



**BENEMERITA UNIVERSIDAD  
AUTONOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA  
COMPUTACION**

**“ARQUITECTURA PARALELA DE PICS PARA EL  
PREPROCESAMIENTO DE IMAGENES”**

**TESIS**

**QUE PARA OBTENER EL GRADO DE :**

**LICENCIADO EN CIENCIAS DE LA  
COMPUTACION**

**PRESENTA:**

**MARIA ELENA FLORES DE ITA**

**ASESOR:**

**M.C. JOSE ESTEBAN TORRES LEON**

**PUEBLA, PUE. AGOSTO 2007**



# INDICE

|  |           |
|--|-----------|
| INTRODUCCION.....  | 1         |
| <b>CAPITULO I Preprocesamiento de imágenes digitales.....</b>                    | <b>3</b>  |
| 1.1 Conceptos básicos.....   | 3         |
| 1.2 Preprocesamiento .....   | 5         |
| 1.3 Tipos de ruido.....  | 6         |
| 1.4 Filtros.....   | 7         |
| 1.4.1 Filtros de suavizado.....  | 8         |
| 1.4.2 Filtros de realce de detalles.....   | 12        |
| 1.4.2.1 Bordes.....  | 13        |
| 1.4.2.2 Filtros paso alto.....   | 14        |
| <b>CAPITULO II Paralelismo, microcontroladores PIC y comunicación serie.....</b> | <b>16</b> |
| 2.1. Paralelismo.....  | 16        |
| 2.2. Microcontroladores.....   | 21        |
| 2.3. Arquitectura interna.....   | 22        |
| 2.4. PIC16F877.....  | 24        |
| 2.5 Programación del microcontrolador PIC16F877.....                             | 27        |
| 2.6. Puerto serie.....   | 27        |
| 2.7 Descripción del Puerto.....  | 27        |
| 2.8. Estructura de datos del puerto.....   | 28        |
| 2.9. Interfaz RS- 232.....   | 28        |
| 2.10. Max232 circuito integrado para conversión de niveles.....                  | 30        |
| <b>CAPITULO III Descripción del sistema.....</b>                                 | <b>31</b> |
| 3.1 Objetivos del trabajo de tesis.....  | 31        |
| 3.2 Análisis y diseño del trabajo de tesis.....                                  | 32        |
| 3.3 Descripción del sistema de procesamiento paralelo.....                       | 34        |
| 3.4. Infraestructura.....  | 35        |
| 3.5. Interfaces.....   | 36        |
| <b>CAPITULO IV Desarrollo del sistema.....</b>                                   | <b>37</b> |
| 4.1. Interfaz de la computadora con el circuito.....                             | 37        |
| 4.2. Recursos utilizados.....  | 37        |
| 4.3. Tarjeta paralela de preprocesamiento de imágenes.....                       | 38        |
| 4.4. Software de los microcontroladores.....                                     | 40        |
| 4.5 Software de la interfaz.....   | 45        |
| <b>CAPITULO V Pruebas, Conclusiones y Perspectivas.....</b>                      | <b>49</b> |
| 5.1. Pruebas de hardware.....  | 49        |
| 5.1.1 Pruebas electricas.....  | 49        |
| 5.1.2 Pruebas de conexión para el puerto serie.....                              | 49        |
| 5.1.3 Pruebas de sincronización con los microcontroladores.....                  | 49        |
| 5.1.4. Pruebas de comunicación.....  | 50        |

|   |    |
|---|----|
| 5.2 Pruebas de software.....                    | 50 |
| 5.2.1 Software de los microcontroladores.....   | 50 |
| 5.2.2 Software de interfaz con el usuario.....  | 50 |
| 5.3 Pruebas del funcionamiento del sistema..... | 50 |
| 5.3.1 Pruebas del usuario.....                  | 51 |
| 5.3.2 Pruebas de realce de detalles.....        | 54 |
| 5.4 Restricciones del sistema.....              | 55 |
| 5.5. Aplicaciones.....                          | 55 |
| 5.6 Conclusiones.....                           | 55 |
| 5.7 Perspectivas.....                           | 55 |

|                          |           |
|--------------------------|-----------|
| <b>BIBLIOGRAFIA.....</b> | <b>56</b> |
|--------------------------|-----------|

# INTRODUCCIÓN

En la actualidad se han construido diversas herramientas para poder tener una información mas completa para el análisis de problemas y se ha vuelto indispensable que dichas herramientas tengan que ver con imágenes digitales ya que con ellas es posible tener una información más amplia.

Las imágenes han tomado mucha importancia en el área de la inteligencia artificial en particular visión artificial. Para la solución de un problema las imágenes debe de estar libres de toda entidad que no se encuentre en la realidad esta es conocida como RUIDO producido por diversos agentes externos, esto hace que se desarrollen métodos para poder eliminar esta imperfección esta técnica es llamada pre-procesamiento de imágenes

Con los microcontroladores en especial se puede desarrollar casi cualquier tipo de proyectos y por lo tanto dar solución a muchas cosas que se dan en la vida diaria.

En general con estos dispositivos se pueden desarrollar muchas aplicaciones que pueden controlar desde electrodomésticos hasta el desarrollo del diseño e implementación de proyectos industriales

Se tiene gran diversidad de marcas en el mercado. Los más populares son Microcontroladores PIC por el fácil acceso a la información acerca de ellos, bajo costo y la difusión de herramientas compatibles con estos dispositivos.

## **El objetivo general de este proyecto es el siguiente:**

Dadas estas características lo que se pretende lograr es diseñar un proyecto en el cual se tenga un arreglo en paralelo de PICs 16F877 que efectúe el preprocesamiento de una imagen sea enviada desde la PC a este dispositivo el cual le eliminara el ruido y realizara los detalles de esta y la volverá a enviar a la PC para ser mostrada nuevamente ya corregida.

## **Los objetivos particulares son:**

- Diseño y desarrollo de una arquitectura basada en PIC
- Diseño de las interfases hardware-software para la transmisión de la imagen original y procesada entre el arreglo de PICs y la computadora.
- Diseño-desarrollo de los algoritmos que procesan una imagen en los PICs

Esta tesis esta conformada por los siguientes capítulos que se describirán a continuación:

**CAPITULO I Preprocesamiento de imágenes digitales.** Se mencionan conceptos básicos de preprocesamiento como los procedimientos de eliminación de ruido el tipo de ruido y las técnicas para realzar los detalles de una imagen.

**CAPITULO II Paralelismo, microcontroladores PIC y comunicación serie.** Se definen algunos modelos de arquitecturas paralelas, se describen las características principales de los PIC 16f877, su funcionamiento y la configuración y características de la comunicación serial.

**CAPITULO III. Diseño del sistema y del dispositivo paralelo.** Se describe el procedimiento que se siguió para el desarrollo tanto del software como del hardware.

**CAPITULO IV. Desarrollo del proyecto** Se presenta el funcionamiento del software y del hardware . Se lista el código de las rutinas que efectúan el preprocesamiento.

**CAPITULO V. Conclusiones** Se describen los resultados de las pruebas que se realizaron en el sistema..

# CAPITULO I

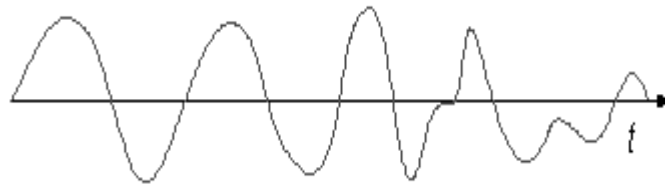
## PREPROCESAMIENTO DE IMÁGENES DIGITALES.

En el presente capítulo se describe lo que es el preprocesamiento de imágenes digitales describiendo diferentes métodos para realizarlo, esto es lo que se trabajará en la tesis.

Es necesario proporcionar algunos de los conceptos básicos que se refieren a imágenes digitales.

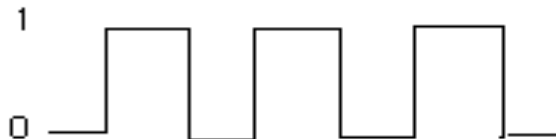
### 1.1 CONCEPTOS BASICOS

**a) SEÑAL ANALÓGICA.** Una señal analógica presenta como característica el que en un momento determinado puede tomar infinitos valores, pues la señal varía de forma continua en el tiempo. Es decir es un voltaje o corriente que varía de forma suave y continuamente por ejemplo el sonido que varía respecto al volumen [4] Ver Imagen 1



**Imagen 1** Ejemplo de señal de audio analógica

**b) SEÑAL DIGITAL.** Se representan con variables discretas las cuales se obtiene de dos niveles de tensión eléctrica distintos representándolos por cero denominado valor bajo y el uno llamado valor alto (ver Imagen.2), permitiendo así aplicar lógica y aritmética binaria. En computación los elementos más pequeños son llamados bits que toman los valores cero o uno, los cuales pueden agruparse en paquetes los más usuales compuestos de 8 bits llamados bytes.[4]



**Imagen 2** ejemplo de una señal digital. (Tren de pulsos)

c) **CONVOLUCION.** En el dominio de las frecuencias, a la multiplicación de dos señales se le llama convolucion. [1]

d) **PIXEL** (Picture element o elemento de una imagen) Se le llama a la menor unidad que compone una imagen digital, el cual se representa por un punto.[1]

Un píxel en escala de grises (blanco y negro), es representado por 8 bits, los cuales permite mostrar  $2^8=256$  tonalidades de grises.

Para que un píxel este a color se utiliza el modelo RGB que es la combinación de 3 colores (rojo, verde y azul) cada uno es representado por 8 bits por lo tanto cada color tiene 256 tonalidades. Combinado todas las posibilidades el total es de 16777216 colores.

f) **IMAGEN DIGITAL:** Es una matriz rectangular representada por columnas y filas de píxeles para imágenes en escala de grises (ver imagen 3) y para imágenes a color se representa por 3 matrices una para tonalidades de verde, otra para azul y otra mas para rojo.[1]

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 0   | 0   | 0   | 0   | 255 | 255 |
| 255 | 255 | 0   | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 0   | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 0   | 0   | 0   | 255 | 255 | 255 |
| 255 | 255 | 0   | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 0   | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 0   | 0   | 0   | 0   | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |
| 255 | 255 | 255 | 255 | 255 | 255 | 255 | 255 |

**Imagen 3.** Donde 255 es blanco y negro 0  
Imagen en escala de grises

g) **CALIDAD. (Resolución)** Se representa por la cantidad de píxeles en una imagen, entre más elementos tenga mas detalles de la imagen se pueden apreciar.[1]

h) **FORMATOS:** Es la forma en la que se guarda un archivo. En imágenes digitales los formatos mas utilizados son: JPG, GIF, PNG y BMP entre otros. [1]

Estos formatos tienen que ver mucho con la calidad de la imagen por ejemplo: el formato JPG elimina muchos detalles con el fin de que la imagen al almacenarla en un dispositivo ocupe poco espacio.

Por el contrario una imagen en formato BMP o mapa de bits requiere de mucho espacio ya que guarda más información porque no esta comprimida como los demás formatos entre mas compresión de datos menos detalles se tienen.

**i) ADQUISICIÓN.** Puede ser por muchos dispositivos por ejemplo escáner o cámara digital, etc. [1]

El proceso de captación de imagen por ejemplo con un escáner es el de iluminar una imagen con un foco y mediante espejos conducir la imagen a un dispositivo llamado: CCD (Charged Couple Device o dispositivo de transparencia de carga) que reacciona ante la luz transformándola en señales eléctricas según sea la intensidad y color de la luz que recibe.

Una cámara digital también cuenta con el sensor CCD después de filtrar los colores rojo, verde y azul. Cada célula fotosensible transforma la luz en una señal eléctrica analógica y la almacena en la memoria de la cámara.

## 1.2 PREPROCESAMIENTO

La visión artificial o compresión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional. Estos procesos se dividen en 6 áreas:

- Adquisición de la imagen.
- **Preprocesamiento.**
- Segmentación
- Descripción
- Reconocimiento.
- Interpretación.

La **adquisición de la imagen** como se menciona capta la imagen mediante algún dispositivo electrónico para poderla interpretar en ceros y unos.

El objetivo de **preprocesamiento** es el de mejorar la imagen y este se divide en dos acciones la eliminación de impurezas y el realce de detalles.

La **segmentación** su objetivo es dividir la imagen en las partes que la constituyen o los objetos que la forman. Por ejemplo en reconocimiento de caracteres, el objetivo de la segmentación es extraer caracteres individuales y palabras del fondo.

La **descripción**, que recibe también el nombre de selección de rasgos se dedica a extraer rasgos que producen alguna información de interés o rasgos que son básicos para diferenciar una clase de objetos de otra.

El **reconocimiento** es el proceso que asigna una etiqueta a un objeto basada en la información que proporciona la descripción.

. La **interpretación** es dar el significado al conjunto de objetos reconocidos.

Cuando una imagen es obtenida por cualquier dispositivo esta no es una copia exacta de la original. Esto se puede presentar a causa de cuestiones ambientales, iluminación o que el dispositivo que la obtuvo no tenga una buena resolución y esto hace que se introduzcan distorsiones.

A toda entidad que contamine una imagen es denominada “ruido”. Para que una imagen pueda ser procesada y obtener un resultado de calidad es necesario eliminar esta contaminación a estos métodos se les llama métodos de suavizado.

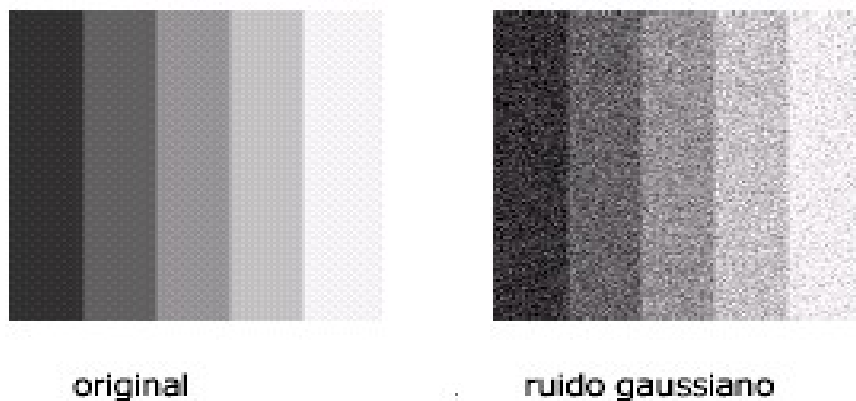
Cuando a las imágenes se les aplican técnicas de suavizado dan como resultado que algunos detalles pueden ser no muy visibles por lo tanto se aplican técnicas de realce. Estas técnicas además de devolver los detalles desdibujados pueden acentuar otros sitios de la imagen que no eran muy visibles en la imagen original y esto hace que se tenga una información más completa.

### 1.3 TIPOS DE RUIDO

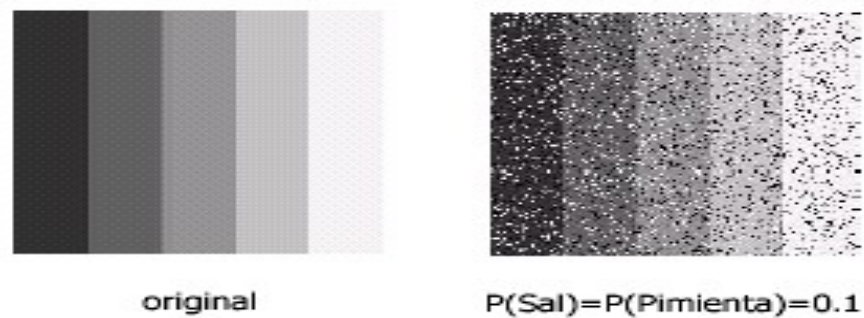
El ruido se puede presentar de diversas formas [1]:

- Gaussiano
- Sal y pimienta
- Ruido uniforme
- Multiplicativo.

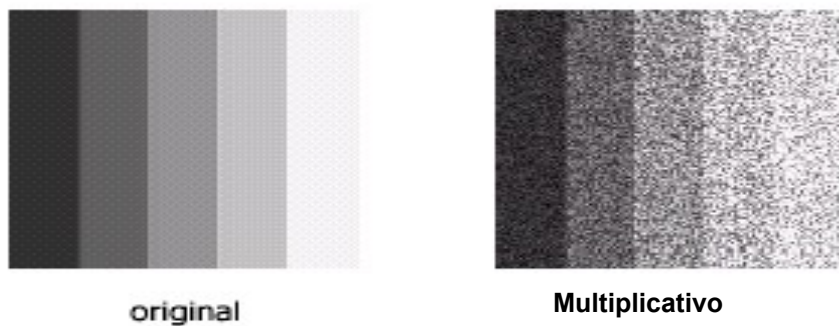
Ver algunos ejemplos en las Imágenes 4, 5 y 6



**Imagen 4.** Ruido Gaussiano.



**Imagen 5** Ruido Sal y Pimienta.



**Imagen 6** Ruido multiplicativo.

## 1.4 FILTROS

Los filtrados mas aplicados son los lineales, no lineales y temporales.

