datos, haciendo uso de la función histrogram2d() y meshgrid haciendo un histograma para cada una de las clases como se muestra en el siguiente código:

Text

Description automatically generated

Los histogramas obtenidos se presentan en las siguientes imágenes:

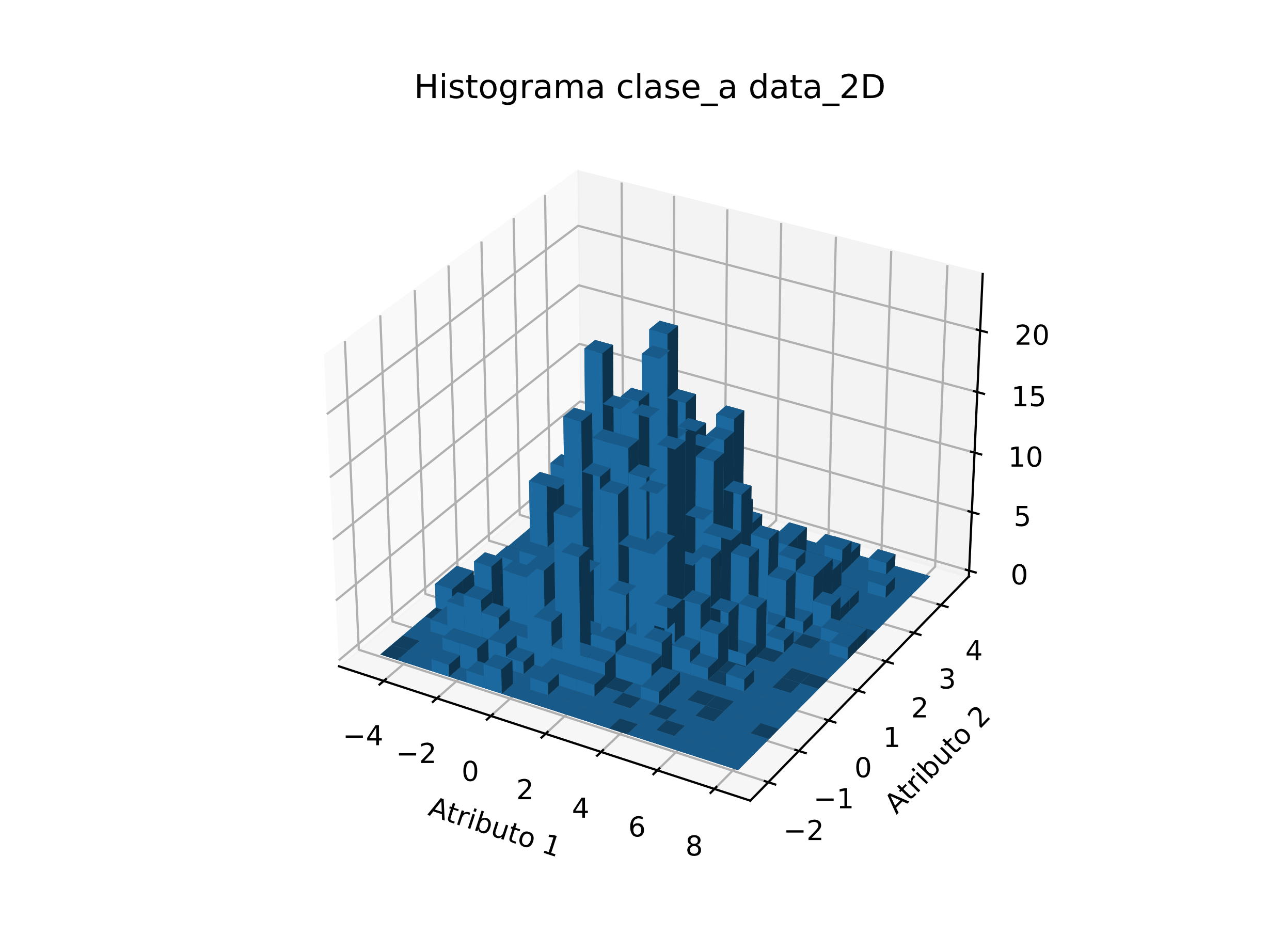


Figura 6. Histograma clase a data\_2D.

