FCI Academia de programación

Manual de prácticas de laboratorio

Programación Avanzada ISC

Jesús Alejandro Flores Hernández,

**Manual Prácticas Estructura de datos**

|  |  |
| --- | --- |
| **Autores** | * **MI Jesús Alejandro Flores Hernández** |
| **Contacto** | * [**jflorespampano@gmail.com**](mailto:jflorespampano@gmail.com) * **Facebook: jflorespampano** |

**Objetivo**: Desarrollar habilidades de implementación, uso y comprensión de la inteligencia artificial y sus aplicaciones, así como la programación en paralelo y programación en tiempo real.

Recursos necesarios instalar el siguiente software (libre):

1. Framwework 3.5
2. Programers NotePad
3. Sharp Develop.
4. JVM
5. SCILAB
6. SWI-PROLOG

Contenido

[Introducción. 3](#_Toc477187861)

[1 Módulo I 4](#_Toc477187862)

[1 Práctica Inteligencia artificial 4](#_Toc477187863)

[Objetivo 4](#_Toc477187864)

[Equipo 4](#_Toc477187865)

[Materiales 4](#_Toc477187866)

[Descripción 5](#_Toc477187867)

[Procedimiento. 5](#_Toc477187868)

[2 Práctica Sistemas expertos. 7](#_Toc477187869)

[Objetivo 7](#_Toc477187870)

[Equipo 7](#_Toc477187871)

[Materiales 8](#_Toc477187872)

[Descripción 8](#_Toc477187873)

[Procedimiento. 11](#_Toc477187874)

[3 Práctica RP – Redes Neuronales. 12](#_Toc477187875)

[Objetivo 12](#_Toc477187876)

[Equipo 12](#_Toc477187877)

[Materiales 12](#_Toc477187878)

[Descripción 13](#_Toc477187879)

[Procedimiento. 23](#_Toc477187880)

[4 Práctica RP- SVM. 23](#_Toc477187881)

[Objetivo 23](#_Toc477187882)

[Equipo 24](#_Toc477187883)

[Materiales 24](#_Toc477187884)

[Descripción 24](#_Toc477187885)

[Procedimiento. 42](#_Toc477187886)

[Módulo II. 43](#_Toc477187887)

[1 Práctica Programación en paralelo. 43](#_Toc477187888)

[Objetivo 43](#_Toc477187889)

[Equipo 43](#_Toc477187890)

[Materiales 43](#_Toc477187891)

[Descripción 43](#_Toc477187892)

[Procedimiento. 46](#_Toc477187893)

[2 Práctica Programación en tiempo real. 46](#_Toc477187894)

[Objetivo 46](#_Toc477187895)

[Equipo 46](#_Toc477187896)

[Materiales 46](#_Toc477187897)

[Descripción 47](#_Toc477187898)

[Procedimiento. 48](#_Toc477187899)

[Referencias: 49](#_Toc477187900)

# Introducción.

La inteligencia artificial son métodos que buscan reproducir las habilidades mentales humanas, son usadas en reconocimiento de patrones, predicción, regresión lineal, sistemas expertos y ámbitos en donde la inteligencia humana juega un papel importante. Los inicios de la inteligencia artificial (IA) se remontan a Alan Turing creador de la máquina enigma, con un artículo que publicó en 1950. En este apartado exploramos las principales técnicas de la IA y sus aplicaciones.

El proceso del pensamiento es muy complejo, es en las últimas décadas que se empieza a entender realmente como funciona nuestro cerebro y aunque aún estamos muy lejos de entenderlo por completo se han realizado muchos avances, lo que hoy sabemos se ha tratado de plasmar en el área de la computación a través de disciplinas que en conjunto trabajan sobre lo que se llama inteligencia artificial, esta es un intento de hacer que la computadora emule los procesos mentales de los seres vivos, preferentemente de los humanos, el problema se ataca desde varias vertientes, una de ellas es el reconocimiento de patrones, nuestro cerebro trabaja con patrones por ejemplo un bebe almacena el patrón visual de sus padres observando que tienen cabeza brazos pies etcétera. Y lo aplica al resto de su mundo para determinar cuándo es una persona la que está cerca de él y cuando un objeto inanimado esto sin duda requiere un complejo proceso mental y un entrenamiento de la mente del niño hasta lograr que su cerebro almacene el patrón de cómo debe verse un humano. Lograr que una computadora discrimine un elemento de otro se viene intentando desde hace años, el problema se ha abordado desde diferentes puntos de vista tales como redes neuronales, programación lógica, algoritmos genéticos y más recientemente las SVM, la solución a este problema se ha dado con escasos resultados. Es de vital importancia resolverlo pues ayudaría a hacer máquinas que pudieran tomar decisiones tal como lo hacen los humanos, el humano determina si un individuo es su hijo en base a un reconocimiento de patrones, un policía dispara a un delincuente en base a reconocimiento de patrones, la exploración en Marte se hace en base a reconocimiento de patrones, se revisa la corrosión en tuberías de gas en PEMEX en base a reconocimiento de patrones, las búsquedas en bancos de datos de huellas digitales se hace en base a reconocimiento de patrones, avanzar en esta área nos permitirá pues resolver muchos problemas asociados al reconocimiento de patrones.

Entre las técnicas usadas en el reconocimiento de patrones se tienen las redes neuronales, estas simulan la distribución neuronal del cerebro por medio de modelos en hardware o software de las neuronas cerebrales, aunque se ha avanzado mucho en esta área aun no estamos en condiciones de simular el comportamiento del cerebro humano, otra técnica usada es la computación evolutiva que consiste en modelar soluciones a problemas que pueden ser modificadas puntualmente para luego evaluar la mejor solución en un proceso de selección natural que simula la selección natural de los seres vivos. Otra técnica usada son las SVM (máquinas de soporte vectorial) son modelos matemáticos que mediante fórmulas determinan si un punto dado pertenece o no a un conjunto, es decir pueden discriminar los elementos. Las SVM nos dan resultados binarios es decir dado un punto nos dice si este pertenece o no a un conjunto dado y en general con esto es suficiente. Dado que tienen un sólido sustento matemático han estado dando buenos resultados, el problema estriba en que determinar las fórmulas adecuadas para que la SVM logre un buen reconocimiento es un proceso difícil y muchas veces lento.

# 1 Módulo I

Objetivo: Explicar los conceptos teóricos que sustentan la disciplina de inteligencia artificial y programar algoritmos de reconocimiento de patrones.

## 1 Práctica Inteligencia artificial

### Objetivo

El alumno será capaz de recordar y explicar los conceptos de inteligencia artificial.

### Equipo

* Computadora
* Internet

### Materiales

Hojas de pales

* Lápiz
* Papel

### Descripción

Sustento teórico

Concepto: En el ámbito de las ciencias de la computación se denomina como inteligencia artificial a la facultad de razonamiento que ostenta un agente que no está vivo.

Desde hace siglos el hombre ha querido reproducir la capacidad del cerebro humano, invento máquinas para facilitar su trabajo físico, pero con el trabajo inteligente no se tienen los logros esperados, el ábaco y las computadoras se inventaron con este fin pero no dejan de ser meros autómatas que no piensan y solo reproducen las ordenes previamente programadas, con la llegada de as computadoras se retoma el deseo de reproducir el pensamiento humano, con la disciplina de la inteligencia artificial, esta es una rama de la computación y relaciona un fenómeno natural con una analogía artificial a través de programas de computador. La inteligencia artificial puede ser tomada como ciencia si se enfoca hacia la elaboración de programas basados en comparaciones con la eficiencia del hombre, contribuyendo a un mayor entendimiento del conocimiento humano.

Concepto: Revise la Definición que sigue: <http://www.definicionabc.com/tecnologia/inteligencia-artificial.php> y realice las actividad siguientes:

Actividades:

1 Actividad 1

Para ayudarle a comprender mejor los inicios de la inteligencia artificial y las repercusiones que en la sociedad puede tener, vea las películas:

Vea la película piso 13

Vea la película Código enigma

2 Actividad 2

Busque 3 definiciones diferentes de inteligencia artificial, compárelas y emita una opinión sobre cual le parece más adecuada y por qué.

3 Actividad 3.

Estudie el siguiente contenido(PDF de inteligencia artificial UNAM) en: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/219/A7.pdf?sequence=7>

### Procedimiento.

Desarrollo de la práctica

Situación problema: Deberá comprender los conceptos de la inteligencia artificial para exponerlos ante un grupo de bachilleres, para asegurarse que los entiende debe realizar los siguientes ejercicios:

Ejercicios:

Ejercicio 1: Diseñe un mapa mental sobre la inteligencia artificial.

Ejercicio 2: Diseñe una línea de tiempo sobre los antecedentes de la inteligencia artificial.

Ejercicio 3: Diseñe 5 preguntas de opción múltiple sobre los conceptos de la IA.

Ejercicio 4 Conteste el siguiente cuestionario:

Propuso un álgebra que trabaja con solo dos valores



Gerge Boole



Alan Turing



Aristoteles

2 Diseño el primer computador electromecánico



Andrei Markov



J. P. Eckert y J W Mauchly



Alan Turing

Final del formulario

Principio del formulario

3 Prueba diseñada para determinar si una máquina es inteligente o no



Prueba de Turing



Algoritmo de Markov



Prueba de Boole

Final del formulario

Principio del formulario

4 Creo el lenguaje REC en México



Gerardo Cisneros



Harold V. McIntosh

1. Final del formulario

Ejercicio 5: Escriba en WORD lo siguiente:

* Resumen de las películas
* 3 definiciones de inteligencia artificial con su opinión comparativa.
* Una línea de tiempo de la evolución de la IA.
* El resultado del cuestionario.
* Las 5 preguntas de opción múltiple.

Entregables

Archivo Word con el resultado de los ejercicios.

Rubrica:

Rubrica películas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rubrica resumen | 0 | 3 | 5 puntos | total |
| Contenido | menos de una cuartilla | una cuartilla | dos cuartillas |  |
| Ortografía y formato | no muestra no ortografía no formato | muestra ortografía y no muestra formato | muestra ortografía y formato |  |
| Emite opinión sobre el la película con un enfoque en el tema | no emite sus opiniones | opina pero no con un enfoque en el tema de la materia | opina enfocado en el tema |  |
| Indica nombre completo(por apellidos), matricula, fecha, materia y profesor | no indica | datos pero incompletos | datos completos |  |

Rubrica definiciones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rubrica definiciones | 0 | 3 | 5 puntos | total |
| entrego a tiempo las definiciones | más de un día de retraso | 1 día de retraso | si |  |
| el archivo en que entrega tiene ortografía y formato | no muestra no ortografía no formato | muestra ortografía y no muestra formato | muestra ortografía y formato |  |
| Emite opinión sobre las definiciones | no emite sus opiniones | opina pero no con un enfoque en el tema de la materia | opina enfocado en el tema |  |
| contesta a las preguntas del profesor en la clase sobre sus definiciones | no contesta | contesta pero incompleto | contesta correctamente |  |

## 2 Práctica Sistemas expertos.

### Objetivo

El alumno será capaz de explicar las aplicaciones de los sistemas expertos y su sustento teórico, así como programar una base de conocimiento sencilla.

### Equipo

* Computadora
* Power Point
* SCI LAB

### Materiales

Hojas de pales

* Lápiz
* Papel

### Descripción

Sustento teórico

Los Sistemas Expertos, es una rama de la Inteligencia Artificial, son sistemas informáticos que simulan el proceso de aprendizaje, de memorización, de razonamiento, de comunicación y de acción en consecuencia de un experto humano en cualquier rama de la ciencia.