- Los **lineales**: Suavizado de una imagen con una plantilla predefinida.
- Los **no lineales**: Son operaciones que utilizan los píxeles del entorno de vecindad comúnmente 3x3 o 5x5. (Ver imagen 7 )
- Los **temporales**: Es el análisis de varias imágenes que son tomadas en diferente momento de la misma escena.[2]

$$\begin{array}{ccc}
 (x-1,y+1) & (x,y+1) & (x+1,y+1) \\
 (x-1,y) & (x,y) & (x+1,y) \\
 (x-1,y-1) & (x,y-1) & (x+1,y-1)
 \end{array}$$

**Imagen 7**Entorno de vecindad 3X3

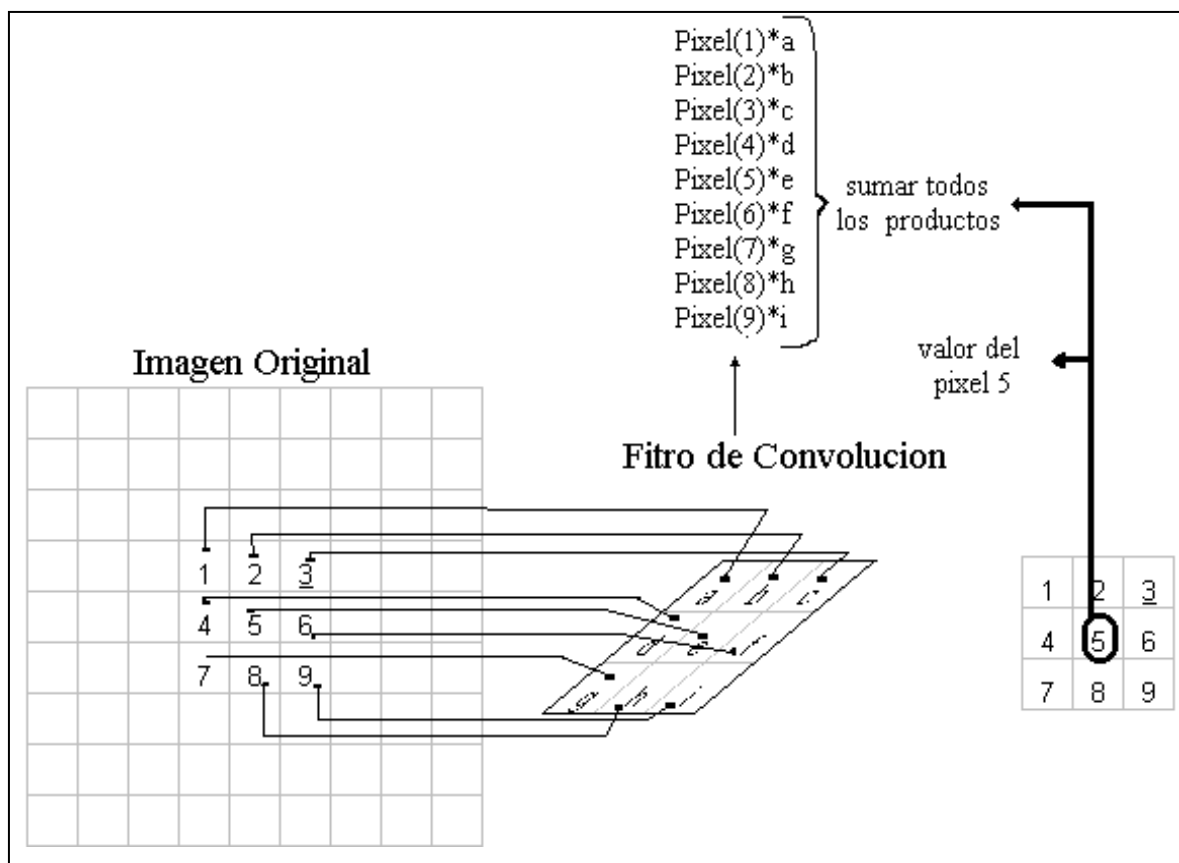
Las técnicas que se aplican para el tratado de imágenes en la mayoría de las ocasiones son los filtros espaciales esto quiere decir que se modifica el valor de un píxel en función del valor de sus vecinos. Para esto se aplica el producto del dominio espacial con las mascararas de convolucion (plantillas o filtros)

El producto de convolucion de f y h (en función de x ) producen una nueva función g y es dada por la siguiente igualdad:

$$g(x) = f(x) * h(x)$$

Operador de convolucion.

La convolucion se aplica en una imagen es: Imagen: F(x, y) donde f(x, y) = nivel de gris de un píxel en la posición (x, y)[2]. Ver imagen 8



**Imagen 8:** Aplicación del producto de convolucion (filtro 3x3)

### 1.4.1 FILTROS DE SUAVIZADO

Los filtros de suavizado mas usados son el de promedio de entorno de vecindad que pertenece a los filtros lineales y el método de la mediana que corresponde a los no lineales

### Promedio del entorno de vecindad.

Se tiene una imagen  $f(x, y)$ , que genera una nueva imagen  $g(x, y)$ , cuyo valor de tono para cada punto  $(x, y)$  se obtiene promediando los valores de intensidad de los píxeles de  $f$  que están definidos en el entorno de vecindad de  $(x, y)$ . La imagen suavizada se obtiene por medio de esta relación [3]:

$$g(x, y) = \frac{1}{P} \sum_{(n, m) \in S} f(n, m)$$

Para todos los  $n$  y  $m$  de  $f(n, m)$ ,  $S$  es el conjunto de coordenadas de el entorno de vecindad  $(n, m)$  predefinido incluyendo  $(n, m)$ .  $P$  es el número total de puntos del entorno de vecindad.

Por ejemplo (ver imagen 9) se tiene un entorno de vecindad de  $3 \times 3$  que es el que mayormente se elige por el tiempo de procesamiento y mayor eficiencia. La mascara seria:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Ejemplo:

| Imagen original   |     |     |     |     |    | Máscara |     |     |
|-------------------|-----|-----|-----|-----|----|---------|-----|-----|
| 90                | 80  | 80  | 100 | 60  | 80 | 1/9     | 1/9 | 1/9 |
| 90                | 100 | 90  | 3   | 80  | 70 | 1/9     | 1/9 | 1/9 |
| 60                | 80  | 5   | 6   | 100 | 90 | 1/9     | 1/9 | 1/9 |
| 80                | 8   | 6   | 2   | 100 | 80 |         |     |     |
| 100               | 4   | 9   | 7   | 100 | 80 |         |     |     |
| 90                | 80  | 120 | 100 | 100 | 90 |         |     |     |
| Imagen resultante |     |     |     |     |    |         |     |     |
| 40                | 58  | 50  | 46  | 44  | 32 |         |     |     |
| 55                | 75  | 60  | 58  | 65  | 53 |         |     |     |
| 46                | 58  | 33  | 43  | 59  | 58 |         |     |     |
| 37                | 39  | 14  | 37  | 63  | 61 |         |     |     |
| 40                | 55  | 37  | 60  | 70  | 58 |         |     |     |
| 30                | 45  | 35  | 48  | 49  | 38 |         |     |     |

Imagen 9 Aplicación de filtro del promedio

Aquí se presenta una imagen con ruido gaussiano y la misma con ruido aleatorio (sal y pimienta) que se aplica este filtro promedio de la vecindad. A cada imagen Ver imagen 10



Imagen 10

**Filtro de la mediana.**

Este es un filtro que pertenece a los no lineales estos reducen un poco las deficiencias de los lineales.

Este es el más utilizado en sectores que requieren un buen tiempo de respuesta y calidad de la imagen.

El filtro de la mediana consiste en generar una imagen de salida,  $g(x, y)$  donde a cada píxel de coordenadas  $(x, y)$  se le asigna un valor de intensidad que es la mediana de los valores de un entorno de vecindad del píxel  $(x, y)$  en la imagen original,  $f(x, y)$  [3].

Que por lo regular se eligen los 8 vecinos adyacentes y el mismo, esto lo hace más eficiente y más rápido de calcular. En otras palabras se ordenan los valores de los píxeles según su valor tomando el valor medio de estos.

Por ejemplo en la imagen 11: Así se elige la mediana:

$$\begin{bmatrix} 19 & 45 & 16 \\ 32 & 12 & 34 \\ 15 & 17 & 26 \end{bmatrix} \quad 12 < 15 < 16 < 17 < \mathbf{19} < 26 < 32 < 34 < 45$$

**Imagen 11** El 19 es el valor que tendrá el píxel de la posición 5 (es decir donde está el número 12 se cambiará por el valor de la mediana que es el 19)

Este tipo de filtro elimina los píxeles cuya intensidad difiere mucho de sus vecinos e influye negativamente en la resolución, textura y contornos de la imagen.

El filtro de la mediana tiene importantes ventajas con respecto al método del promedio del entorno de la vecindad. Es más eficiente debido a que no desdibuja los contornos de los objetos, su desventaja principal es que el tiempo de cálculo es más elevado que el del promedio del entorno de vecindad y la implementación de algoritmos es más compleja y requiere un alto número de instrucciones que esto se traduce en un mayor tiempo de cálculo y esto exige un menor tiempo de proceso real. Ejemplo de su aplicación en [3] la imagen 12.

|     |     |     |     |     |    |
|-----|-----|-----|-----|-----|----|
| 90  | 80  | 80  | 100 | 60  | 80 |
| 90  | 100 | 90  | 3   | 80  | 70 |
| 60  | 80  | 5   | 6   | 100 | 90 |
| 80  | 8   | 6   | 2   | 100 | 80 |
| 100 | 4   | 9   | 7   | 100 | 80 |
| 90  | 80  | 120 | 100 | 100 | 90 |

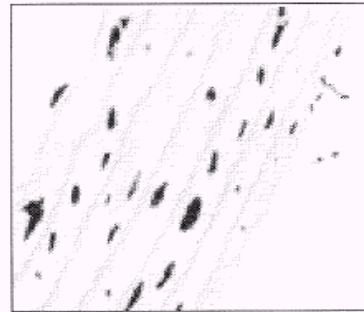


imagen original

|    |    |    |     |     |    |
|----|----|----|-----|-----|----|
| 80 | 80 | 80 | 80  | 80  | 70 |
| 80 | 80 | 80 | 80  | 80  | 80 |
| 80 | 80 | 8  | 80  | 80  | 80 |
| 60 | 9  | 6  | 9   | 80  | 90 |
| 80 | 80 | 8  | 100 | 80  | 80 |
| 80 | 80 | 80 | 100 | 100 | 80 |

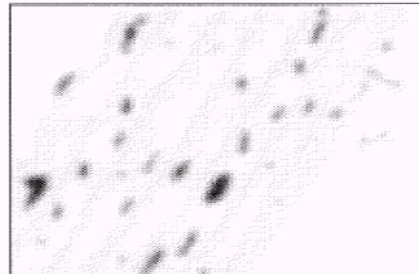


imagen mediana

**Figura 12** Aplicación del método mediana

Existe también otra forma de eliminar el ruido en la imágenes digitales haciendo la combinación de los dos filtros este métodos se llama alfa trim o mediana recortada esto es porque se ordenan los datos y se eliminan datos de los extremos y se realiza un promedio con los datos restantes es mas eficiente ya que eliminar datos que no sea muy parecidos a la mayoría.[3] Ver ejemplo en imagen 13

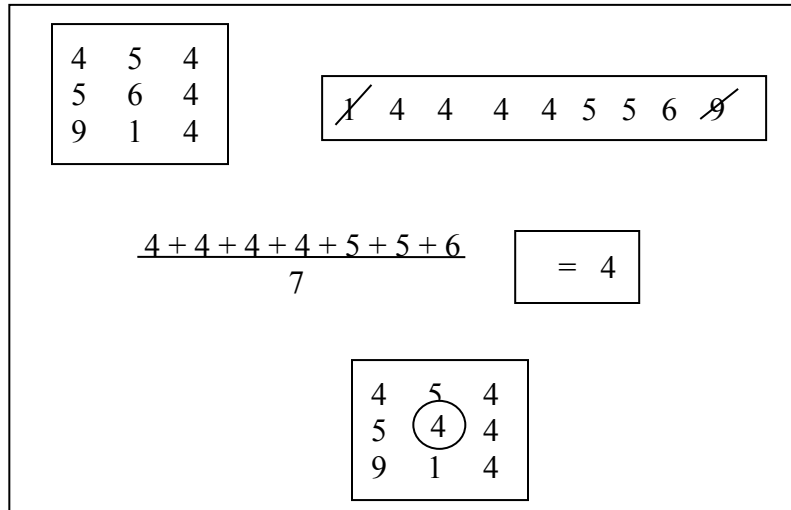


Imagen13 Procedimiento del método alfa trim

## 1.4.2 FILTRADO ESPACIAL: REALCE DE DETALLES

Al aplicar filtros de suavizado para la eliminación del ruido se presentan los siguientes factores:

- Disminución de la nitidez
- Aumento de borrosidad.
- Perdida de detalles.

Por lo tanto se tienen que utilizar técnicas para realzar detalles para su posterior estudio. El objetivo principal del realce es el de destacar los detalles finos de una imagen o intensificar detalles que han sido difuminados, bien sea por error o bien por efecto natural del método de adquisición de la imagen. El realce de una imagen aumentará su contraste.

Las utilidades del realce de las imágenes son variadas e incluyen aplicaciones que van desde la impresión electrónica y las imágenes médicas hasta las inspecciones industriales e incluso la detección autónoma de objetivos en las armas inteligentes.

Para la implementación del realce se utilizan, lo mismo que para el suavizado, técnicas basadas en la aplicación de filtros del dominio de espacio.[1]

Por lo general, la máscara de convolución usada comúnmente en el realce, es máscara de filtro paso alto, tiene un coeficiente positivo en su centro y coeficientes sobre todo negativos alrededor del borde externo.

Los detalles mayormente perdidos son algunos bordes al aplicar métodos de reducción de ruido. Y son los que se intensifican con los filtros de realce

### 1.4.2.1 BORDES

Se considera borde cuando en un conjunto de puntos de la imagen aparece un cambio significativo en la intensidad luminosa Existen básicamente dos formas de obtener los bordes de una imagen, estas son:

**Método basado en la primera derivada.** En este método primero se miden los cambios de intensidad en todos los puntos de la imagen, haciendo uso de la primera derivada, después se seleccionan como puntos de bordes aquellos puntos en los que el cambio de intensidad rebasa algún umbral preestablecido. El método mas elemental es el gradiente con valor absoluto es el de combinar dos derivadas, una que extrae bordes verticales y otra que extrae los horizontales [1] Ver imagen 14. La expresión del gradiente para hallar los bordes bit a bit en las dos direcciones es esta:

$$z_1 \leftarrow |z_2 - z_1| \text{OR} |z_3 - z_1|$$

**Método basado en la segunda derivada** Consiste en determinar en que puntos de la imagen aparecen los picos de iluminación. Los puntos en los que la intensidad tiene un máximo se anula la segunda derivada de la intensidad tomada en la dirección en que se tiene lugar la transición de claro a oscuro [1]

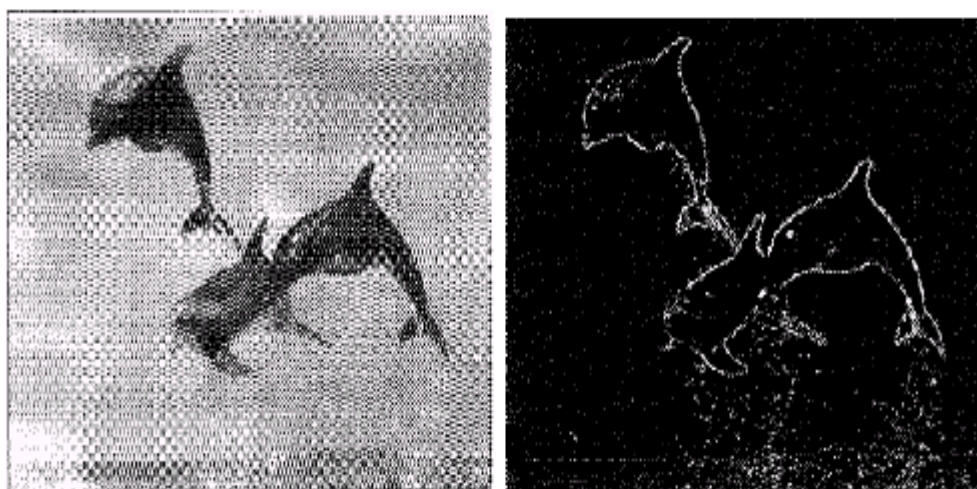


Imagen Original

Bordes Verticales y Horizontales

Imagen 14 Adquisición de bordes

### 2.4.2.2 FILTROS PASO ALTO

Un filtro paso alto eliminará los componentes bajos de frecuencia (como el medio de la imagen) y mostrará sólo los detalles altos.

También permite pasar las componentes de altas frecuencias y diluir las de baja frecuencia, es necesario que el filtro posea coeficientes negativos en la periferia y positivos en el centro. Así, cuando la máscara se encuentra sobre una zona uniforme, la salida proporcionada por la máscara será 0 o próxima a dicho valor.

Este tipo de filtro elimina también el término de frecuencia 0 con lo que la imagen resultante deberá tener valores de intensidad negativos. Como sólo estamos considerando niveles positivos de gris, los resultados del filtrado paso alto necesariamente implican alguna forma de desplazamiento o cambio de escala para que al final los niveles de gris queden dentro del rango.[1], [3]

Las mascararas más utilizada de paso alto son los siguientes:

|             |    |    |
|-------------|----|----|
| 1           | -2 | 1  |
| -2          | 4  | -2 |
| 1           | -2 | 1  |
| Paso Alto 1 |    |    |
| -1          | -1 | -1 |
| -1          | 8  | -1 |
| -1          | -1 | -1 |
| Paso Alto 2 |    |    |
| 0           | -1 | 0  |
| -1          | 4  | -1 |
| 0           | -1 | 0  |
| Paso Alto 3 |    |    |

Por ejemplo con el filtro paso alto 2 este filtro realiza una mejora en los bordes de imágenes en fondos oscuros ver imagen 15

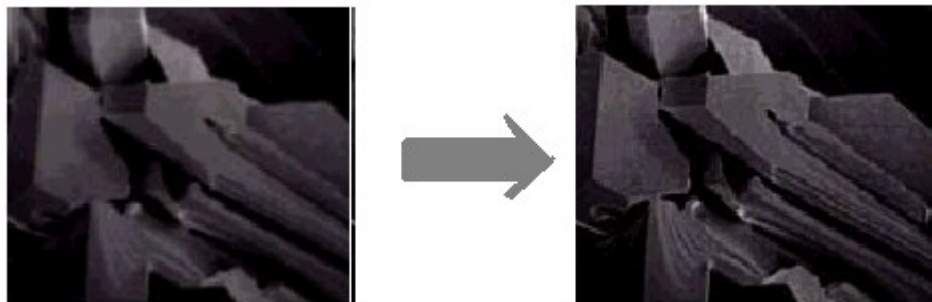


imagen 15. Aplicación del filtro pasó alto 2

En estas máscaras paso alto, hay que destacar que la suma de los coeficientes es cero. Así cuando la máscara está sobre una zona de la imagen de poco contraste o pequeña variación del nivel de gris, la salida de la máscara es cero o muy pequeña. Esto se traduce en que zonas uniformes con distintos niveles de gris son pasadas a un mismo nivel (el cero), con lo que perdemos información de la imagen. Además, eliminar los términos de

baja frecuencia produce una disminución de la media de los niveles de gris, reduciendo significativamente el contraste global de la imagen.