En el histograma realizado para la clase a, se puede observar que la mayor frecuencia de datos se encuentra entorno al valor central de los datos muy cercano a (1,1).

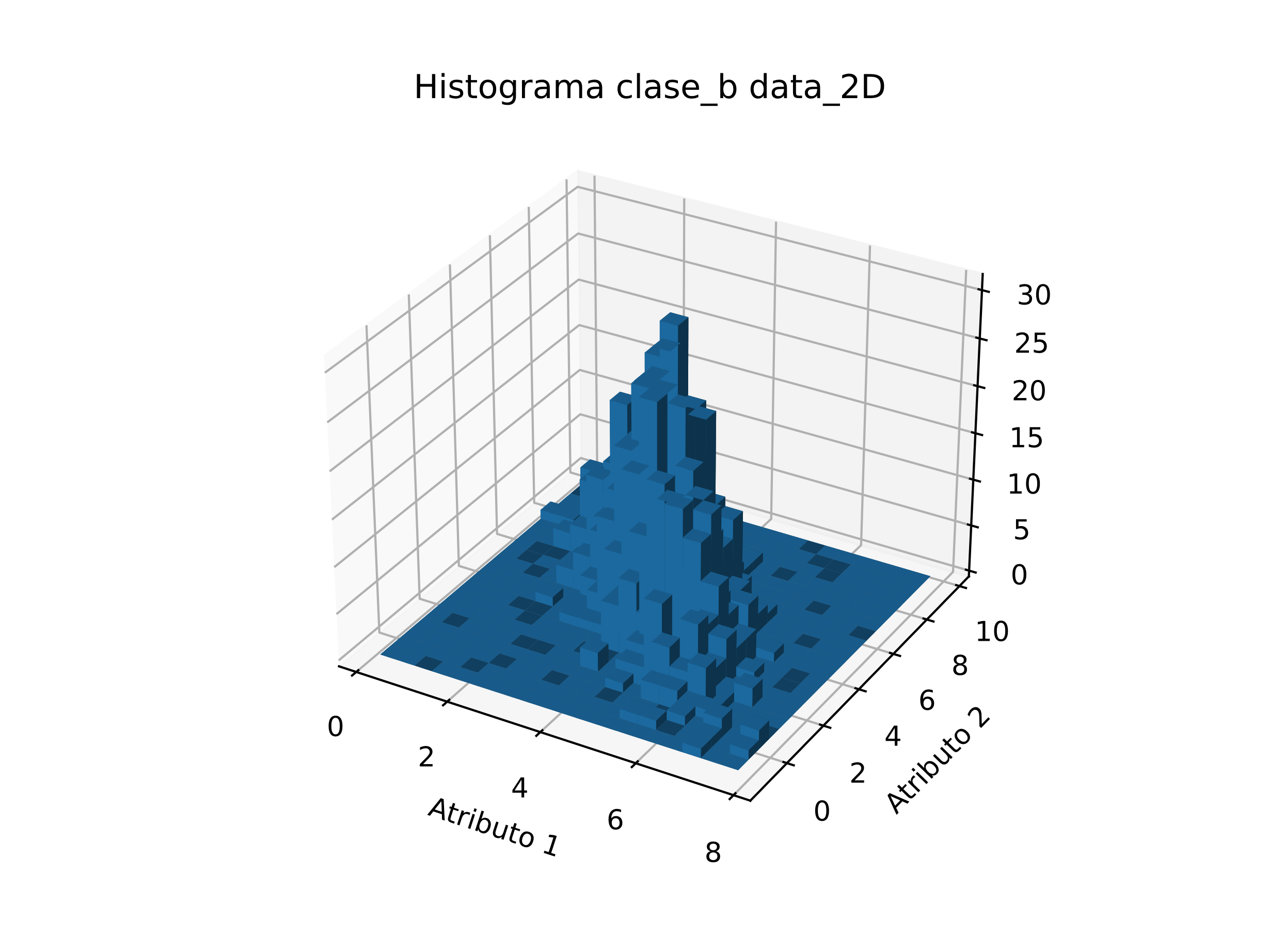


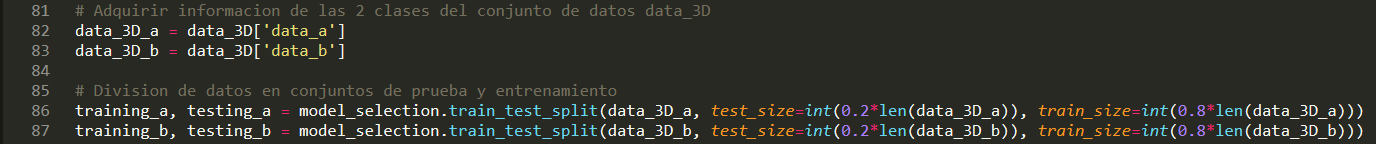
Figura 7. Histograma clase b data\_2D.

En el histograma realizado para la clase b, se puede observar que la mayor frecuencia de datos se encuentra entorno al valor central de los datos muy cercano a (4,4).

# Conjunto de datos data\_3D

## Divida los datos aleatoriamente en conjunto de prueba (20 %) y de entrenamiento (80 %).

La división de datos se realizó adquiriendo inicialmente los datos de cada una de las clases para el conjunto data\_3D, seguidamente se utilizó la función train\_test\_split() del módulo sklearn como se muestra en la imagen:



En este, para el parámetro test\_size, se encontró el 20% del número total de datos de cada clase y para el parámetro train\_size se encontró el 80% del número total de datos de cada clase.

## Visualice el conjunto de entrenamiento con un color para cada clase.

Para graficar el conjunto de entrenamiento de cada clase, se utilizó la función scatter3D pasando como parámetro las columnas de las variables de entrenamiento creadas para cada clase como se muestra en el siguiente código:

Text

Description automatically generated

La gráfica obtenida es la siguiente:

