# 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

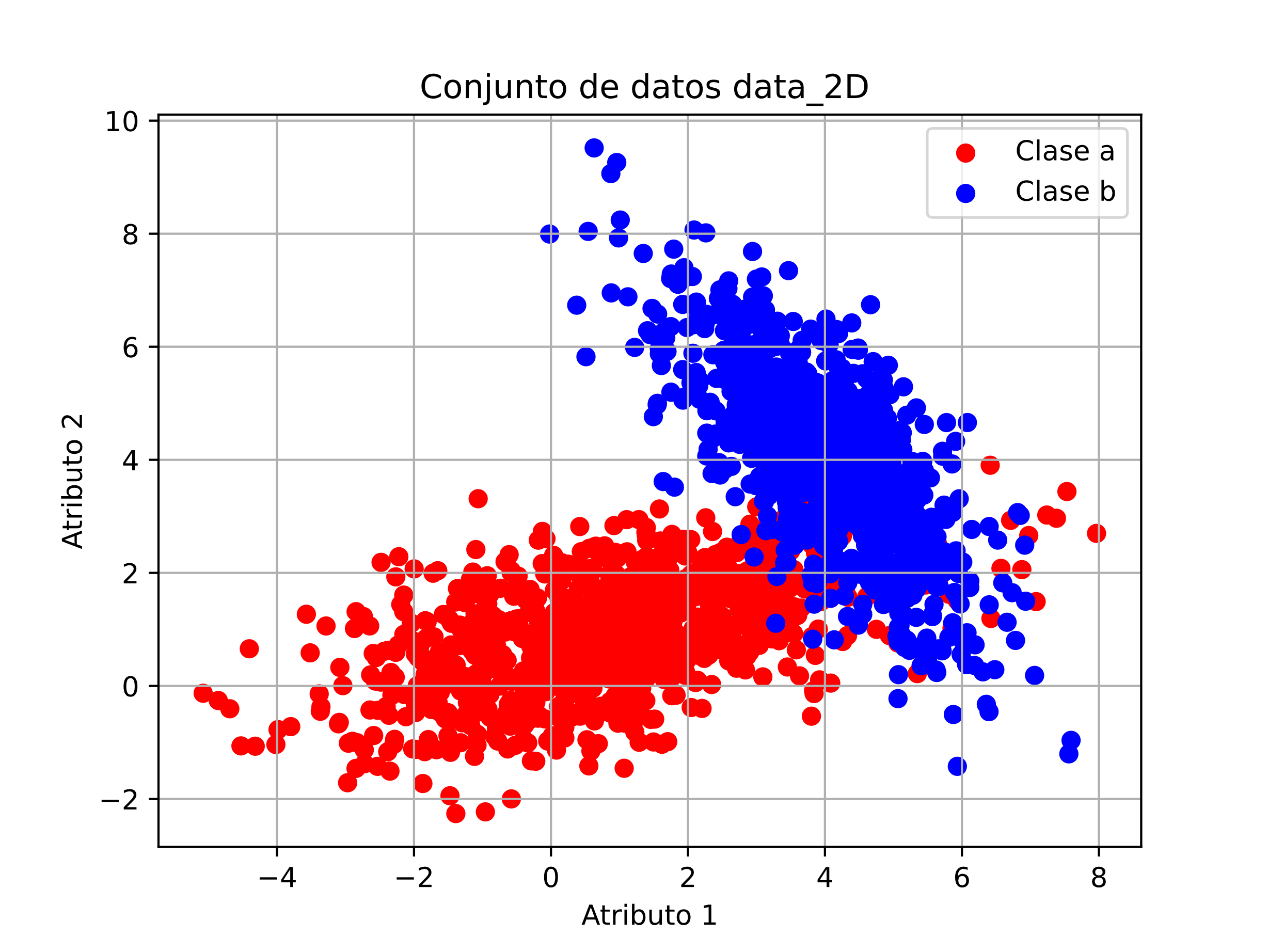
Una vez separados los conjuntos de datos, se extrajeron los valores de las clases a y b y se realizó la gráfica de estas dos clases usando el módulo matplotlib

Text

Description automatically generated

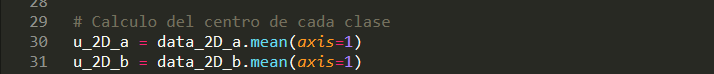
En este caso, se eligió el color rojo para la clase a y el color azul para la clase b para poder distinguirlos adecuadamente en la gráfica

La gráfica obtenida se presenta en la siguiente imagen:



## 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

A screenshot of a computer

Description automatically generated with medium confidence

## 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

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

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.

