PRUEBA 1

INTEGRANTES:
Freddy Abad, Tania Landivar, Elvis Calle.

1. **¿Qué clases existen?**
   A, B y C

2. **Indique si los datos son linealmente separables entre las clases del dataset (D) o si se identifican ciertos outliers que indican que no son totalmente linealmente separables.**
   Sí, los datos son linealmente separables. Y no, no existen outliers, los datos están bien agrupados. Esto se puede apreciar claramente en las Gráficas 3, 4, 5, 6 respectivamente al verlos en un eje coordenado de tres ejes y sus respectivas proyecciones en dos ejes (ejes: xy, xz, yz).

3. **En el caso de que las clases no sean linealmente separables. ¿Podría usted recomendar utilizar un perceptrón para separar a las clases (...)?**
   Si y No, no por la siguiente razone:
   El algoritmo del perceptrón separa solamente de forma lineal. Si utilizamos el mismo para datos no separables linealmente los valores de los pesos en el algoritmo no convergerían.
   Si por las siguiente razon:
   Citando a Raschka en su libro “Python Machine Learning”, en la pagina 32 menciona que: *“la convergencia es uno de los mayores problemas del perceptron. Frank Rosenblatt probó matemáticamente que el la regla de aprendizaje de perceptron converge si las dos clases pueden ser separados por un hiperplano lineal. Sin embargo, si las clases no pueden separarse perfectamente por un límite de decisión tan lineal, los pesos nunca dejarán de actualizarse a menos que establezcamos un máximo número de épocas.”*
   Asi que si los datos nos son separables linealmente, se puede usar perceptrón dando un rango alto de número de iteraciones para que grafique una función lineal, pero el nivel de error no dependeria ante las iteraciones ya que se mantendria actulizando, sin dar un error minimo.

4. **Si es posible utilizar un perceptrón o más perceptrones para separar los datos, indique cuál será su estrategia para realizar eso.**
   La estrategia consistiría en:
   1. Para el uso de un plano de dos ejes, la técnica consistiría en separar los datos en dos grupos: en el primer grupo estaría solamente una clase y en el segundo grupo las otras dos, tomando a todas estas como una sola entrada, lo que se busca es que el perceptrón separe primero estos dos grupos, luego se realiza las agrupaciones pendientes para que el perceptrón proceda a separar.
   Las agrupaciones serán:
   - (Clase A = -1) y ((Clase B y C) = 1)
   - (Clase B = -1) y ((Clase A y C) = 1)

   Si estas dos agrupaciones quedan separadas por un perceptrón, entonces por defecto va a quedar separadas todas las clases.

Usando esta tecnica el entrenamiento de los datos seria mas facil, para el nivel del curso, es decir, se aprovecha la tactica para hacerla mas facil de entrenar, obteniendose como resultado los dos perceptrones.

5. **Explique todo lo que hizo con los datos antes de entrenar su modelo (preprocesamiento).**

Primero se separó por listas cada uno de los patrones de entrenamiento (x1, x2, x3), para así graficar en planos cartesiano de dos ejes para tener proyecciones en el plano x1x2 -xy-, x1x3 -xz-, x2x3 -yz-.

Segundo, para graficar las clase A, B y C, hicimos A = 1, B = -1 y C = 2, haciendo más fácil las comparaciones al momento de graficar y gracias a esta gráfica se pudo determinar que no existía ningún outliers, y que los datos son linealmente separables, Gráfica 1.

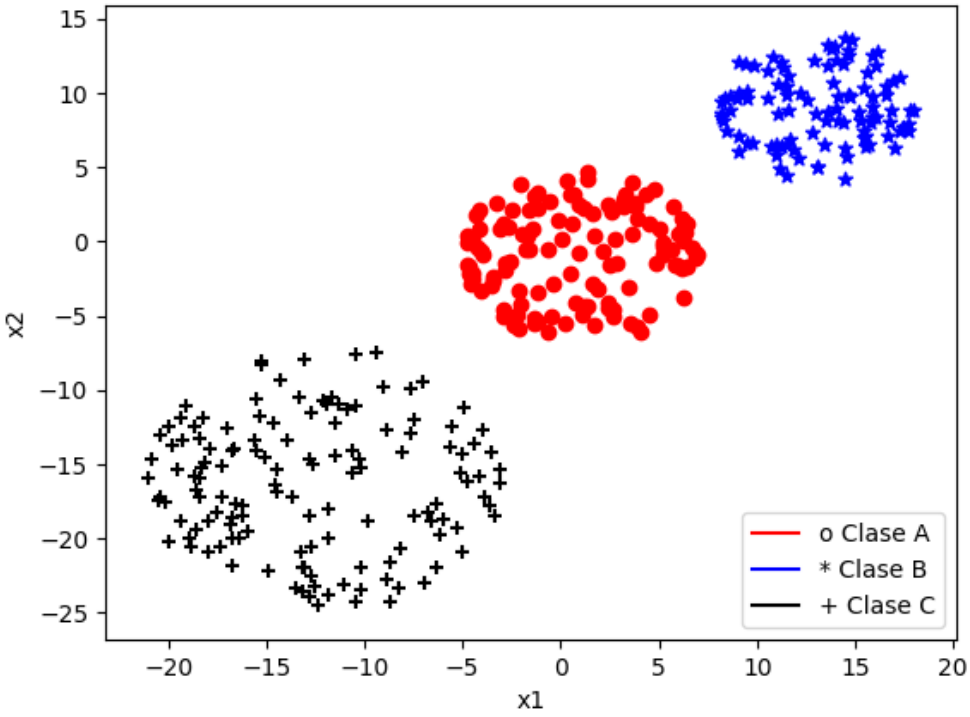
Tercero, se entrenó el perceptrón con las coordenadas (las proyecciones en el plano, ya sea x1x2, x1x3, x2x3) con las debidas etiquetas, permitiendo clasificar los datos correctamente en un plano de dos ejes.

```
ppn = Perceptron(eta=0.1, n_iter=15)
ppn.fit(coordenadas, etiquetas)
plt.plot(range(1, len(ppn.errors_) + 1), ppn.errors_, marker='o')
plot_decision_regions(coordenadas, etiquetas, classifier=ppn)
```