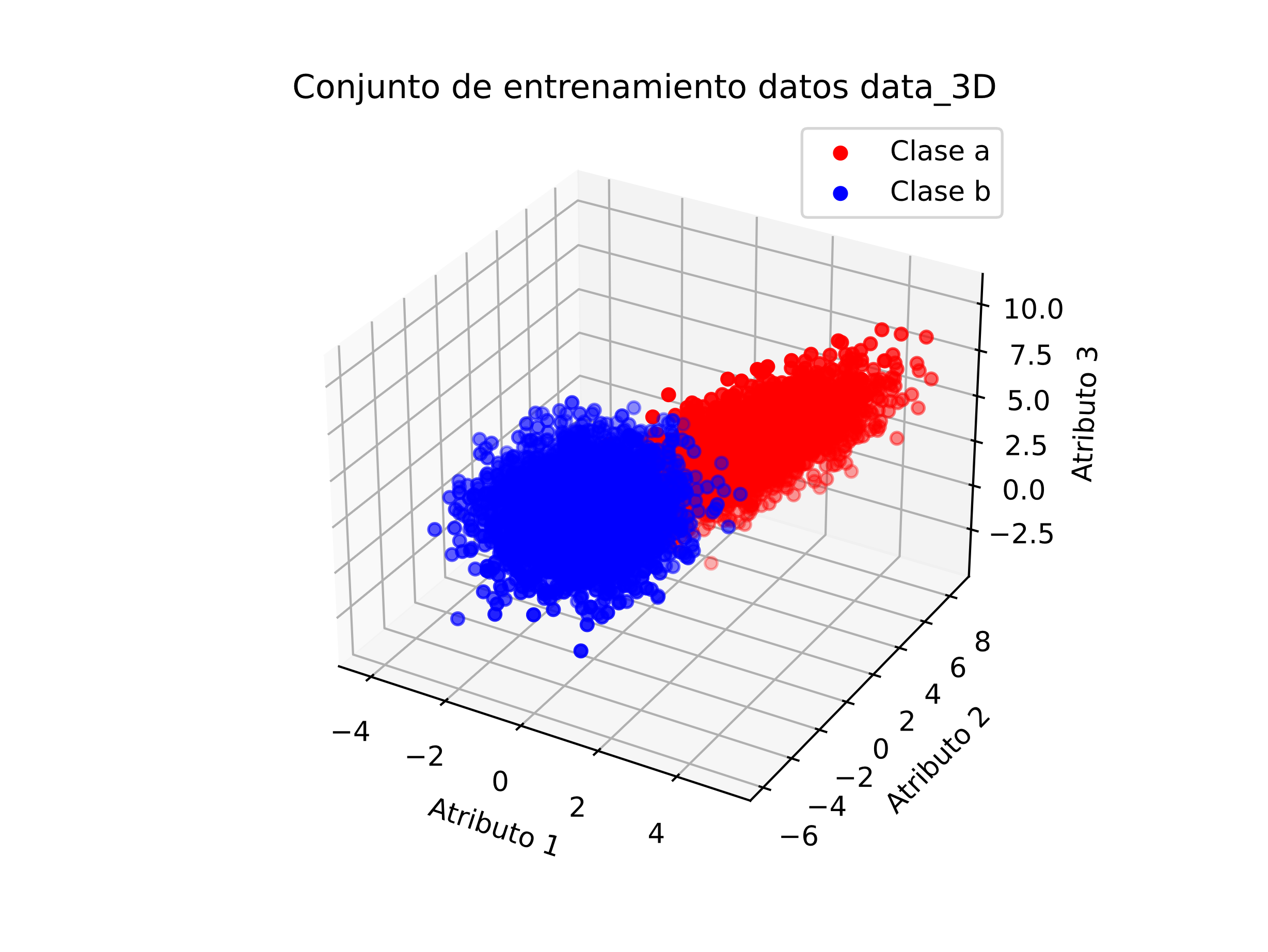


Figura 8. Conjunto de entramiento data\_3D.

## Implemente un clasificador Bayesiano Gaussiano, y:

1. Estime la función de verosimilitud de cada clase:

Inicialmente para la implantación de la función de verosimilitud presentada en la siguiente ecuación:

Se encontró el número de atributos y las probabilidades a priori de cada una de las clases, en este caso, asumiendo que estaban distribuidas homogéneamente:

Text

Description automatically generated

Posteriormente, se encontraron los valores centrales y las matrices de covarianza para cada una de las clases.

Text

Description automatically generated

Al tener 3 atributos se tiene un vector con 3 valores de media, uno por cada atributo.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 9. Vector medias para clase a.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 10. Vector medias para clase a.

Adicionalmente, se tienen matrices de covarianza de 3x3, debido al número de atributos:

A screenshot of a computer

Description automatically generated with medium confidence

Figura 11. Matriz de covarianza clase a.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 12. Matriz de covarianza clase b.

De las matrices de covarianza, se puede observar que en caso de la clase a, se tiene una dependencia entre atributos, mientras que para el caso de la clase b, se podría decir existe una dependencia mínima entre estos.