Al aplicar los filtros paso alto se disminuye considerablemente el número de grises presentes en la imagen original, con lo que se reduce el contraste global de la imagen. Para realizar un mejor realce de la imagen se emplean los filtros Sharpen que es una variación de las máscaras de realce de paso alto. Las máscaras Sharpen o también llamados filtros de enfoque más comunes son las siguientes [1],[3]:

|           |    |    |           |    |    |           |    |    |
|-----------|----|----|-----------|----|----|-----------|----|----|
| 1         | -2 | 1  | -1        | -1 | -1 | 0         | -1 | 0  |
| -2        | 5  | -2 | -1        | 9  | -1 | -1        | 5  | -1 |
| 1         | -2 | 1  | -1        | -1 | -1 | 0         | -1 | 0  |
| Sharpen 1 |    |    | Sharpen 2 |    |    | Sharpen 3 |    |    |

## CAPITULO II

# PARALELISMO, MICROCONTROLADORES PIC Y COMUNICACIÓN SERIE

En el presente capítulo se describe los que son las arquitecturas paralelas, comunicación serie y microcontroladores haciendo hincapié en tipos de paralelismo, microcontroladores PIC 16F877 y características de circuito integrado max232 y puerto serie debido a que son elementos que forman parte del presente trabajo de tesis.

### 2.1 PARALELISMO

Cualquier computadora, sea en serie o paralelo, opera ejecutando instrucciones sobre datos. Un algoritmo indica a la computadora lo que debe realizar en cada paso. El flujo de datos es afectado por dichas instrucciones.

Algunas clasificaciones de los sistemas en paralelo son los siguientes: el propuesto por Michael J. Flynn (1966) que es el más utilizado se basa en el número de instrucciones simultáneas y flujo de datos computados por el procesador, el esquema de *Feng* (1972) se basa en la confrontación procesamiento serie vs. Procesamiento paralelo y la clasificación de *Händler* (1977) se determina por el grado de paralelismo y encauzamiento (*pipelining*) en diferentes niveles de los subsistemas.

El procesamiento paralelo es un término que se usa para denotar tareas simultáneas de procesamiento de datos con el fin de aumentar la velocidad computacional de un sistema de computadoras. En un sistema de procesamiento paralelo puede ejecutar procesamiento concurrente de datos para conseguir un menor tiempo de ejecución en lugar de procesar cada instrucción en forma secuencial como es una computadora convencional.

En un sistema paralelo las hay tres cuestiones fundamentales que hay que considerar:

1. Describir la naturaleza, tamaño y número de los elementos procesadores.
2. Describir la naturaleza, tamaño y número de los módulos de memoria.
3. Describir la estrategia de interconexión entre procesadores y memoria.

Los procesadores pueden variar desde unas cuantas unidades aritmético lógicas hasta unidades centrales de procesamiento completas, también se encuentran los sistemas con muchos procesadores, cada uno de los cuales cuenta con su propia memoria.

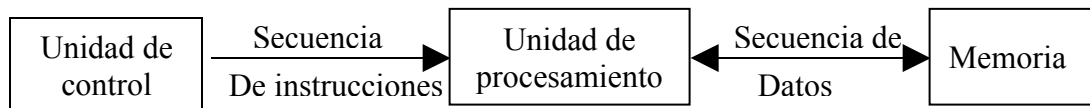
La clasificación de Flynn que se basa en los conceptos: **flujo de instrucciones** que corresponde a los contadores de programa. Un sistema con  $n$  unidades centrales de procesamiento posee  $n$  contadores, y por consiguiente,  $n$  flujos o corrientes de instrucción y el **flujo de datos** consiste de un conjunto de operandos. Ver imagen 16.

Existen 4 tipos de computadoras las cuales pueden procesar secuencias de instrucciones y secuencias de datos que pueden ser sencillas o múltiples. Los modelos son los siguientes:

| Flujo de instrucciones | Flujo de datos | Nombre | Ejemplos  |
|------------------------|----------------|--------|---|
| 1                      | 1              | SISD   | Máquinas clásicas de Von Neumann                        |
| 1                      | Múltiple       | SIMD   | Supercomputadoras vectoriales, procesadores de arreglos |
| Múltiple               | 1              | MISD   | Presumiblemente ninguna                                 |
| Múltiple               | Múltiple       | MIMD   | Multiprocesadores                                       |

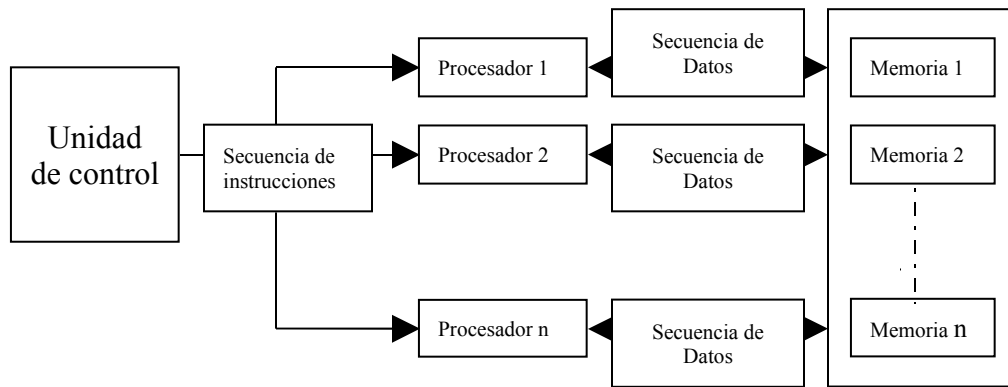
**Imagen 16** Clasificación de Flynn de procesamiento de datos e instrucciones

- **SISD (una instrucción, un dato).** En este modelo solo tiene un procesador y solo se va procesando instrucción por instrucción. En cada paso el procesador ejecuta una instrucción sobre un dato obtenido de la memoria. Las computadoras comunes usan este modelo inventado por John von Neumann. Todo algoritmo que se ejecuta en una computadora SISD se denomina secuencial (o serial), y no contiene ningún tipo de paralelismo. Ver imagen 17



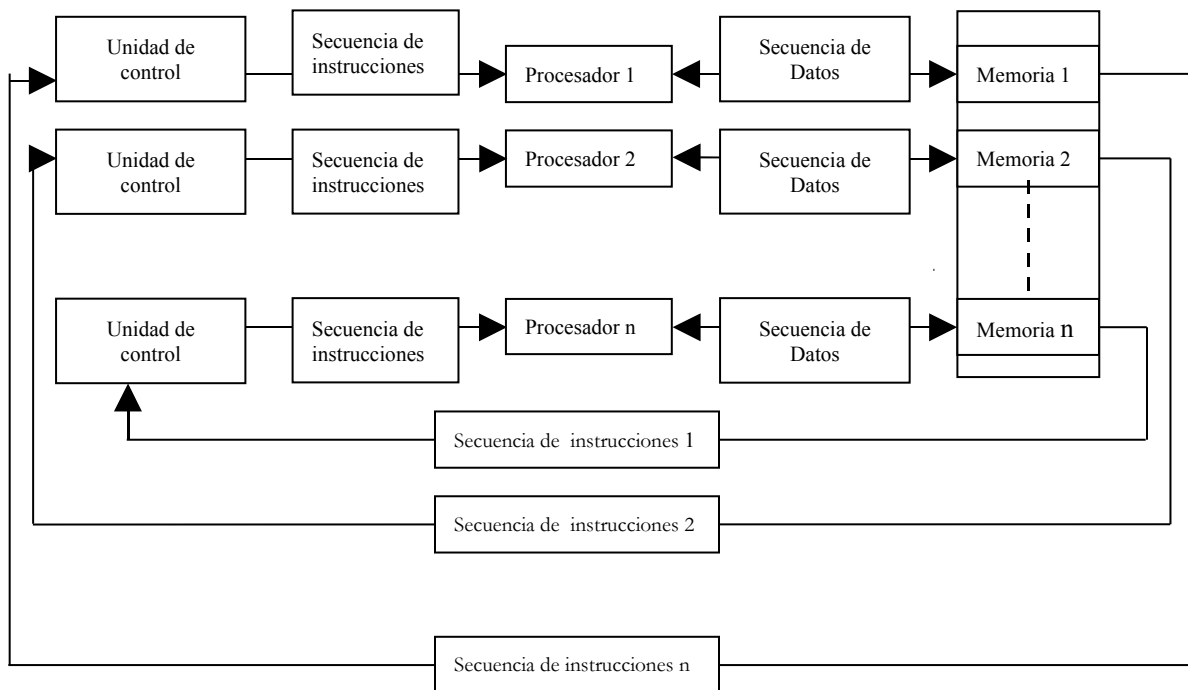
**Imagen 17** Modelo SISD (una sola instrucción un solo dato)

- **SIMD (sola instrucción, múltiples datos).** Este modelo consiste en N procesadores idénticos, cada uno con su unidad de memoria local, donde puede almacenar datos. Todos los procesadores trabajan bajo el control de una única instrucción ordenada por la unidad central de control. Hay N datos, cada uno por procesador. Los procesadores operan sincrónicamente: en cada paso, todos los procesadores ejecutan la misma instrucción sobre datos distintos. Ver imagen 18



**Imagen 18** Modelo Una sola instrucción muchos datos

- MIMD (múltiple instrucción, múltiples datos)** Esta clase de computadoras en paralelo es la más general y la más poderosa en la clasificación de Flynn. Aquí existen N procesadores con N instrucciones y N datos. Cada procesador tiene su unidad de control y su memoria local. Cada procesador cada instrucción ordenada por su unidad de control, por lo tanto los procesadores están potencialmente ejecutando diferentes programas con diferentes datos mientras están resolviendo diferentes subproblemas de un único problema. Esto significa que los procesadores usualmente operan asincrónicamente. Ver imagen 19



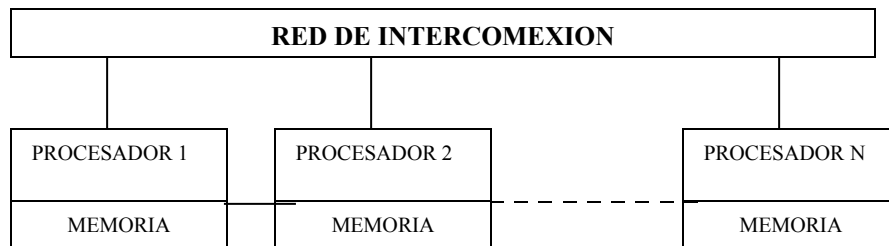
**Imagen 19** Modelo general de MIMD Múltiple instrucción, múltiples datos

Por lo tanto concluimos que sus principales características de esta arquitectura son:

- Todos los elementos de procesamiento (EP) poseen su propia unidad de control.
- Todas las unidades de proceso ejecutan su propio programa.

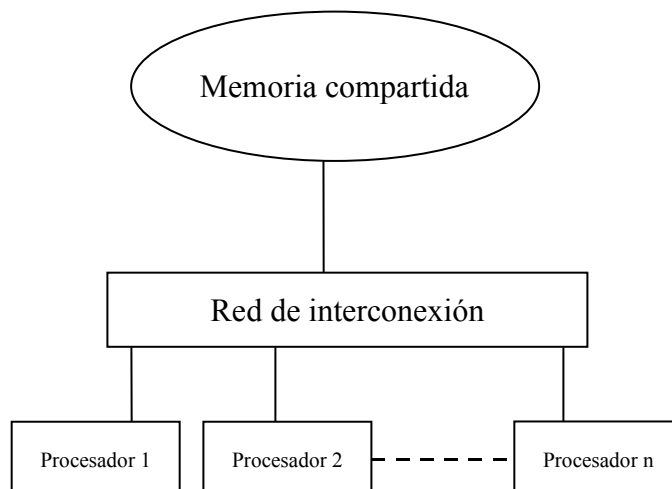
Los modelos paralelos MIMD se clasifican por el tipo de memoria

1. **Arquitectura de paso de mensajes (Message Passing Arq.)** Llamada también memoria compartida distribuida, cada procesador tiene su propia memoria local o privada, cada procesador se puede comunicar con los demás con solo paso de mensajes Ver imagen 20



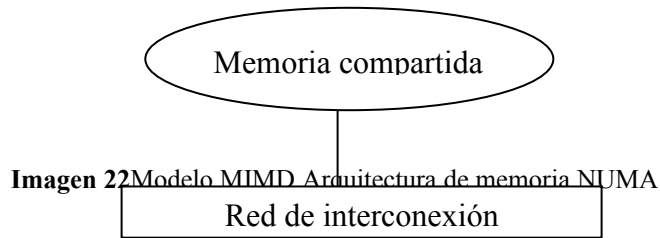
**Imagen 20** Modelos MIMD arquitectura de paso de mensajes

2. **Arquitectura de memoria compartida (Shared Address Space Arq.)** Todos los procesadores comparten toda la memoria, este es un ejemplo típico de una de las computadoras de esta arquitectura Ver figura 16.1



**Imagen 21** Modelo MIMD. Memoria compartida.

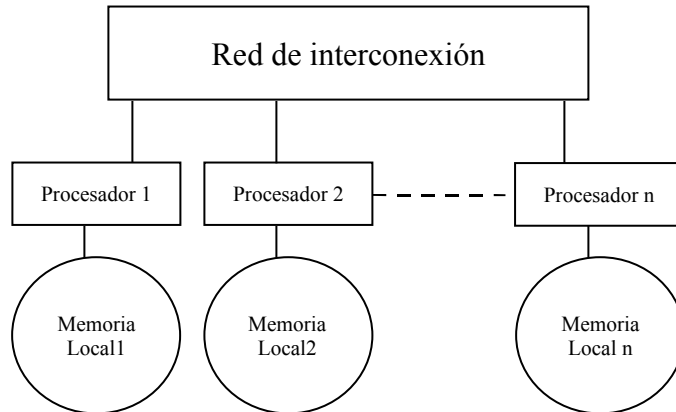
- c) **Arquitectura NUMA. (Non Uniform Memory Architecture)**  
 Además de la memoria compartida, cada procesador tiene una memoria local para guardar el programa y los datos no compartidos. En este caso disminuye el acceso a la red por parte de cada procesador haciendo más eficiente el acceso a memoria. Ver imagen 22



**Imagen 22** Modelo MIMD Arquitectura de memoria NUMA

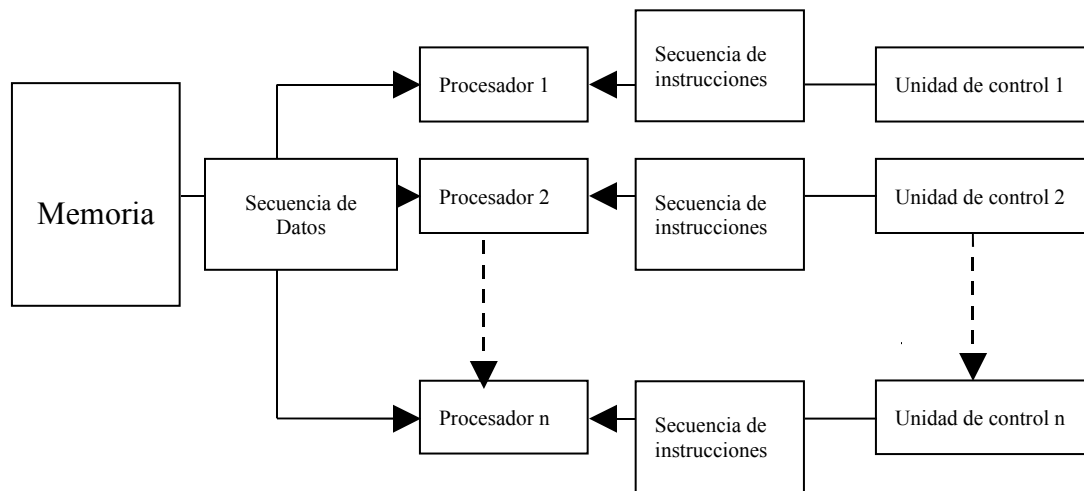
- d) **Arquitectura COMA. (Cache Only Memory Architecture)** Solo quedan las memorias locales y cada procesador puede acceder a la memoria de cualquier otro (memoria virtual). Ver Imagen 23

- A estos se les llama memoria local.
- Pueden empujar y tirar datos a la memoria de paso de otros procesadores.
- Pero son más costosos.