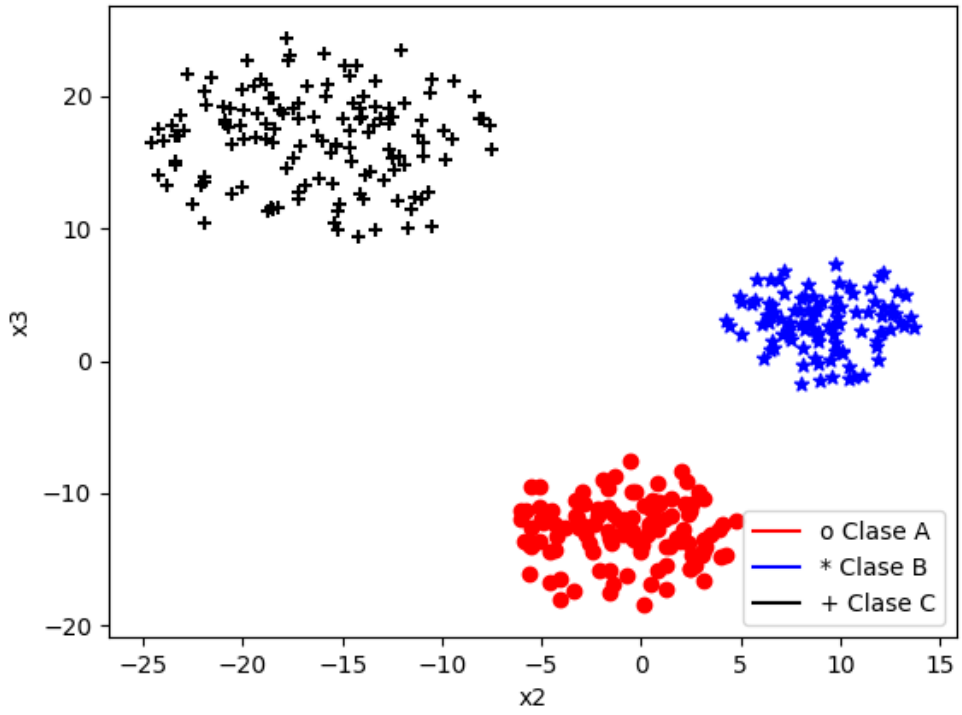
Se repitio el proceso con las otras proyecciones en el plano, y obteniendo los perceptrones en cada caso.

Tercero, se uso librerias graficas que permitieran graficar los datos en un plano xyz. Las librerias usadas son mencionadas un poco despues de las graficas.