Una vez encontradas cada una de las constantes necesarias para encontrar la función, se creó una matriz de prueba que contenía los conjuntos de prueba de la clase a y b, se creó un vector con las etiquetas de clasificación reales para posteriormente calcular el error de la clasificación y finalmente un vector para almacenar la salida de la clasificación a realizar:

Text

Description automatically generated

Finalmente, se implementó la función, asignando un 0 para la clase a y un 1 para la clase b como se muestra en el siguiente código:

A screenshot of a computer

Description automatically generated

1. Visualice la clasificación realizada sobre el conjunto de prueba:

Para la visualización del conjunto de prueba, inicialmente, se crearon 2 nuevas matrices para cada una de las clases en base a la clasificación realizada y seguidamente se graficó el conjunto haciendo uso de la función scatter3D():

Text

Description automatically generated

La gráfica obtenida es la siguiente:

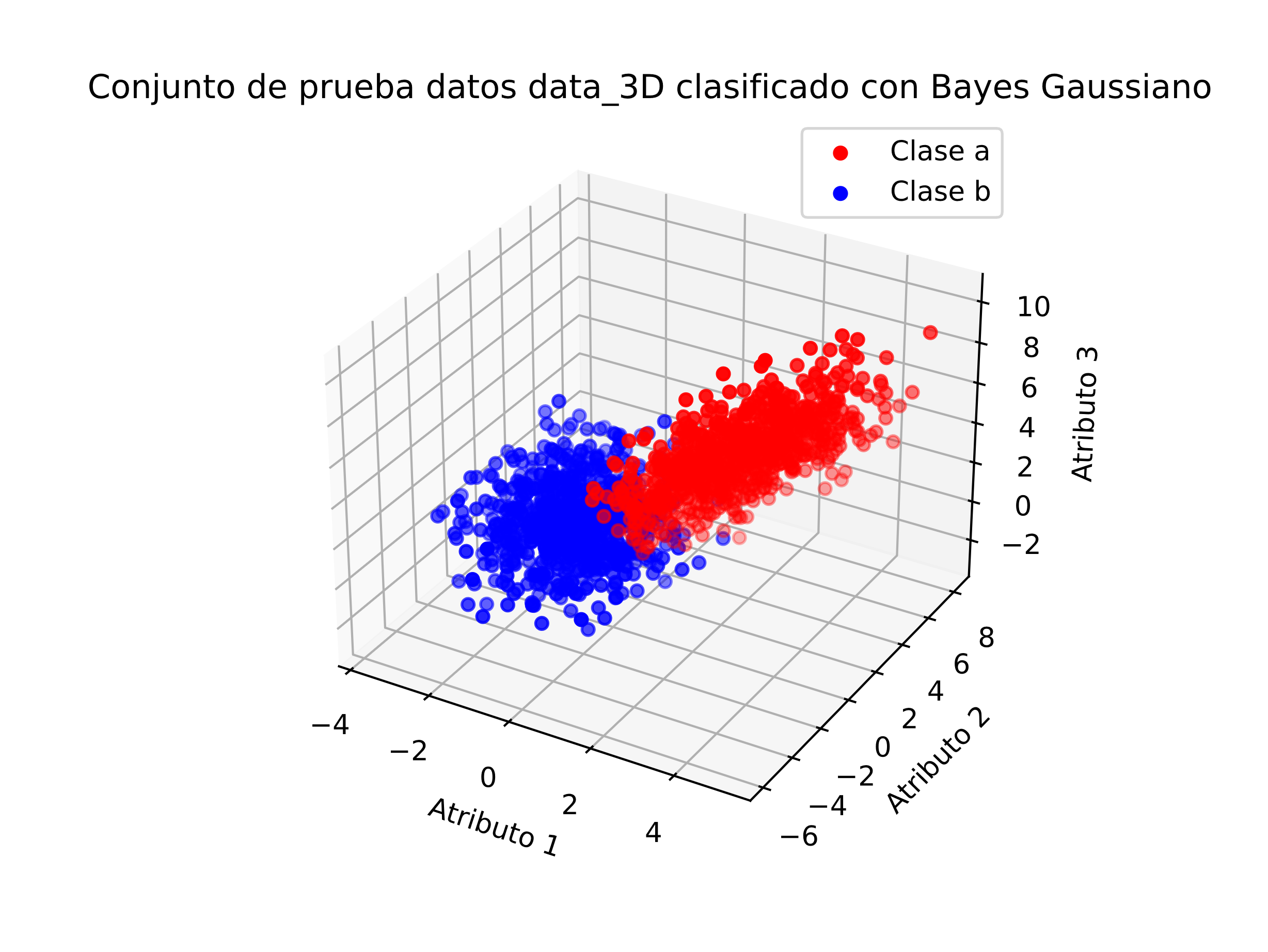


Figura 13. Conjunto de prueba clasificado con Bayes Gaussiano.

