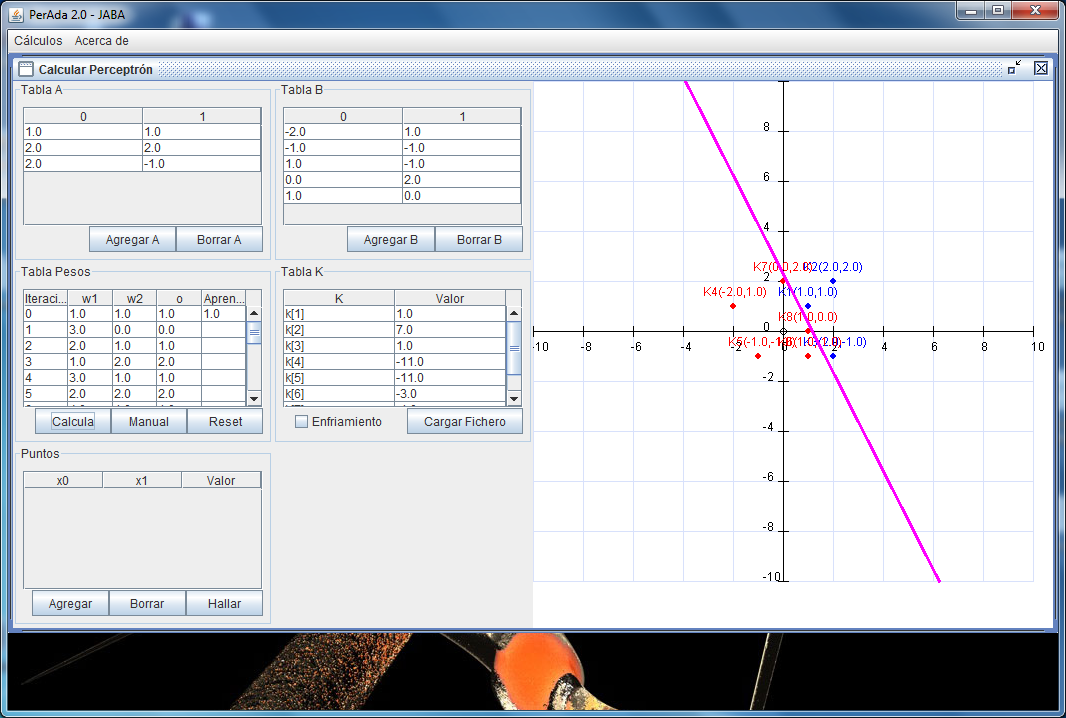
*Perceptrón Adaline*

( Desarrollado en el entorno Eclipse en el lenguaje JAVA )



Jose Alberto Benítez Andrades

71454586A

Redes Neuronales y Algoritmos Genéticos

Universidad de León

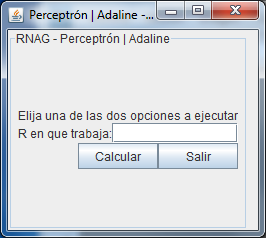
# Manual de Utilización de PerAda 2.0 – JABA