**Imagen 23** Modelo MIMD con arquitectura COMA

- **MISD (múltiple instrucción, un solo dato).** Se tienen  $N$  procesadores, cada uno con su unidad de control, comparten una única unidad de memoria. En cada paso, un dato es recibido desde la memoria y procesado simultáneamente por todos los procesadores, cada uno de ellos ejecutando la instrucción recibida de su unidad de control. El paralelismo se logra permitiendo a cada procesador hacer diferentes cosas sobre el mismo dato.. Ver Imagen 24



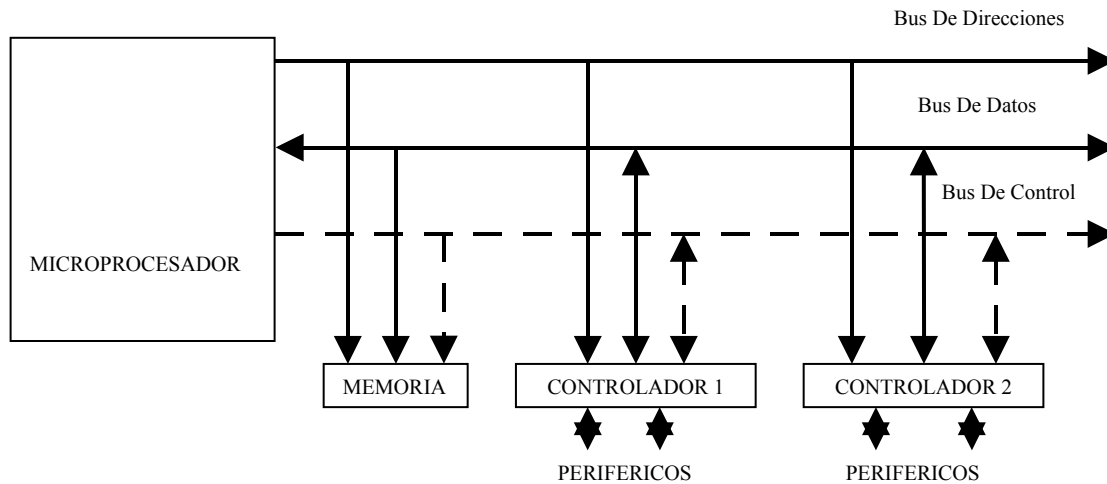
**Imagen 24** Modelo MISD múltiple instrucción, un solo dato

## 2.2 MICROCONTROLADORES

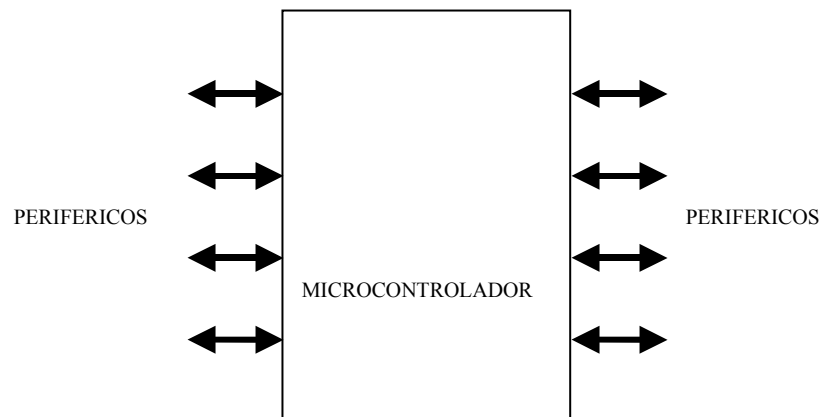
Es un circuito integrado programable que contiene todos los componentes de una computadora. Se emplea para controlar el funcionamiento de una tarea determinada. Este circuito integrado es dedicado. En su memoria solo reside un único programa que manipula una aplicación determinada.

Estos circuitos integrados se emplean en numerosos aparatos electrónicos incluso en los periféricos de una computadora como el ratón, impresoras etc. están regidas por el programa que se ejecuta en un microcontrolador.

La diferencia entre un microcontrolador y un microprocesador es que este último es un sistema abierto con el que se puede construir una computadora con las características que se desee, acoplándole los módulos necesarios Ver Imagen 25 .Y un microcontrolador es un sistema cerrado que contiene una mini computadora completa de prestaciones un poco limitadas que no se pueden modificar un circuito integrado que contiene una unidad central de proceso que interpreta las instrucciones y la única comunicación con lo exterior es por medio de sus terminales [7] Ver imagen 26



**Imagen 25** Estructura de un sistema abierto



**Imagen 26** Estructura de un sistema cerrado

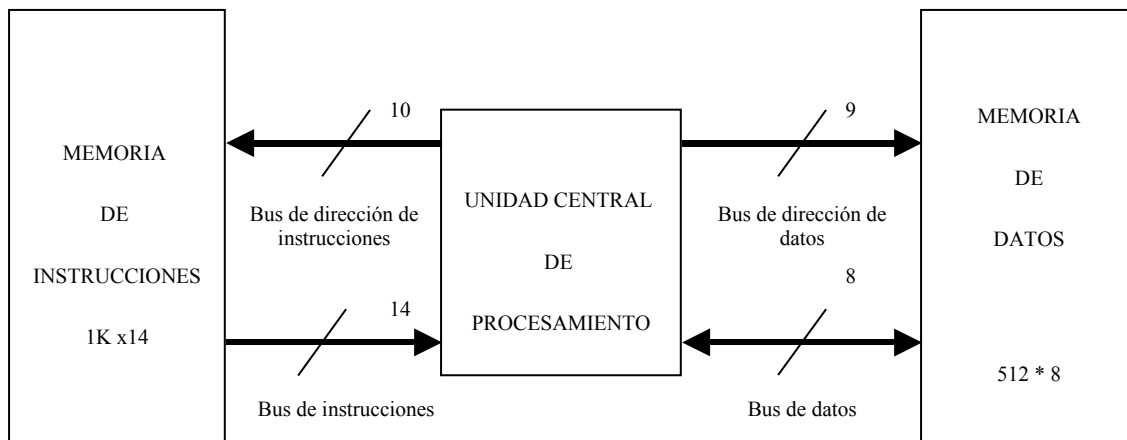
## 2.3 ARQUITECTURA INTERNA

Un microcontrolador posee todos los componentes de una computadora, pero con las características fijas que no se pueden alterar. Las partes principales son:

1. Procesador
2. Memoria no volátil para contener el programa
3. Memoria de lectura y escritura para guardar los datos
4. Líneas de entrada salida
  - Comunicación paralela

- Comunicación serie
  - Diversos puertos para la comunicación con periféricos.(PC, USB, etc.)
5. Recursos auxiliares.
- Circuito de reloj.
  - Temporizadores.
  - Perro guardián (wachdog).
  - Conversores analógico digital y digital a analógico..
  - Comparadores analógicos.
  - Protección ante fallos de alimentación.
  - Estado de reposo o bajo consumo

**PROCESADOR.** Se compone de arquitectura Harvard donde son independientes la memoria de instrucciones y la memoria de datos y tiene su sistema de buses para su acceso y la capacidad de cada memoria es diferente ver imagen 27. También pertenece también a la arquitectura RISC (computadoras de juego de instrucciones reducido) tiene un pequeño número de instrucciones máquina simples lo cual hace que la mayoría de las instrucciones se efectúen en un ciclo de instrucción. Y cuenta con el paralelismo implícito, que consiste en la segmentación de procesador (pipe line), descomponiéndolo en etapas para poder procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez [7].



**Imagen 27** Arquitectura Harvard

**MEMORIA DE PROGRAMA** En esta memoria se guardan todas las instrucciones del programa de control. No hay posibilidad de utilizar memoria externas de ampliación. Como en programa siempre es el mismo, debe de estar grabado de manera permanente.

**Tipos de memoria:**

1. ROM con mascara Este tipo de memoria se graba en el proceso de fabricación
2. EPROM La grabación de esta memoria se realiza por medio de un dispositivo externo de la computadora llamado grabador. Se puede borrar mediante rayos ultravioleta esto se efectúa enviando los rayos a la superficie de la cápsula en la cual se tiene un cristal
3. OTP Solo se graba una vez por parte del usuario utilizando el mismo procedimiento que la EPROM, posteriormente no se podrá borrar
4. EEPROM. Este se puede grabar y borrar eléctricamente tantas veces se requiera.
5. FLASH Es una memoria no volátil. Solo que el borrado se realiza por bloques.

**MEMORIA DE DATOS.** Los datos que manejan los programas varían continuamente entonces esta memoria es de lectura y escritura, por lo tanto tiene una memoria volátil RAM. También hay microcontroladores como el PIC 16F877 que cuenta con una memoria no volátil del tipo EEPROM que tiene 256 bytes para contener datos.[7]

**RECURSOS AUXILIARES.** Los más comunes:

- Circuito de reloj: Es el encargado de generar los impulsos que sincronizan el funcionamiento de todo el sistema
- Temporizadores: Orientados a controlar tiempos.
- Perro guardián (“Wachdog”), destinado a provocar una reinicialización cuando el programa queda bloqueado.
- Convertidores analógico digital y digital analógico: Para poder recibir y enviar señales analógicas.
- Comparadores analógicos .Para poder verificar el valor de una señal analógica
- Sistema de protección ante fallos de alimentación.
- Estado de reposo: En el que el sistema queda congelado y el consumo de energía se reduce al mínimo.

## PROGRAMACIÓN DE MICROCONTROLADORES

Los lenguajes más utilizados para la programación de los microcontroladores PIC es el lenguaje ensamblador con el cual los programas se ejecutan más rápido, también ya son frecuentes programas de alto nivel lenguaje C [5] y basic

### 2.4 PIC16F877

El circuito microcontrolador PIC16F877 de la compañía Microchip ver imagen 28. Este dispositivo cuenta con diversos elementos que le hacen adecuado para emplearse en labores didácticas. Cuenta con 4 pines los cuales tienen una función principal y una alternativa lo cual lo hace más versátil en su uso vea imagen 29 [6]

## Pin Diagram

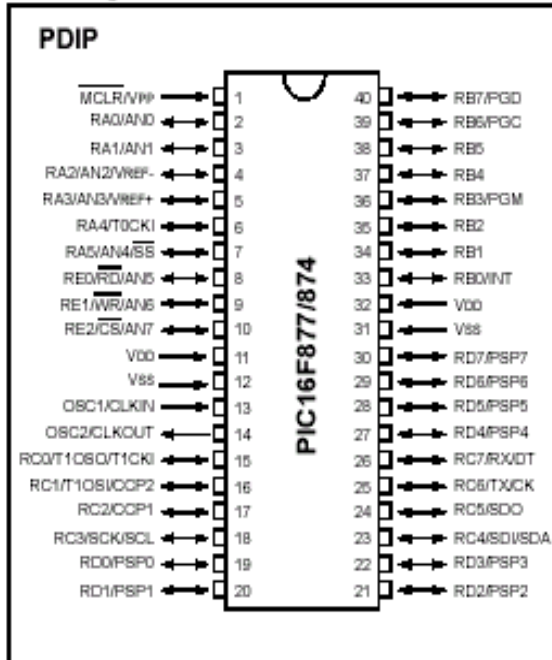


Imagen 28 Microcontrolador PIC16f877

- cpu risc
- arquitectura harvard
- 35 instrucciones de ensamblador
- ejecucion en 1 ciclo máquina
- máxima vel 20mhz
- mem programa 8k-palabras (flash)
- 368 kb mem ram de datos
- 256 kb mem eeprom de datos
- 1 watchdog
- 3 timers (8,16,8bits)
- 2 canales pwm
- 8 conver. a/d de 10bits
- puertos serie:
  - spi
  - iic
  - usart
- puerto paralelo esclavo de 8 bits
- 33 pines e/s en 5 puertos
- pila hardware
- debugger in circuit

| <i>Elemento</i>         | <i>Función Periférica Principal</i>   | <i>Función Alternativa</i>   |
|-------------------------|---|--|
| Puerto A<br>(RA0 - RA5) | - Entradas para conversión de señales analógicas a digitales. Incluye ajustes para referencias de voltajes.<br>- Comparte una línea con el sistema SPI.<br>- Entrada del temporizador básico.         | Comunicación en paralelo (todas las líneas pueden ser bi direccionales). |
| Puerto B<br>(RB0 - RB7) | - Entradas que generan interrupciones individuales en el sensado de cambios de nivel.<br>- Elementos para la programación del dispositivo y depuración interna  | Comunicación en paralelo (todas las líneas pueden ser bi direccionales). |
| Puerto C<br>(RC0 - RC7) | - Funciones de temporizado: Capturas de entradas, Comparaciones de salidas.<br>- Comunicación serial síncrona SPI.<br>- Comunicación serial síncrona IIC.<br>- Comunicación serial asíncrona (USART). | Comunicación en paralelo (todas las líneas pueden ser bi direccionales). |
| Puerto D<br>(RD0 - RD7) | - Puerto Paralelo Esclavo de 8 bits.  | Comunicación en paralelo (todas las líneas pueden ser bi direccionales). |
| Puerto E<br>(RE0 - RE2) | - Controles de acceso al Puerto Paralelo Esclavo.   | Comunicación en paralelo (todas las líneas pueden ser bi direccionales). |

Imagen 29 Funciones principales y alternativas del microcontrolador PIC 16F877

A continuación se enlistan las características de cada pin del microcontrolador.

### **Puerto A**

- puerto de e/s de 6 pines
- ra0 è ra0 y an0
- ra1 è ra1 y an1
- ra2 è ra2, an2 y vref-
- ra3 è ra3, an3 y vref+
- ra4 è ra4 (salida en colector abierto) y t0cki(entrada de reloj del modulo timer0)
- ra5 è ra5, an4 y ss (selección esclavo para el puerto serie síncrono)

### **Aplicaciones del Puerto B**

- Puerto e/s 8 pines
- Resistencias pull-up programables
- RB0 è Interrupción externa
- RB4-7 Interrupción por cambio de flanco
- RB5-RB7 y RB3 è programación y debugger in circuit.

### **Aplicaciones del Puerto C**

- Puerto e/s de 8 pines
- RC0 è RC0, T1OSO (Timer1 salida oscilador) y T1CKI (Entrada de reloj del modulo Timer1).
- RC1-RC2 è PWM/COMP/CAPT
- RC1 è T1OSI (entrada osc timer1)
- RC3-4 è IIC
- RC3-5 è SPI
- RC6-7 è USART

### **Aplicaciones del Puerto D**

- Puerto e/s de 8 pines
- Bus de datos en PPS (Puerto paralelo esclavo)

### **Aplicaciones del Puerto E**

- Puerto de e/s de 3 pines
- RE0 è RE0 y AN5 y Read de PPS
- RE1 è RE1 y AN6 y Write de PPS

- RE2 è RE2 y AN7 y CS de PPS

### **Registros de dirección.**

- Configuran si un pin es de entrada o de salida  
Los registros que llevan acabo esta tarea son:  
–TRISA –TRISC –TRISE  
–TRISB –TRISD
- 0 significa Output (salida)
- 1 significa Input (entrada)

Un uso frecuente del sistema microcontrolador es en la adquisición de datos. Una dificultad específica del microcontrolador PIC16F877 es su escasa cantidad de localidades de memoria RAM disponible, alrededor de 300 localidades.

## **2.5 PROGRAMACIÓN DEL MICROCONTROLADOR PIC16F877**

El esquema de programación del sistema se fundamenta en las herramientas que la compañía que fabrica al dispositivo, Microchip, pone a disposición de sus usuarios. La herramienta principal es el medio ambiente de desarrollo MPLAB. En este paquete se hacen disponibles al diseñador un editor de texto, un programa ensamblador y un sistema simulador. También se cuenta con una emulación de una terminal serial y elementos para iniciar el proceso de grabado de los programas diseñados en los dispositivos físicos. Este paquete de cómputo ofrece una organización de las aplicaciones en proyectos. Lo anterior puede facilitar el empleo de compiladores de lenguajes de alto nivel con una mezcla de rutinas en lenguaje ensamblador.

Para solucionar el problema anterior el microcontrolador PIC16F877 cuenta con mecanismos alternos de programación. Uno de ellos le permite grabar información en su zona de código a partir de un pequeño programa “pre-cargado”, que es lo único con que debe contar antes de ensayar aplicaciones. [5]

## **2.6 PUERTO SERIE**

A las comunicaciones serie. Las comunicaciones serie se utilizan para enviar datos a través de largas distancias, ya que las comunicaciones en paralelo exigen demasiado cableado para ser operativas.

## **2.7 DESCRIPCIÓN DEL PUERTO**

El puerto serie de la PC es compatible con el estándar RS-232C. Este estándar fue diseñado en los 60s para comunicar un equipo terminal de datos o DTE (Data Terminal Equipment, la PC en este caso) y un equipo de comunicación de datos o DCE (Data Communication Equipment, habitualmente un modem). El estándar especifica 25 pines de

señal, y que el conector de DTE debe ser macho y el conector de DCE hembra., pero muchos de los 25 pines no son necesarios. Por esta razón son más usuales los DB-9 macho.

Los voltajes para un nivel lógico alto están entre -3V y -15V. Un nivel lógico bajo tendrá un voltaje entre +3V y +15V. Los voltajes más usados son +12V y -12V.

## **2.8 LA ESTRUCTURA DE DATOS DEL PUERTO SERIE RS-232**

La comunicación de datos en un puerto serial, se usa normalmente para efectuar comunicaciones sincrónicas, es decir, sin tiempo preestablecido para iniciarse, es decir, los datos llegan en ráfagas ó paquetes de información, normalmente cada paquete es de 8 bits=1 byte (equivalente a un carácter en código ASCII), algunos equipos envían carácter por carácter, otros guardan muchos caracteres en la memoria y cuando les toca enviarlos, los envían uno tras otro.[9]

Uno de los parámetros más importantes en la comunicación serie, es la velocidad con la que los datos se transmiten, para el caso del RS-232, pueden transmitir de los 300 Baudios (1 Baudio = 1 bit / seg) hasta 115,200 Baudios, la velocidad depende de los equipos conectados en el puerto serie y la calidad y longitud de los cables.