1. Determine el error de clasificación sobre el conjunto de prueba:

Se determinó el error de clasificación, encontrando la cantidad de clasificaciones mal realizadas dividido el número de elementos clasificados.

Text

Description automatically generated

Los errores porcentuales obtenidos en 3 ejecuciones del programa fueron:



## Implemente un clasificador Bayesiano Gaussiano Naive, y:

1. Estime la función de verosimilitud de cada clase:

En este caso, la función implementada en se presenta en la siguiente ecuación:

Se encontró el número de atributos y las probabilidades a priori de cada una de las clases, en este caso, asumiendo que estaban distribuidas homogéneamente. Estas son las mismas que en el caso anterior. Adicionalmente, se encontraron las desviaciones estándar para cada una de las clases.

Text

Description automatically generated

Al tener 3 atributos, se obtiene 1 desviación estándar por cada uno de los atributos:

A screenshot of a computer

Description automatically generated

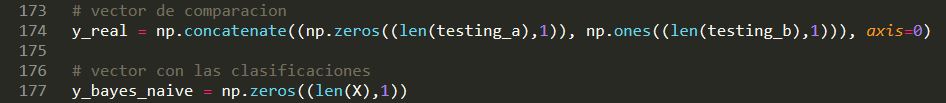
Figura 14. Desviaciones estandar clase a.

A screenshot of a computer

Description automatically generated with medium confidence

Figura 15. Desviaciones estandar clase b.

Se creó un vector con las etiquetas de clasificación reales para posteriormente calcular el error de la clasificación y finalmente un vector para almacenar la salida de la clasificación a realizar:



Finalmente, se implementó la función, asignando un 0 para la clase a y un 1 para la clase b como se muestra en el siguiente código:

Text

Description automatically generated

1. Visualice la clasificación realizada sobre el conjunto de prueba:

Para la visualización del conjunto de prueba, inicialmente, se crearon 2 nuevas matrices para cada una de las clases en base a la clasificación realizada y seguidamente se graficó el conjunto haciendo uso de la función scatter3D():

Text

Description automatically generated

La gráfica obtenida es la siguiente:

