

## Reconocimiento de Caras Humanas mediante una Red Neuronal con Ada95

Araceli Sánchez Sánchez  
Universidad de Salamanca  
Plaza de la Merced s/n  
37008 Salamanca

Enrique Cabello Pardos  
Universidad de Salamanca  
Plaza de la Merced s/n  
37008 Salamanca

Luis Pastor Pérez  
U. P. M.  
Facultad de Informática  
Campus de Montegancedo  
28860 Madrid

### INTRODUCCION

El rostro humano está lleno de estímulos visuales, detalles y matices que hacen que seamos diferentes unos de otros, incluso en el caso de los gemelos es posible encontrar algún detalle que los diferencie. Sin embargo, en el mundo de los bloques empleado habitualmente en Visión Artificial todo suele presentar una estructura mucho más regular y lo único que suele interesar es que el robot pueda moverse libremente sin chocar con los obstáculos. La identificación de personas por medio de su cara es un problema complejo de resolver por los muchos matices que presenta el rostro humano, sin embargo el cerebro resuelve este problema de forma rápida y fiable [12].

Desde el siglo pasado ha parecido interesante construir un sistema automático para la identificación de individuos a través de su cara. En 1889 Sir Galton [1] buscaba fórmulas matemáticas para solucionar el problema. Hay que considerar que cuanto más información se disponga del individuo mejor será nuestro sistema, así si además de contar con una imagen de su cara dispusiéramos de información relativa a la voz o imágenes con movimiento en las que se pudiera apreciar detalles de la forma de andar del sujeto, nuestro sistema será mucho más eficiente. También hay que tener en cuenta

que las personas se equivocan a la hora de identificar a otras, por lo que no es realista esperar que el sistema construido pudiera alcanzar un 100 % de resultados positivos.

Los contextos en los que podría ser útil un sistema de estas características son :

**Entornos policiales** o similares, donde lo que se pretende es localizar una persona en una base de datos de caras. En este caso la base de datos estaría formada por un elevado número de individuos de forma que de cada uno de ellos se dispone de un número reducido de imágenes. El sistema no proporcionaría una respuesta en tiempo real y la respuesta correspondería a aquellos individuos de la base de datos que más se parezcan al individuo buscado.

**Entornos de control de accesos**, en este caso lo que se busca es la identificación de personas en tiempo real. Para estos sistemas la base de datos estará formada por un número mucho menor de personas que en el caso anterior, pero se dispondrá de más imágenes por persona. El sistema proporcionaría una respuesta en tiempo real que correspondería con el individuo

de la base de datos que más se aproxima al buscado.

El presente trabajo representa sólo a los procesos relacionados con el reconocimiento de individuos, es decir, comprende las etapas correspondientes al preprocesamiento de las imágenes y las relativas al empleo de una Red Neuronal, en las que se incluye el entrenamiento y prueba de la red.

En una aplicación real el sistema engloba todo el proceso que hay desde la captación de la imagen del sujeto hasta su reconocimiento, nuestro trabajo se está desarrollando en dicha línea, buscando incluir la etapa de adquisición de las imágenes que incluirá el empleo de una cámara CCD que capture las imágenes y las pase a la etapa de identificación del sujeto.

## DATOS EMPLEADOS

La base de datos empleada está formada por imágenes correspondientes a 30 personas, con 10 imágenes por sujeto. Todos los individuos son de raza blanca, y de sexo masculino, presentando algunos de ellos rasgos asiáticos. La base de datos utilizada cuenta con individuos con barba, bigote, gafas, pelo largo, corto, etc, intentando siempre que varios individuos presenten la misma característica para dificultar el reconocimiento.

Las imágenes tienen la siguiente distribución : 2 mirando de frente a la cámara, 2 mirando a la derecha de la cámara, 2 mirando a la izquierda de la cámara, 2 mirando hacia arriba, 2 mirando hacia abajo.

El tamaño original de las imágenes es de 512 x 342 bytes y provienen de la Universidad de Berna ( Suiza ) [2].

La siguiente figura nos muestra las 10 imágenes empleadas correspondientes a uno de los individuos de la base de datos.



**Fig. 1 :** *Imágenes pertenecientes a uno de los individuos empleados.*

Las imágenes aparecen borrosas ya que son las empleadas en la red de Neuronas tras la

etapa de preprocesamiento en la que se les ha aplicado el operador de Gauss.