Estas características le permiten almacenar datos y conocimiento, sacar conclusiones lógicas, tomar decisiones, comunicarse con expertos humanos, explicar el porqué de las decisiones tomadas y realizar acciones como consecuencia de todo lo anterior.

Técnicamente un sistema experto, contiene una base de conocimientos que incluye la experiencia acumulada de expertos humanos y un conjunto de reglas para aplicar ésta base de conocimientos en una situación particular que se le indica al programa. Cada vez el sistema se mejora con adiciones a la base de conocimientos o al conjunto de reglas.

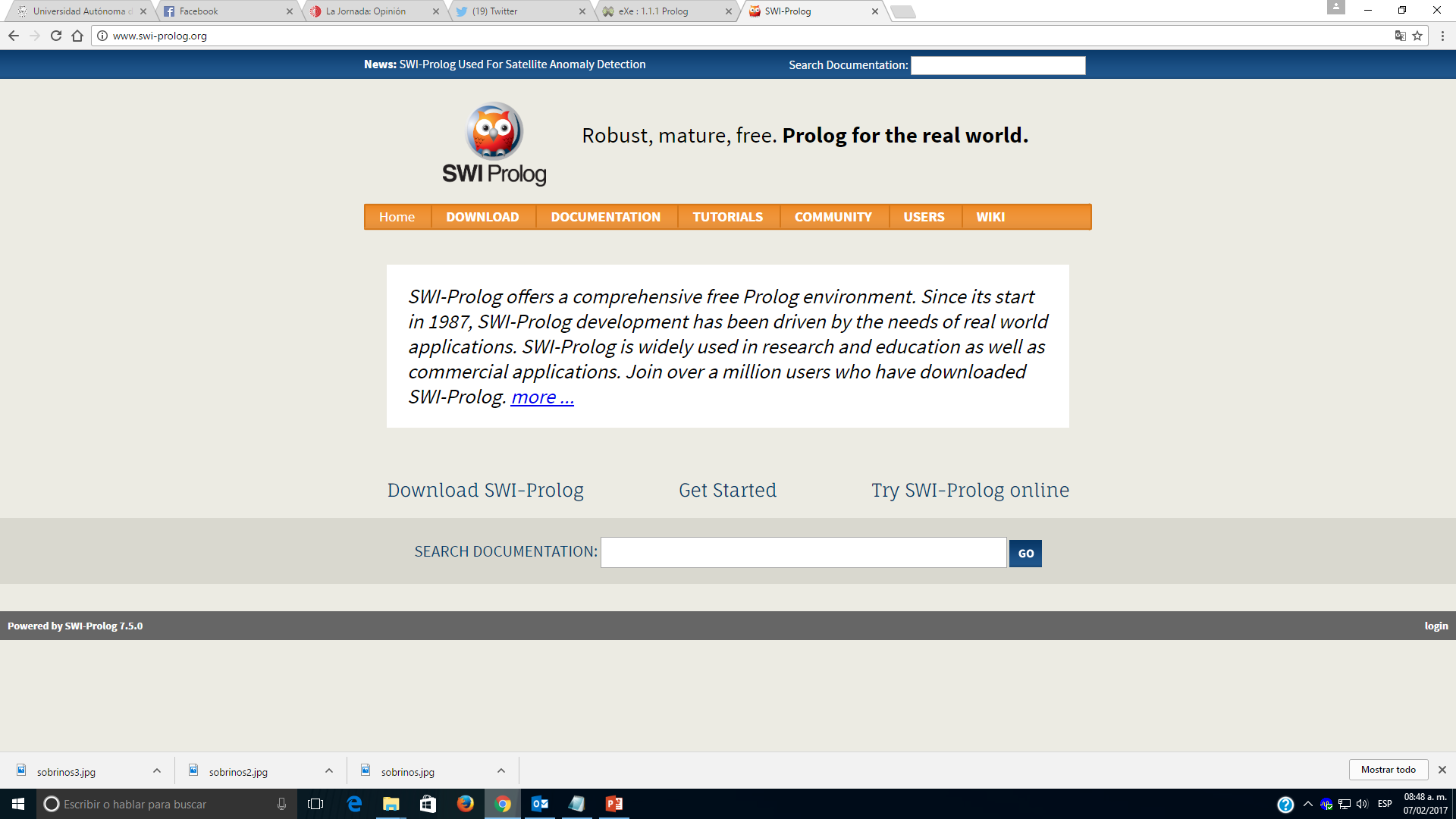
Usos de los SE

* Con la ayuda de un Sistema Experto, personas con poca experiencia pueden resolver problemas que requieren un "conocimiento formal especializado".
* Los Sistemas Expertos pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos.
* Los Sistemas Expertos razonan, pero en base a un conocimiento adquirido y no tienen sitio para la subjetividad.
* Se ha comprobado que los Sistemas Expertos tienen al menos, la misma competencia que un especialista humano.
* El uso de Sistemas Expertos es especialmente recomendado en las siguientes situaciones:
* Cuando los expertos humanos en una determinada materia son escasos.
* En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas.
* Cuando es muy elevado el volumen de datos que ha de considerarse para obtener una conclusión.

Estudie los siguientes archivos:

* Ingeniería del conocimiento
* Introducción a prolog

De la página WEB que se muestra en la imagen descargue el software SWI PROLOG:



Ejemplo 1 codificar hechos y reglas:

En un editor de código escriba el siguiente código:

|  |
| --- |
| Archivo programar.pl |
| /\*hecho 1 ana aprobo programacion 1\*/  aprobo\_programacion1(ana).  /\*hecho 2 rosa reprobo programacon 1\*/  reprobo\_programacion1(rosa).  /\*regla 1 X sabe programar si X aprobo programacion 1\*/  sabe\_programar(X):-aprobo\_programacion1(X). |

Este archivo muestra los siguientes hechos:

hecho 1 Ana aprobó programación 1

hecho 2 Rosa reprobó programación 1

Y una regla:

X sabe programar si X aprobó programación 1

Cargue la base de conocimientos “programar.pl”

|  |
| --- |
|  |

Realice las siguientes consultas:

|  |
| --- |
|  |

### Procedimiento.

Desarrollo de la práctica

Situación problema: Suponga que necesita explicar los conceptos de sistemes expertos a un grupo de bachilleres.

Ejercicios:

1 Elabore una presentación con los conceptos principales de los sistemas expertos.

2 Suponga que un tutor realiza las siguientes actividades:

* Aplica un examen a sus tutorados sobre un conjunto de temas.
* En base al examen determina que curso requiere el alumno para reforzar su aprendizaje.
* Tiene un conjunto de reglas como: “si el alumno X aprobó el curso Y entonces el alumno X tiene la habilidad Z”
* Diseñe una base de conocimiento para representar esas reglas (complete la información con entrevistas a sus profesores).
* Dado el ejemplo, para un grupo de alumnos en particular diseñe un conjunto de hechos.

3 Represente la información del punto 2 en SWI PROLOG.

Entregables

Código fuente de la aplicación.

Presentación en PPT expuesta al grupo.

Rubrica:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rubrica Presentación | 20 | 15 | 25 puntos | total |
| Contenido | El contenido es pertinente al tema | poco pertinente | muy pertinente |  |
| Ortografía y formato | no muestra no ortografía no formato | muestra ortografía y no muestra formato | muestra ortografía y formato |  |
| El diseño de cada diapositiva es adecuado en cuanto a visibilidad del texto,  contenido de imágenes que apoyan la explicación, la dosificación de texto es  adecuada, los colores y tamaño de la letra permiten una visibilidad desde el  extremo del salón | diapositivas mal diseñadas | cumplen algunos requerimientos como uso de imágenes  pero no dosificación adecuada de texto o viceversa. | cumple con todos los requerimientos |  |
| Participan todos los integrantes del equipo, exponen con seguridad sin uso de muletillas, contestan las preguntas satisfactoriamente | no participan todos o ni exponen con seguridad ni contestan preguntas satisfactoriamente | participan todos pero no exponen con seguridad o  no contestan adecuadamente | participan todos exponen con seguridad y  contestan satisfactoriamente |  |

## 3 Práctica RP – Redes Neuronales.

### Objetivo

Entender las bases teóricas y prácticas de las redes neuronales para codificar algoritmos que soluciones problemas de reconocimiento de patrones.

### Equipo

* Computadora
* SCI LAB

### Materiales

Hojas de pales

* Lápiz
* Papel

### Descripción

Sustento teórico

Introducción Redes neuronales

“Se estima que el cerebro humano contiene más de cien mil millones (1011) de neuronas y 1014 sinapsis en el sistema nervioso humano. Estudios sobre la anatomía del cerebro humano concluyen que hay más de 1000 sinapsis a la entrada y a la salida de cada neurona. Es importante notar que, aunque el tiempo de conmutación de la neurona (unos pocos milisegundos) es casi un millón de veces menor que en los actuales elementos de las computadoras, ellas tienen una conectividad miles de veces superior que las actuales supercomputadoras” [1]. El elemento básico de la red neuronal es la neurona, en esta se distinguen tres partes: el cuerpo de la neurona, ramificaciones llamadas dentritas, y axón que conecta la salida de la neurona con otras neuronas.

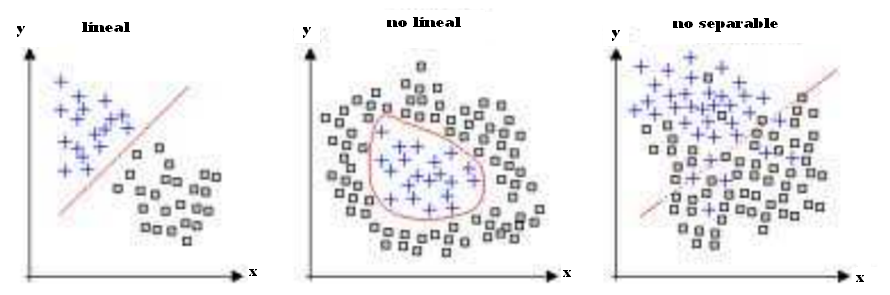
En computación las redes neuronales artificiales son sistemas que emulan el funcionamiento de los sistemas biológicos, particularmente el del cerebro humano, las primeras referencia a este tema aparecen en 1943 cuando McCulloch y Pitts crean un modelo computacional de actividad nerviosa, su elemento básico en el modelado de una neurona artificial que simula el funcionamiento de las neuronas animales, en este sentido es muy importante comprender el funcionamiento de la neurona biológica y de las redes neuronales del cerebro animal, en este campo es importante mencionar que el español Santiago Ramón Y Cajal desarrolla la teoría neuronal. “Esta teoría postula que el tejido nervioso está formado por billones de células (neuronas) independientes y separadas entre sí, pero capaces de comunicarse entre ellas” ver: http://spin.udg.edu/rn12/santiago-ramon-y-cajal/#sthash.2g2OJSvi.dpuf. En un tiempo en el que se consideraba que todos los tejidos estaban células independientes, con excepción del tejido nervioso. En este último caso, al hacer las observaciones microscópicas solo se veía una red enmarañada, pero no se podía visualizar en ningún momento células independientes hasta que Cajal expuso su teoría. Esta nos llevó al conocimiento del cerebro que se tiene hoy día y a partir del cual se ha tratado de emular las neuronas biológicas y las redes en las que participan como una técnica de IA para resolver problemas como reconocimiento de patrones y predicción, son técnicas que han demostrado alto grado de valor, en este capítulo veremos qué son y cómo se aplican las redes neuronales artificiales.