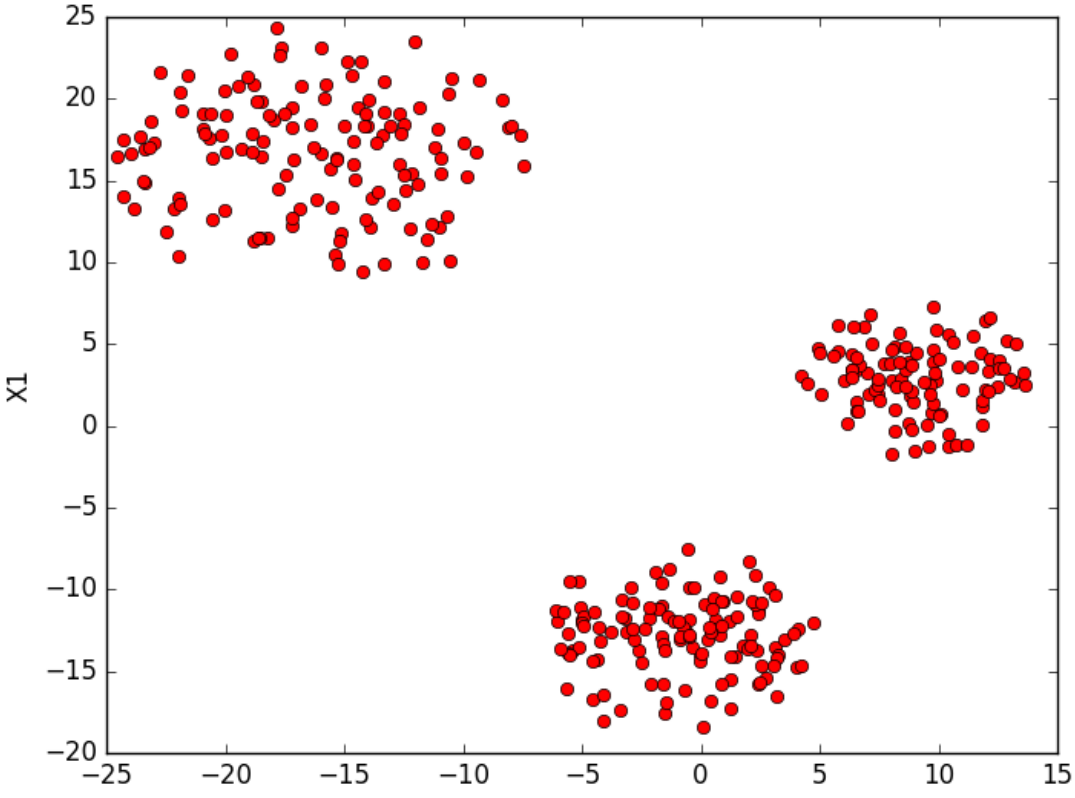
6. **Recurso gráfico**



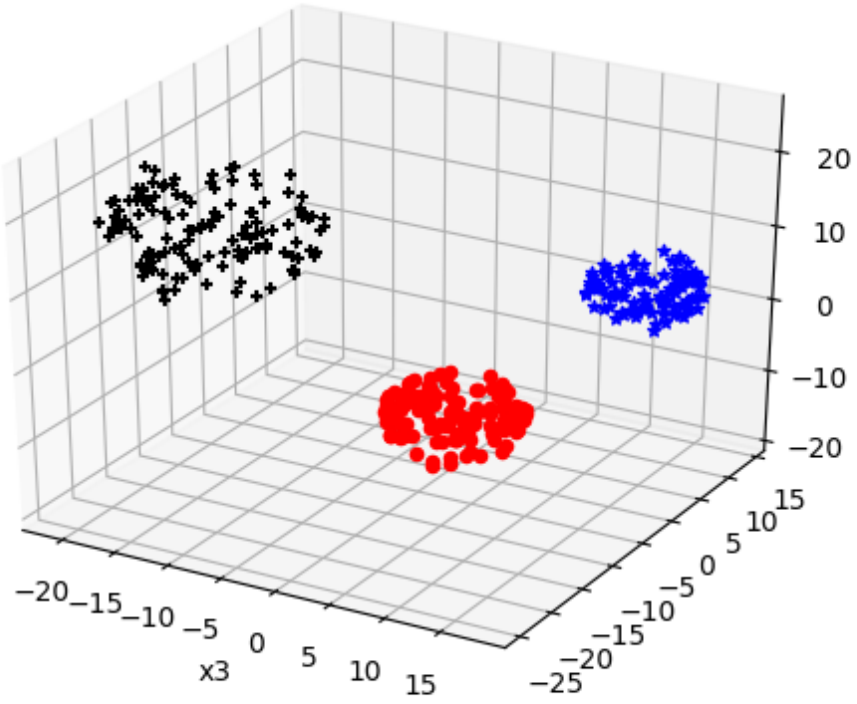
Gráfica 1



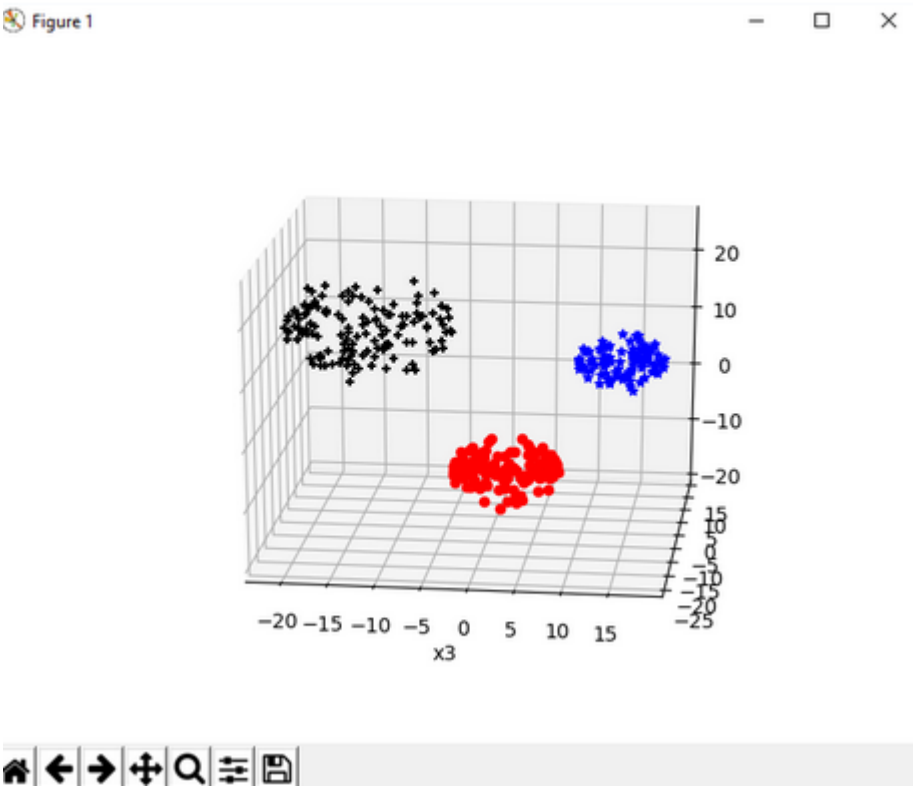
Gráfica 2



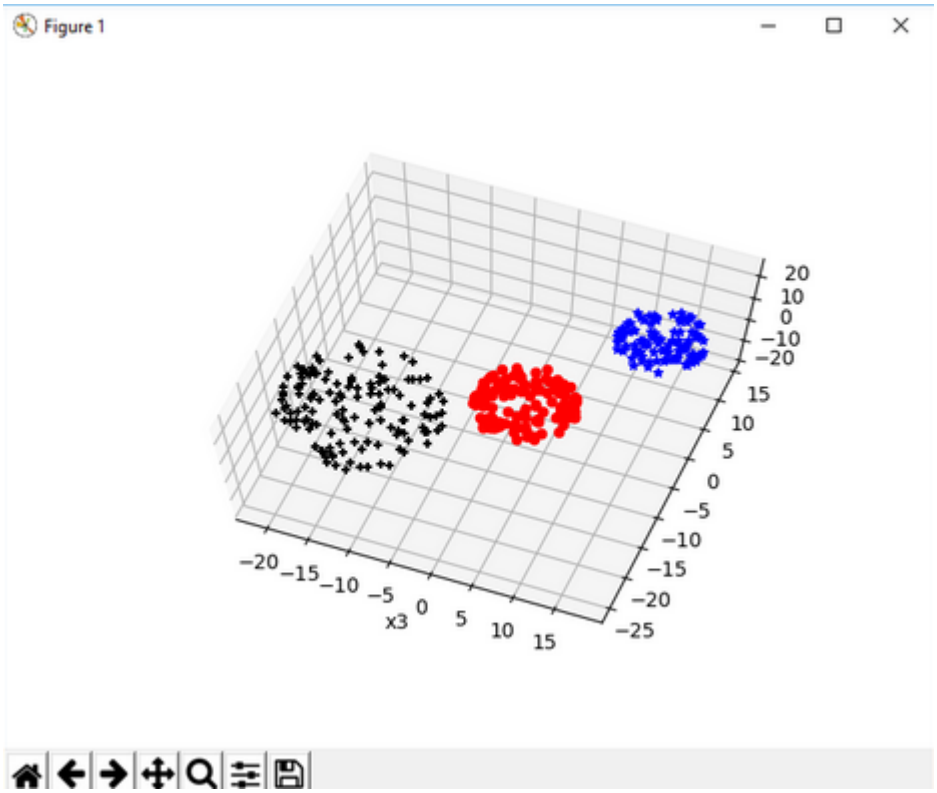
Grafica 3



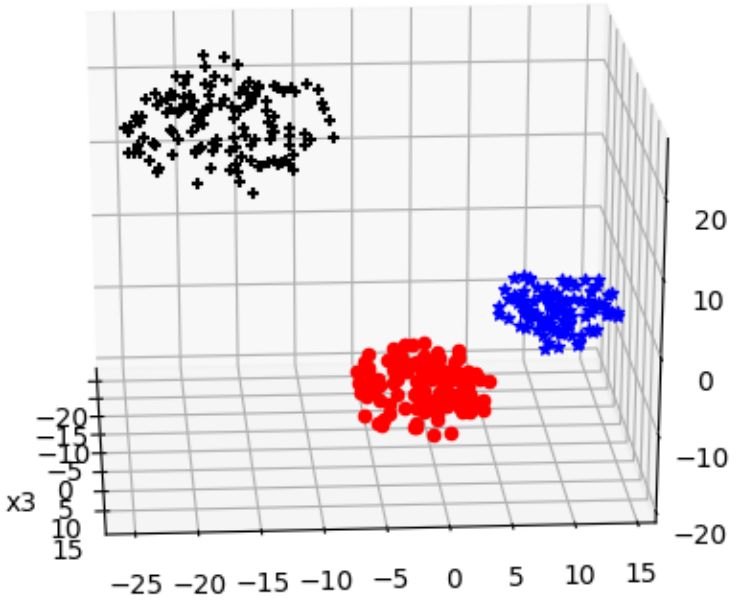
Gráfica 4



Grafica 5



Grafica 6



Grafica 7

Para la gráfica en 3 dimensiones utilizamos las librerías:

from matplotlib import pyplot
from mpl_toolkits.mplot3d import Axes3D

Entrenamiento del modelo

1. **¿Qué es exactamente una época?**
   Una época es cada iteración del algoritmo en las que haya un ajuste de las variables o pesos. Durante un época los pesos del algoritmo cambian de un estado A a un estado B luego de un determinado procesamiento.

