

IV.3 DISEÑO Y CONSTRUCCIÓN DE SISTEMAS INTERACTIVOS APLICANDO EXPERIENCIAS BASADAS EN EL TRATAMIENTO DE IMÁGENES

*Fernando Galindo Soria**

*Jesús Antonio Jimenez Aviña***

Resumen

La Interactividad estudia los mecanismos para lograr que dos o más dispositivos o sistemas se relacionen reciproca y simultáneamente entre si, de tal manera que las acciones e información generada por unos repercutan en los otros. Es un área en la que se manejan cientos de miles de millones de dólares y en la que se involucran la robótica, realidad virtual, sistemas evolutivos, telecontrol, simuladores, biótica, multimedia, sistemas de comunicación, control distribuido, música por computadora, etc.

La Interactividad es una de las áreas de mayor desarrollo a nivel mundial, sin embargo en nuestras escuelas prácticamente no existen cursos sobre este tema, Lo anterior es en parte por desconocimiento y en parte porque se cree que requiere muchos recursos. Lo ultimo es relativamente cierto, por ejemplo, si se quiere realizar un sistema de realidad virtual o de reconocimiento de voz normalmente no se cuenta con los recursos necesarios para hacerlo.

Sin embargo, al analizar múltiples sistemas interactivos nos damos cuenta que manejan componentes parecidos y que es relativamente fácil aplicar los conocimientos adquiridos al desarrollar unos sistemas en el desarrollo de otros.

Se podría decir que es distinto manejar sonido, presión, temperatura o imágenes, pero lo único distinto es la forma como lo percibimos, ya que si manejamos imágenes tendremos que sacarlas por el monitor y en el caso de sonido por la tarjeta de sonido o por la bocina, pero la computadora los trata a todos como cadenas de 0's y 1's.

* Fernando Galindo Soria y Jesús Antonio Jiménez Aviña escribieron este trabajo en 1996 cuando estaban en la Escuela Superior de Computo del IPN y lo presentaron como ponencia y publicaron en las memorias de la Segunda Conferencia de Ingeniería Eléctrica CIE/96, organizado por el CINVESTAV del IPN, en la Cd. de México en Septiembre de 1996.

Por ejemplo, normalmente lo único que sabemos del sistema de graficación, es que consta de un monitor y una tarjeta de video, pero si empezamos a profundizar nos damos cuenta que es una mina de información sobre interactividad y que prácticamente todo lo que se maneja en un sistema interactivo, tiene una representación concreta en un sistema de graficación.

Es por lo anterior que en este trabajo proponemos que un camino factible para introducir a los estudiantes de Informática y Computación en la interactividad es a través del estudio y manejo de los sistemas de graficación.

Uno de los problemas que se presentan en esta propuesta, es la gran fragmentación del conocimiento ya que mientras en una fuente explican cómo graficar en algún lenguaje, en otras presentan que las imágenes se almacenan en algún tipo especial de formato, en otras hablan de diferentes tipos de tarjetas gráficas, en fin otras presentan la Teoría del Color. Si se profundiza mas, se encuentra que la información sobre formatos gráficos, arquitectura de las tarjetas y sobre su manejo esta muy dispersa, y se detecta que se requieren conocimientos que normalmente se dan desvinculados en los cursos de Arquitectura de Computadoras, Sistemas Operativos y Ensambladores.

Es por lo anterior que en este trabajo presentaremos ejemplos en los cuales se ataca el problema del manejo de imágenes y veremos que es similar al del sonido (es prácticamente igual ya que también se manejan interrupciones, puertos, formatos, memoria, etc. y fácilmente una persona que aprendió a manejar imágenes pasa a sonido) y al de otros tipos de mecanismos interactivos.

Entre otros temas se ve como cambiar de un modo gráfico a otro usando interrupciones y un ejercicio sencillo para introducir al manejo de puertos y a la programación de tarjetas, que consistente en cambiar los valores de la paleta de colores programando directamente los puertos que controlan al DAC de la tarjeta de video.

Otro de los problemas clásicos dentro de la Interactividad se presenta en el manejo masivo de información, en particular cuando se manejan imágenes o sonido se esta tratando con cantidades masivas de información, ya que por ejemplo, una sola imagen contiene alrededor de 256 mil caracteres, por lo que se mostraran ejemplos de manejo de sonido e imagen en forma masiva dentro de

una computadora, incluyendo el manejo de formatos de video y sonido, con el fin de introducir al tema y mostrar que los diferentes problemas de interactividad siguen siendo similares.