Las redes neuronales son una técnica usada en la inteligencia artificial donde sistemas de software y hardware simulan a las neuronas del cerebro humano y permiten crear redes entre esas neuronas de la misma forma en que el cerebro humano lo hace, estas técnicas son eficazmente usadas en reconocimiento de patrones y predicciones, tienen la ventaja de que son adaptativas y robustas, tienen alta tolerancia a fallos, capacidad de procesar en paralelo, tienen la desventaja de que son lentas de entrenar y no existen reglas para ello, a pesar de que tienen la capacidad de procesar en paralelo están limitadas por que nuestros procesadores solo son capaces de procesar en línea. A continuación, se muestra material multimedia para introducirnos en los conceptos de las redes neuronales y sus aplicaciones, siga las instrucciones que a continuación se indican.

Ejemplo uso de una red neuronal:

Objetivo: Clasificar un conjunto de puntos:

Tipos de clasificación:



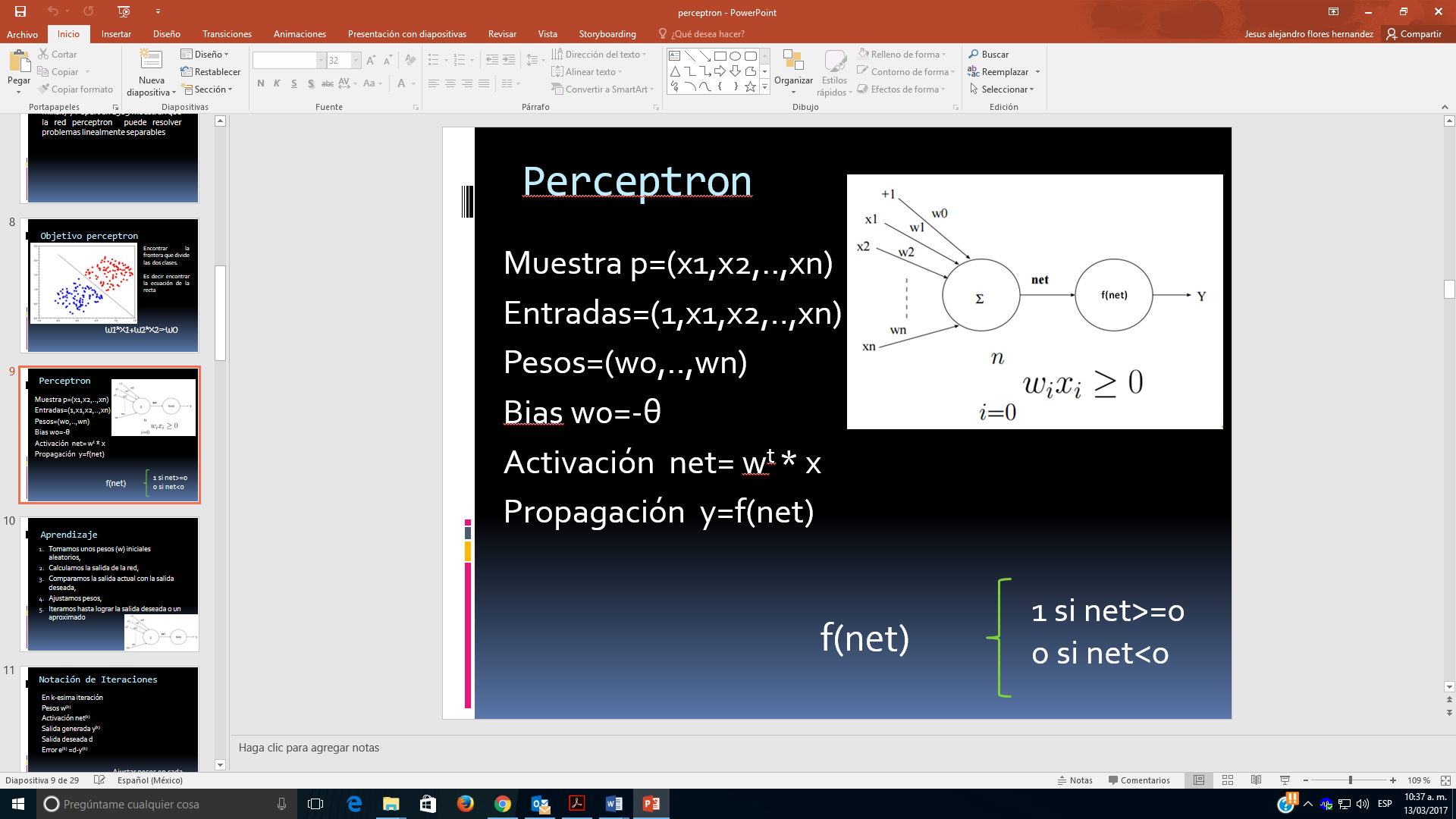
Este ejemplo se centra en la clasificación lineal.

Decimos que un conjunto es linealmente separable en dos subconjuntos en 2D si existe una recta que puede trazarse entre los dos subconjuntos, en 3D buscaríamos un plano.

Red neuronal básica consistente en una sola neurona:



La neurona tiene un conjunto de entradas xi (muestra) que son multiplicadas por un conjunto de pesos wi, la salida de la neurona (net) es la sumatoria de todos los xi \* wi, a esa salida le aplicamos una función de decisión f(net) que nos da 1 si la salida en >=1 y 0 en caso contrario:



Aprendizaje de la red

1. Tomamos unos pesos (w) iniciales aleatorios,
2. Calculamos la salida de la red,
3. Comparamos la salida actual con la salida deseada,
4. Ajustamos pesos,
5. Iteramos hasta lograr la salida deseada o un aproximado

Algoritmo:

1. En k-esima iteración
2. Pesos w(k)
3. Activación net(k)
4. Salida generada y(k)
5. Salida deseada d
6. Error e(k) =d-y(k)
7. Ajuntar pesos

Ajuste de pesos:

Si e(k) =0 fin

En caso contrario

Caso 1 e(k) =1, cuando d=1 , y(k) =0

Aumentar pesos

Caso 2 e(k) =-1, cuando d=0 , y(k) =1

Disminuir pesos

Es decir:

Caso 1 e(k) =1, tenemos nx

Caso 2 e(k) =-1, tenemos –nx

Es decir, tenemos: n \* e(k) \* x

Con n un factor de aprendizaje

Así:

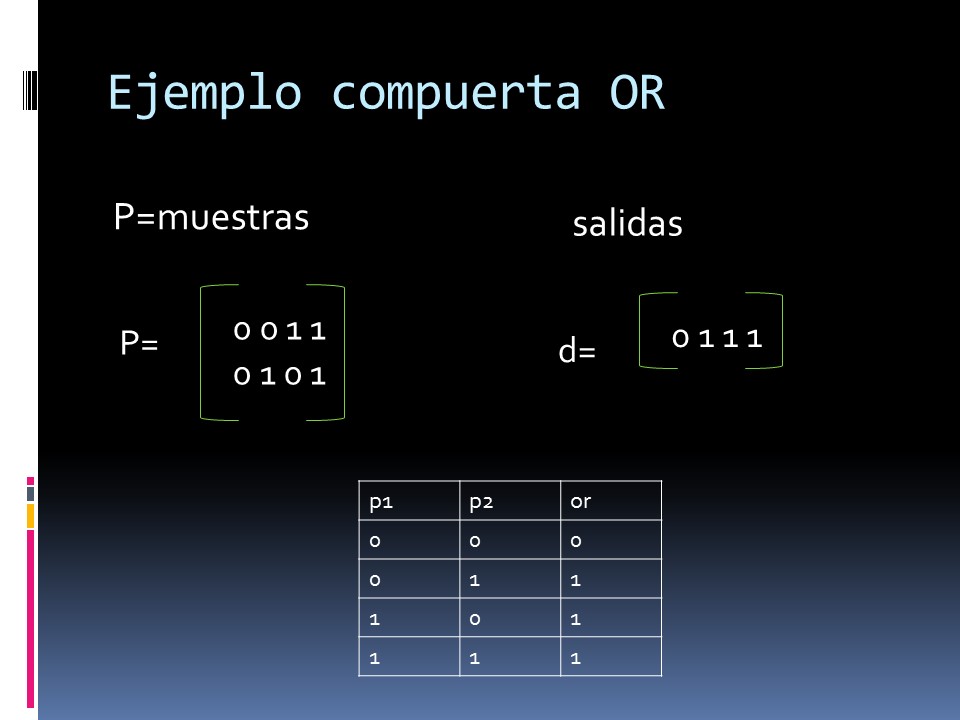
w(k+1) = w(k) +n\* e(k) \* x

Por tanto, la fórmula de ajuste de pesos es:

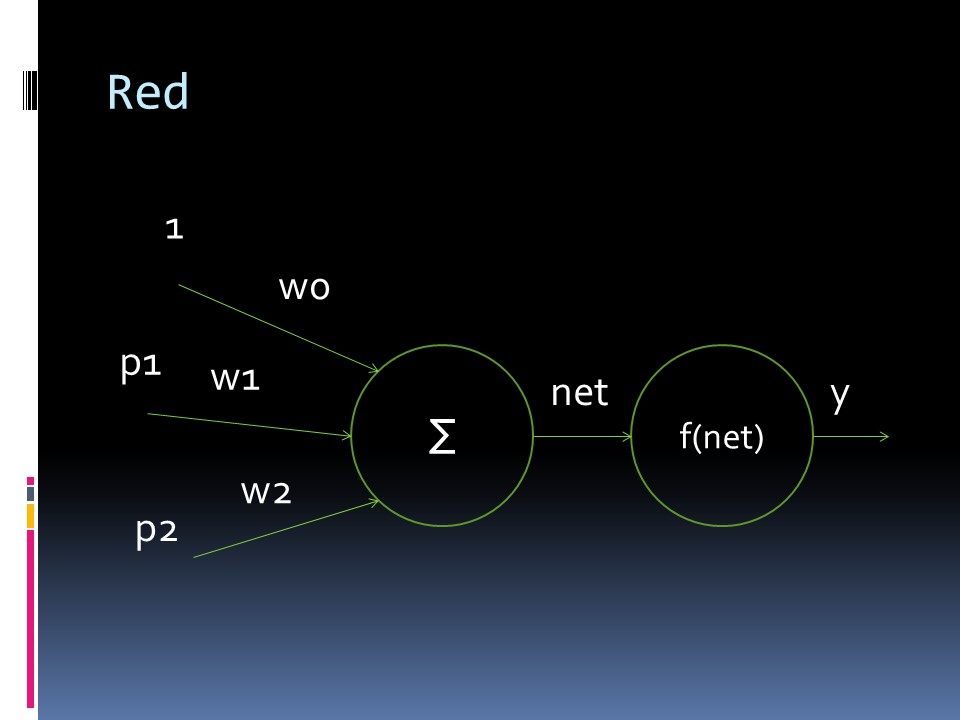
w(k+1) = w(k) +n\* e(k) \* x

|  |
| --- |
| Algoritmo |
| Entrada xi , yi  Salida vector de pesos w  Inicio  k=0  inicializar w(k), n  repetir  para cada (xi,yi) hacer  yi(k)=f(w(k)’\*xi)  ei(k) =di-yi(k)  actualizar w(k+1) = w(k) +n\* ei(k) \* xi  siguiente  k=k+1  hasta que ||(ei(k-1)) ||<Є  fin |