2. **¿Cómo convergió el modelo en el entrenamiento con respecto al número de épocas?**
   Se uso como practicas dos tasas de aprendizaje, uno de 0.1 y otro de 0.6 tratado de dar una variedad que enriqueciera la practica, y asi obtener mejores conclusiones, se obtuvo que ante los 359 datos del Dataset de nuestro grupo se obtuvo una convergencia de 2 a 3 iteraciones a pesar de que los patrones de convergencia variaron 0.5 unidades una de otra.

3. **Si utilizó una tasa de aprendizaje, ¿qué tasa de aprendizaje utilizó en el entrenamiento?**
   Se uso tasas de aprendizaje de 0.1 y 0.6.

4. **¿Utilizó un número fijo de iteraciones máxima?**
   Como se menciono preguntas atrás, ya que los datos son linealmente divisibles, y las tasas de aprendizaje fueron de 0.1 y 0.6, con convergencia en 2, 3 epocas no fue necesario variar el numero de iteraciones, aunque en codigo fuente se puso 10 como numero de iteraciones.

5. **¿Utilizó un error relativo? Si sí, ¿cuál fue y por qué ese valor?**
   Si, se uso un error relase destivo de 0.0, porque para la cantidad de datos del dataset, para las circunstancias en las que se desenvolveria el algoritmo, el error relativo debia tender a 0, o ya ser un 0 absoluto.

6. **¿Utilizó tanto un número máximo de iteraciones y un error relativo?**
   Si y si

7. **¿Cuándo terminó el entrenamiento?**
   El entrenamiento termino al tener un maximo de iteraciones o actualizaciones ya que el proceso matematico (sacar un z :

$$z = w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{j=0}^{m} \boldsymbol{x}_j \boldsymbol{w}_j = \boldsymbol{w}^T \boldsymbol{x}$$

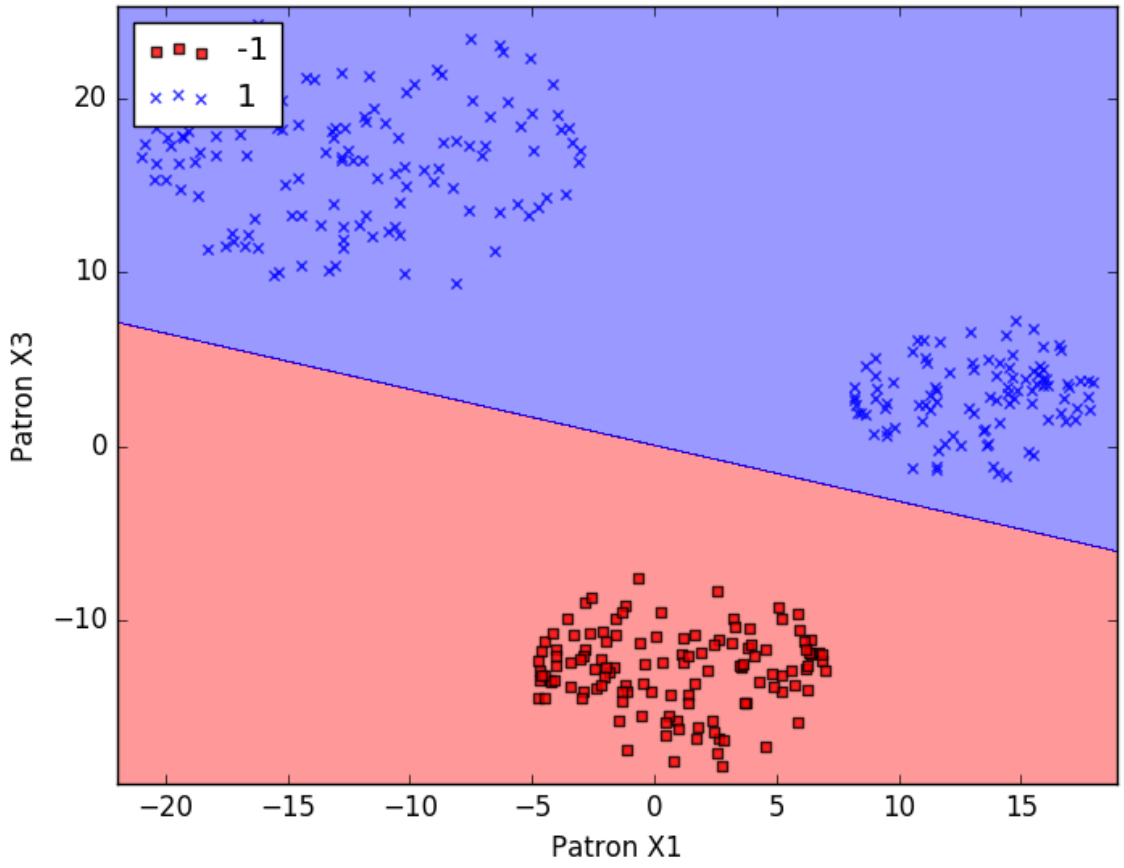
   para todos los datos) se dio hasta la ultima epoca.

8. **¿En el número máximo de iteraciones o cuando se llegó al error relativo mínimo?**
   En este caso no fue necesario, ya que el error relativo se lleno antes de la mitad de iteraciones impuestas, asi que el error relativo en casos como este no es directamente relacionado con el numero de iteraciones
9. **¿Esto le da una idea de si los datos son o no linealmente separables (discuta)?**
   Si, si no tuvieramos el chance de graficar en un plano cartesiano, esto nos ayudaria a ver que los datos son linealmente separables.