Destacar de la figura que para que se puedan apreciar mejor los detalles de las imágenes han sido ampliadas un 300 % con respecto a su tamaño original.

## PREPROCESAMIENTO

Las imágenes se someten a un procesamiento de bajo nivel [3,4] con el fin de :

*Evitar la pérdida de información útil.* Si perdemos detalles decisivos a la hora del reconocimiento, lógicamente estamos provocando la disminución del porcentaje de aciertos que podría tener el sistema, porcentaje que nos interesa que sea lo más elevado posible.

*Disminuir el tiempo de procesamiento de las imágenes.* Incrementar el tratamiento al que sometamos las imágenes lo que provoca es el incremento del tiempo de funcionamiento del sistema. Este tiempo no puede ser reducido en todas las etapas debido al carácter crítico de las mismas (la etapa de entrenamiento de la red, por ejemplo) por lo que resulta interesante tratar de reducirlo en las que nos lo permiten para conseguir así un proceso de identificación mucho más rápido.

El hecho de trabajar con imágenes completas proporciona gran cantidad de información visual. El tamaño de las imágenes empleadas es de 512 x 342 bytes con una resolución de 256 niveles de gris lo que viene a

suponer 2 Megabytes por sujeto aproximadamente. Hemos de recordar que la base de datos empleada esta formada por 30 individuos, si esto lo aplicásemos a una base de datos policial o de una empresa los problemas de almacenamiento y tratamiento de datos aumentan rápidamente. El procesamiento al que se sometan las imágenes debe estar encaminado a reducir el espacio necesario.

Es por ello que se optó por reducir el tamaño de las imágenes hasta 32 x 22 pixeles, ya que además se simplificaba la estructura de la red neuronal necesaria.

Para realizar la reducción del tamaño de las imágenes se emplea una pirámide jerárquica o pirámide de Gauss de 5 niveles[5,6]. Una pirámide jerárquica está formada por un conjunto de imágenes del mismo objeto, pero con diferentes resoluciones ( tamaños ) donde en cada nivel se reduce la resolución anterior a la mitad. Con los 5 niveles empleados obtenemos una pirámide con resoluciones:

$$512 \times 342 \Rightarrow 256 \times 171 \Rightarrow 128 \times 85 \Rightarrow 64 \times 42 \Rightarrow 32 \times 22$$

Para la construcción de la pirámide se emplea el operador de Gauss, cuya expresión es la que aparece a continuación :

$$G = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

**Ecuación 1 : Operador de Gauss**

El operador de Gauss hace que cada pixel de la imagen englobe información de los que le rodean, lo que origina la aparición de información redundante, por lo que tras la aplicación de dicho operador es necesaria la eliminación de filas y columnas de la imagen,

consiguiendo así reducir la resolución inicial de la imagen a la mitad.

Un ejemplo que ilustra la aplicación de la pirámide gaussiana sobre una de las imágenes de la base de datos lo muestra la siguiente figura:



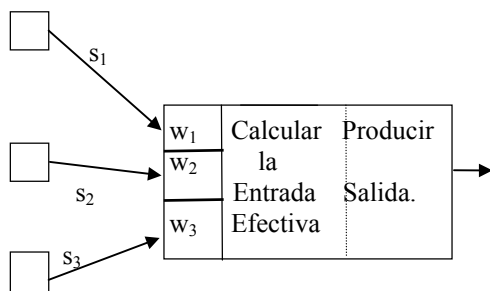
**Fig. 2 :** Pirámide Gaussiana de una de las imágenes.

Como se aprecia en la figura la aplicación del operador de Gauss ha eliminado detalles secundarios de la misma manteniendo los rasgos más importantes.

## REDES NEURONALES

Las redes de neuronas están formadas por múltiples unidades de procesamiento llamadas neuronas, siguiendo el modelo de las redes de neuronas de los sistemas biológicos vivos [14].

Una **neurona** es un dispositivo simple formado por una serie de entradas y una única salida. La figura siguiente nos muestra el esquema general de una neurona :



**Fig. 3 :** Esquema general de una neurona

Cada neurona acepta como entrada las salidas procedentes de otras neuronas, siendo la entrada efectiva a la neurona la suma ponderada de la entradas reales a dicha neurona.