INTRODUCCIÓN

La *Interactividad* o el área de los sistemas interactivos estudia los mecanismos y la forma para lograr que *dos o más dispositivos o sistemas se relacionen recíproca y simultáneamente entre sí*, de tal manera que las *acciones e información* generada por unos repercutan en los otros. Es un área en la que se manejan cientos de miles de millones de dólares y en la que se involucran la robótica, realidad virtual, sistemas evolutivos, telecontrol, simuladores, biotica, multimedia, sistemas de comunicación, control distribuido, música por computadora, etc.

Los sistemas interactivos se encuentran presentes en prácticamente todas las actividades cotidianas. Por ejemplo, la interrelación entre el sensor de temperatura y un calentador para regular la temperatura del agua, la forma como se relacionan una cámara de video con una televisión, una computadora con un sistema de audio, en los sistemas para detectar la presencia de personas en un cuarto, en un simulador (de vuelo, del metro, de una planta termoeléctrica), en sistemas de telecontrol, etc.

Esta es una de las áreas de mayor desarrollo mundial, sin embargo en nuestras escuelas prácticamente no existen cursos sobre este tema. Lo anterior es *en parte por desconocimiento y en parte porque se cree que se requiere muchos recursos*. Lo último es relativamente cierto, por ejemplo, si se quiere realizar un sistema de realidad virtual o de reconocimiento de voz normalmente no se cuenta con los recursos necesarios para hacerlo.

Sin embargo, al analizar *múltiples sistemas interactivos* nos damos cuenta que *manejan componentes parecidos y que es relativamente fácil aplicar los conocimientos adquiridos al desarrollar unos sistemas en el desarrollo de otros*. Prácticamente en todos los casos se construyen o

manejan aparatos, interfaces y programas y se requiere usar interrupciones, puertos, archivos con formato, manejadores de entrada/salida, microcontroladores, circuitos programables, etc.

Si se logra crear conciencia entre los estudiantes de las similitudes entre estos sistemas, es relativamente fácil para ellos aplicar los conocimientos adquiridos para desarrollar unos en el desarrollo de los otros.

Y una cosa curiosa y muy interesante es que, *no se requiere contar con una gran cantidad de recursos para aprender a construir este tipo de sistemas*. Ya que en los sistemas de cómputo actuales existen sistemas interactivos completos que en su momento pueden servir para introducir a la gente en la interactividad sin necesidad de conseguir recursos externos. Tal es el caso de los sistemas de graficación y en general de manejo de imágenes de las computadoras.

Comúnmente lo único que sabemos del sistema de manejo de imágenes de las PC's, es que consta de un monitor y una tarjeta que comunica el monitor con la computadora, pero si empezamos a profundizar nos damos cuenta que estamos entrando en una mina de información sobre interactividad y que *prácticamente todo lo que se maneja en un sistema interactivo, tiene una representación concreta en un sistema de graficación por computadora*.

Es por lo anterior que en este trabajo proponemos que *un camino factible para introducir a los estudiantes de Informática y Computación en la interactividad es a través del estudio y manejo de los sistemas de graficación*.

Uno de los problemas que se presentan en esta propuesta, es la gran fragmentación del conocimiento ya que mientras en una fuente explican cómo graficar en algún lenguaje, en otras presentan que las imágenes se almacenan en algún tipo especial de formato (pero sólo lo mencionan y no indican en qué consiste), en otras hablan de diferentes tipos de tarjetas gráficas (pero no explican como se programan), en fin otras presentan la Teoría del Color. Si se profundiza más, se encuentra que la información sobre formatos gráficos, arquitectura de las tarjetas

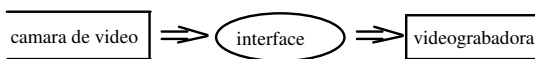
y sobre su manejo esta muy dispersa, y se detecta que se requieren conocimientos que normalmente se dan desvinculados en los cursos de Arquitectura de Computadoras, Sistemas Operativos y Ensambladores.