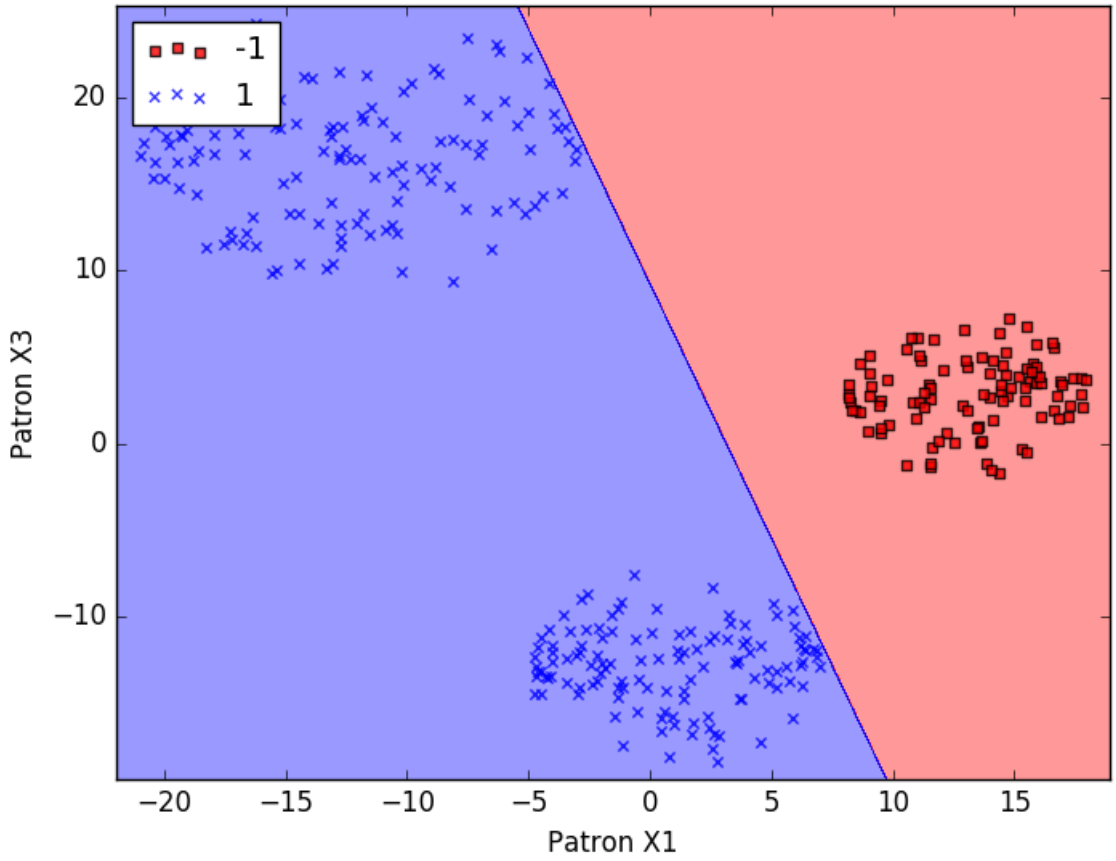
10. **Si su perceptrón fue entrenado pero los datos no son totalmente linealmente separables, ¿cuál es el porcentaje de error? ¿cómo fue evolucionando en cada época (discuta)?**
    En el caso de nuestro dataset, los datos fueron linealmente separables.

Clasificación con datos de prueba
Perceptrón para cada una de las agrupaciones.

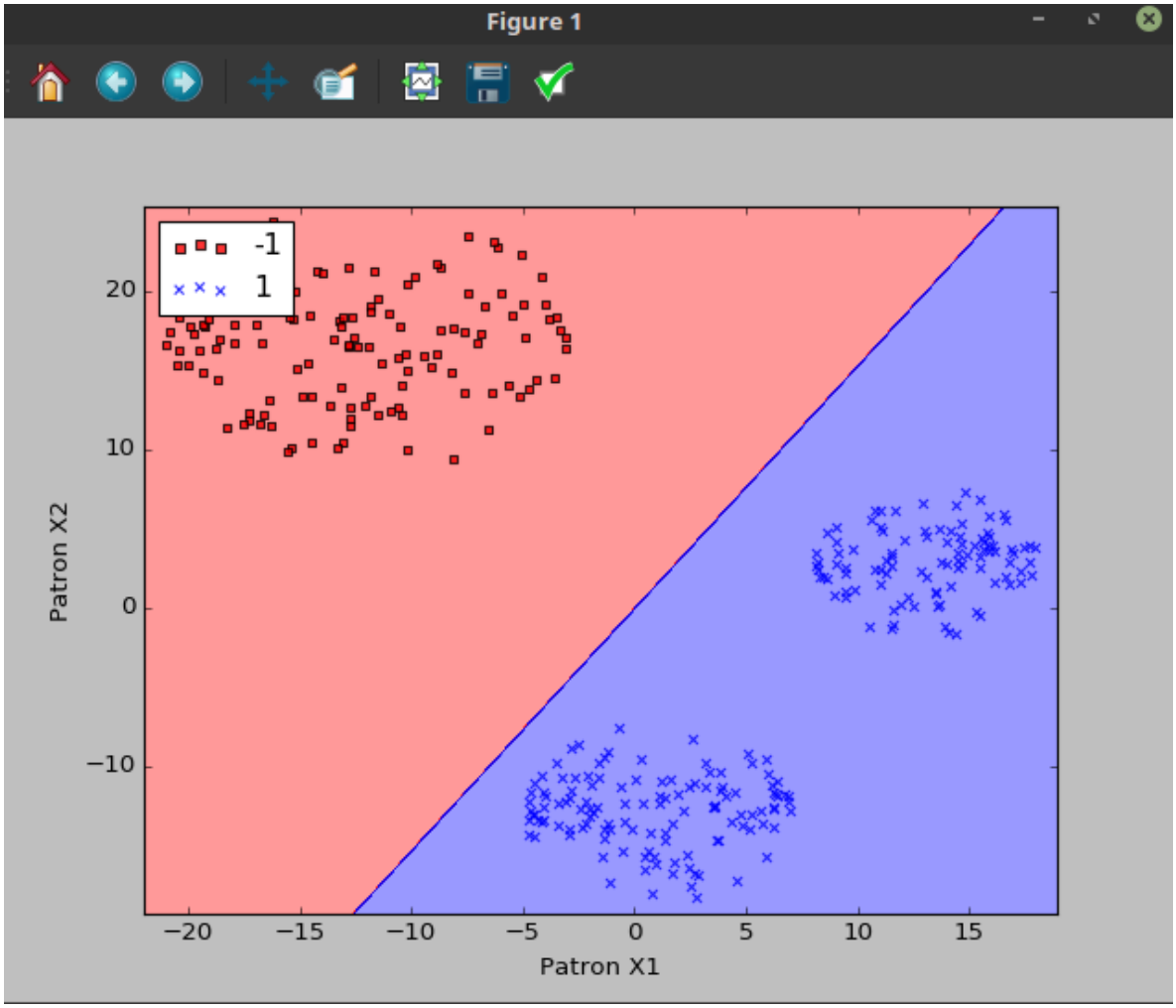
- (Clase A = -1) y ((Clase B y C) = 1)



- (Clase B = -1) y ((Clase A y C) = 1)