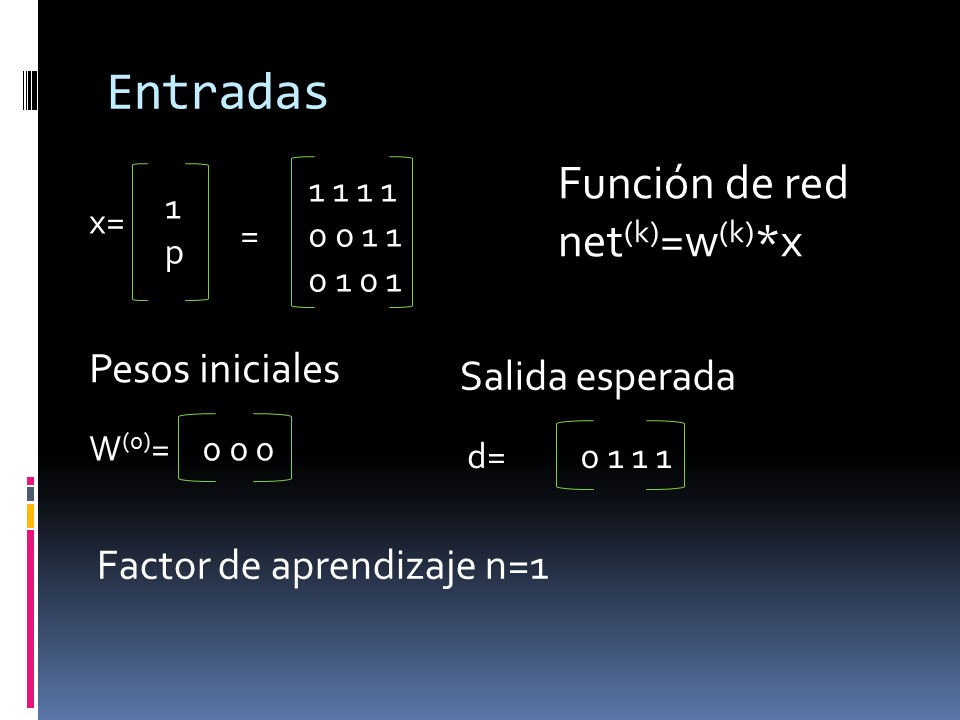
Ejemplo Compuerta OR



En la imagen se puede ver las muestras P las salidas d y la tabla de la función OR. Se requiere una red neuronal de tipo perceprón como se muestra a continuación:

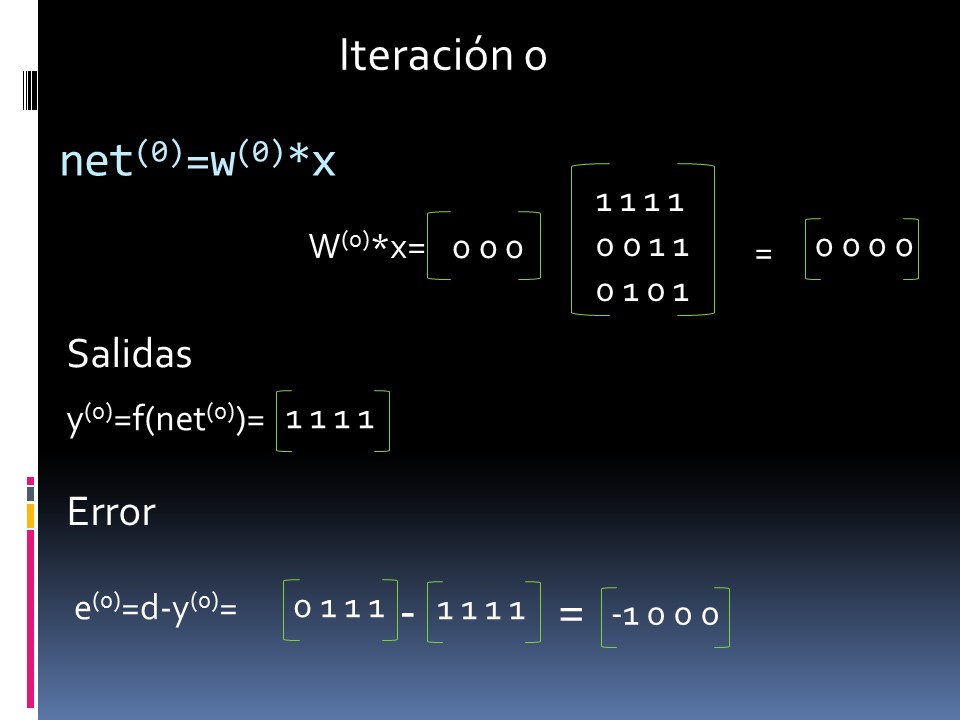


A continuación, se muestran la función de red net(k) las entradas y la salida esperada.



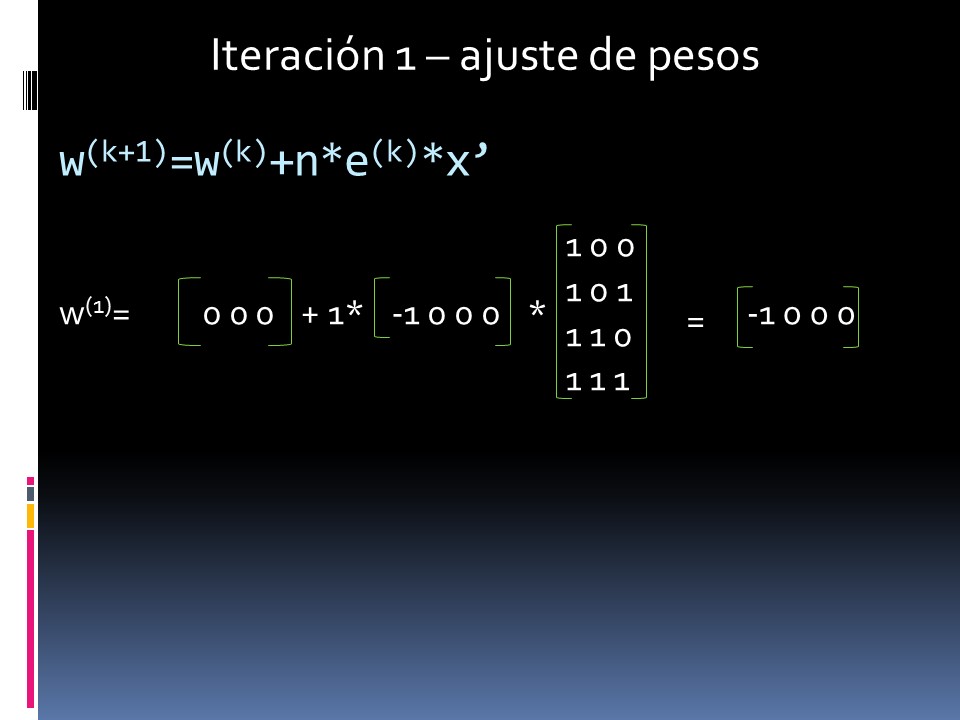
Empecemos las iteraciones:

Iteración 0:

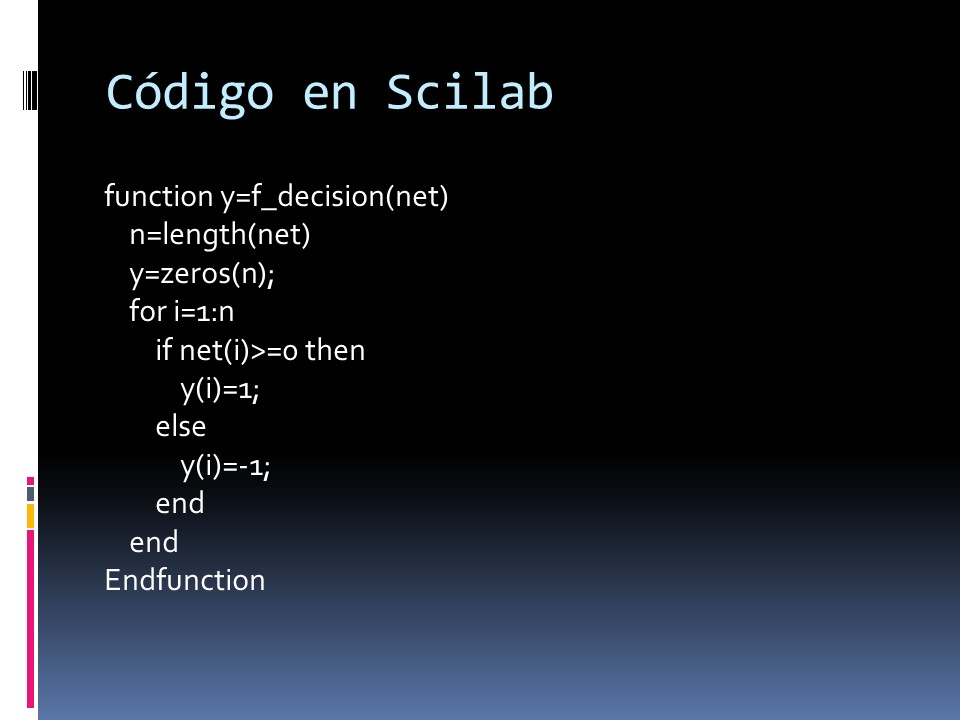


Co este error en la iteración siguiente ajustamos pesos:

Iteración 1:



Con esa sola iteración se tiene la solución correcta:



### Procedimiento.

Desarrollo de la práctica

Situación problema: se requiere la simulación del comportamiento de las compuertas lógica OR y AND.

Ejercicios:

1. Programa la función OR en SCI LAB usando perceptrón.
2. Programe la función AND en SCI LAB usando perceptrón

Entregables

Código fuente de la aplicación.

Resultado de los ejercicios.

## 4 Práctica RP- SVM.

### Objetivo

El alumno será capaz de identificar entender e implementar una cola usando una lista ligada en un lenguaje de programación.

### Equipo

* Computadora
* SCI LAB

### Materiales

Hojas de pales

* Lápiz
* Papel

### Descripción

Sustento teórico

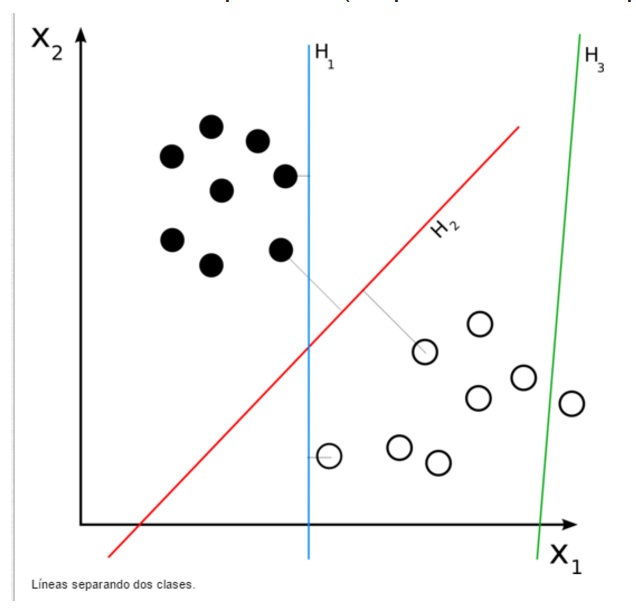
**SVM**

Los SVM son una herramienta usada en la clasificación de objetos de dos clases que ha resultado ser muy poderosa, se basa en encontrar una  superficie de clasificación determinada  por ciertos puntos de un conjunto de entrenamiento, este conjunto de vectores debe estar en la frontera entre los dos subconjuntos en que se clasificarán los puntos, estos vectores son llamados vectores de soporte y describen una línea de separación (en la imagen se muestran una línea roja H2 y una azul H1, ambos separan a los puntos de la clase negra y blanca respectivamente), si estamos en tres dimensiones el separador no será una línea será definido por un plano y en una dimensión mayo a tres definen un hiperplano de separación, en general le llamaremos hiperplano, la idea es buscar un hiperplano con  una “franja” con un margen lo más ancho posible, que separe a los dos subconjuntos. Las SVM son un paradigma aparte de la Redes Neuronales, pero a pesar de tener similitudes están mejor fundamentadas en la teoría y tienen mucho mejor capacidad de generalización.