Otro de los parámetros de trascendencia es el bit de inicio, es decir, el bit que le indica al puerto receptor que va a llegar un Byte de información. Hay dos tipos de paridad adicional que se usan y estos son: Marca (mark) El bit de paridad que se intercala siempre es un uno. Espacio (space) El bit de paridad que se intercala siempre es un cero y el número de bits que se emplean para cada paquete, pueden ser 5, 6, 7 u 8. Es así como la comunicación serie RS-232 es la comunicación de Datos más empleada en el mundo, ya que utiliza pocos alambres ó cables para lograrlo y mediante los módems, es la forma de intercomunicar computadoras, comunicarse a través de Internet, control a distancia y muchas otras aplicaciones.

## **2.9 INTERFAZ RS-232**

Este estándar fue desarrollado por la EIA (Electronics Industries Association) hacia 1969. Equivale al estándar V.24 de la CCITT. Esta recomendación especifica un límite de velocidad de 20.000 bps. Inicialmente utilizaba conectores DB-25 pero en la actualidad se usan mucho conectores DB-9.[9] Ver Figura (32)

Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Ver figura 30

| Pin | Función                   |
|-----|---------------------------|
| TXD | (Transmitir Datos)        |
| RXD | (Recibir Datos)           |
| DTR | (Terminal de Datos Listo) |
| DSR | (Equipo de Datos Listo)   |
| RTS | (Solicitud de Envío)      |
| CTS | (Libre para Envío)        |
| DCD | (Detección de Portadora)  |

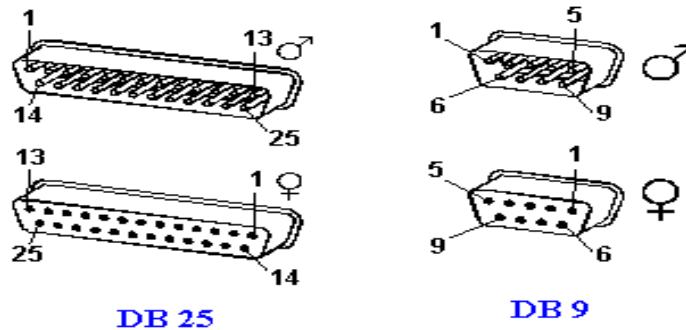
**Imagen 30** Funciones mas importantes de RS232

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS y DCD son de entrada. La masa de referencia para todas las señales es SG (Tierra de Señal). En la siguiente imagen 31 se muestra la comparación de los PINES del conector DB25 con el DB9. Ver conectores en la imagen 32.

| Numero En DB-25 | De Pin En DB-9 | Señal                        | Descripción            | E/S |
|-----------------|----------------|------------------------------|------------------------|-----|
| 1               | 1              | -                            | Masa chasis            | -   |
| 2               | 3              | TxD                          | Transmit Data          | S   |
| 3               | 2              | RxD                          | Receive Data           | E   |
| 4               | 7              | RTS                          | Request To Send        | S   |
| 5               | 8              | CTS                          | Clear To Send          | E   |
| 6               | 6              | DSR                          | Data Set Ready         | E   |
| 7               | 5              | SG                           | Signal Ground          | -   |
| 8               | 1              | CD/DCD (Data) Carrier Detect |                        | E   |
| 15              | -              | TxC(*)                       | Transmit Clock         | S   |
| 17              | -              | RxC(*)                       | Receive Clock          | E   |
| 20              | 4              | DTR                          | Data Terminal Ready    | S   |
| 22              | 9              | RI                           | Ring Indicator         | E   |
| 24              | -              | RTxC(*)                      | Transmit/Receive Clock | S   |

(\*) = Normalmente no conectados en el DB-25

**Imagen 31** comparación de DB25 con DB9



**Imagen 32** Conectores serie

## 2.10 MAXRS- 232 CIRCUITO INTEGRADO PARA CONVERSIÓN DE NIVELES.

El MAX232 es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL (5V que representan señales altas unos y 0V señales en bajo ceros ) y viceversa. Lo interesante es que sólo necesita una alimentación de 5V, ya que genera internamente algunas tensiones que son necesarias para el estándar RS232. [9] Ver figura 33

El MAX232 soluciona la conexión necesaria para lograr comunicación entre el puerto serie de una PC y cualquier otro circuito con funcionamiento en base a señales de nivel TTL es decir, 0V equivale a un cero +5V a 1,

El circuito integrado posee dos convertidores de nivel TTL a RS232 y otros dos convertidores que, a la inversa, convierten de RS232 a TTL.

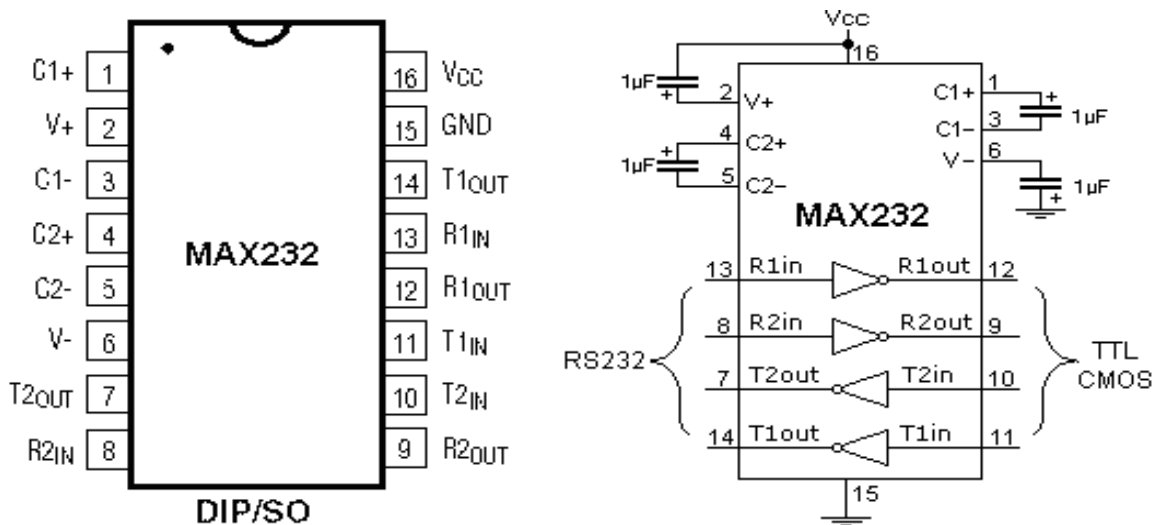


imagen 33 Max232

Estos convertidores son suficientes para manejar las cuatro señales más utilizadas del puerto serie del PC, que son TX, RX, RTS y CTS.

TX es la señal de transmisión de datos, RX es la de recepción, y RTS y CTS se utilizan para establecer el protocolo para el envío y recepción de los datos.

# CAPITULO III

## DESCRIPCIÓN DEL SISTEMA

Para la solución de un sistema es necesario un buen planteamiento del problema para poder realizar el análisis y el diseño que son pasos previos a la búsqueda de la solución o conjunto de soluciones.

El análisis se refiere a determinar cuales son los objetivos generales y cuales los objetivos específicos. Los objetivos generales establecen que se tiene que hacer para llegar a la solución. Y los objetivos específicos detallan cada paso que se tiene que hacer; se puede decir que se divide en subproblemas y llegar a la solución de cada uno.

En cuanto se tiene el análisis lo siguiente es realizar la parte del diseño esto es: el como llevar acabo cada uno de los objetivos específicos para su posterior desarrollo y llegar así al objetivo del proyecto

En este capitulo 3 se describen los objetivos generales y específicos que se requieren para la realización del proyecto. Son presentados los módulos del sistema que estará encargado de la comunicación de la computadora con el dispositivo y también los módulos que componen el programa de los microcontroladores PIC. Se detalla cada función del sistema de preprocesamiento de imágenes paralelo. También se proporciona la información en cuanto a las partes que componen infraestructura del hardware y las interfaces de entrada y salida.

### 3.1 OBJETIVOS DEL TRABAJO DE TESIS

#### **Objetivos Generales.**

- El diseño y desarrollo de un arreglo paralelo de dos microcontroladores PIC16F877 con el cual se pueda preprocesar una imagen binaria en paralelo.
- Comunicación dispositivo-computadora
- Visualizar la imagen original y la imagen preprocesada.

#### **Objetivos Específicos.**

Diseño y desarrollo de:

- Una arquitectura paralela basada en dos microcontroladores PICs.
- Un algoritmo para sincronizar a los microcontroladores
- Las interfaces hardware-software para la transmisión de la imagen original y procesada entre el arreglo de dos microcontroladores PICs y la computadora.
- Algoritmos que preprocesan imágenes digitales.

### 3.2 ANALISIS Y DISEÑO DEL TRABAJO DE TESIS

El lenguaje que se utilizara para la interfaz de la computadora con el dispositivo paralelo será en el lenguaje Borlan Delphi 6. Se utilizara la librería RSCOM que permite realizar la comunicación asíncrona RS232 para poder transmitir y recibir la imagen digital. Para el programa de los microcontroladores se utilizara la combinación del lenguaje C y ensamblador para facilitar su desarrollo.

Los módulos del Programa en Delphi 6

|   |   |
|---|---|
| MODULO CONSTRUCTOR                              | En este modulo se realiza la creación de un mapa de bits para la manipulación de la imagen y se realiza la configuración del puerto serie.  |
| MODULO DE CARGAR IMAGEN                         | Aquí se realiza la apertura de una imagen existente para su visualización en pantalla. Se extraen las medidas de la imagen y se llama al modulo de transformación a matriz.                           |
| MODULO TRANSFORMAR MAPA DE BITS A MATRIZ        | En este modulo se realizara la transformación del mapa de bits a una matriz para poder manipular su envío a los Microcontroladores. Ya que cuando se abre la imagen se encuentra en un mapa de bits.. |
| MODULO GUARDAR COMO                             | Este modulo guarda la imagen recibida en una ruta especificada por el usuario   |
| MODULO QUE TRANSFORMA LA MATRIZ EN MAPA DE BITS | Se realiza la transformación de la matriz recibida a un mapa de bits para mostrarla en pantalla   |
| MODULO ENVIAR                                   | En este modulo se realiza el envío en bloque de la imagen a cada microcontrolador.  |
| MODULO RECIVIR                                  | En este modulo se realiza la recepción en bloques de la matriz preprocesada por los microcontroladores.   |

|                 |  |
|-----------------|--|
| MODULO PROCESAR | Es el modulo principal ya que realiza las particiones de la imagen las envía y recibe por medio de los módulos enviar y recibir y ensambla la imagen recibida. |
|-----------------|--|

|                               |   |
|-------------------------------|---|
| MODULO DE ELECCION DE PROCESO | En este modulo se realiza la elección del filtro de eliminación de ruido o del filtro de realce de bordes |
|-------------------------------|---|

|              |   |
|--------------|---|
| MODULO SALIR | Libera el puerto serie y cierra la aplicación |
|--------------|---|

A continuación se muestran los módulos del programa de los microcontroladores PICs

|                          |  |
|--------------------------|--|
| MODULO DE INICIALIZACIÓN | En este modulo se configura el puerto C para la comunicación serial, puerto b como salida, y se indican los bits D0y D1 uno como entrada y otro como salida. |
|--------------------------|--|

|                      |  |
|----------------------|--|
| MODULO ESPERAR TURNO | En este modulo hace que uno de los microcontroladores espere por el uso de la transmisión o recepción mientras el otro microcontrolador está enviando o recibiendo. Cuando obtiene su turno activa la transmisión serie. |
|----------------------|--|

|                        |  |
|------------------------|--|
| MODULO LIBERA CONEXIÓN | En este modulo cuando uno de los microcontroladores termina de transmitir o recibir desactiva su conexión serie y envía una señal al otro microcontrolador para indicarle que libero la conexión |
|------------------------|--|

|                   |  |
|-------------------|--|
| MODULO TRANSMITIR | Envía bytes a la computadora por el puerto serie |
|-------------------|--|

|                |   |
|----------------|---|
| MODULO RECIBIR | Recibe de la computadora bytes por el puerto serial |
|----------------|---|

|                  |   |
|------------------|---|
| MODULO DE REALCE | En este modulo se lleva acabo el proceso de aplicación de mascararas para el realce de detalles |
|------------------|---|

|                     |  |
|---------------------|--|
| MODULO DE SUAVISADO | En este modulo se realiza el proceso de suavizado elegido ya sea por promedio por alfa trim o mediana. |
|---------------------|--|

MODULO PRINCIPAL

Es el modulo que se conjugan todos los procesos coordinándolos haciendo llamadas a ellos para realizar la recepción, transmisión, escribir, leer de la memoria EEPROM, sincronizar bloques de recepción y hacer el proceso elegido.

Con la descripción de los módulos que componen el sistema de interfaz se describe el sistema de la siguiente manera como se muestra en la imagen 34

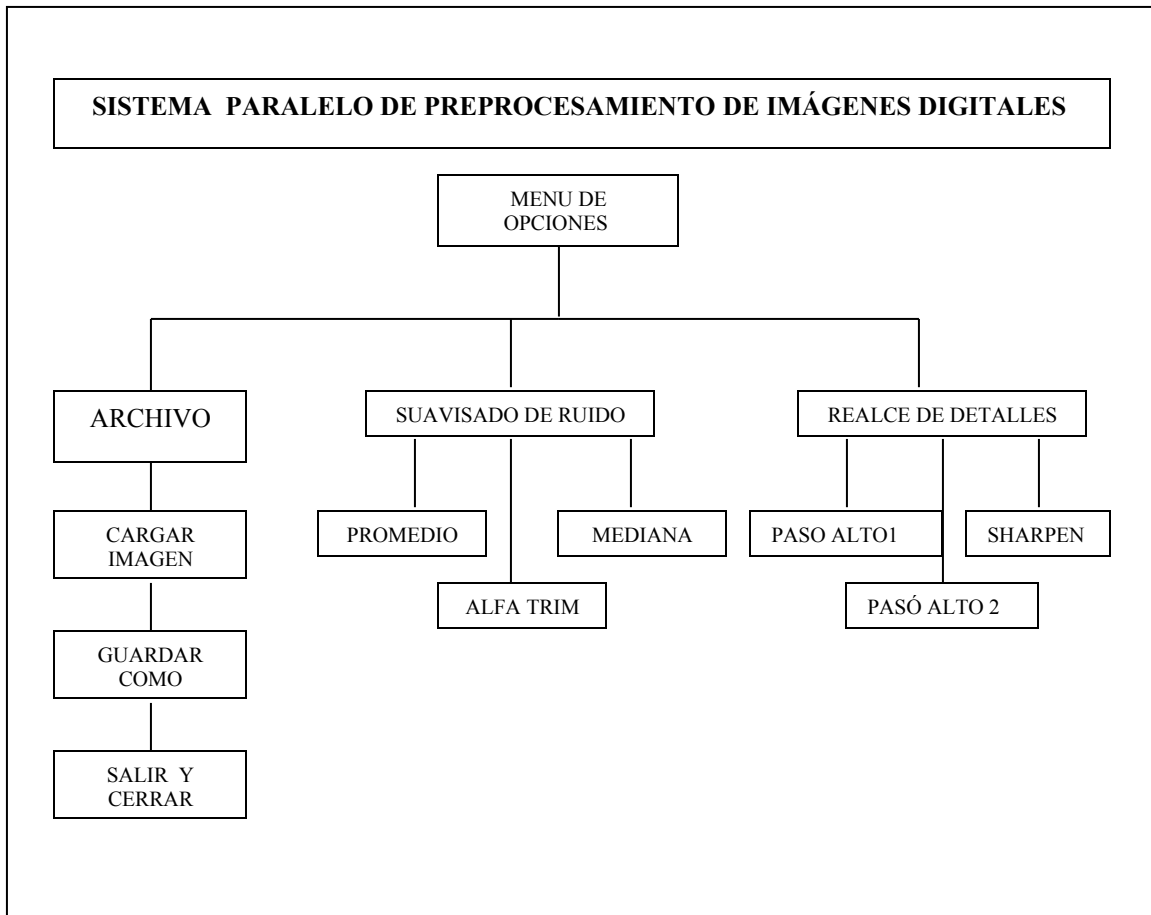


Imagen 34 De contenido del sistema preprocesamiento de imágenes digitales.

### 3.3 DESCRIPCION DEL SISTEMA DE PREPROCESAMIENTO PARALELO

**Menú de Opciones:** Se encuentran todas las opciones a elegir que manejan el sistema. Y son las siguientes:

1. **ARCHIVO:** Se despliegan las opciones siguientes:

- **Cargar Imagen**: Se carga en pantalla una imagen existente elegida por el usuario.
- **Guardar Como**: Se guarda la imagen preprocesada con un nombre elegido por el usuario en la ruta especificada por el.
- **Salir y cerrar**: Se cierra la aplicación y se libera el puerto serie.

2. **SUAVISADO DE RUIDO**: Se despliegan opciones de eliminación de ruido

- **Promedio**: Si es elegida se envía al PIC esta opción y se empieza a enviar la imagen
- **Afa Tim**: Es lo mismo que la anterior solo que envía esta opción.
- **Mediana**: También envía al los PIC la opción y se envía la imagen.

3. **REALCE DE DETALLES**: Se despliegan las opciones de suavizado para elegir

- **Paso alto1**: envía solicitud al los PIC para realce con esa mascara de convolucion y envía la imagen
- **Paso alto2**: envía Solicitud al los PIC para que realce los detalles con esta mascara y envía imagen
- **Sharpening**: También envía la solicitud a los PIC para realce con esta mascara y enviar la imagen.