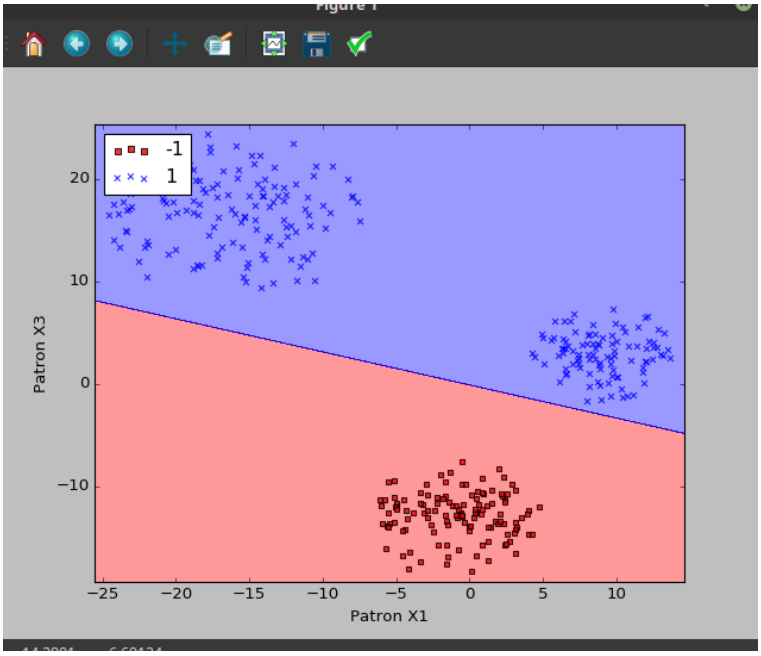


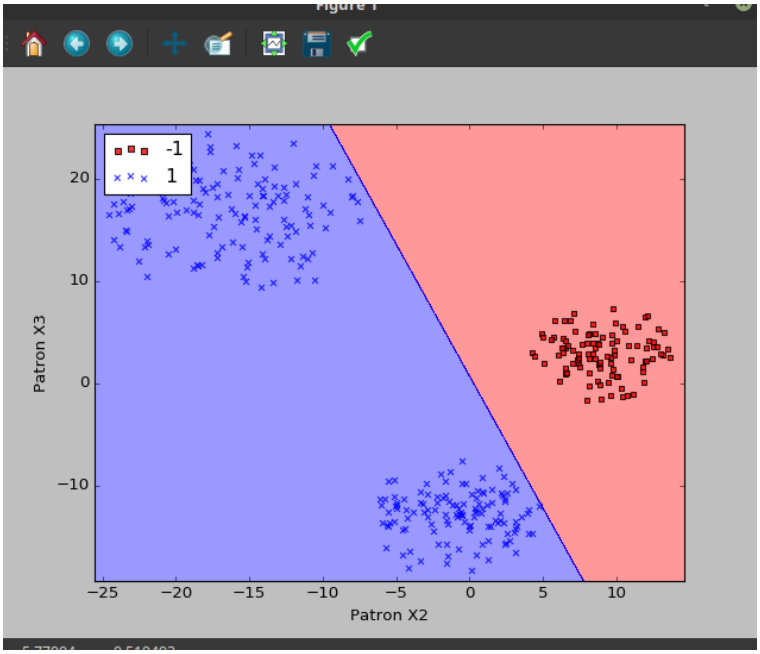
- (Clase A = -1) y ((Clase B y C) = -1)

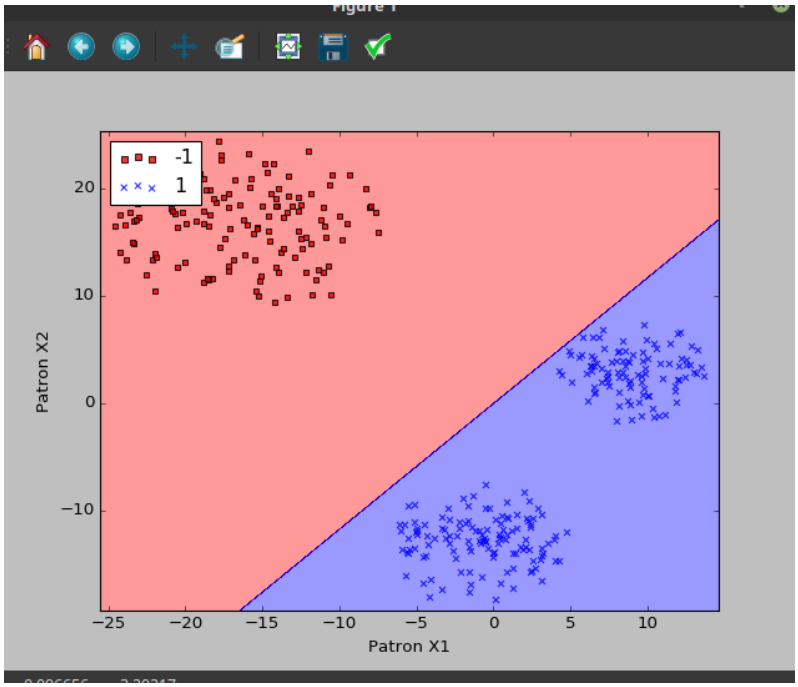


- (Clase C= -1) y ((Clase A y B) = -1)