Y prácticamente en todos los casos se requieren construir o conseguir los aparatos, las interfaces y desarrollar los programas para manejar los sistemas interactivos. Para una gran cantidad de casos se requiere conocer y manejar: interrupciones, puertos, archivos con formato, manejadores de entrada/salida, microcontroladores, circuitos programables, lectura de discos, programación en binario de los registros de las tarjetas, manejo de memoria, etc.

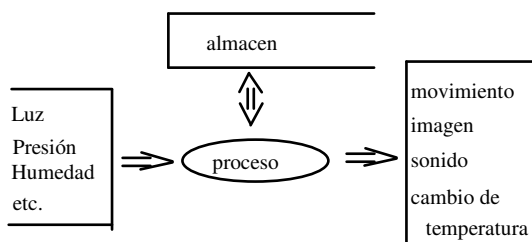
Es por lo anterior que en este trabajo presentaremos ejemplos sobre manejo de imágenes y veremos que el problema es similar al del sonido (es prácticamente igual ya que también se manejan interrupciones, puertos, formatos, memoria, etc. y normalmente una persona que aprendió a manejar imágenes pasa directamente a sonido) y al de otros tipos de mecanismos interactivos.

I ARQUITECTURA GENERAL DE UN SISTEMA INTERACTIVO

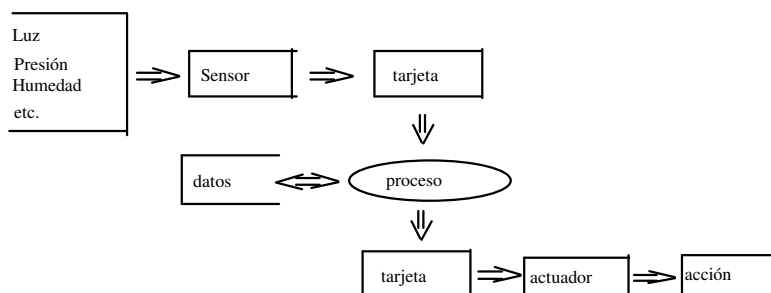
Como primer punto comentaremos que, en un sistema interactivo es necesario lograr que al menos dos objetos se interrelacionen, por ejemplo una cámara de video con una videograbadora, para lo cual se requiere normalmente un mecanismo de interfase entre los dos objetos.



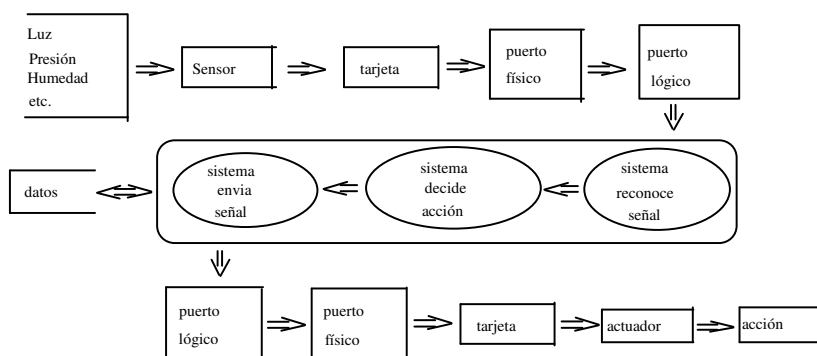
En particular en el caso de las computadoras, la arquitectura de un sistema interactivo es:



Y más específicamente



Si detallamos mas los componentes del sistema interactivo se llega a una arquitectura como la siguiente



Como se puede ver, las principales actividades que se realizan en un sistema interactivo son comunes a cualquier tipo de sistema, pero sin embargo en el caso particular de estos sistemas se tienen ciertas características que conviene resaltar: una es la *capacidad de procesamiento en tiempo real*, ya que un sistema interactivo tiene que

tener un tiempo de respuesta muy rápido, otra es que comúnmente las *interfaces con el sistema no son las tradicionales*, sino que usan prácticamente cualquier mecanismo de adquisición y salida de información.

Otra característica común a este tipo de sistemas es que normalmente *manejan grandes cantidades de información*, ya que, por ejemplo en el envío de una imagen o de una canción entre dos dispositivos se esta manejando un mayor volumen de datos que los manejados en muchos de los procesos administrativos tradicionales.

Una ultima característica consiste en que, *si se desarrolla un sistema interactivo específico, la experiencia particular se puede migrar fácilmente a muchos otros*, ya que en todo los casos se requiere desarrollar mecanismo de captura, procesamiento, almacenamiento y salida de los datos.