## Instalación del programa

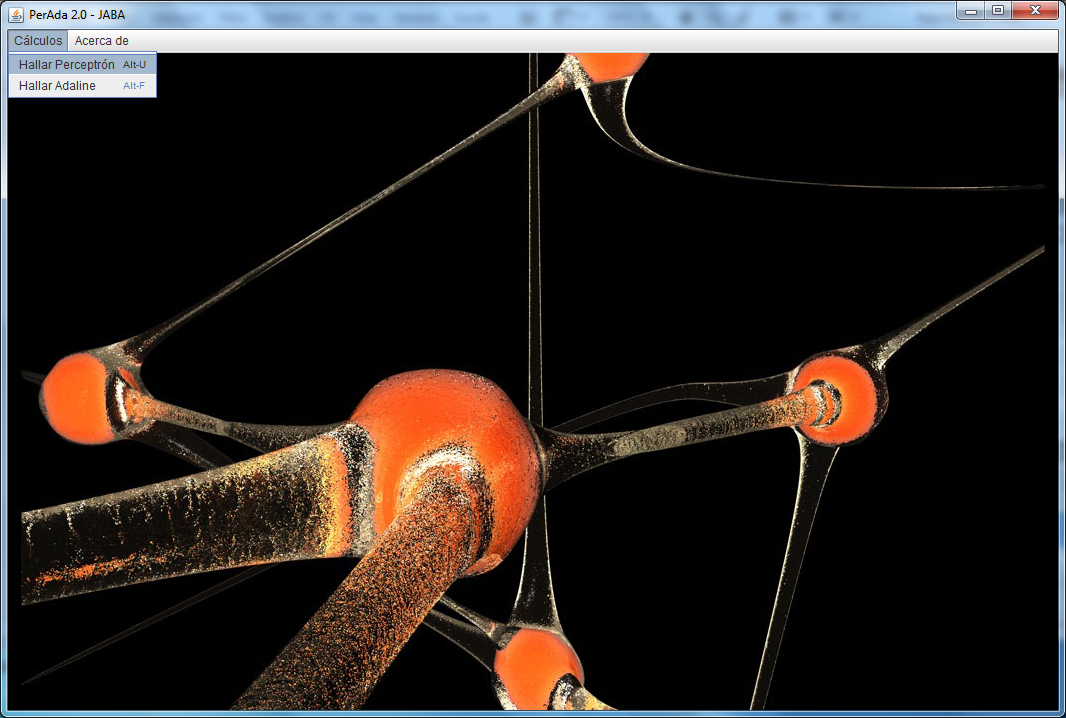
Para poder ejecutar el programa, debemos tener instalada la máquina virtual de JAVA, en su versión 6.X, que se puede descargar desde la página web de SUN (<http://java.sun.com/javase/downloads/index.jsp>) no obstante, en el CD viene un ejecutable.

## Ejecución del programa

Una vez esté instalada la máquina virtual, bastará con hacer doble click sobre el fichero .jar incluido en el CD ( PerAda.jar) y ya podremos ver la primera ventana, en la que pide en qué *R* queremos trabajar.

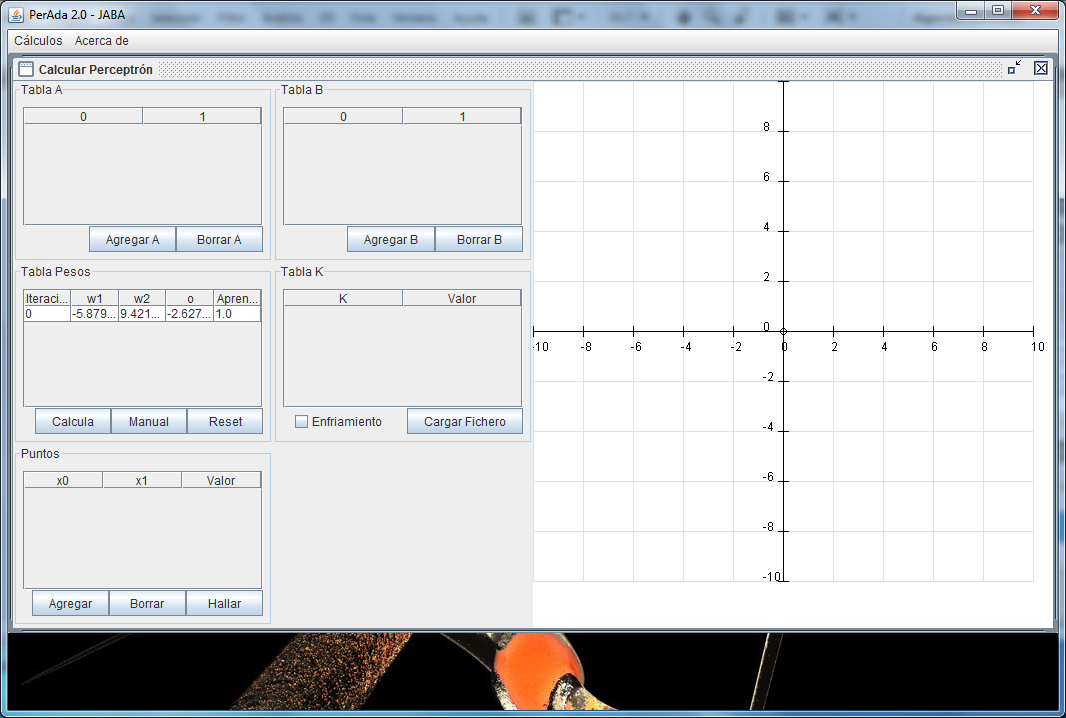


* Introducido el R en el que trabajar, accederemos a la pantalla principal, desde la cual podemos elegir si queremos calcular un *Perceptrón*  o un *Adaline.*