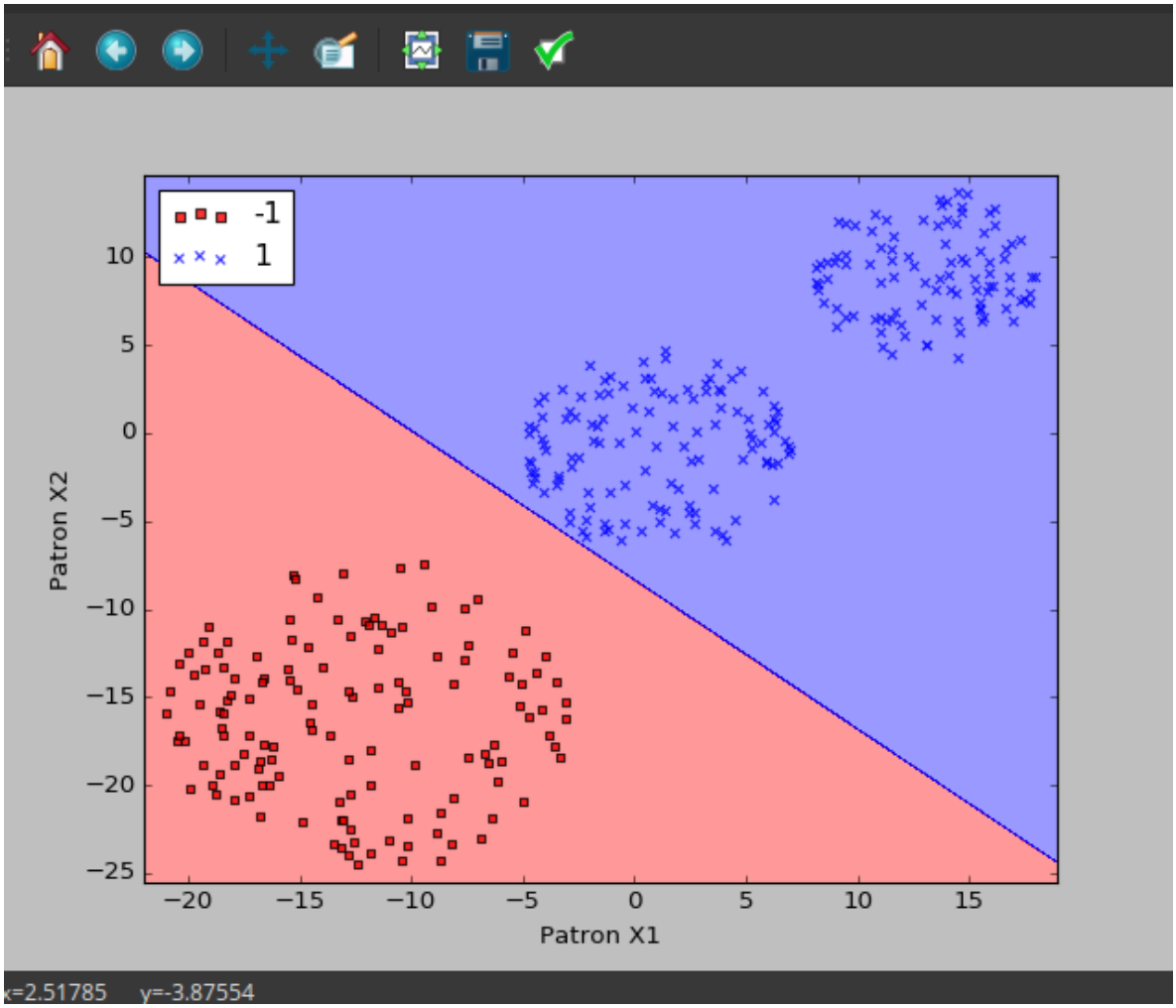
EJE YZ

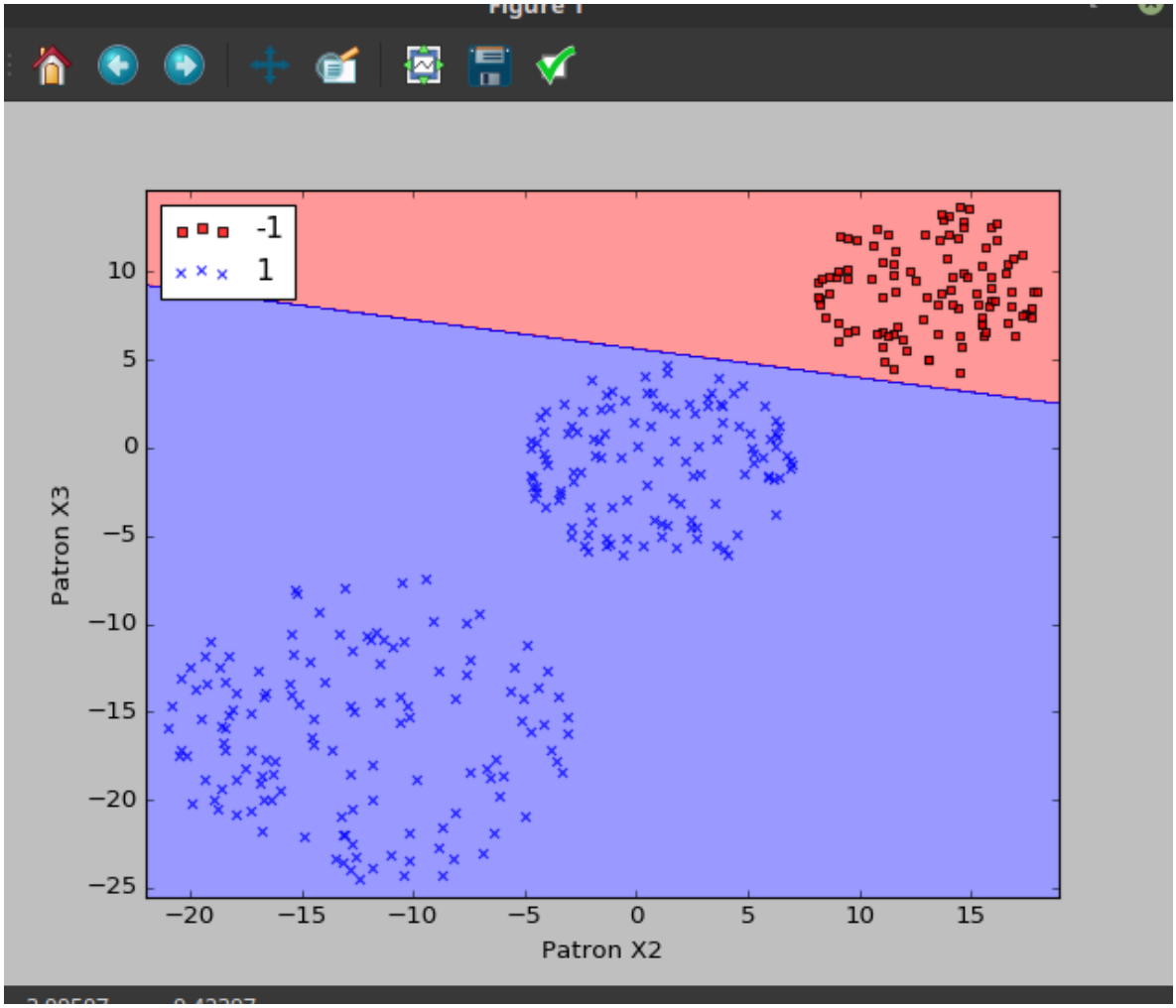

-1
1
20
10
0
−10
Patron X3
−25
−20
−15
−10
−5
0
5
10
Patron X1


-1
1
20
10
0
−10
Patron X3
−25
−20
−15
−10
−5
0
5
10
Patron X2

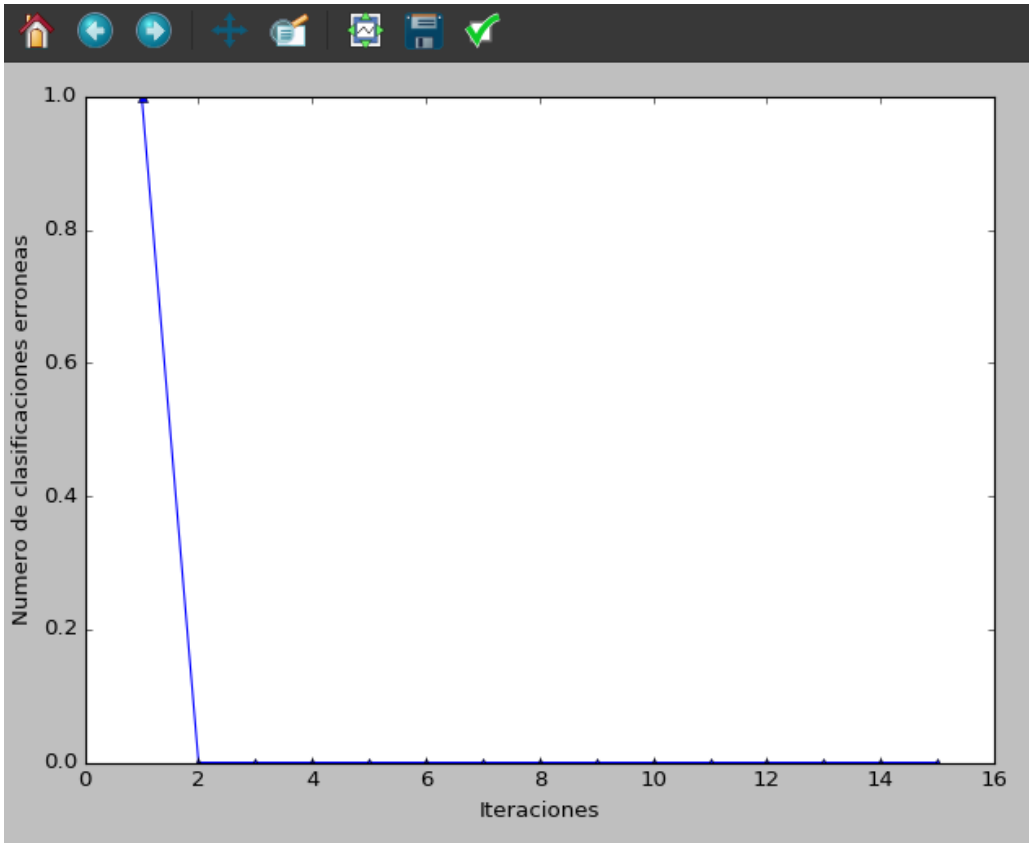

-1
1
20
10
0
−10
Patron X2
−25
−20
−15
−10
−5
0
5
10
Patron X1

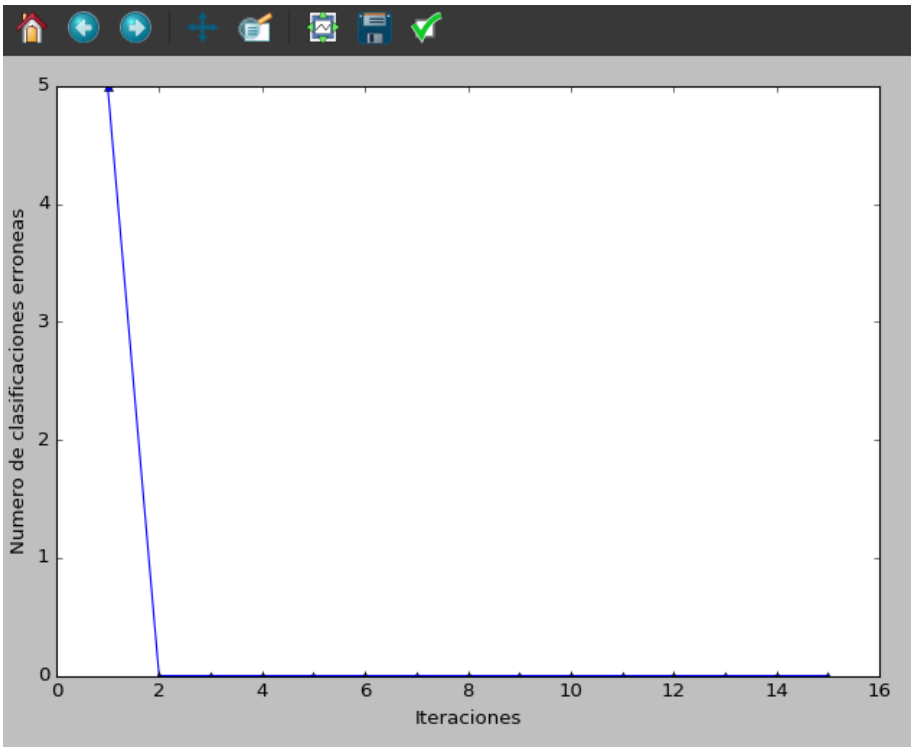
EJE X1 X2