Por lo que a pesar de que es muy fácil rendirse y no entrar al área de la interactividad por falta de recursos, ya que, no siempre se cuenta con todo el equipo que se quisiera, esta posición no es valida, ya que todas las computadoras cuentan con una tarjeta de video, por lo que podemos introducirnos a la interactividad programando la tarjeta de video y posteriormente migrando esta experiencia a otros tipos de dispositivos.

Se eligió la tarjeta de video porque los problemas involucrados en su manejo son problemas clásicos que se presentan al manejar cualquier otro dispositivo interactivo como una tarjeta de sonido o de adquisición de datos, ya que todas estas tarjetas se pueden programar directamente y prácticamente en todos los casos en forma parecida.

Por lo que, a continuación se muestra como se enfrentan los problema de entrada/salida y almacenamiento de información en un sistema interactivo, usando como casos particulares la programación de la tarjeta de video para que realice algunas funciones específicas y el almacenamiento y despliegue de imágenes en una computadora.

La programación de cualquier dispositivo interactivo puede ser extremadamente compleja y en su caso existen libros completos de cientos de paginas para programar algunos de estos, por lo que no es el objetivo dar un curso sobre el tema, sino simplemente mostrar con ejemplos sencillos como se puede programar un dispositivo interactivo y crear conciencia de que este proceso se puede realizar con muchos otros dispositivos.

II CONTROLADORES DE VÍDEO

Se podría decir que es distinto manejar sonido, presión, temperatura o imágenes, pero lo único distinto que tienen es la forma como lo percibimos, ya que la computadora los trata de la misma forma, como cadenas de 0's y 1's. Si manejamos imágenes tendremos que sacar la información por medio del monitor y en el caso del sonido por medio de la tarjeta de sonido o por la bocina interna de la PC.

Para manejar cualquiera de estos tipos de información, además de que es aconsejable conocer como se transmite el sonido o como se generan los colores (ya que, entender esto facilita el manejo de la información), debemos saber también como programar los dispositivos de entrada y salida.

Por ejemplo para manejar gráficas por computadora los compiladores de *C y Pascal* ya cuenta con controladores de video como los **.bgi*, los cuales nos permiten manejar el modo gráfico para varios tipos de tarjetas, desde *CGA* y *EGA* hasta *VGA*.

Pero a pesar de que en una tarjeta normal VGA la memoria consta de 256K estos controladores solo permiten una resolución máxima de 640X480 pixeles y 16 colores. Si queremos realizar un trabajo de mejor calidad tenemos que ocupar controladores como el *VGA256.BGI* o *SVGA256.BGI*, que aumenta el manejo de colores de 16 a 256. y dependiendo del tipo de tarjeta aumenta la resolución.

Otra opción consiste en construir nuestro propio controlador, para lo cual debemos programar directamente la tarjeta de video. Ahora bien, la información para programarla se encuentra muy

dispersa y es difícil de conseguir. Por lo que a continuación se da una introducción al tema.

Si contamos con una tarjeta VGA tenemos ya por default 256 colores y una resolución de 320X200 (modo 13h), lo cual da un total de 64K de memoria convencional, Por lo que, un primer ejercicio de programación de la tarjeta de video consiste en cambiar nuestra tarjeta al modo 13h (dependiendo del lenguaje los números hexadecimales se indican con un \$ en Pascal, 0x en C o una h en ensamblador)

1 CAMBIO DEL MODO DE VÍDEO

Para cambiar la computadora de un modo de video a otro usando interrupciones se puede usar la siguiente rutina que llama a la *interrupción 10h* que es la interrupción del video del BIOS:

```
void modogra(int modo) //inicia el modo de video, para tarjetas hasta VGA
{
    union REGS reg;
    reg.h.ah=0;
    reg.h.al=modo;
    int86(0x10,&reg,&reg);
}
```

y si dentro del programa se llama a la instrucción:

modogra(0x13)

la maquina se pone en modo 13h

Antes de terminar el programa es conveniente regresar a la computadora a un modo normal, para lo cual se puede usar por ejemplo las instrucciones:

**modogra(0x12); //regresa a modo de video 12h 640*480*16
colores**

ó

textmode(C80); //pasa a modo texto

2 MEMORIA DE VÍDEO

Una de las características mas interesantes de la programación en modo 13h es que asociado a cada pixel de la pantalla se tiene una localidad de memoria, o sea que si se coloca un valor en esa localidad de memoria automáticamente aparece un color en una posición de la pantalla con lo que se pueden sacar imágenes en la pantalla programando directamente la memoria.