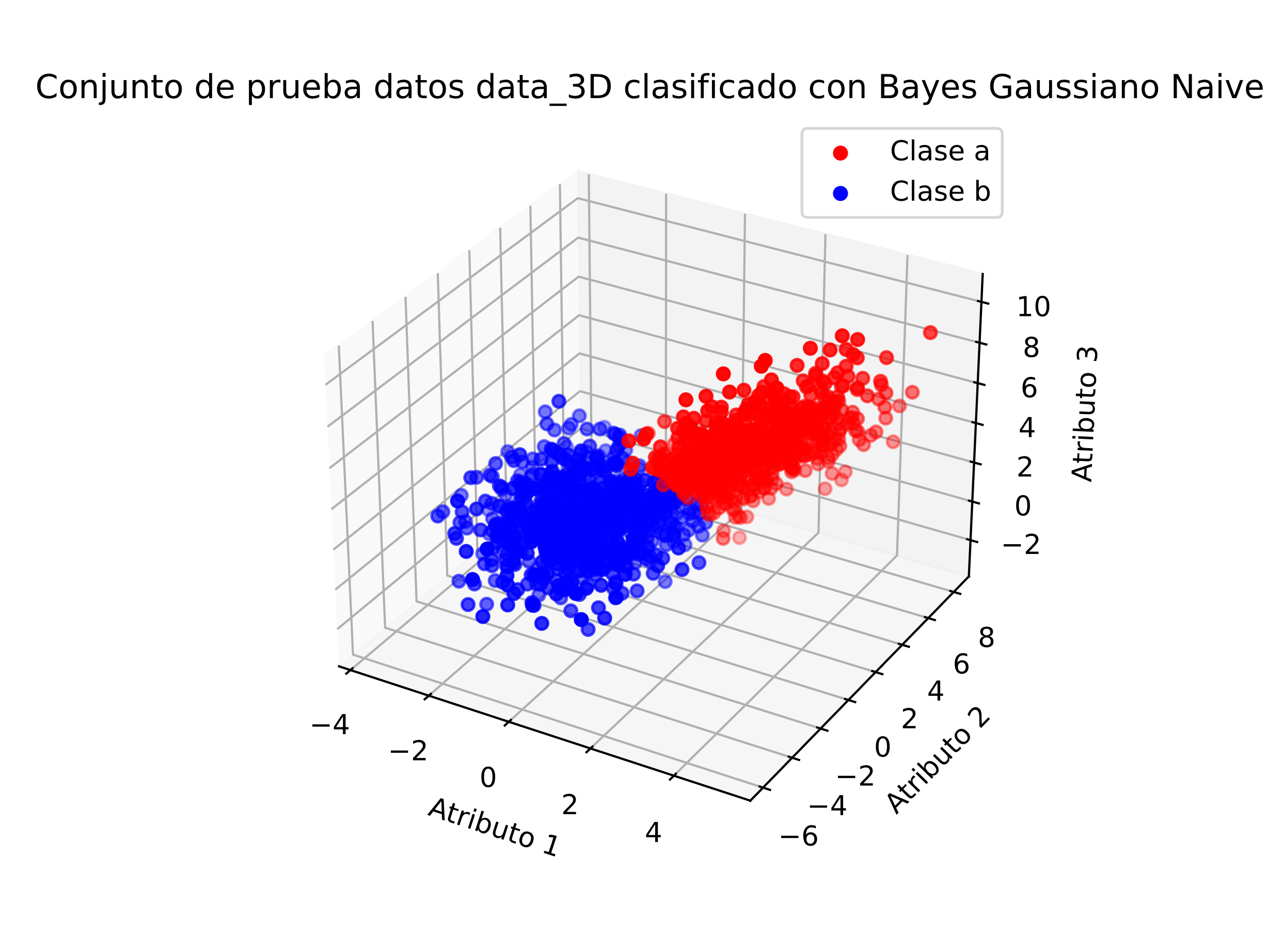
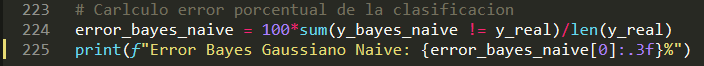


Figura 16. Conjunto de prueba clasificado con Bayes Gaussiano Naive.

1. Determine el error de clasificación sobre el conjunto de prueba:

Se determinó el error de clasificación, encontrando la cantidad de clasificaciones mal realizadas dividido el número de elementos clasificados.



Los errores porcentuales obtenidos en 3 ejecuciones del programa fueron:







# Concluya sobre los resultados obtenidos de los clasificadores 2.c y 2.d

Partiendo de los errores de clasificaciones encontrados, se puede decir que el clasificador Bayesiano Gaussiano funciona correctamente teniendo errores menores al 5%.

En el caso del clasificador a Bayesiano Gaussiano Naive, se siguen obteniendo buenos resultandos con errores menores al 10%. Este incremento en el error se debe a la aproximación que se hace al decir que los atributos son independientes entre sí, ya que como se pudo observar en la matriz de covarianza de la clase a, estos sí presentan una dependencia.

En general, se puede decir que ambos clasificadores tienen un buen rendimiento y al observar los resultados de clasificación.