### 3.4 INFRAESTRUCTURA

- **Computadora. De escritorio** con las siguientes características:  
 Microprocesador ATLON 2800  
 Memoria RAM de 1Gigabyte  
 Disco Duro de 123.5 Gigabytes  
 Monitor a color UVGA .28 no entrelazado 15''  
 Tarjeta de video 128 Megabytes  
 Puerto COM1
- **Software**  
 Sistema Operativo Windows ME  
 Borland Delphi 6  
 Lenguaje C para PICS  
 MPLAB  
 Libreria RSCOM:dll
- **Hardware**  
 2 microcontroladores PIC 16F877A  
 Fuente de poder 5V  
 Programador PICSTART PLUS  
 1 integrado MAX232  
 Cable comunicación serial

### 3.5 INTERFACES

Las interfases están dadas por la siguiente asignación:

#### INTERFAZ DE ENTRADA:

- **Ratón:** Se eligen las opciones por medio de este periférico.

#### INTERFACES DE SALIDA:

- **Pantalla:** donde se muestra la imagen preprocesada
- **Puertos B:** en estos puertos se conectaron LEDs como apoyo a pruebas con los PIC.

## CAPITULO IV

### DESARROLLO DEL SISTEMA

En este capitulo se narra el funcionamiento del proyecto de tesis se divide en dos partes:

- **HARDWARE:** Se describe la construcción de la tarjeta de preprocesamiento paralelo que se encarga de recibir la imagen aplicar el filtro elegido y transmitirla a la computadora para mostrarla.
- **SOFTWARE:** En esta parte se detallan los algoritmos utilizados en los PIC para el preprocesamiento de la imagen y el algoritmo del sistema de interfaz con el usuario

#### HARDWARE

##### **4.1 INTERFAZ DE LA COMPUTADORA CON EL DISPOSITIVO PARALELO.**

La interfaz se establece mediante el puerto serial COM1 de la computadora el cual por omisión esta configurado con una velocidad de transmisión de 9600 baudios. Se conecta un cable que en cada extremo tiene un conector DB9 uno hembra y en el otro un DB9 macho. El extremo con el conector DB9 hembra esta conectado al puerto serie de la computadora y el extremo DB9 macho se conecta al dispositivo paralelo.

Estos conectores solo usan 3 de los Pines: el 2 que es RXD para la recepción de datos, el PIN 3 que es TXD transmisión de datos y el PIN 5 que es GND señal de tierra.

##### **4.2 RECURSOS UTILIZADOS**

- Computadora personal
- 2 microcontroladores PIC16F877.
- 1 integrado Max232
- 4 capacitores electrolíticos de 10 micro faradios
- 1 capacitor electrolítico de 1 micro faradio
- 1 capacitor electrolítico de 470 micro faradios
- 2 conectores DB9 Hembra
- 1 conector DB9 Macho
- 1 cable UTP de 1.50m
- 2 cristales de 4 Mhz
- 4 capacitares cerámicos de 33 pico faradios
- 5 capacitares cerámicos de 104 pico faradios
- 2 resistencias de 100 ohms
- 16 resistencias de 1 KOHM
- 17 Leds
- 1 push button

### 4.3 TARJETA PARALELA DE PREPROCESAMIENTO DE IMÁGENES DIGITALES.

La tarjeta esta construida por una entrada DB9 hembra el PIN 5 esta conectado a tierra, el PIN 2 RX conectado al PIN 14 de MAX232 y el PIN 3 TX conectado al PIN 13 de Max232 ver imagen 35

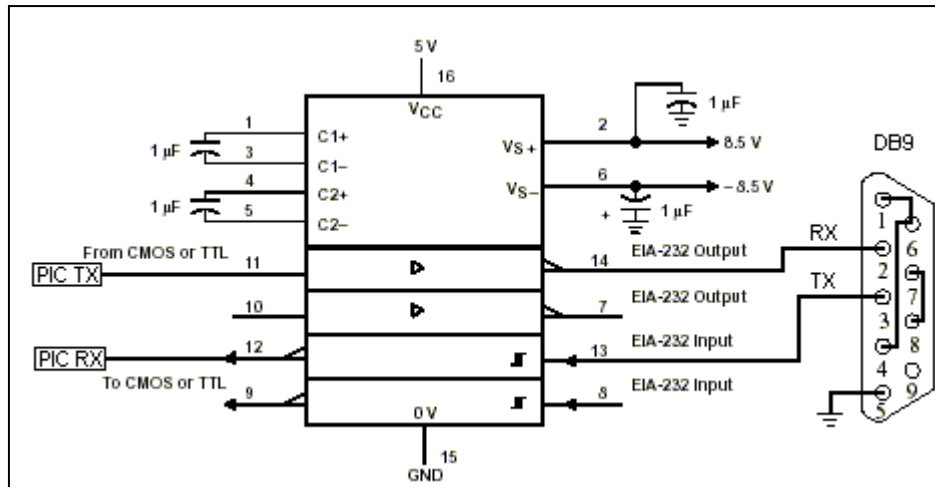


Imagen 35 Conexiones MAX232 con el conector DB9

Para llegar a la construcción final de la tarjeta paralela se tubo que realizar una tarjeta con arquitectura SISD (una sola instrucción un solo dato) Ver imagen 36., para facilitar las pruebas. Tanto a la arquitectura paralela y la arquitectura SISD a los microcontroladores se les coloco en el puerto b (PIN 36 al PIN 37) leds para poder realizar pruebas con los programas de preprocesamiento de imágenes

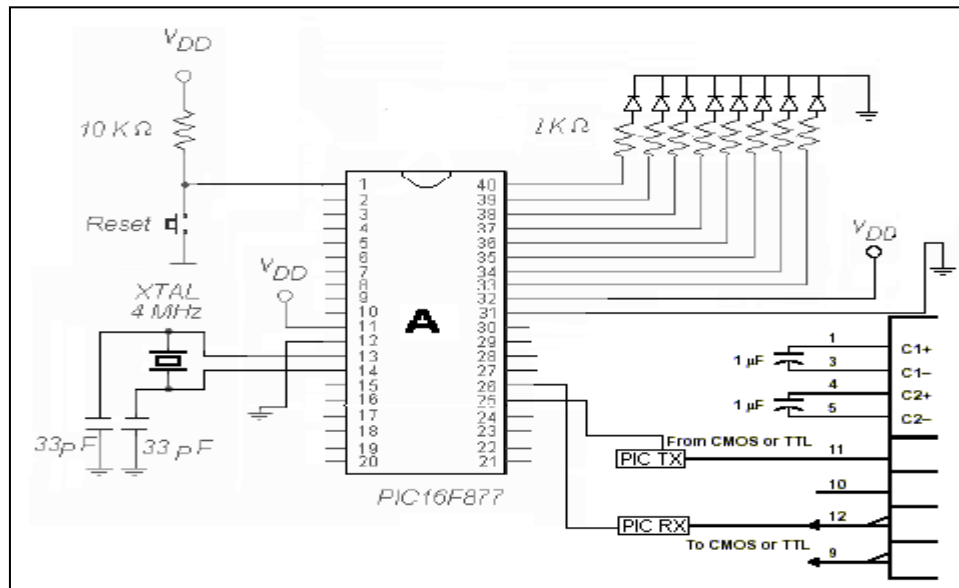


Imagen 36 Arquitectura SISD de prueba

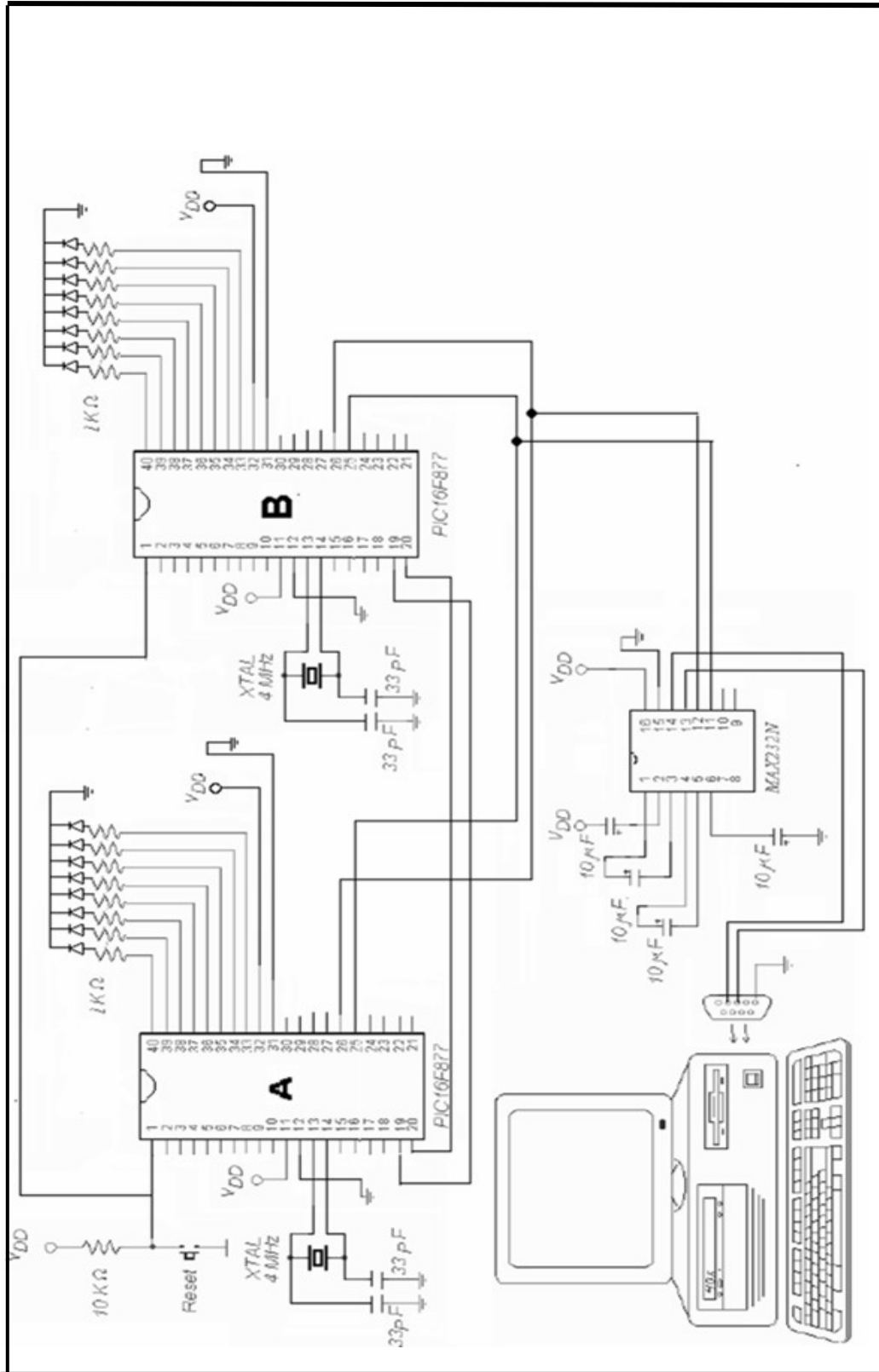


Imagen 37 Arquitectura final del proyecto de tesis

## SOFTWARE

En esta parte del capítulo se explican los procedimientos que componen la programación de los microcontroladores y de la interfaz con el usuario, para la explicación de estos procedimientos se utilizara pseudocódigo.

El pseudocódigo describe un algoritmo utilizando una mezcla de frases en lenguaje común y palabras clave que definen las estructuras básicas. para expresar instrucciones de programación

### 4.4 SOFTWARE DE LOS MICROCONTROLADORES

**Procedimiento inicializar** Este procedimiento inicializa puerto B para poder visualizar información en las pruebas, configurar transmisión-recepción asíncrona para uso del puerto serie y la comunicación entre los microcontroladores

#### PSEUDOCODIGO INICIALIZAR

“DIRECCION DE PUERTO B”

- Fijar puerto B como salida
- Limpiar puerto B

“CONFIGURACIÓN RS232”

Ir a banco 1

- Habilita PIN 6 del puerto C como transmisión
- Habilita PIN 7 del puerto C como recepción
- Habilitar transmisión asíncrona en velocidad rápida
- Fijar velocidad de comunicación en Baudios a 9600

Ir a banco 0

- “Si es microcontrolador inicial”
- Habilitar puerto serial y recepción continua

“CONFIGURAR COMUNICACIÓN ENTRE MICROCONTROLADORES “

Ir a banco 1

“Microcontrolador inicial “

- Habilita PIN 0 del puerto D como salida
- Habilita PIN 1 del puerto D como entrada

“Microcontrolador que espera señal”

- Habilita PIN 0 del puerto D como entrada
- Habilita PIN 1 del puerto D como salida

Ir a banco 0

- Inicializar salida en 0

Inicializar entrada en 1

**Procedimiento Transmisión** este procedimiento transmite por en PIN 6 del puerto C de modo asíncrono un BYTE usando una variable llamada DATO de tamaño BYTE.

#### **PSEUDOCODIGO TRANSMICION**

“Mueve DATO al registro de salida (TXREG)”  
TXREG <= DATO  
“Verificar envió del DATO”  
Si el registro esta lleno  
    Volver a verificar hasta que este vació en registro  
Sino  
    Salir del ciclo de verificación el DATO se transmitió

**Procedimiento Recibir** este procedimiento recibe por el PIN 7 del puerto C de modo asíncrono un BYTE usando también la variable DATO para guardar lo recibido.

#### **PSEUDOCODIGO RECIBIR**

“Verificar si llego un BYTE”  
Si llego un BYTE en el registro de entrada RCREG  
    Guardar en DATO  
Sino volver a verificar hasta que llegue un BYTE  
Enviar El DATO al procedimiento que llamo la rutina

**Procedimiento liberar puerto:** este procedimiento es parte de la sincronización de los dos microcontroladores al terminar de realizar cualquier tarea que requiera transmitir o recibir es necesario liberar el puerto para que el otro microcontrolador pueda utilizarlo si lo requiere.

#### **PSEUDOCODIGO LIBERAR PUERTO**

Verificar si esta en el banco cero de la memoria  
“Asignar cero a al registro que habilita la recepción para deshabilitar puerto”  
RCSTA = 0  
“Poner en uno al PIN D0 para que el otro microcontrolador como aviso de que el puerto fue liberado”  
D0 <= 1  
“Para asegurar que se reciba la señal en el otro microcontrolador “  
    Retarda su siguiente tarea un milisegundo

**Procedimiento esperar turno:** en este procedimiento se realiza el monitoreo del PIN de entrada del puerto D para que el microcontrolador pueda verificar si el puerto de comunicación esta libre o esta ocupado.

#### PSEUDOCODIGO ESPERAR TURNO

```
Si puerto de entrada esta en cero
    Esperar hasta que reciba uno
Sino
    Envía cero en PIN de salida
“Para continuar siguiente tarea esperar para que el otro microcontrolador se le envíe
mensaje de que se ocupo el puerto
    Esperar un milisegundo
    IR al banco cero
    Habilita puerto de comunicación serial
```

**Procedimiento aplicar Filtro Promedio** En este procedimiento se va recorriendo los datos guardados en la memoria EEPROM de nueve en nueve para juntar la vecindad después se asignan los valores a un arreglo y se suman y se realiza el promedio

#### PSEUDOCODIGO FILTRO PROMEDIO

```
K = 0
desde j1=0 hasta j1<12
    desde j = 0 hasta j<16 inicio
        suma=0;
        i=k;
        M[0]=lee posición (i) eeprom M[1]= lee posición (i +1) de eeprom
        M[2]=lee posición (i+2) de eeprom M[3]=lee posición(i+18) de eeprom
        M[4]=lee posición (i+19)de eeprom M[5]=lee posición (i+20)de eeprom
        M[6]= lee posición (i+36)de eeprom M[7]= lee posición (i+37)de eeprom
        M[8]= lee posición (i+38) de eeprom
        Suma desde M[0] hasta M[9]
        T = suma/9;
        Escribe nuevo valor P en posición (i+16) de EEPROM;
    k++;
fin
k = k+2;
fin
```

**Procedimiento aplicar Filtro Mediana** En este procedimiento se va recorriendo los datos guardados en la memoria EEPROM también asignándolos a un arreglo

#### PSEUDOCODIGO FILTRO MEDIANA

```
K = 0
```

```

desde j1=0 hastaj 1<12
  desde j = 0 hasta j<16 inicio
    suma=0;
    i=k;
    M[0]=lee posición (i) eeprom M[1]= lee posición (i +1) de eeprom
    M[2]=lee posición (i+2) de eeprom M[3]=lee posicion(i+18) de eeprom
    M[4]=lee posición (i+19)de eeprom M[5]=lee posicion (i+20)de eeprom
    M[6]= lee posición (i+36)de eeprom M[7]= lee posición (i+37)de eeprom
    M[8]= lee posición (i+38) de eeprom
    Ordenar arreglo M
    T = M[4];
    Escribe nuevo valor P en posición (i+16) de EEPROM;
    k++;
  fin
k = k+2;
fin

```

**Procedimiento aplicar filtro Alfa Trim** En este procedimiento se va recorriendo los datos guardados en la memoria EEPROM también asignándolos a un arreglo

#### PSEUDOCODIGO FILTRO ALFA TRIM

```

K = 0
desde j1=0 hastaj 1<12
  desde j = 0 hasta j<16 inicio
    suma=0;
    i=k;
    M[0]=lee posición (i) eeprom M[1]= lee posición (i +1) de eeprom
    M[2]=lee posición (i+2) de eeprom M[3]=lee posicion(i+18) de eeprom
    M[4]=lee posición (i+19)de eeprom M[5]=lee posicion (i+20)de eeprom
    M[6]= lee posición (i+36)de eeprom M[7]= lee posición (i+37)de eeprom
    M[8]= lee posición (i+38) de eeprom
    Ordenar arreglo M
    Sima desde M[1] hasta M[7]
    T =suma/7;
    Escribe nuevo valor P en posición (i+16) de EEPROM;
    k++;
  fin
k = k+2;
fin

```

**Procedimiento aplicar filtros Paso alto** En estos procedimientos se va recorriendo los datos guardados en la memoria EEPROM multiplicandos por el filtro que se haya decidido también asignándolos a un arreglo y después sumando el arreglo y guardando el resultado en la posición  $i + 19$ .