Las neuronas se caracterizan por su estado de activación, que es un valor que oscila entre 0.0 y 1.0, de forma que si el estado de activación de la neurona es 0.0 indica que tal neurona no esta activa, mientras que cualquier valor distinto de este corresponde a una neurona

activa. La salida de la neurona es el estado de activación.

Una **Red de Neuronas** esta formada por múltiples capas de neuronas donde cada neurona de cada capa esta conectada con todas las neuronas de la capa siguiente. Esta conexión esta ponderada mediante un peso, que es el que determina, junto con la salida, la activación o no de la neurona.

En una red siempre vamos a distinguir una capa de entrada, a la que se le proporcionan los datos, una capa de salida, que es la que produce los resultados, y un número indeterminado de capas intermedias.

El número de neuronas en la capa de entrada depende de la cantidad de información de que dispongamos, las de la capa de salida, dependen del problema que tratemos de resolver, siendo un buen criterio el disponer en la capa de salida de tantas neuronas como elementos queramos diferenciar. Para las capas intermedias no existe un criterio para determinar ni el número de capas, ni el número de neuronas en cada capa.

Teniendo en cuenta estas consideraciones se construyó una red de neuronas para resolver el problema formada por una capa de entrada de 674 neuronas, una capa intermedia de 100 neuronas y una capa de salida de 30.

El sistema de entrenamiento empleado es de tipo supervisado que consiste en que durante la fase de entrenamiento se le proporcionan a la red tanto los patrones que queremos que diferencie, como los resultados

que debería proporcionar cada patrón, de forma que la red va reajustando los pesos de las conexiones en función del error cometido.

El algoritmo concreto empleado es el Backpropagation ( retropropagación ) por ser el mejor conocido, más extendido y del que se conoce por bibliografía que proporciona buenos resultados.

El conjunto empleado para las pruebas es diferente del empleado para el entrenamiento de la red, de las 10 imágenes se seleccionó una en la que el individuo aparecía mirando frente a la cámara para realizar el reconocimiento, utilizando las 9 restantes para el entrenamiento de la red.

Para entrenar la red se estableció un número de iteraciones sobre los patrones de entrada, tras las cuales se comprobó si la red había logrado aprender o no. Si la respuesta es negativa, se iniciaba un nuevo ciclo de entrenamiento con más iteraciones.

### DISEÑO HOOD

Como metodología de diseño se empleo HOOD [7,8,11].

El problema a resolver consiste en implementar un simulador de redes neuronales para una topología de red y pensando en un algoritmo de entrenamiento concreto. A grandes rasgos el problema se podría describir como un sistema al que el usuario le proporciona unos datos, la red los procesa y genera unos resultados. La siguiente figura representa esta situación :

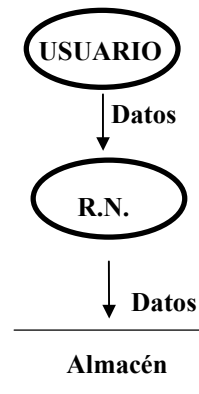


Fig. 4 : Esquema General del Sistema.

El diagrama HOOD correspondiente es el que aparece en la siguiente figura :

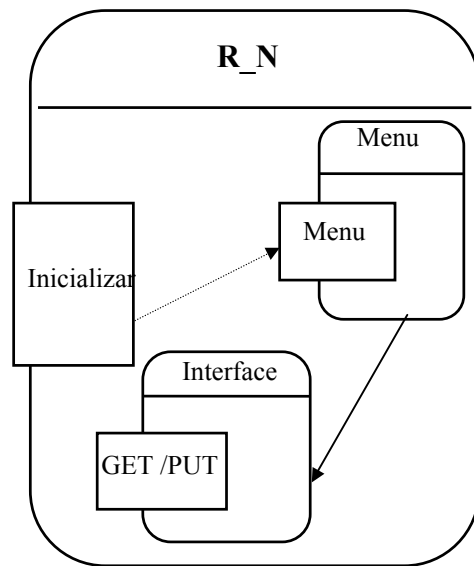
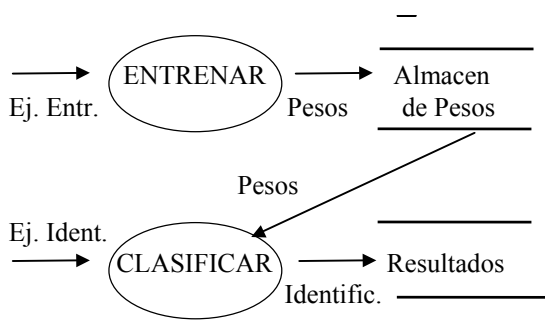


Fig. 5 : Diagrama HOOD estricto de la aplicación.