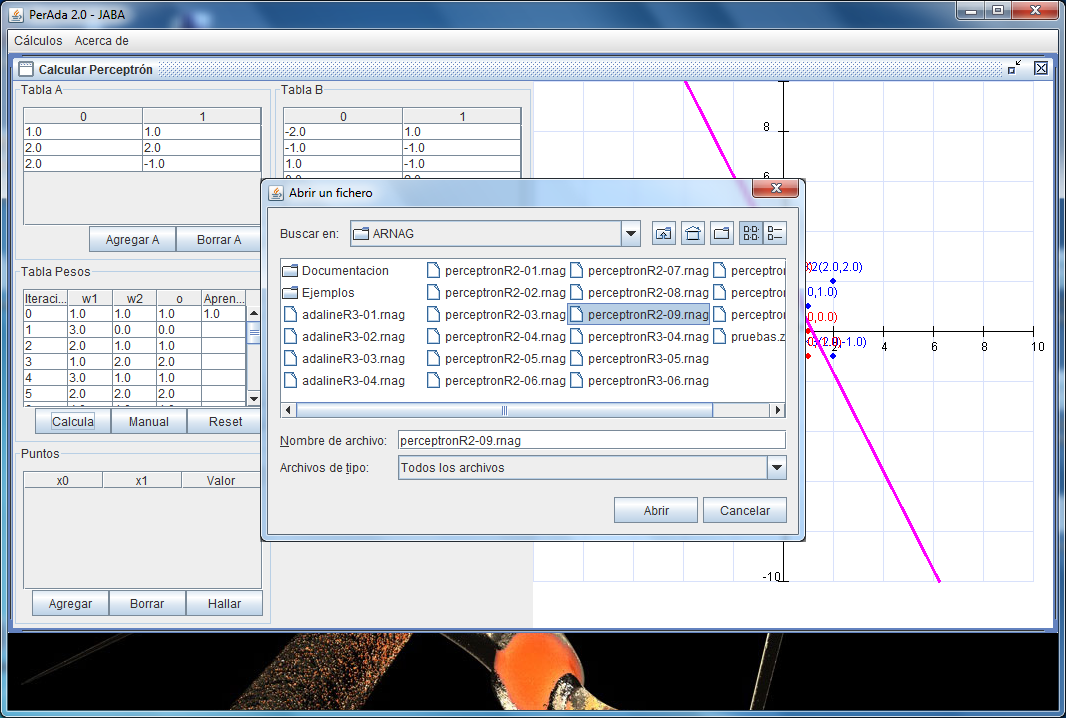


## Perceptrón

* Eligiendo la opción Perceptrón, podremos observar un *frame* como el de la siguiente imagen, en el que tendremos 5 tablas, y una gráfica (caso de introducir R = 2) o solamente 5 tablas, que contienen la siguiente información:
  + Tabla A: Pulsando *Agregar A* se insertará una fila en la tabla A, en la cual insertaremos las coordenadas que pertenecen a la tabla de los +1. Si queremos borrar la última fila, pulsamos *Borrar A.*
  + Tabla B: Pulsando *Agregar B* se insertará una fila en la tabla B, en la cual insertaremos las coordenadas que pertenecen a la tabla de los -1. Si queremos borrar la última fila, pulsamos *Borrar B.*
  + Tabla Pesos: En ella, podemos visualizar los pesos iniciales, junto con las constante de aprendizaje y el umbral. Además, una vez hayamos pulsado *Calcula* podremos ver todos los pesos que se han ido calculando mediante el algoritmo.
  + Tabla K: En ella podremos ver los valores reales obtenidos con cada coordenada, una vez estén bien hallados todos los pesos.
  + Puntos: En esta tabla, después de haber hecho el aprendizaje de la neurona, podremos poner puntos, y pulsando el botón *Hallar*, nos dirá si pertenece a la tabla A o al a tabla B.