#### PSEUDOCODIGO FILTROS ALTO Y SHARPEN

```

K = 0
F[9] a este arreglo se le asignan los valores del filtro deseado
desde j=0 hasta j<12 inicio
desde j = 0 hasta j<16 inicio
suma=0;
i=k;
M[0]=lee posición (i) eeprom multiplica F[0]
M[1]= lee posición (i +1) de eeprom multiplica por F[1]
M[2]=lee posición (i +2) de eeprom multiplica por F[2]
M[3]=lee posición(i+18) de eeprom multiplica por F[3]
M[4]=lee posición (i+19)de eeprom multiplica por F[4]
M[5]=lee posición (i+20)de eeprom multiplica por F[5]
M[6]= lee posición (i+36)de eeprom multiplica por F[6]
M[7]= lee posición (i+37)de eeprom multiplica por F[7]
M[8]= lee posición (i+38) de eeprom multiplica por F[8]
Ordenar arreglo M
Sima desde M[1] hasta M[7]
Escribe nuevo valor de la suma en la posición (i + 19) de EEPROM;
k++;
fin
k = k+2;
fin

```

**Procedimiento Principal:** en este procedimiento se realizan las llamadas a los procedimientos para el funcionamiento del preprocesamiento de la imagen.

#### PSEUDOCODIGO PRINCIPAL

```

llamar Procedimiento inicializar
"Hacer los siguientes procedimientos hasta terminar "
Recibir DATO
Filtro = DATO
Esperar turno
Para i = 0 hasta i menor a 225 hacer
    Recibir DATO
    Escribe en la memoria EEPROM en la dirección i el dato
Liberar puerto
Para i = 0 hasta i menor a 225 hacer
    leer la posición i de la memoria EEPROM
    mover BYTE leído a DATO
    Transmitir DATO
    Esperar un milisegundo para que se termine de enviar dato
Liberar recepción
Limpia puerto B

```

## 4.5 SOFTWARE DE LA INTERFAZ

Para la comunicación serie se utilizo la librería COMPORT.dll para delphy y C++ de la cual los siguientes procedimientos fueron utilizados en el sistema.

```
Function OPENCOM(openStrin:PChar):integer; stdcall; external 'RSCOM.DLL';  
Procedure CLOSECOM(); stdcall;external 'RSCOM.DLL';  
Procedure SENDBYTE(Dat:integer);stdcall; external 'RSCOM.DLL';  
Function READBYTE():Integer; stdcall; external 'RSCOM.DLL';
```

**Procedimiento constructor** en el se realiza la configuración de los valores para abrir el puerto y se crea un mapa de bits.

### PSEUDOCODIGO CONSTRUCTOR

Se crea mapa de bits BM  
Se dan los valores para la comunicación  
Puerto COM1  
Baudios: 9600  
Datos 8  
Se abre la comunicación con OpenCOM

**Procedimiento cerrar** Se cerrar se llama al la función CloseCom para que libere el puerto Com1 y se cierra la aplicacion

**Procedimiento cargar imagen** al dar clic el botón se abre un cuadro de dialogo y se elige la ruta y la imagen deseada para mostrarse en pantalla.

### PSEUDOCODIGO CARGAR IMAGEN

Abrir dialogo  
Se asigna a un mapa la imagen a un mapa al mapa de bits  
Se muestra en pantalla la imagen  
Se extraen las medidas de la imagen  
Renglones  $\leq$  altura de la imagen  
Columnas  $\leq$  ancho de la imagen  
Se da las dimensiones a la matriz donde se trabajara la imagen  
Se pasa el mapa de bits a una matriz para que se pueda precisar.

**Procedimiento convertir mapa de bits a una matriz** se realiza el cambio de formato para poder manipular los datos de la imagen.

#### **PSEUDOCODIGO MAPA DE BITS A MATRIZ**

Desde  $y = 0$  a renglones-1  
A:= se asigna un vector del mapa que contiene los 3 planos de color  
Desde  $x:=0$  a columnas-1  
K:= cada 4 datos de x es necesario un plano solo imágenes en escala de gris  
 $M[x,y] = A[k]$

**Procedimiento convertir mapa de bits a una matriz** se realiza el cambio de formato para poder manipular los datos de la imagen.

#### **PSEUDOCODIGO MATRIZ A MAPA DE BITS**

Desde  $y = 0$  a renglones-1  
A:= se asigna un vector del mapa que contiene los 3 planos de color  
Desde  $x:=0$  a columnas-1  
K:= cada 4 datos de x es necesario un plano solo imágenes en escala de gris.  
Se asignan los valores a los 3 planos  
 $A[k] = M[x,y];$   
 $A[k+1] = M[x,y];$   
 $A[k+2] = M[x,y];$

**Procedimiento enviar datos** en el se describe el envío de los datos por bloques cada bloque es dividido en dos para que cada microcontrolador filtre una parte

#### **PSEUDOCODIGO ENVIAR DATOS**

Un bloque de 476 datos se reparte entre los dos PICs  
Desde  $K = 1$  hasta 2 do  
Se envía filtro a utilizar  
desde  $y$  que se asigna comienzo del bloque  
hasta el término de la mitad +1 del bloque  
desde  $X$  que se le asigna el inicio del bloque hasta  
el término de la mitad mas 1 del bloque  
Enviar  $M[x,y]$   
fin del envío de la primera parte de la imagen a un PIC  
se establece que se envíe nuevamente dos columnas de la primera parte para que se pueda tratar la orilla derecha y se envía la otra parte de la imagen al otro PIC

**Procedimiento enviar datos** en el se describe la recepción de los datos de los microcontroladores y la unión de estos datos.

#### **PSEUDOCODIGO RECIBIR DATOS**

Se prepara para recibir dos bloques en orden de como fueron enviados

Desde  $y$  índice inicial de del largo hasta el fin del primer bloque menos uno  
Desde  $x$  índice inicial del bloque hasta el final  
Se recibe datos  $y$  se asignaba a la matriz reconstruida  
Mat2 [ $x, y$ ]:=datos.  
Fin del primer envió  
Se establecen los índices para la recepción del segundo bloque

**Procedimiento Procesar imagen** se describe como es la partición de la imagen para que se pueda filtrar con los microcontroladores.

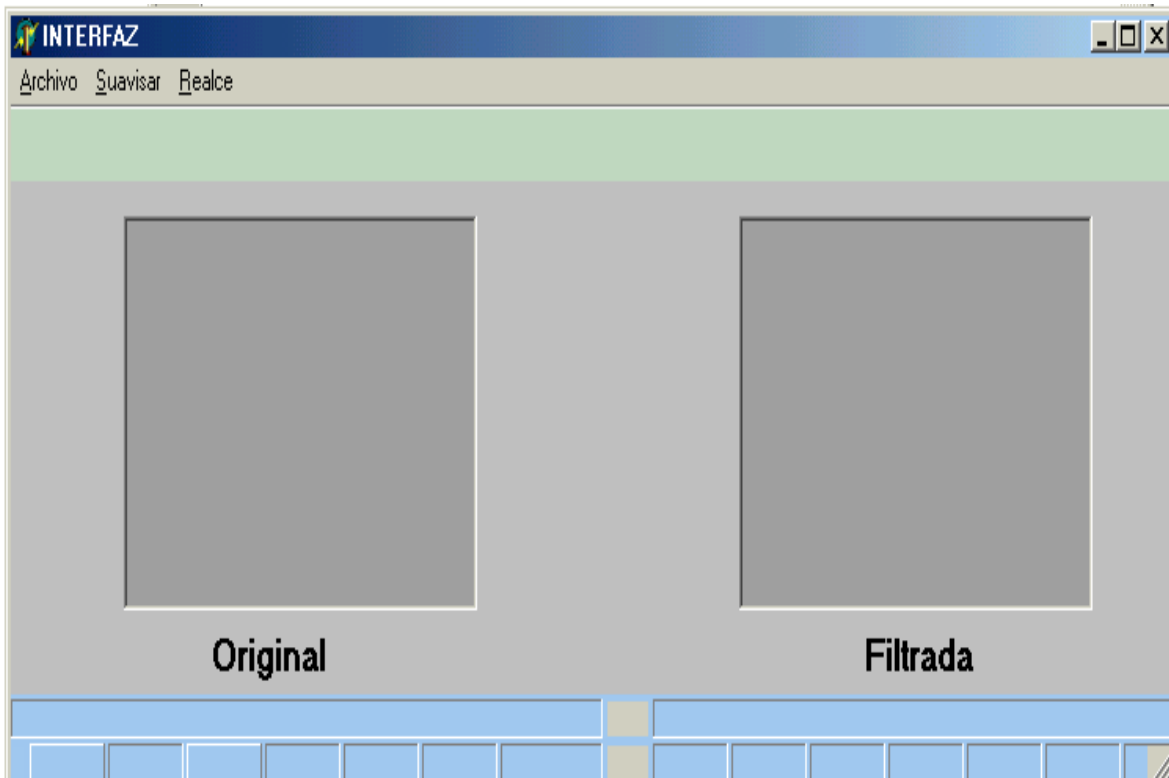
#### PSEUDOCODIGO PROCESAR IMAGEN

Se asigna el ancho total de la matriz menos uno a la variable ancho  
Se asigna el largo total de la matriz menos uno a la variable largo  
Se establece el tamaño del bloque que se enviara será de 14 de largo por 34 de ancho  
A las variables  $al$  y  $an$  respectivamente  
A la variable Filtro se le asigna el numero de procedimiento a usar  
Repite hasta que se llegue al alto total de la matriz

Si el ultimo bloque es mas chico que 14 de alto entonces  
Ajustar el alto del bloque a enviar  
Repite hasta que se llegue al ancho total de la matriz  
Si el ultimo bloque es mas pequeño que 34 de ancho entonces  
Ajustar el ancho del bloque a enviar  
Se llama al procedimiento enviar primer bloque a los dos microcontroladores  
Una vez enviados los datos se llama al procedimiento recibir  
Se retroceden dos columnas para tratar las orillas del bloque anterior  
Se retroceden dos renglones para tratar las orillas del bloque anterior  
Una vez terminada la recepción se convierte la matriz a un mapa de bits  
Se muestra en pantalla el resultado

Ver imagen de la interfaz y de la tarjeta paralela imagen 38

Ver imagen de la tarjeta paralela imagen 39



**Imagen 38** interfaz con el usuario en la cual puede elegir la imagen que se desea preprocesar y el filtro mostrando en el lado derecho el resultado



**Imagen 39** Tarjeta final del proyecto de tesis

## **CAPITULO V**

### **PRUEBAS, CONCLUSIONES Y PERSPECTIVAS**

En el presente capitulo se derriben las pruebas realizadas al sistema de preprocesamiento paralelo. Como el sistema esta combinado por Hardware y software se tuvieron que realizar por separado las pruebas para cada parte.

Para el hardware se realizaron pruebas en primera instancia con un solo microcontrolador para verificar así la comunicación con la computadora y la programación para microcontroladores en lenguaje C para PICS una vez que se verifico el sistema con un solo microcontrolador se instalo otro microcontrolador para que funcionaran de forma paralela por lo tanto también se realizaron pruebas para la sincronización de los PIC.

En cuanto al software se relazaron pruebas de cada bloque por separado tanto del programa de los microcontroladores como el de la interfaz con el usuario con cosas más sencillas para la facilitar la detección de los errores en cada programa.

#### **51. PRUEBAS DE HARDWARE**

Se relazan las pruebas eléctricas, pruebas del puerto serie y pruebas con los microcontroladores

##### **5.1.1 PRUEBAS ELECTRIAS**

Se construyo una pequeña fuente de poder para la alimentación de los microcontroladores y el circuito integrado MAX232 la cual tenía que emitir 5V. Se relazaron mediciones con un multmetro digital para la comprobación del voltaje requerido.

Se relazaron también las pruebas eléctricas en el diseño del sistema paralelo con el multmetro digital en los sitios que requerían alimentación de 5V para verificar que estaban siendo alimentados los circuitos correctamente

##### **5.1.2 PRUEBAS DE CONEXIÓN PARA EL PUERTO SERIE**

También se realizaron pruebas de continuidad al construir el cable de comunicación serie usando el multmetro comprobando que estuvieran unidos los conectores DB9 por medio de los pines de recepción, transmisión y tierra.

##### **5.1.3 PRUEBAS DE SINCRONIZACIÓN DE LOS MICROCONTROLADORES**

Para relazar esta prueba se utilizo el puerto b de los microcontroladores (de hecho para las pruebas con los microcontroladores se utiliza el puerto B para verificar el buen funcionamiento de los mismos) verificando que emitiera un dato por los leds conectados al puerto b cuando cada uno estuviera utilizando el puerto.

#### **5.1.4 PRUEBAS DE COMUNICACIÓN**

Se relazaron las primeras pruebas con un diseño de un solo microcontrolador para verificar las conexiones necesarias para su funcionamiento Después de verificar su funcionamiento se agrego el circuito integrado MAX232 para establecer la comunicación.

Para realizar esta comprobación se utilizo un programa llamado comunica.exe enviando daros a los microcontroladores verificando por el puerto B que fueran recibidos también fue utilizado este programa para enviar datos desde los microcontroladores a la computadora comprobando la comunicación.

### **5.2 PRUEBAS DE SOFTWARE**

Unas vez que se comprobó el funcionamiento del hardware tanto en conexiones como en comunicación se realizo las pruebas del software.

#### **5.2.1 SOFTWARE DE LOS MICROCONTROLADORES**

Se relazaron pruebas por separado con el programa comunica.exe de cada modulo del programa de los microcontroladores al principio con nueve datos simulando las vecindades de 3X3 aplicando los filtros para verificar que realizaran los microcontroladores las operaciones.

Una vez verificado esto se realizo la transmisión de la imagen para que la preprocesaran con cada filtro por separado, después de esto se unieron en un solo programa los filtros para que fueran elegidos desde la computadora.

#### **5.2.2 SOFTWARE DE LA INTERFAZ CON EL USUARIO**

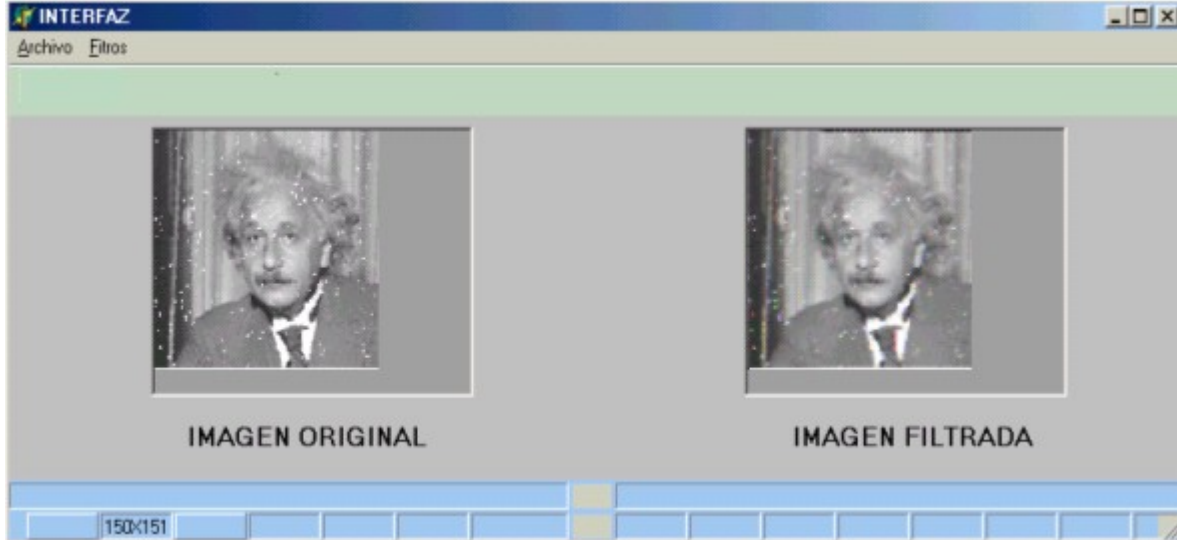
Para la prueba de envió en bloques de la imagen se rehizo con una matriz pequeña de enteros simulando la transmisión con archivos es decir cuando se tenia que enviar ese trozo de la matriz se guardaba en un archivo y al final se leían los datos guardados reconstruyendo la matriz.

Comprobando esto se realizo el envió de esta matriz de prueba a los microcontroladores sin inconvenientes entonces se hizo la prueba con una imagen.