Como se aprecia en la figura es necesario un objeto Menu, que nos va a permitir seleccionar la operación a realizar con el simulador, también aparece un objeto Interface, que se va a encargar de la comunicación con el usuario.

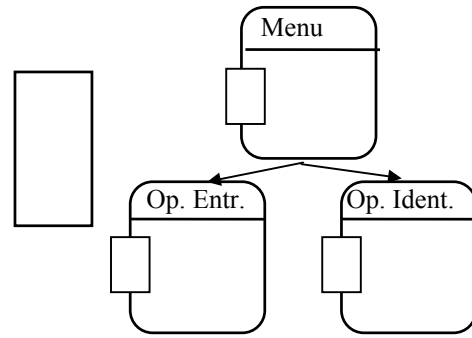
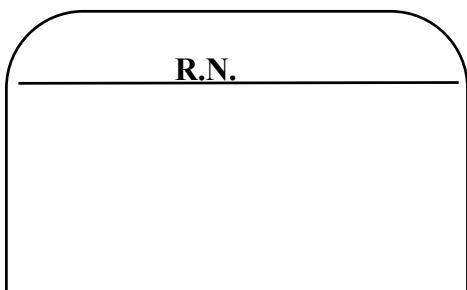
La aplicación que nos ocupa forma parte de un proyecto más amplio cuyo objetivo final es encontrar el método más adecuado para el reconocimiento de individuos a través de su cara, por lo que la construcción del simulado se hizo con el único fin de conseguir resultados para compararlos con los proporcionados por otras técnicas empleadas. Es por esto por lo que finalmente se prescindió del objeto Interface, realizándose la comunicación con el usuario de forma textual.

Se podría profundizar tanto en la descripción del problema, como en el diagrama HOOD considerando que a este simulador el usuario le va a suministrar una serie de datos en la etapa de entrenamiento de la red, al finalizar esta etapa la red nos proporciona unos pesos adecuados para las conexiones. Durante la fase de pruebas, a la red se le entregara el patrón de un determinado individuo para la identificación del mismo empleando los pesos conseguidos en el entrenamiento. Esto es lo que nos muestra la siguiente figura :



**Fig. 6** Esquema simplificado del problema

El diagrama HOOD [13] correspondiente es el que muestra la figura siguiente :



**Fig. 7** : Esquema simplificado del sistema.

Como vemos es necesario un objeto Menú que va a ser el encargado de llamar a los objetos que realizan cada una de las operaciones que vamos a poder realizar con el simulador.

## RESULTADOS

Se recogieron datos durante los periodos de entrenamiento y de pruebas de la red, en el primer caso se tomaban valores acerca del error cuadrático medio que se producía en función del número de iteraciones y en el segundo caso los datos eran relativos al porcentaje de aciertos conseguido.

La siguiente tabla nos muestra los valores del error cuadrático medio en función del número de iteraciones :

| Nº DE ITERACIONES | ERROR CUADRATICO MEDIO |
|-------------------|------------------------|
| 1                 | 0.0619                 |
| 100               | 0.0130                 |
| 400               | 0.0104                 |
| 700               | 0.0082                 |
| 1000              | 0.0071                 |

**Tabla 1 :** *Error Cuadrático Medio en función del número de iteraciones.*

Destacar que se emplearon 1000 iteraciones para entrenar la red, así como la gran diferencia que existe entre el error de la iteración 1 y la 100 que es muy superior a la que se encuentra entre cualquier otro par de iteraciones. Esta diferencia es debida a que cuando se comienza el entrenamiento de una red se le asignan valores aleatorios a los pesos de las conexiones lo que provoca que la primera iteración tenga un error tan elevado.