* Además de insertar los datos a mano, mediante los botones de *Agregar A, Agregar B*, podemos cargar un fichero, que debe tener una estructura concreta, mediante el botón *Cargar Fichero* que mostrará una ventana de diálogo para seleccionar un fichero de nuestro ordenador.



* Teniendo ya los datos en ambas tablas, y estando bien escritos, podremos realizar el aprendizaje de 2 formas, de manera automática clickando en *Calcula* o de manera manual, pasando por distintas iteraciones y pudiendo observar movimientos en la gráfica, pulsando *Manual*.
* Al algoritmo que sigue para calcular el perceptrón, es de la siguiente manera:

Mientras NO haya terminado && contador < 50000

Mientras i < coordenadas de la tabla A

Calcular valor real

Si valor real <= 0

Recalcular pesos

Mientras i < coordenadas de la tabla B

Calcular valor real

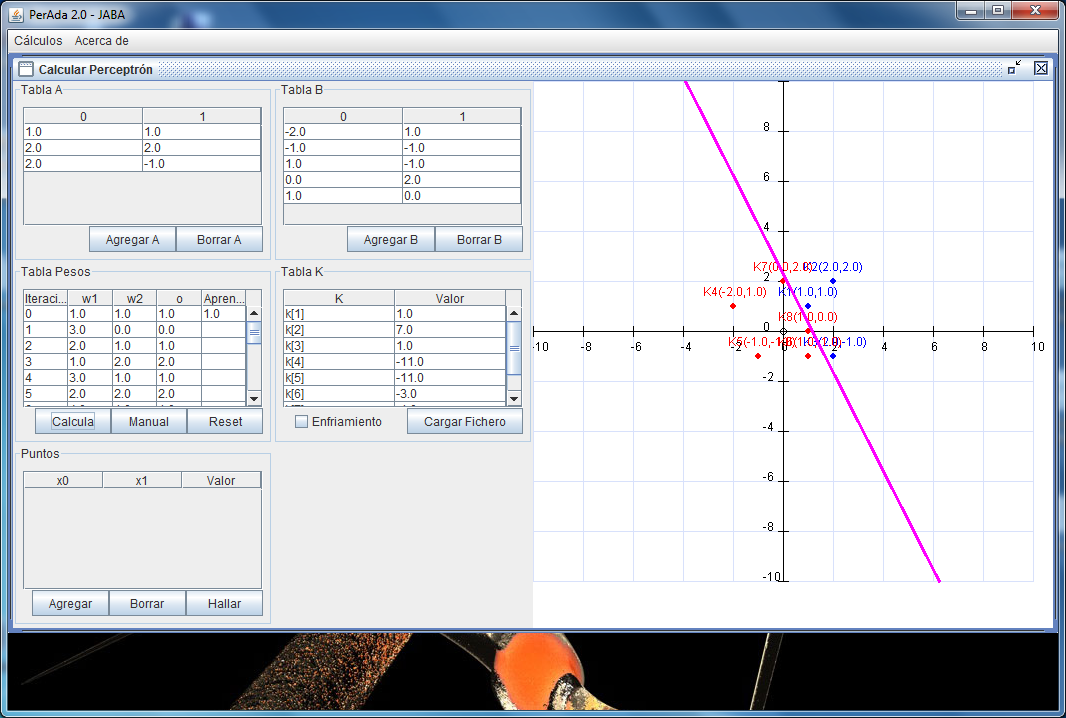
Si valor real >= 0

Recalcular pesos

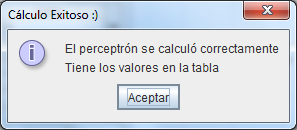
Si i == total de coordenadas

Terminado = true

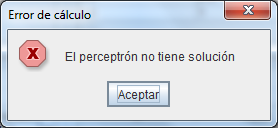
* Una vez calculado uno de los ejemplos que tengo en los ficheros de prueba, se obtiene una imagen como la siguiente:



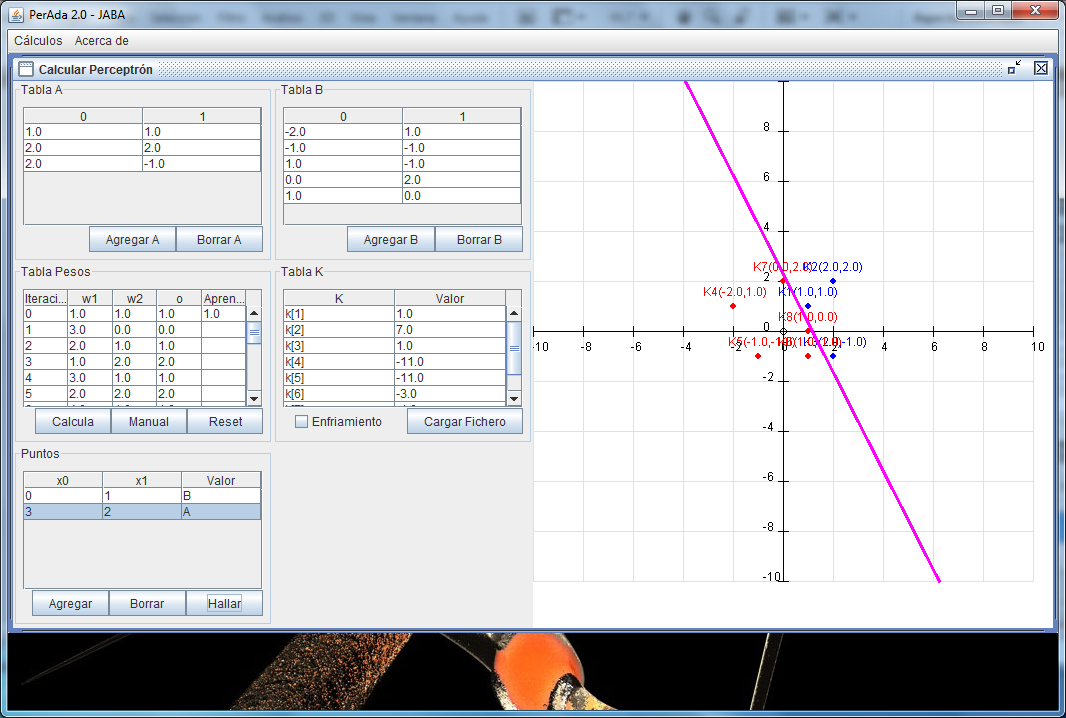
* Con el perceptrón bien calculado, obtendremos un mensaje del siguiente tipo:



* Si por el contrario, el perceptrón no tuviera solución, o al menos, no se pudiese calcular en menos de 5000 iteraciones, mostraría el siguiente mensaje:

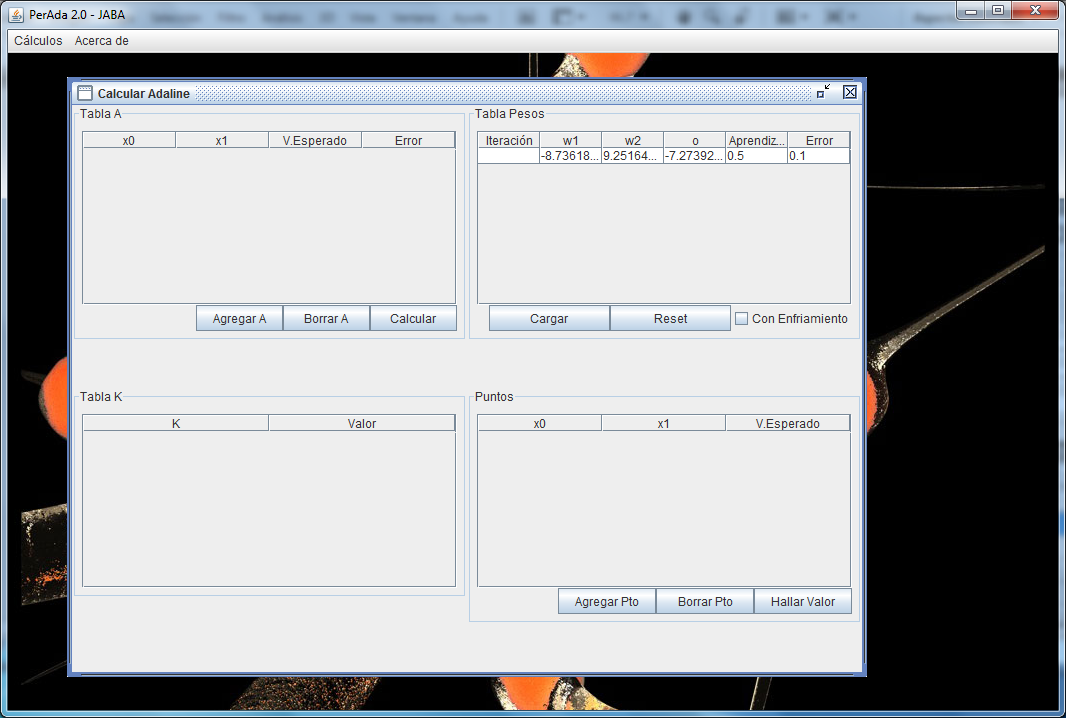


* Después de haber realizado el aprendizaje del perceptrón, tendremos la posibilidad de hallar, si un punto que nosotros queramos, pertenece a la tabla A, o a la tabla B. Pulsaremos el botón *Agregar* de la tabla de Puntos, insertaremos las coordenadas, y pulsando el botón *Hallar*, obtendremos el resultado de la tabla a la que pertenece.

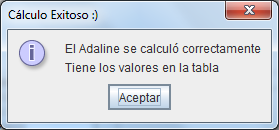


## Adaline

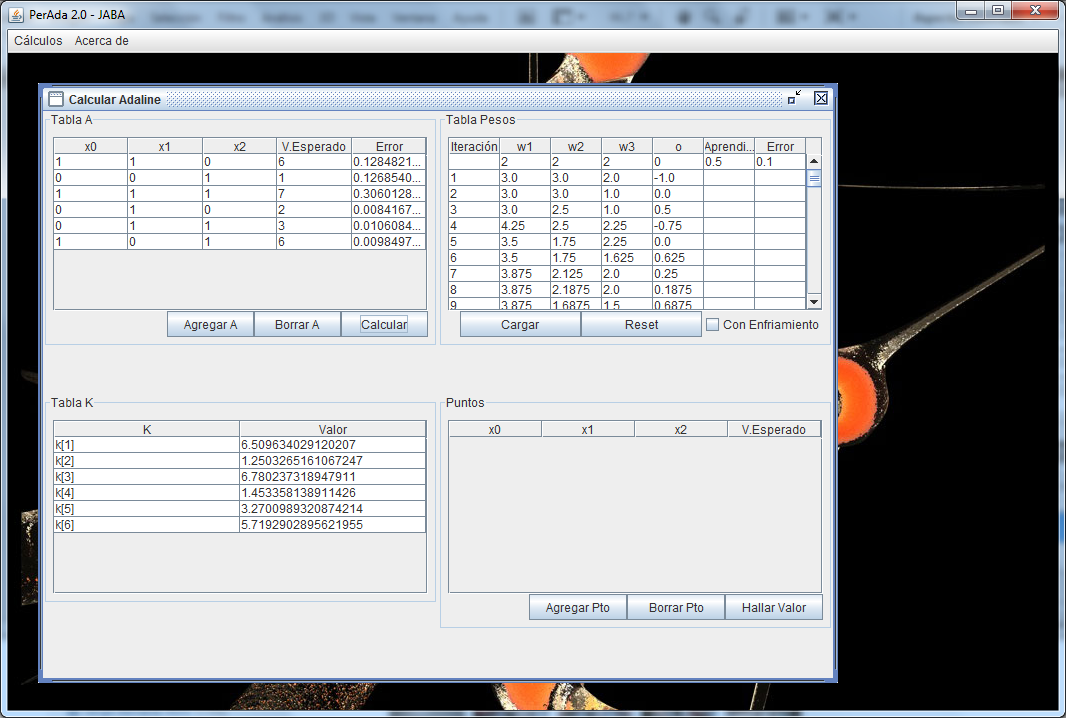
* Si cerramos el frame de calcular perceptrón, y pulsamos en el menú *Cálculos > Calcular Adaline*, accederemos a la pantalla mediante la cual podemos calcular un Adaline, en el R que queramos.
* Las tablas que aparecen son las siguientes:
  + Tabla A: Tabla en la cual insertaremos las coordenadas, junto con el valor esperado, y en la que se mostrará el error individual de cada coordenada, una vez realizado el cálculo.
  + Tabla Pesos: Contiene la información de los pesos iniciales, el umbral, el aprendizaje, y el error permitido.
  + Tabla K: Muestra los valores reales obtenidos, una vez realizado el cálculo de los pesos finales.
  + Tabla Puntos: Al igual que en el perceptrón, permite agregar una serie de puntos, y pulsando el botón *Hallar* nos dice qué valor es el esperado para ese punto.