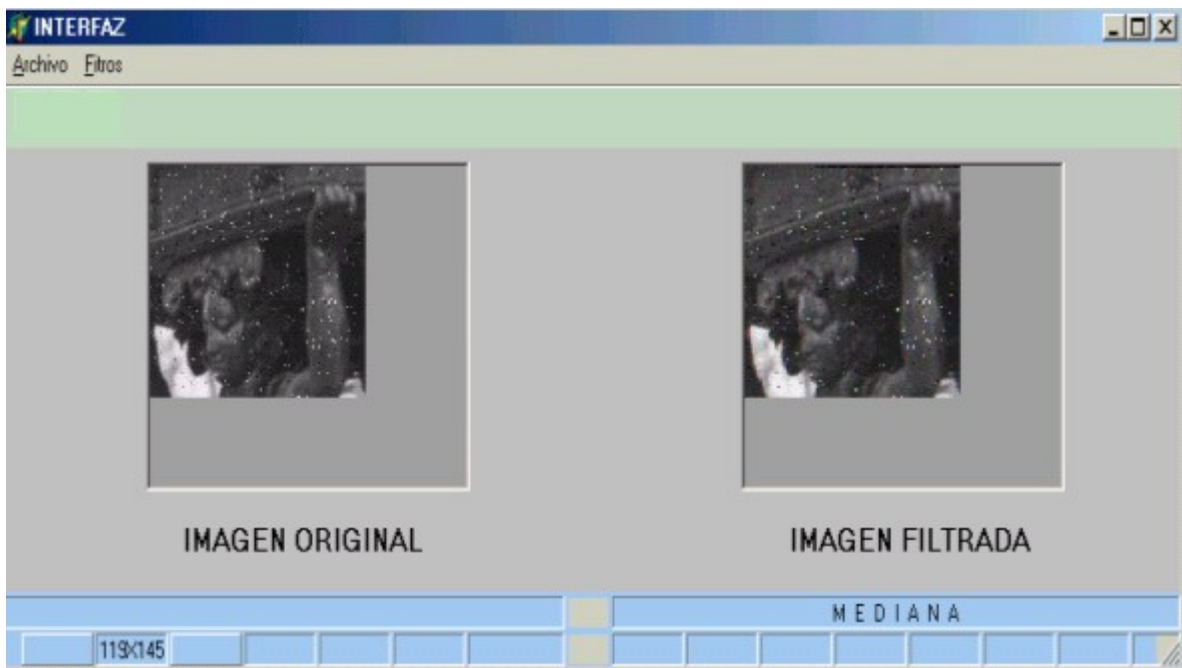
### **5.3 PRUEBAS DEL FUNCIONAMIENTO DEL SISTEMA**

Una vez que se comprobó el funcionamiento de la sincronización de los microcontroladores y de cada modulo de cada programa (interfaz y de los microcontroladores) se hicieron pruebas finales que se ilustran a continuación

### 5.3.1 PRUEBAS DE SUAVIZADO

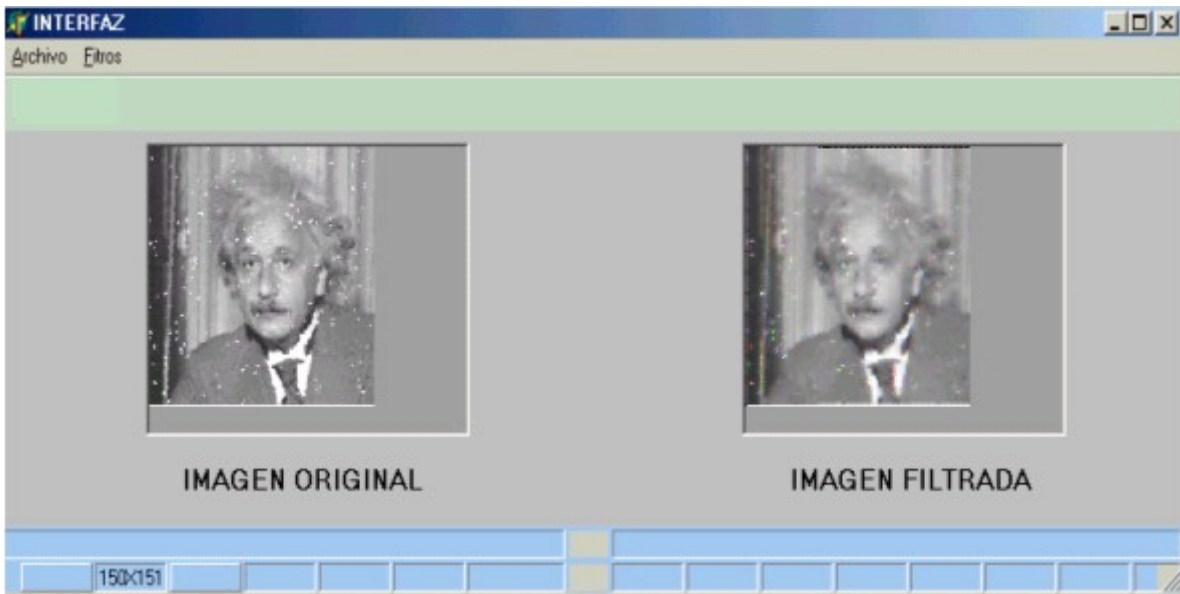


**Imagen 40** filtro Promedio se muestra que la imagen esta borrosa

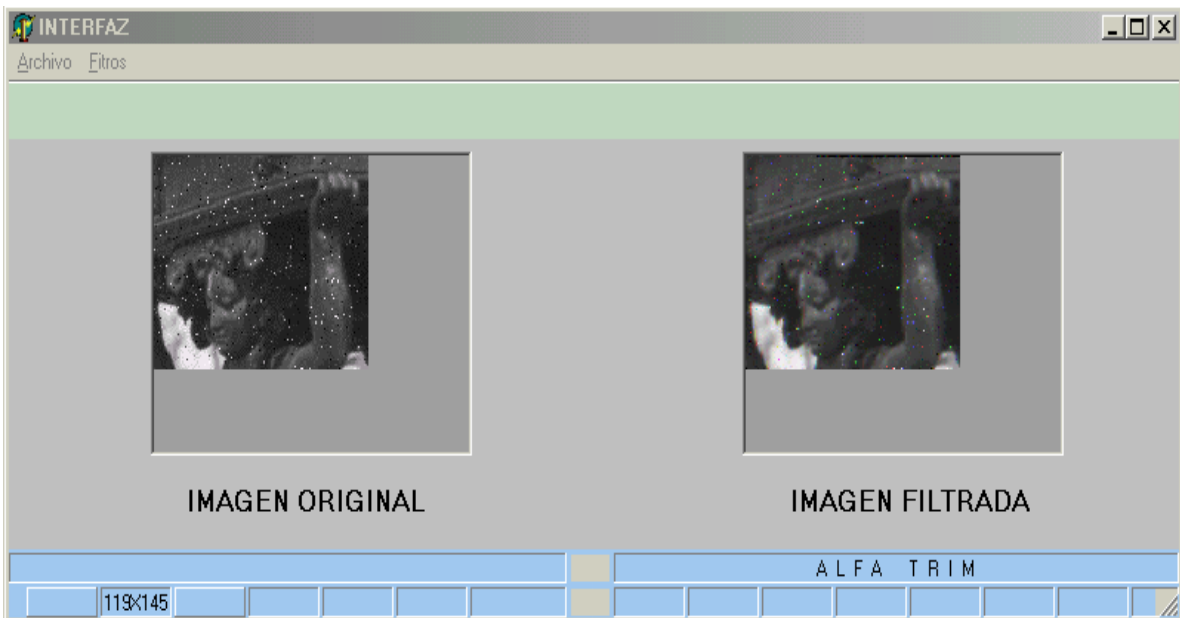


**Imagen41** Filtro de la mediana

Se puede realizar el filtrado otra vez sobre la imagen ya tratada se demuestran algunos ejemplos continuación:



**Imagen42** se muestra la imagen tratada dos veces con el filtro del promedio ha perdido mas ruido pero presenta mas borrosa.

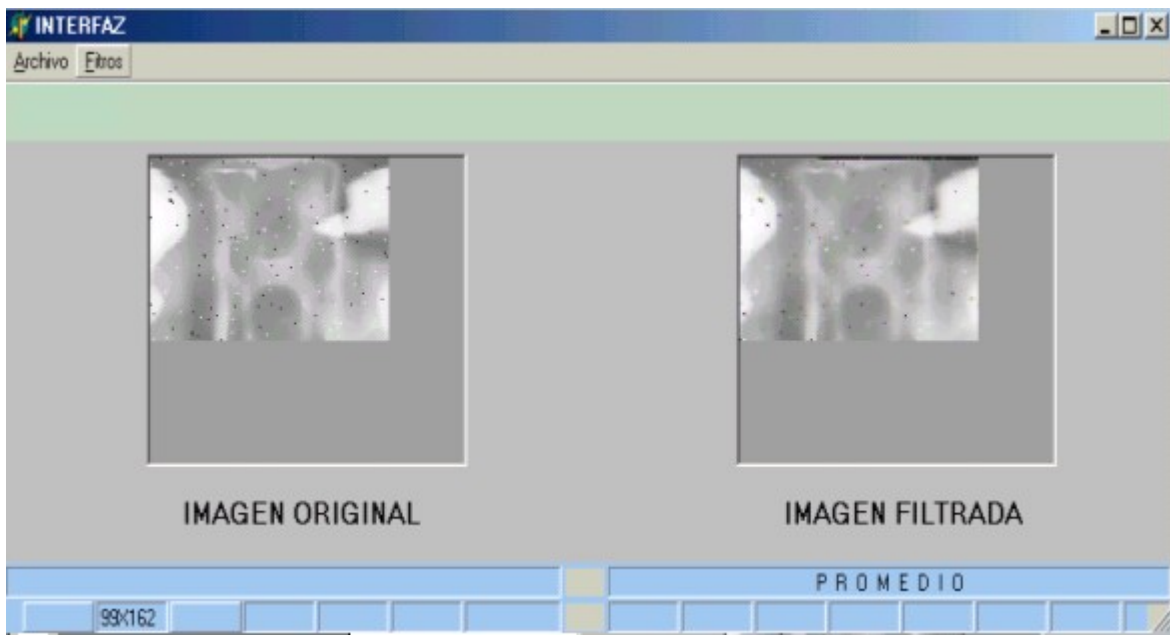


**Imagen 43** Se aplico 3 veces el filtro de la mediana se comprueba que ha desaparecido mas puntos

También se pudo hacer combinaciones de filtro por ejemplo aplicando filtro alfa trim y mediana ver imagen 44

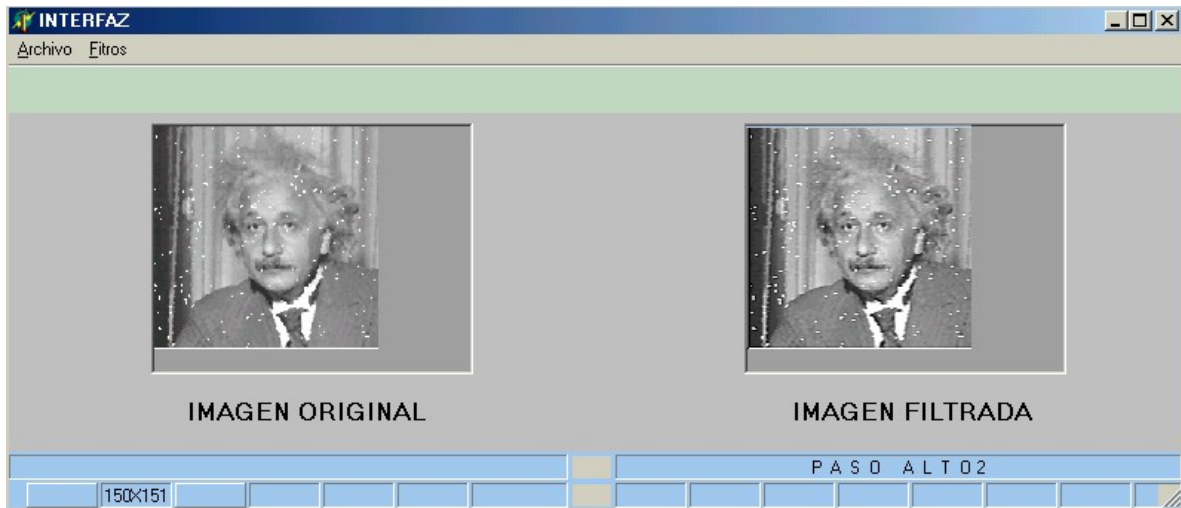


**Imagen 44** se muestra el funcionamiento del filtro alfa trim aplicado 3 veces



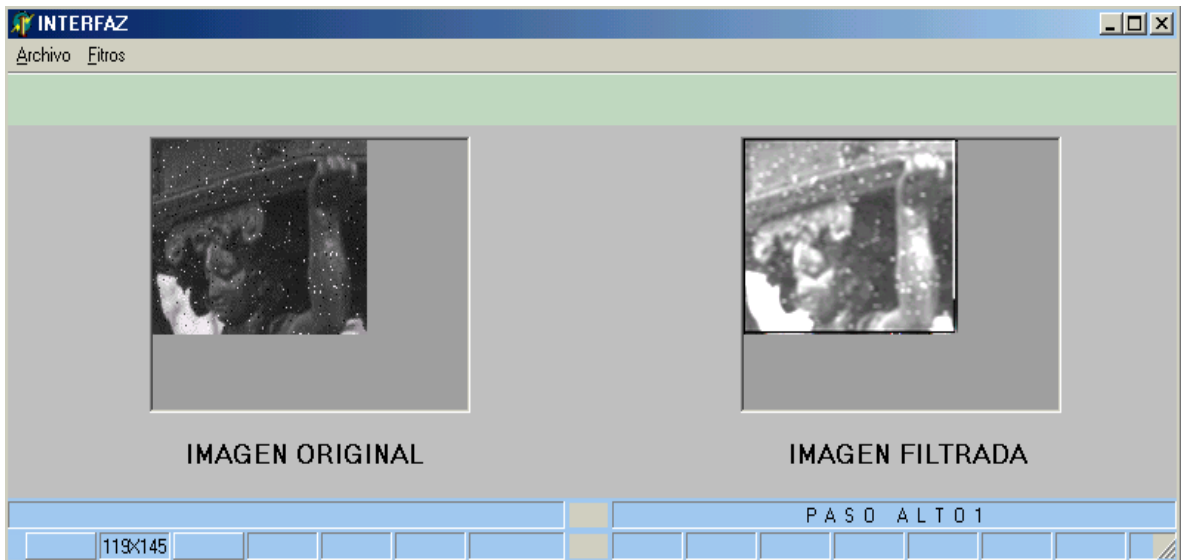
**Imagen 45** se muestra la aplicación de la combinación de 2 filtros alfa trim y promedio ha desaparecido mas ruido

### 5.3.2 PRUEBAS DE REALCE DE DETALLES



**Imagen 46** Filtro paso alto 2 que se muestra abajo

|        |        |        |
|--------|--------|--------|
| $-1/9$ | $-1/9$ | $-1/9$ |
| $-1/9$ | $8/9$  | $-1/9$ |
| $-1/9$ | $-1/9$ | $-1/9$ |



**Imagen 47** Filtro paso alto 1

|        |        |        |
|--------|--------|--------|
| $1/4$  | $-1/2$ | $1/4$  |
| $-1/2$ | $5/4$  | $-1/2$ |
| $1/4$  | $-1/2$ | $1/4$  |

## 5.4 RESTRICCIONES DEL SISTEMA

- Solo trabaja con imágenes en escala de grises
- En imágenes grandes el tiempo de proceso es muy grande
- Solo cuenta con 3 filtros de suavizado y 3 de realce.
- Poca memoria EEPROM en los microcontroladores lo cual hace lento el reprocesamiento ya que tiene que ser dividida la imagen por varios bloques.

## 5.5 APLICACIONES

Se podría utilizar esta clase de sistema en visión artificial ya que no sería necesario tener una computadora para que un robot pudiera tener una mayor calidad en las imágenes que capturara.

## 5.6 CONCLUSIONES

Al finalizar este trabajo de tesis se puede verificar con las pruebas vistas en este capítulo que se alcanzaron los objetivos planteados en el capítulo 3 en un 100%.

Se desarrolló un arreglo paralelo de 2 microcontroladores el cual puede utilizarse para resolver diversos problemas no solo en el de preprocesar una imagen, ya que se pueden reprogramar los microcontroladores para la solución de otra tarea en forma paralela.

A lo largo de esta tesis también se observó que se dividió en varias partes este sistema para llegar a obtener los resultados deseados tanto en hardware como en software.

A su vez el software también se dividió en dos partes: en una interfaz la cual cumplió con el objetivo de comunicar al usuario con el dispositivo electrónico y el software de los microcontroladores que realiza la eliminación de ruido y realce de imágenes.

Esto aplica el paradigma de *divide y vencerás* para alcanzar los objetivos generales y específicos mencionados. Se concluye también que se pueden construir aplicaciones con materiales de bajo costo para resolver diversas tareas.

## 5.7 PERSPECTIVAS

En esta sección se listan las propuestas de las mejoras que se pudieran hacer en el presente trabajo.

- Se puede desarrollar un sistema en el que pueda preprocesar imágenes a color.

- Extender el arreglo paralelo a más microcontroladores para manipular un volumen mayor de datos.
- Aumentar el numero de filtros para eliminación de ruido y realce de imágenes
- Implementar mas tareas para procesar imágenes como el aumentar una imagen el rotarla etc.

## BIBLIOGRAFIA

- [1] R Molina. Departamento de Ciencias de la Computación e I.A Universidad de granada  
“Introducción Al Procesamiento y Análisis de Imágenes Digitales”
- [2] Universidad Miguel Hernández Div. Ingeniería de Sistemas y Automática  
“Visión por computador”  
<http://isa.umh.es/isa/es/asignaturas/rvc/Tema%204.%20Reduccion%20del%20ruido.pdf>.
- [3] Dr. Manuel Martín Ortiz “Procesamiento Digital de Imágenes”, Benemérita universidad Autónoma de Puebla <http://www.cs.buap.mx/~mmartin/pdi>.
- [4] robótica: control, detección, visión e inteligencia ks. fu gonzález edit. mc. graw hill
- [5] Custom Computer Services Inc. Febrero 2002 “C Compiler Refence Manual”
- [6] Microchip ,PIC 16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH microcontrollers.
- [7] Amgulo Usategui. José Ma, Angulo Matines Ignacio. “Microcontroladores <<PIC>> Diseño Practico de Aplicaciones”, McGraw-HILL.
- [8] <http://delhipage.free.fr/portserie.html>
- [9] <http://www.ortodoxism.ro/datasheets/maxim/MAX220-MAX249.pdf>