La siguiente tabla nos muestra el porcentaje alcanzado para las distintas iteraciones consideradas :

| Nº DE ITERACIONES | PORCENTAJE DE ACIERTOS |
|-------------------|------------------------|
| 3                 | 23.33                  |
| 23                | 50                     |
| 30                | 53.33                  |
| 200               | 63.33                  |
| 1000              | 73.33                  |

**Tabla 2 :** *Porcentaje de aciertos en función del número de iteraciones.*

Como se aprecia en la tabla al aumentar el número de iteraciones aumenta el porcentaje de aciertos, pero no de forma proporcional.

El presente trabajo se engloba dentro de otro más amplio que lo que pretende es determinar cual es el mejor método para el reconocimiento de caras humanas a través de imágenes, la siguiente tabla nos muestra los

resultados comparativos obtenidos con las tres técnicas empleadas para ello:

Red Neuronal de una capa que es la red implementada.

Red Neuronal de dos capas, construida empleando un simulador existente [9].

Técnica LVQ ( Learning Vector Quantization ) [10].

|       | RED NEURONAL 1 CAPA | RED NEURONAL 2 CAPAS | LVQ     |
|-------|---------------------|----------------------|---------|
| Gauss | 73.33 %             | 43.00 %              | 96.00 % |

**Tabla 3 :** *Resultados de los métodos empleados.*

Además destacar que la red de una capa empleó 1000 iteraciones para el entrenamiento, tardando en esta etapa del orden de 10 días. La red de dos capas empleó 30.000 iteraciones con un tiempo del orden de semanas. La técnica LVQ utilizó 18.000 iteraciones en tan solo minutos.

## CALIDAD DEL SOFTWARE

Nuestro grupo de investigación emplea métodos de evaluación para ver la calidad del Software de los proyectos que está desarrollando.

Existen varios métodos que permiten medir la calidad de un Software determinado, para ello hay que medir una serie de características. Las empleadas para evaluar este trabajo se basan en la norma ISO / IEC 9126, junto con algunas modificaciones realizadas.



|  |             |
|--|-------------|
|  | Testability |
|--|-------------|

Considerando el entorno educativo e investigador en el que nos movemos se seleccionaron las siguientes características :

- Funcionalidad ( Functionality )
- Usabilidad ( Usability )
- Cambiabilidad ( Changeability )

La primera fue seleccionada ya que el Software debe satisfacer las necesidades definidas o implícitas. Nos interesa que el Software “ funcione “ de forma adecuada.

Muchas veces el Software que desarrolla una persona es utilizado por muchos otros dentro del grupo, o bien se incluye en un proyecto mayor. Es muy importante, por lo tanto, que sea fácil de utilizar por otros usuarios del grupo.

A veces, un proyecto surge de una línea inicial bastante aproximada, pero imprecisa o también puede suceder que en el Software desarrollado sea preciso modificar alguna parte. Por lo tanto esta es una característica que creemos que debe ser medida en este tipo de proyectos.

Cada característica principal que se pretende medir se divide en varias subcaracterísticas. Las utilizadas en este trabajo han sido :

| CARACTERISTICA | SUBCARACTERISTICA                                  |
|----------------|----------------------------------------------------|
| Functionality  | Suitability<br>Accurateness                        |
| Usability      | Understandability<br>Learnability<br>Operatibility |
| Changeability  | Simplicity<br>Analyzability<br>Changeability       |

**Tabla 4 :** *Tabla de Características y Subcaracterísticas empleadas.*

Para evaluar cada subcaracterística se formularon dos cuestiones puntuadas de 1 a 5 (1 para el caso más desfavorable, 5 para el más favorable). Las preguntas fueron contestadas por el programador de la aplicación y por el usuario de dicha aplicación de forma independiente uno de otro.