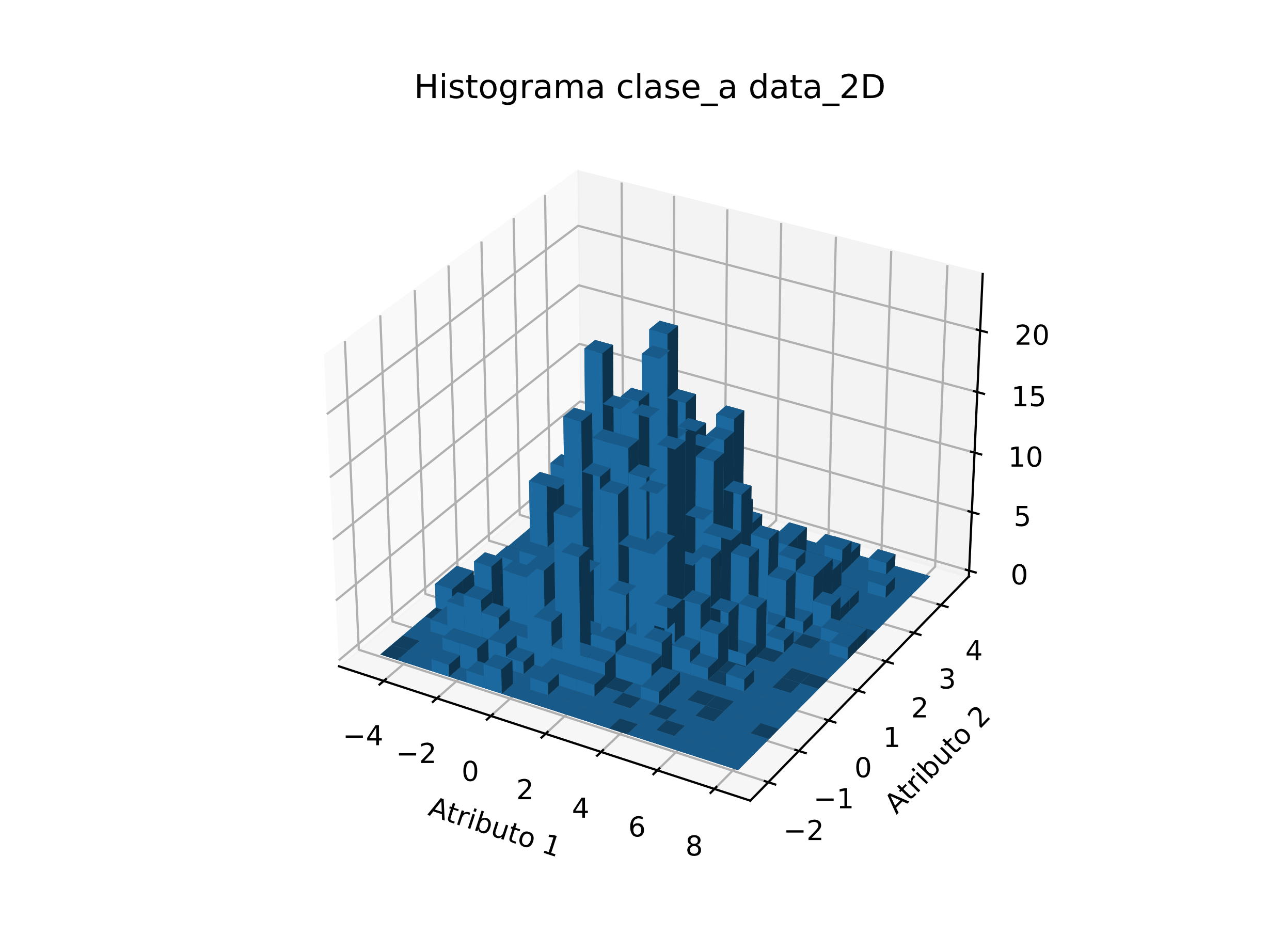
## Determine y visualice el histograma de los datos.