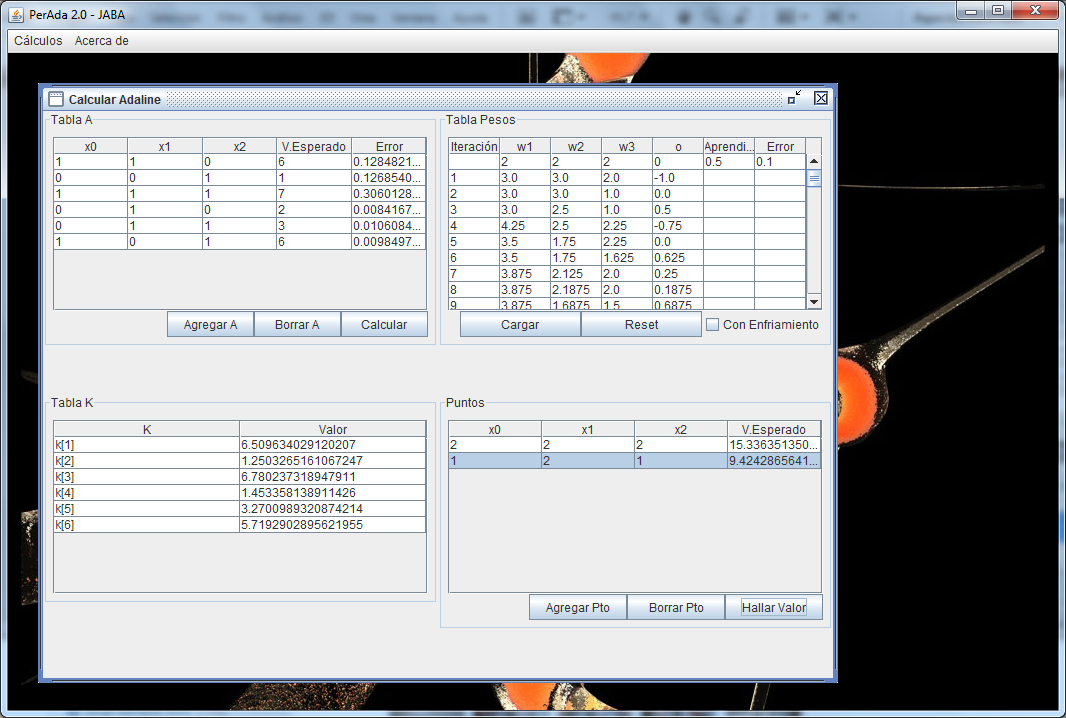
* Cuando se realizan los cálculos correctamente, muestra un mensaje de información como el del perceptrón, en caso contrario, muestra un mensaje de error.



* Una vez realizado el cálculo, obtendremos toda la información en las tablas como en la siguiente imagen:



* Como podemos observar en la siguiente imagen, habiendo hallado el adaline e insertando dos coordenadas, pulsando *Hallar Valor* , se obtienen los resultados de los valores esperados para esas coordenadas, como se puede observar en la siguiente imagen.



### Otras funciones del programa

* Si deseamos iniciar un nuevo cálculo, tanto de perceptrón, como de Adaline, pulsando el botón Reset, reiniciaremos todos los valores.
* Si queremos cambiar el *R* en el que trabajamos, cerrando la ventana principal, volveremos a la primera ventana en la que nos pide el R en el que trabajar.
* La estructura de los ficheros para cargar un perceptrón es de la siguiente manera:
  + Si trabajamos en R2

Peso1

Peso2

Umbral

A

X1

X2

X1

X2

…

B

X1

X2

X1

X2

….

En las primeras líneas pesos y umbral, en las siguientes, A y las coordenadas que se deseen, y en las últimas B y las coordenadas de la tabla B que se deseen.

* El formato del fichero a cargar en Adaline es de la siguiente manera (en R3 por ejemplo):

Peso1

Peso2

Peso3

Umbral

Aprendizaje

Error Esperado

X1

X2

X3

X1

X2

X3

…

## Otra información

* El programa se ha realizado en el entorno *Eclipse* en lenguaje JAVA.