Los resultados que se muestra reflejan los resultados obtenidos 3 proyectos desarrollados en el entorno de Visión Artificial por el simulador implementado junto con los de otras aplicaciones de similares características en cuanto a número de líneas de código, complejidad de la aplicación y tiempo de desarrollo necesario. Los resultados aparecen en la siguiente tabla :

|                    | Ada95           | C                                           | C++                    |
|--------------------|-----------------|---------------------------------------------|------------------------|
| S. O.              | UNIX / DOS      | UNIX / DOS                                  | UNIX / DOS             |
| Tipo de Aplicación | Red de Neuronas | Segmentación Correspondencia entre Imágenes | Calibración de Cámaras |
| Functionality      | 4,75            | 3,6                                         | 4,3                    |
| Usability          | 4,6             | 3,7                                         | 4                      |
| Changeability      | 4,7             | 3,1                                         | 3,8                    |

**Tabla 5 :** *Resultados de la Calidad del Software*

Como se aprecia en la tabla anterior el software que presenta mejores resultados es el implementado utilizando el lenguaje de programación Ada95, no siendo sólo los mejores resultados sino que están muy próximos al valor máximo (5). La aplicación implementada con C++ presenta también buenos resultados, mientras que los del sistema desarrollado en C son bastante pobres.

Las conclusiones a las que llegamos son que la utilización de objetos en un proyecto

de investigación o desarrollo aumenta la calidad del mismo. El lenguaje C ofrece unos resultados de cambiabilidad muy pobres, mientras que la inclusión de objetos permite actualizar las distintas versiones que pueden ir desarrollándose de manera más sencilla.

## CONCLUSIONES

Las conclusiones a las que llegamos se resumen en los siguientes puntos :

1.- Es posible el diseño de un sistema de reconocimiento de caras humanas basado en imágenes, utilizando una reducción Gaussiana.

2.- Los resultados muestran que las Redes de Neuronas son un método adecuado para resolver el problema del reconocimiento.

3.- Las Redes Neuronales requieren un elevado tiempo de aprendizaje frente a un tiempo de reconocimiento muy pequeño.

4.- Se consiguen buenos resultados con imágenes que presentan ángulos de inclinación altos, lo que demuestra la robustez del método empleado.

5.- El uso de la metodología HOOD y el lenguaje de programación Ada95 han desembocado en un Software de gran calidad.

## BIBLIOGRAFIA

[1] Sir Francis Galton. Numeralized Profiles for Classification and Recognition. Nature 83, pp. 127-130, 31 March, 1910.

[2] Facial Data Base. University of Bern (iamftp.unibe.ch ). Switzerland, 1995.

[3] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, Andrew D. Back. Face Recognition: A Hibrid Neural Network Approach. Technical Report. CS - TR - 3608. 1996.

[4] D. Marr. Vision. W.H. Freeman, San Francisco, 1982.

[5] Andrés Bueno Sanz. Mecanismo Atencional Multinivel para la Correspondencia de Imágenes. Grado de Salamanca, 1995.

[6] Andrés Bueno Sanz y Enrique Cabello Pardos. A Matching Algorithm for a Stereo System Simulating Human Vision. Proceedings of the Fourteenth IASTED International Conference. Innsbruck, Austria, 20 - 22 Feb. 1996.

[7] Peter J. Robinson. Hierarchical Object - Oriented Design. Prentice Hall Object - Oriented Series. 1992.

[8] John Barnes. Programming in Ada95. Addison - Wesley, 1995.

[9] Institute for Parallel and Distributed High Performance Systems. Manual SNNS. University of Stuttgart. Report No. 3/94.

[10] LVQ Programming Team. The Lvq Program Package. University of Technology. Helsinki. 1995.

- [11] HOOD Working Group. HOOD Reference Manual. September, 1989.
- [12] Jonh B. Reppas, Anders M. Dale, Martin I. Sereno and Roger B. H. Tootell. La Visión, Una Percepción Subjetiva. Mundo Científico. Octubre 1996.
- [13] Enrique Cabello Pardos, M<sup>a</sup> Araceli Sánchez Sánchez y Luis Pastor Pérez. Una Red Neuronal con Ada95. Aplicación al Reconocimiento de Caras Humanas. Sexta Jornada Técnica de Ada-Spain. Feb. 1997
- [14] Jeffrey R. Carter, B. I. Sanden. Ada Design of a Neural Network. ACM Ada Letters, May / Jun 1994, pp. 61 - 73, Vol. XIV, No. 3.

## **AGRADECIMIENTOS**

Este trabajo ha sido parcialmente financiado por la Comisión Interministerial de Ciencia y Tecnología, dentro del proyecto titulado "Reconocimiento de Objetos Tridimensionales. Instrumentación en Multiprocesadores de Memoria Distribuida" Referencia TAP94 - 0305 C03 -02.