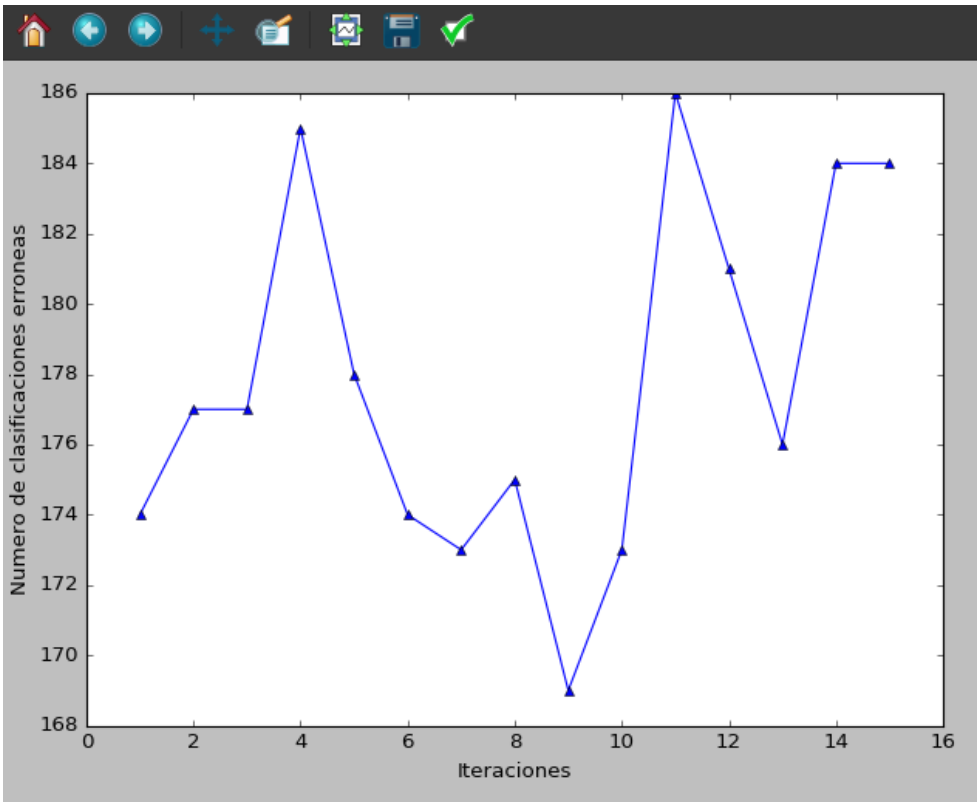

-1
1
10
5
0
−5
−10
−15
−20
−25
Patron X2
−20
−15
−10
−5
0
5
10
15
Patron X1
x=2.51785
y=-3.87554



Epocas vs Num Clasifiaciones Erroneas (en el orden de las graficas anteriores)




Numero de clasificaciones erroneas
Iteraciones
0
1
2
3
4
5
0
2
4
6
8
10
12
14
16


Numero de clasificaciones erroneas
Iteraciones
168
170
172
174
176
178
180
182
184
186
0
2
4
6
8
10
12
14
16