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:

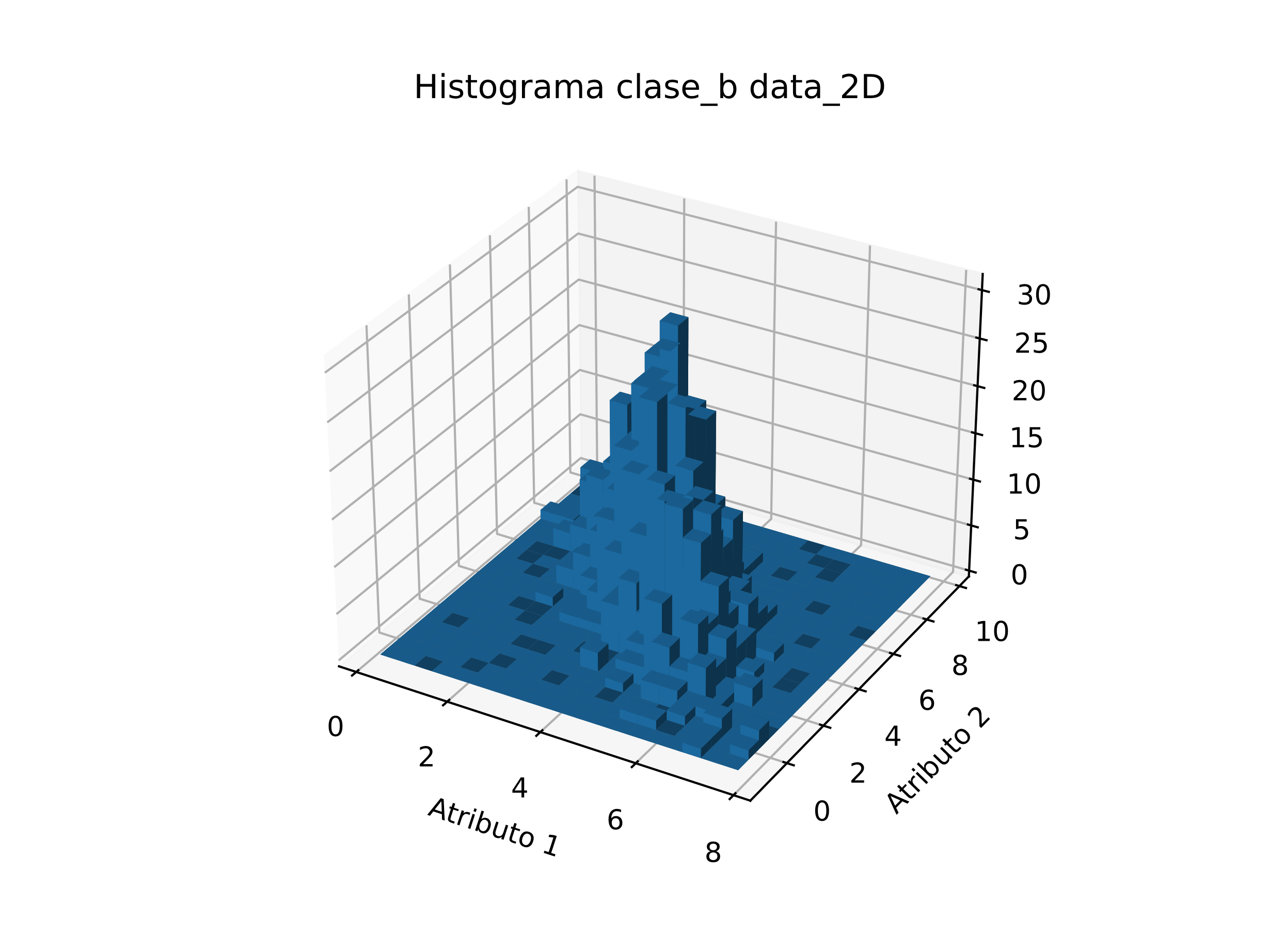
Text

Description automatically generated

Los histogramas obtenidos se presentan en las siguientes imágenes:



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).

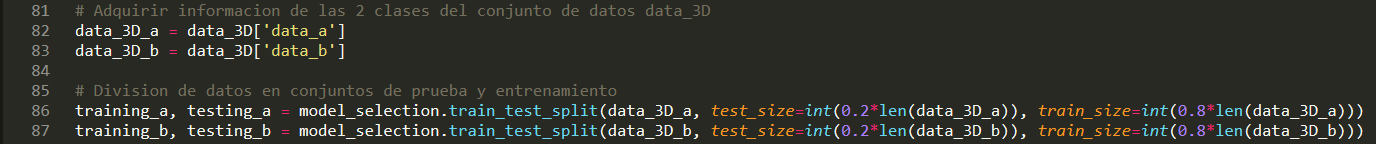


En el histograma realizado para la clase b, se puede observar que la mayor frecuencia de datos se encuentra entorno al valor central de los datos muy cercano a (4,4).