En las SVM dados dos conjuntos de datos de clases diferentes se trata de buscar un conjunto de puntos llamados vectores de soporte  sobre los que se define una frontera de decisión a través de un hiperplano de separación, esto no siempre es sencillo debido a la distribución de los datos por lo que puede requerirse aplicar transformaciones lineales por medio de fórmulas llamadas kernel () mediante las cuales se lleva el conjunto de puntos a un hiperespacio donde puedan ser linealmente separables.

Para obtener la superficie de separación se deberá resolver un problema de programación cuadrática convexa, con restricciones lineales.  Para este problema se tienen dos casos:

**1 Linealmente separable (se puede trazar un plano de separación, se muestra en rojo)**



**2 No separable linealmente.**

Este caso tenemos vectores en 2 dimensiones R2, se lleva a R3 donde es linealmente separable y se traza un plano de separación (en verde).

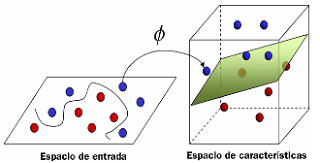
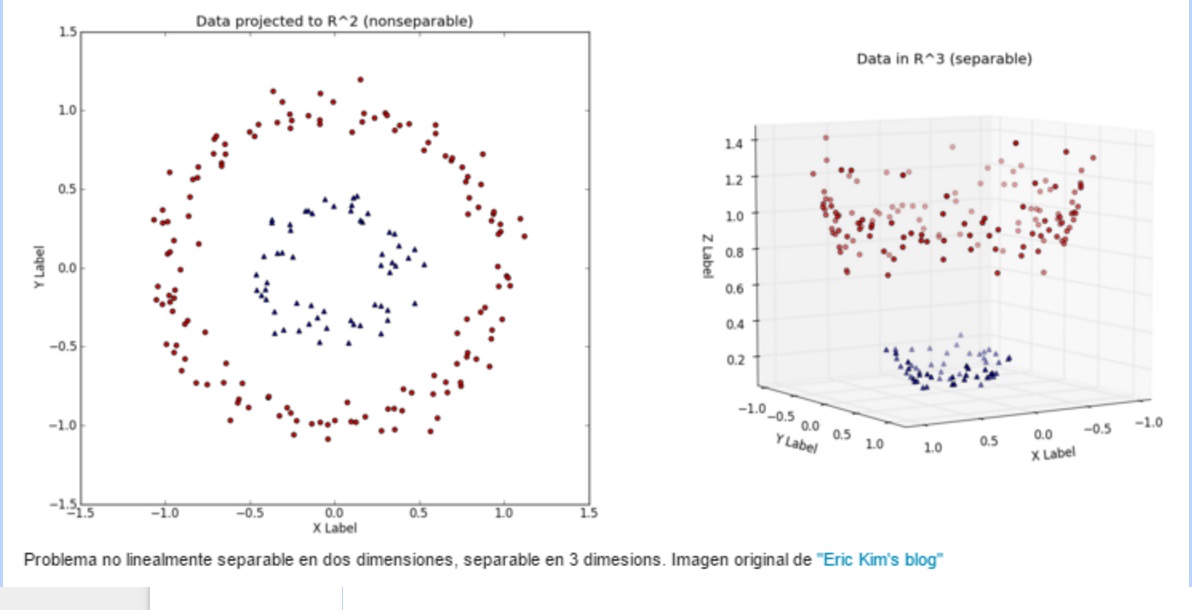
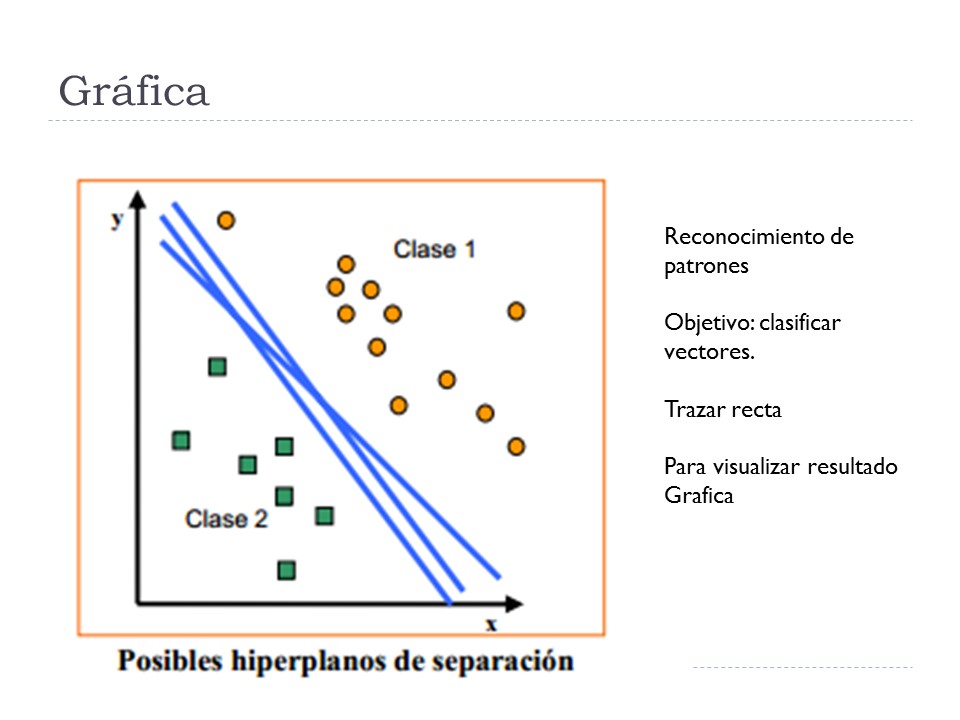


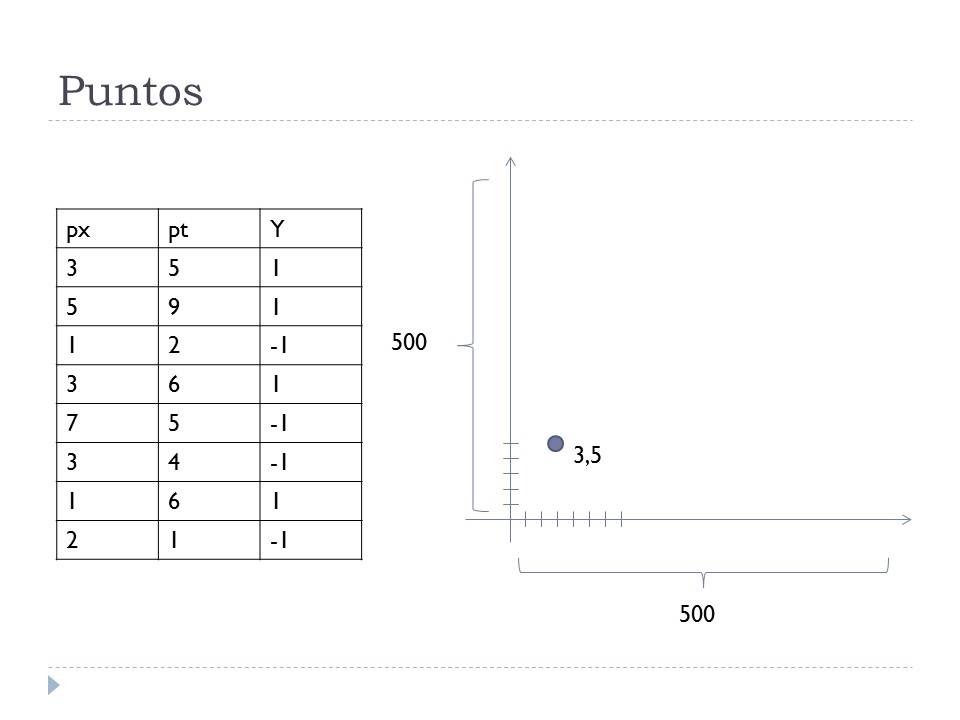
figura de un conjunto de puntos no separable linealmente, está en R2, se lleva a R3 y es linealmente separable

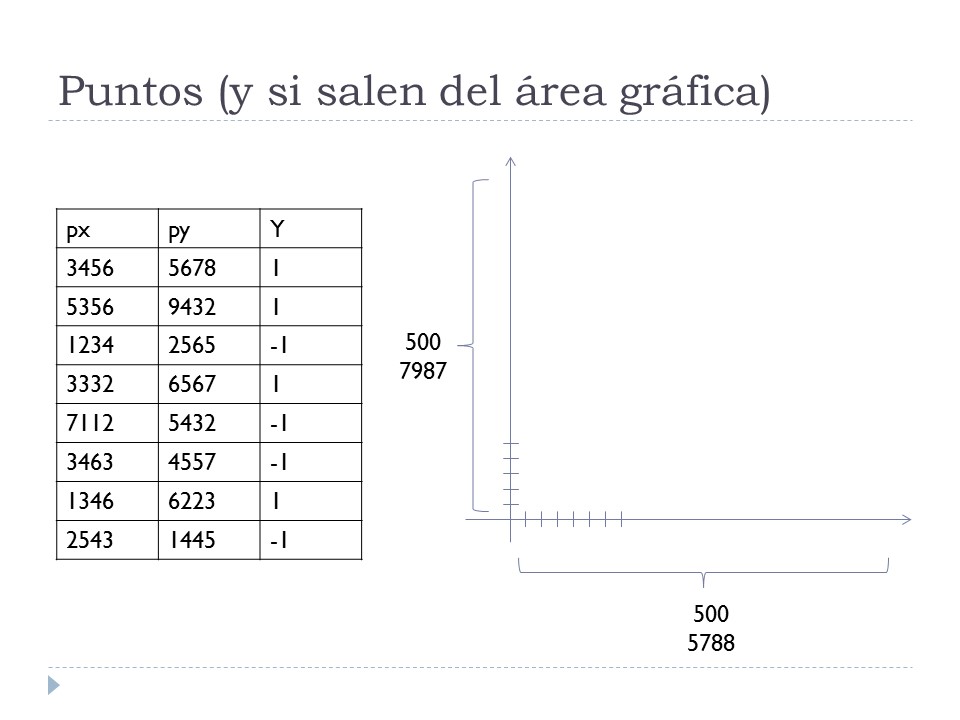


Ejemplo:

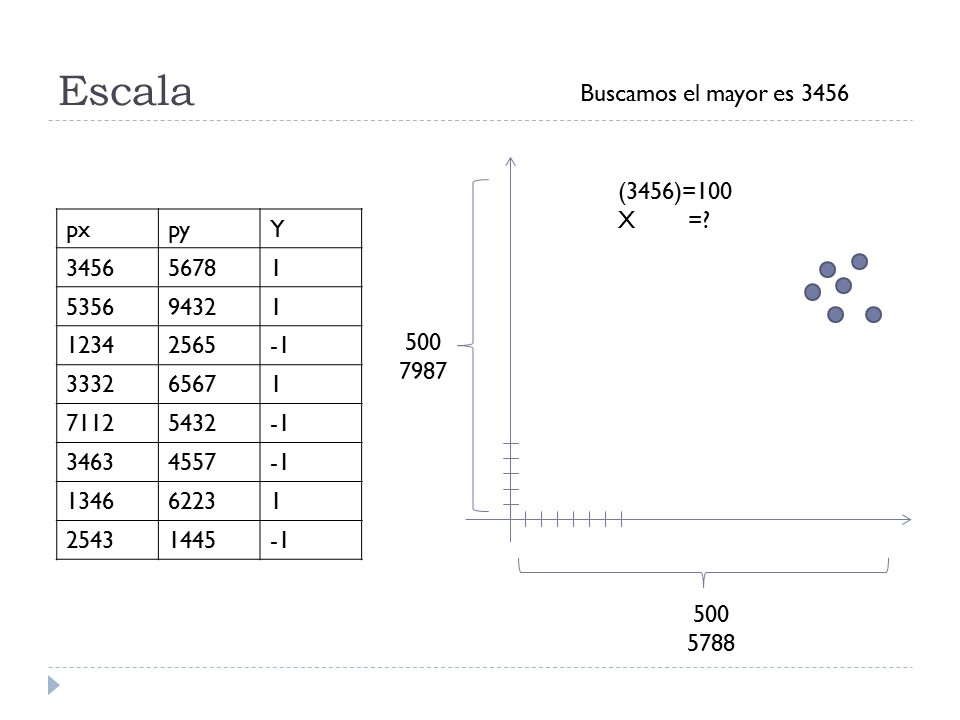


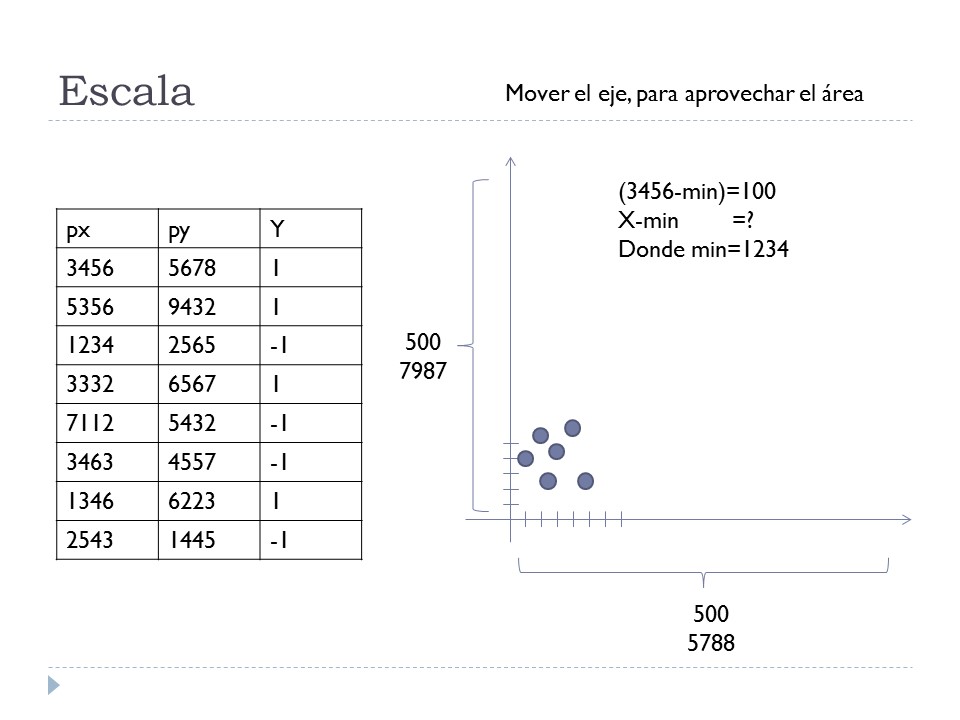
Tomemos un conjunto de puntos aleatorios linealmente separables:



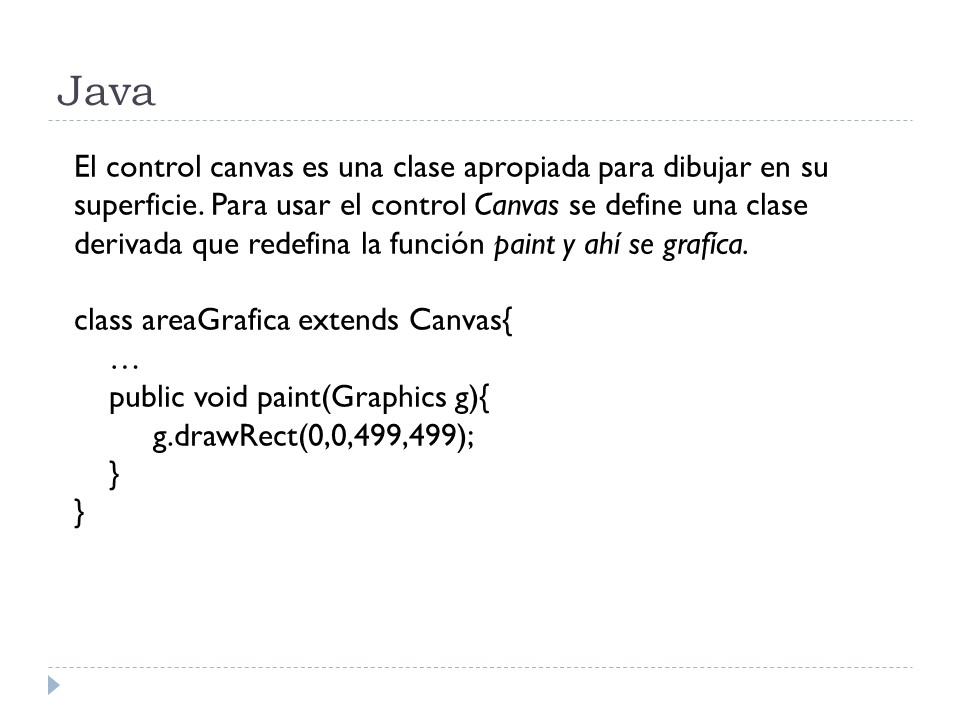


Les aplicamos una escala para visualizarlos mejor





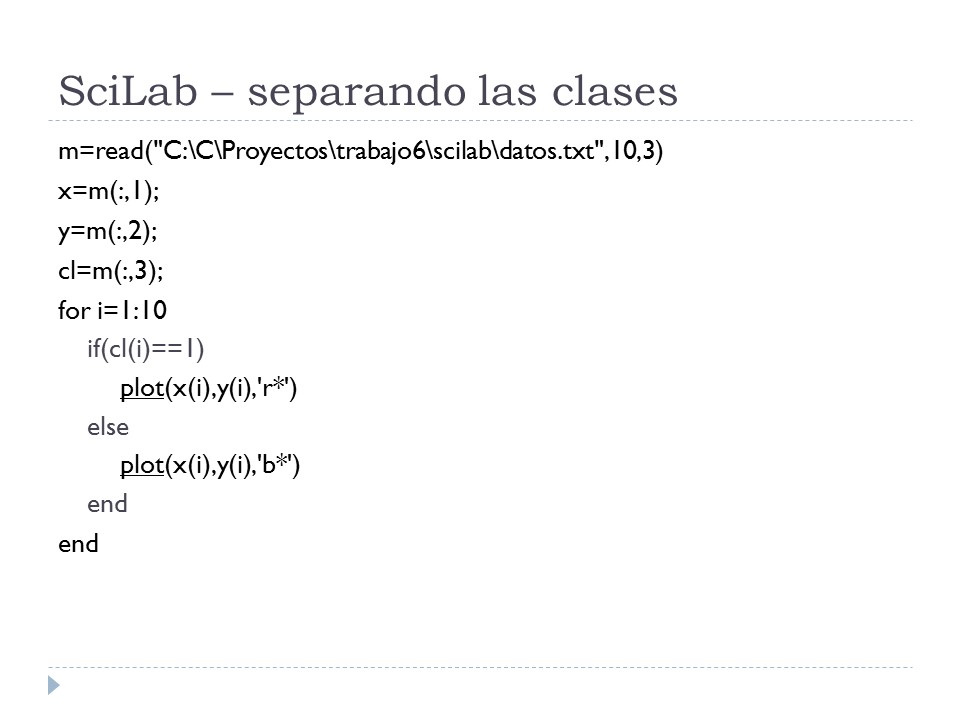
Esto se puede graficar en SCILAB de manera sencilla, pero en Java se puede también graficar utilizando el control Canvas.



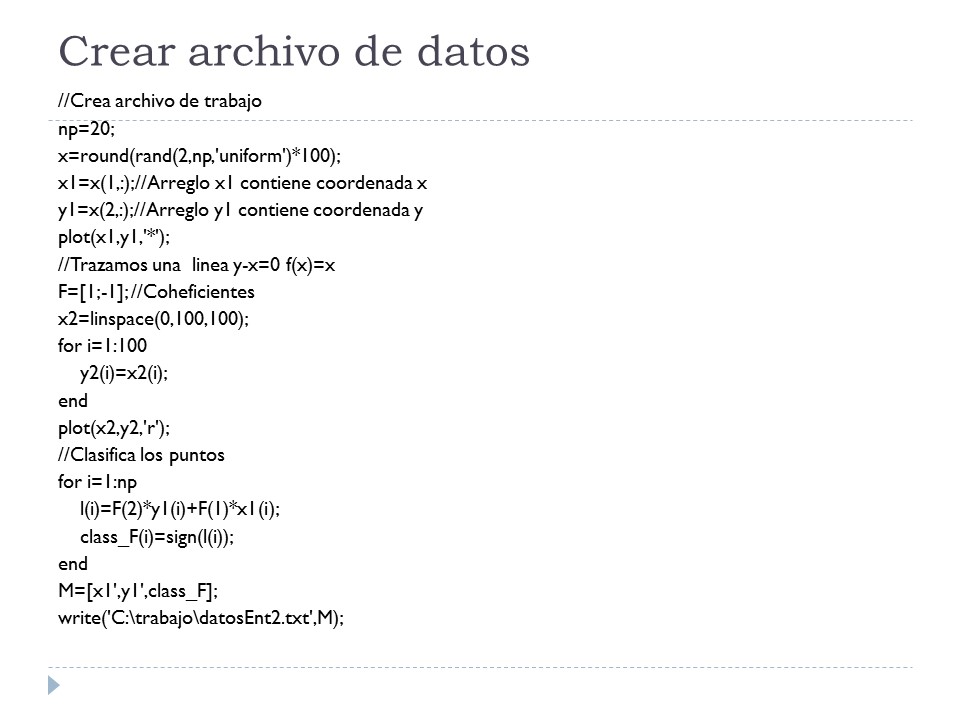
En SCI LAB a partir de un archivo como este:

|  |
| --- |
| Archivo: datos.txt |
| 5.2 2.3 1  5 1.9 1  5.2 2 1  4.9 2 1  5.1 1.8 1  5.1 1.9 1  4.8 1.8 1  5.8 2.2 1  6 1.8 1  6.3 1.8 1  5.8 1.8 1  4.9 1.8 1  5.1 2 1  5.3 1.9 1  5.5 2.1 1  5 2 1  5.1 2.4 1  5.3 2.3 1  5.5 1.8 1  6.1 2.3 1  4.7 1.4 -1  4.5 1.5 -1  4.9 1.5 -1  4 1.3 -1  4.6 1.5 -1  4.5 1.3 -1  4.7 1.6 -1  3.3 1 -1  4.6 1.3 -1  3.9 1.4 -1 |

La forma de graficar esos puntos en SCXI LAB es con el siguiente código (mostrando ya las clases separadas para verificar el comportamiento del método):



Para generar el archivo de datos en SCI LAB puede usar el siguiente código:



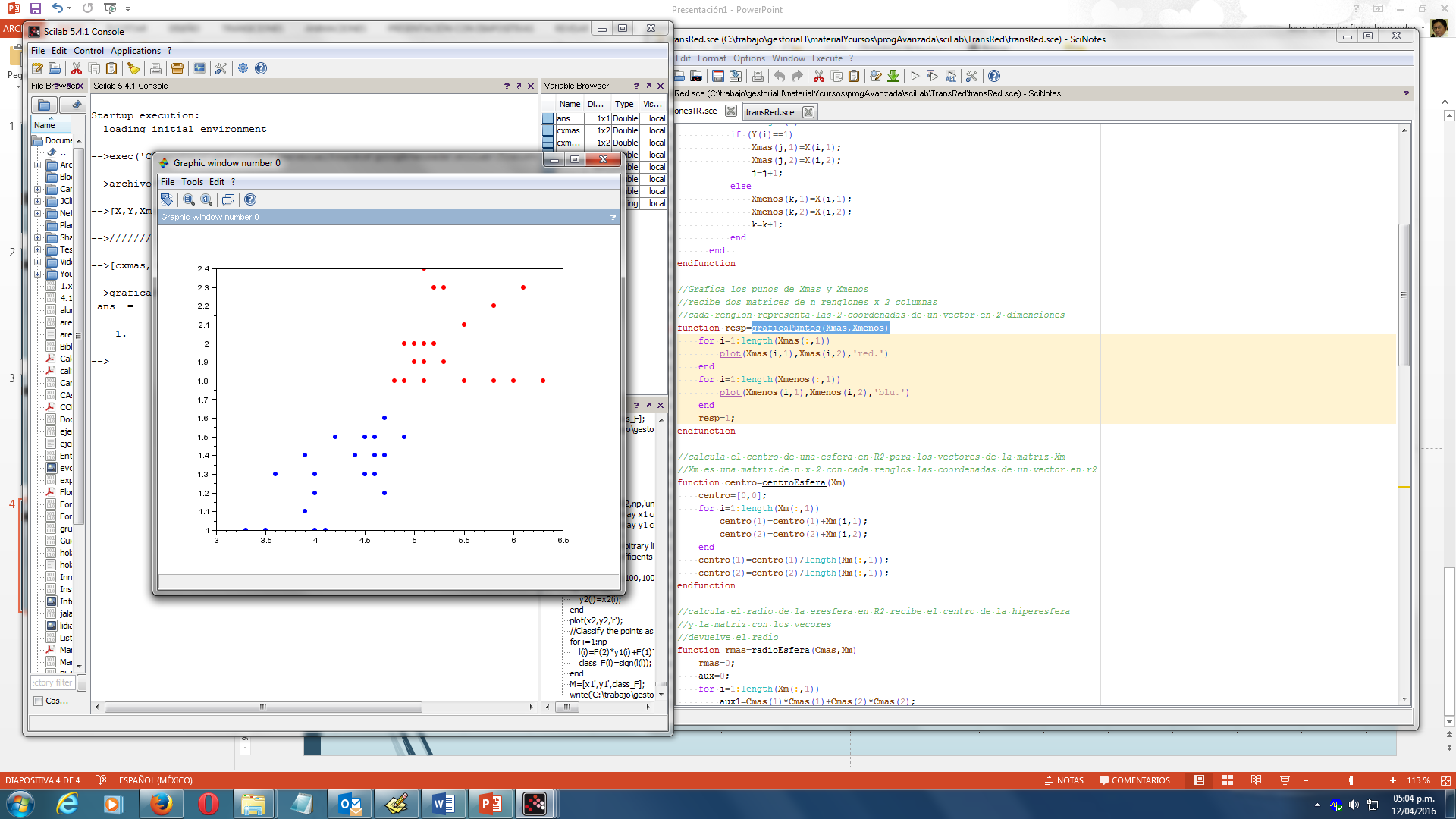
Entrenamiento de SVM



Suponga que tiene el ***l*** vectores independientes e idénticamente distribuidos con una distribución de probabilidad desconocida.

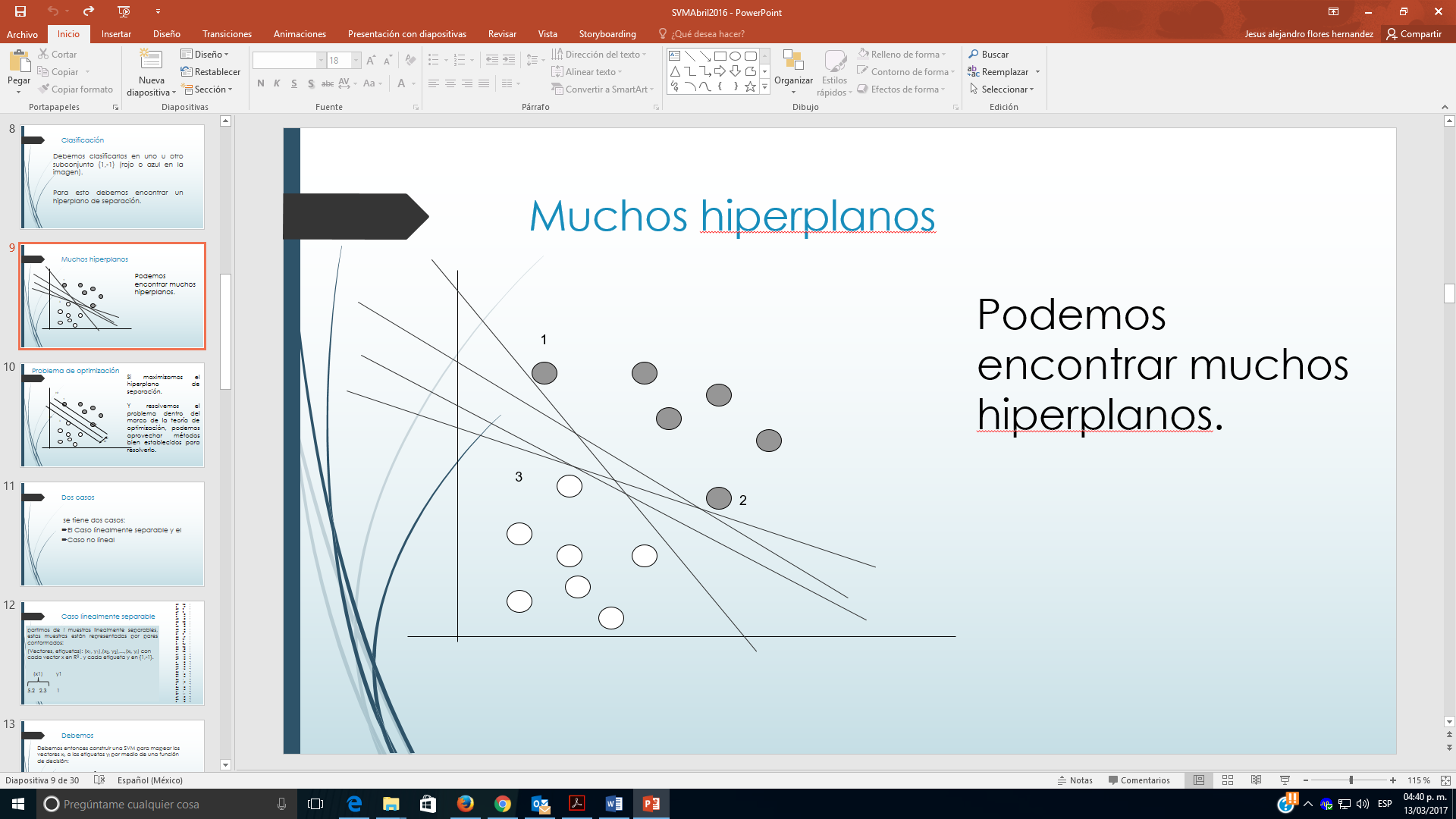
Que mapean un conjunto de vectores en R2 a un conjunto de etiquetas en R1

x 🡪 y



Debemos clasificarlos en uno u otro subconjunto {1,-1} (rojo o azul en la imagen).

Para esto debemos encontrar un hiperplano de separación. Podemos encontrar muchos hiperplanos:



Si encontramos el hiperplano de máxima anchura podemos convertir esto en un problema de maximización.

M+

M-

2

3

1

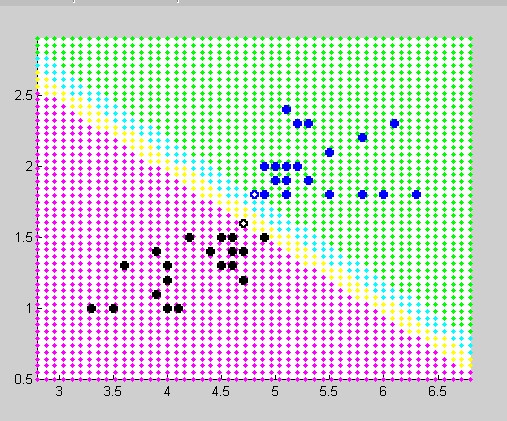
H1

H2

En el caso linealmente separable, los datos pueden separarse por un hiperplano orientado también llamado hiperplano de separación con ecuación: w x +b=0, con w un vector de pesos y b(w0 en la definición de hiperplano orientado) se conoce como sesgo, la distancia del hiperplano al origen se define como:



En la figura se muestra un conjunto de vectores linealmente separables:



Fig(1.3.2) Vectores Linealmente Separables

Fig Vectores Linealmente Separables

Conjunto de vectores linealmente separables, el conjunto de puntos azules y negros representan los vectores xi, aquí se muestran clasificados como y en {1,-1} representado por negro =1, azul= -1.

En la figura 1.3.1, se muestra la distancia entre el hiperplano de separación hasta el dato positivo más próximo M+, lo mismo para M-, que representa la distancia entre el hiperplano de separación y el dato negativo más próximo. M=M++M- se conoce como el margen del hiperplano de separación. Debemos encontrar el hiperplano de separación con margen mayor.

Un conjunto de entrenamiento es linealmente separable si existen  y , tales que todos los datos de entrenamiento satisfacen las siguientes restricciones:

Para el caso linealmente separable se tiene una solución de programación cuadrática convexa:



Sujeto a las restricciones:



La solución de este problema de optimización se encuentra en el punto silla de la función lagrangiana o Lagrangiano primal dada por:



Donde la para i=1..l, son los multiplicadores de lagrange, representando esta solución en su forma dual, tenemos:



Que equivale a:



Sujeto a las mismas restricciones, y donde H es una matriz de *l x l* tal que Hij=yiyjxixj, y d es un vector con todos los elementos con valor de -1.

Para el caso no lineal, el espacio de entrada es demasiado restrictivo para un uso práctico. En este caso se puede mapear los puntos a un espacio superior buscando luego el hiperplano óptimo de separación en dicho espacio.

Para el mapeo de los datos se usa una función  utilizando una función simétrica llamada kernel

Como se sabe que según el teorema de Lagrange, un problema de optimización no lineal puede (con alguna restricción) resolverse por medio de la función de lagrange dual (Lagrangiano Dual):



Sujeto a:

0≤ α ≤ C



En el problema de separar las dos clases, puede ser difícil encontrar un separador lineal por lo que las SVM pueden mapear los vectores de entrada hacia un espacio de dimensión alta por medio de un *Kernel,* por lo general los datos serán linealmente separables en algún espacio de rasgos de muy alta dimensión, se tomará un kernel que haga un mapeo de los vectores hacia un espacio de rasgos con la menor dimensión posible.

La idea de la función kernel es permitir la realización de operaciones en el espacio de entrada en vez de hacerlo en el espacio de rasgos de alta dimensión. Como el producto punto no necesita ser evaluado en el espacio de rasgos, el costo computacional es mucho menor.

Un producto punto en el espacio de rasgos tiene un kernel equivalente en el espacio de entrada . Si k es una función simétrica positiva definida en el espacio L2 que cumple con la condición



Para toda g ≠ 0 para la cual 

Tiene una expansión



En L2, donde 

Esto es conocido como Condiciones de Mercer. Los kerneles más conocidos y probados son:

* Polinomiales de grado d.
* Funciones de Base Radial.
* Perceptrones multicapas.