# 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 numero total de datos de cada clase y para el parámetro train\_size se encontró el 80% del número total de datos de cada clase.

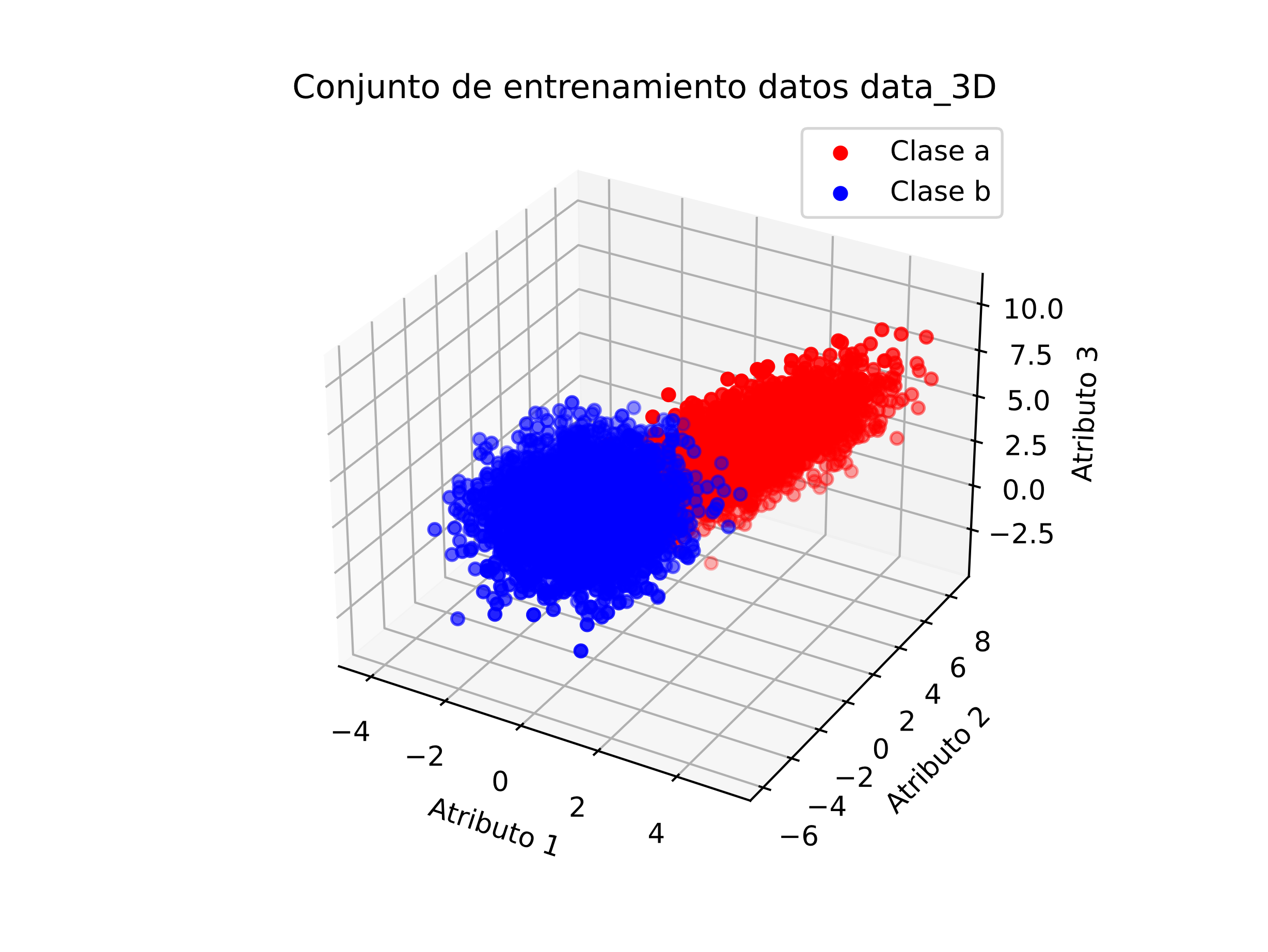
## Visualice el conjunto de entrenamiento con un color para cada clase.

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:



## 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 tienen matrices de covarianza de 3x3 y un vector de 3 elementos para los valores centrales:

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

\A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

Una vez encontradas cada una

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

## Implemente un clasificador Bayesiano Gaussiano Naive, y:

1. Estime la función de verosimilitud de cada clase:
2. Visualice la clasificación realizada sobre el conjunto de prueba:
3. Determine el error de clasificación sobre el conjunto de prueba:

# Units