El mapeo polinomial es muy usado en la modelación no lineal, existen dos formas de hacerlo:





El segundo kernel es frecuentemente elegido porque evita problemas de que el hesiano llegue a 0. Si se toma el primer tipo de kernel con d=1, tendremos un producto punto en el mismo espacio de entrada. Se suele llamar e esto un kernel lineal.

Las funciones de Base Radial (RBF) Gaussianas. Emplean algún método para determinar un subconjunto de centros, su forma es:



### Procedimiento.

Desarrollo de la práctica

Situación problema: Suponga que necesita separa un conjunto de datos usando máquinas de soporte vectorial.

Ejercicios:

1. Escriba un programa para la generación del conjunto de datos en JAVA, apóyese en el ejemplo dado en SCILAB.
2. Escriba un programa que grafique el conjunto de entrada en JAVA, apoyese en el ejercicio dado en SCILAB.
3. Para un conjunto de datos linealmente separable aplique los algoritmos de SVM para encontrar un hiperplano de separación.
4. Genere un conjunto de datos aleatorio y busque el hiperplano de separación usando SVM, repita usando Redes neuronales, compare.

Entregables

Código fuente de la aplicación.

# Módulo II.

Objetivo: Aplicar los conceptos programación en paralelo y programación en tiempo real en sus programas.

## 1 Práctica Programación en paralelo.

### Objetivo

El alumno será capaz de comprender los conceptos de programación e paralelo y programar algoritmos que utilicen subprocesos y técnicas de programación en paralelo.

### Equipo

* Computadora
* Framework .NET 3.5 instalado
* Sharp Develop instalado

### Materiales

Hojas de pales

* Lápiz
* Papel

### Descripción

Sustento teórico

La necesidad que surge para resolver problemas que requieren tiempo elevado de cómputo origina lo que hoy se conoce como computación paralela. Mediante el uso concurrente de varios procesadores se resuelven problemas de manera más rápida que lo que se puede realizar con un solo procesador. Una computadora paralela es un conjunto de procesadores que son capaces de trabajar cooperativamente para solucionar un problema computacional. Esta definición es muy extensa e incluye supercomputadoras que tienen cientos o miles de procesadores, redes de estaciones de trabajo o estaciones de trabajo con múltiples procesadores. La programación paralela es diferente para cada arquitectura. La programación con hilos se aplica principalmente a las arquitecturas de multiprocesadores con memoria compartida y la programación con paso de mensajes a las arquitecturas de multicomputadoras.

A nivel de hardware para aprovechar que los circuitos del CPU trabajen en paralelo se utiliza la técnica de pipeline esta es un conjunto de elementos procesadores de datos conectados en serie, en el cual la salida de un elemento está conectado en la entrada del que sigue. los elementos de esta técnica son ejecutados en paralelo, debe haber un tipo buffer insertando los elementos.

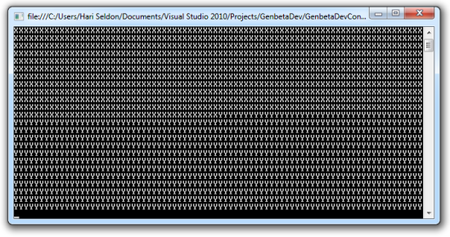
En NET

Una de las formas de hacer paralelismo en por medio de ejecutar nuestra aplicación en diferentes y concurrentes hilos de ejecución. Los llamados Tread.

Veamos un ejemplo:

|  |
| --- |
|  |
| using System;  using System.Threading;  namespace GenbetaDevConsolePararell  {  class Program  {  static void Main()  {  escribeUnaLetraX();  escribeUnaLetraY();  Console.ReadKey();  }  static void escribeUnaLetraX()  {  escribeUnaLetra('X');  }  static void escribeUnaLetraY()  {  escribeUnaLetra('Y');  }  static void escribeUnaLetra(Char letra)  {  for (int i = 0; i < 1000; i++) Console.Write(letra);  }  }  } |

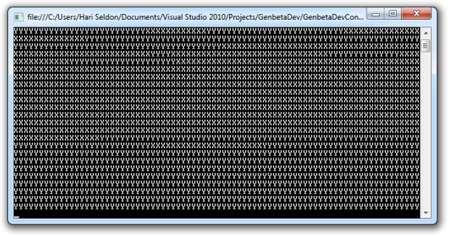
Si ejecutamos este código tendremos la salida:



Apliquemos ahora un hilo, sustituimos la función main por la siguiente:

|  |
| --- |
|  |
| static void Main()  {  Thread t = new Thread(escribeUnaLetraY);  t.Start();  escribeUnaLetraX();  Console.ReadKey();  } |

Si ejecutameos veremos la salida:



Vemos ahora que la salida se entrelaza, esto debido a que los procesos trabajan en paralelo.

### Procedimiento.

Desarrollo de la práctica

Situación problema: Suponga que quiere reducir el tiempo de cómputo para el procesamiento de datos.

Ejercicios:

1. Suponga que quiere reducir el tiempo de ordenamiento de un gran volumen de datos. Investigue sobre un algoritmo de ordenamiento paralelizable.
2. Programe el algoritmo que encontró en el ejercicio 1.
3. Investigue sobre otros algoritmos paralelizables para procesamiento de grande volúmenes de información.

Entregables

Código fuente de la aplicación.

## 2 Práctica Programación en tiempo real.

### Objetivo

El alumno será capaz comprender los conceptos de programación en tiempo real.

### Equipo

* Computadora
* Framework .NET 3.5 instalado
* Sharp Develop instalado

### Materiales

Hojas de pales

* Lápiz
* Papel

### Descripción

Sustento teórico

Un sistema es determinístico si para cada estado y cada conjunto de entradas pueden ser determinados un único conjunto de salidas y el próximo estado del sistema. El tiempo entre la presentación de un conjunto de entradas a un sistema y la aparición de todas las salidas asociadas se llama tiempo de respuesta del sistema. Un sistema en falla es un sistema que no puede satisfacer uno o más de los requisitos presentados en la especificación del sistema.

Un sistema de tiempo real puede definirse, entonces, como un sistema que debe satisfacer restricciones explícitas en el tiempo de respuesta o arriesgarse a severas consecuencias, incluida la falla. Por lo tanto, un sistema de tiempo real es un sistema que responde a un estímulo externo dentro de un tiempo especificado. Su eficiencia no solo depende de la exactitud de los resultados de cómputo, sino también del momento en que los entrega. La predictibilidad es su característica principal. A diferencia de los sistemas tradicionales, que tienden a distribuir en forma equitativa los recursos disponibles entre las diferentes tareas a ejecutar, los sistemas de tiempo real deben asegurar la distribución de recursos de tal forma que se cumplan los requerimientos de tiempo.

Los sistemas de tiempo real pueden dividirse en dos tipos diferentes, en función de su severidad en el tratamiento de los errores que puedan presentarse:

Sistemas de tiempo real blandos o Soft real-time systems: pueden tolerar un exceso en el tiempo de respuesta, con una penalización por el incumplimiento del plazo.

Sistemas de tiempo real duros o Hard real-time systems: la respuesta fuera de término no tiene valor alguno, y produce la falla del sistema.

La idea de sistema de tiempo real no debe asociarse únicamente con la velocidad de respuesta del sistema. En cambio, tiempo real implica sí necesariamente que los tiempos de respuesta estén acotados. Debe conocerse exactamente el tiempo que le tomará al sistema responder a un determinado evento. Este tiempo debe ser invariable, fundamentalmente, y además debe ser lo suficientemente rápido como para no producir una falla, por supuesto.

Si bien los primeros trabajos en sistemas de tiempo real se enfocaron a arquitecturas simples y dedicadas, los actuales sistemas de computadoras de propósitos generales y sus respectivas aplicaciones se están transformando en sistemas complejos que generalmente requieren los atributos de los sistemas de tiempo real: múltiples tareas coexisten en el sistema y cada una de ellas tiene diferentes requerimientos de tiempo; se requiere de mecanismos eficientes de comunicación entre las tareas; acceso simultáneo a dispositivos y redes; configurabilidad; adaptabilidad; etc.

Presentemos algunas definiciones más antes de asociar la anterior definición de sistema de tiempo real con la idea de sistemas de computadoras.

Llamaremos sistema de computadora a todo sistema que incluya un microprocesador, tanto una computadora personal, como un sistema cerrado.

Un sistema operativo es una colección especializada de programas que provee los mecanismos para interrelacionar los distintos componentes de un sistema de computadora.

Un embedded system, o sistema dedicado, es un software utilizado para controlar un hardware determinado que es parte de un sistema mayor. Generalmente se trata de sistemas cerrados en los cuales el usuario solo tiene acceso a un conjunto limitado de funciones, o a ninguna. Estos sistemas tienen un conjunto específico de tareas asignadas que no puede modificarse si no es por medio de una reprogramación total. Un ejemplo clásico de un sistema embebido es el software de un microcontrolador que es parte de una planta, y que está a cargo, por ejemplo, de controlar la temperatura de la misma.

Un sistema operativo de propósitos generales difiere de un sistema embebido en que no tiene asignado un único conjunto de tareas a realizar. Puede ser reprogramado completamente por el usuario. Tal es el caso de una PC con sistema operativo UNIX, con un programa ejecutándose. Este sistema operativo permite ejecutar tareas tan disímiles como un procesador de texto o un control de temperatura.

### Procedimiento.

Desarrollo de la práctica

Situación problema: Suponga que necesita determinar las características de los sistemas de tiempo real para poder hacer más eficientes los procesos de su organización.

Ejercicios:

1. Investigue sobre las aplicaciones de los sistemas de tiempo real.
2. Identifique 5 ámbitos en los que es necesario la aplicación de estos sistemas.
3. Investigue las primitivas para la aplicación de restricciones de tiempo real en Java.
4. Proponga un diseño de sistema de tiempo real para el control de un robot.

Entregables

Archivo Word con el resultado de los ejercicios.

## Referencias:

1. [flores 2003] Flores Hernández Jesús Alejandro, "Entrenamiento de SVM", Tesis.
2. [itam 2001] <http://biblioteca.itam.mx/estudios/estudio/estudio10/sec_16.html>
3. [unam2005] <http://www.comoves.unam.mx/numeros/articulo/2/la-inteligencia-artificial-hacia-donde-nos-lleva>
4. [iit01] <http://www.iit.upcomillas.es/pfc/resumenes/467f930a0aa05.pdf>
5. Video:  <https://www.youtube.com/watch?v=1NxnPkZM9bc&list=PLdWEvb4DrLQ4595qWwQhBCyKFYhbqLBCJ>
6. video : <https://www.youtube.com/watch?v=LXGaYVXkGtg>
7. [buap01] <http://www.cs.buap.mx/~mtovar/doc/ProgConc/ProgramacionParalela.pdf>).
8. [uach01] <http://atc2.aut.uah.es/~eliseo/Tema_3.pdf>
9. [uv01] <http://www.uv.es/gomis/Apuntes_SITR/Programacion.pdf>
10. [unlp01]<http://sedici.unlp.edu.ar/bitstream/handle/10915/50091/Documento_completo.pdf?sequence=1>
11. [Flores y Barrera 2007] <https://books.google.com.mx/books?id=rbCUa-nkXU8C&pg=PA51&lpg=PA51&dq=algoritmo+transred+maquinas+de+soporte+vec&source=bl&ots=WffjGkAXUU&sig=y1FoO6T4MD4GXBVYiWbM9c2gvy4&hl=es-419&sa=X&ved=0ahUKEwj1-eCPv97KAhUN0WMKHQIdAqgQ6AEILzAD#v=onepage&q=algoritmo%20transred%20maquinas%20de%20soporte%20vec&f=false>
12. <https://www.genbetadev.com/paradigmas-de-programacion/breve-introduccion-a-la-programacion-en-paralelo-en-net>