El área de la computadora donde se almacenan las imágenes que salen por el video se conoce como *memoria de video y empieza en la localidad A000:0000h*.

Si estamos en *modo 13h* y mandamos escribir un valor en esa localidad podremos ver un punto en la parte superior del monitor. La ventaja de mandar a escribir directamente a memoria, es que agiliza el manejo de información. (*en otros modos no es tan directa la programación de la memoria de video por lo que se recomienda dejar ese problema para el momento en que se tenga mayor manejo del tema*)

Para accesar a la memoria de video en C se debe declarar un apuntador mediante una instrucción como la siguiente:

```
char far *apun= (char far *) 0xA0000000;
```

Para mandar escribir un punto a una localidad se usa una instrucción como

```
*(apun+j*320+i)=c;
```

donde **apun** es la base o inicio de la memoria de video, **i** nos indica la columna, **j** el renglón donde se quiere poner el punto y **c** es el índice del color que se quiere poner.

Con lo anterior ya se tiene una base mínima para manejar directamente el video, sin embargo para la gran mayoría de actividades relacionadas con los dispositivo interactivos, se requiere de la programación directa de puertos, por lo que a continuación se muestra un pequeño ejemplo de esta actividad.

3 PALETA DE COLORES

Un ejercicio sencillo para introducirnos al manejo de puertos y la programación de tarjetas, consiste en cambiar directamente los valores de la paleta de colores, para lograr que tome valores entre 256,000 posibles colores, programando directamente los puertos que controlan al DAC (Digital Analogic Converter) de la tarjeta de video.

La paleta de colores no es mas que una tabla de 256 registros con 3 componentes para cada registro. Los componentes son R, G, B (rojo, verde, azul).

La tarjeta de video cuenta con un DAC el cual permite el acceso a la paleta de colores mediante los puertos 3C7h, 3C8h, 3C9h.

Un *puerto* es un dispositivo Físico/Lógico que permite la conexión entre dos aparatos, programando directamente los puertos obtenemos la respuesta en un menor tiempo. Al aumentar la velocidad de respuesta se pueden realizar por ejemplo animaciones en tiempo real, y una interactividad mejor.

El manejo de puertos es muy simple pero puede ser peligroso y en su momento se dificulta su uso porque, para cada aparato que se comunica mediante un puerto se requiere conocer el numero de puerto y los valores de las señales que permiten que funcione, y gran parte de esta información es poco accesible

Cuando se programan puertos se tiene que tomar en cuenta que muchas veces se esta programando directamente un pequeño procesador y sino se sabe manejar se corre el peligro de dañar un dispositivo, por lo que, es conveniente no realizar experimentos ni jugar con los puertos ya que son delicados, por lo que *en los programas que siguen se recomienda tener cuidado y no ponerse a "hacer experimentos"* y si se quiere continuar profundizando sobre el tema, verificar siempre las instrucciones que se quieren usar.

Las funciones de los puertos del DAC son las siguientes:

3C7h En este puerto se debe almacenar el *índice del color* del que se quiere leer sus *componentes* R, G, B

3C8h En este puerto de debe almacenar el *índice del color* que se va a *modificar* en sus *componentes* R, G, B

3C9h *Componentes del color* (R, G, B)

Cada vez que se hace la llamada al puerto 3C9h, automáticamente apunta al siguiente registro de color

Con un par de funciones podemos modificar los colores o crear nuevas tonalidades que antes no podíamos realizar.

//Rutina para leer un color específico

```
void R_C(int C, int R, int G, int B)
{
    outportb(0x3c7,C); //Escribimos el índice del color
    R= inportb(0x3c9); //Leemos componente Roja
    G= inportb(0x3c9); //Leemos componente Verde
    B= inportb(0x3c9); //Leemos componente Azul
}
```

//Rutina para modificar un color con componente R, G, B.

```
void W_C(int C, int R, int G, int B)
{
    outportb(0x3c8,C); //Escribimos el índice del color
    outportb(0x3c9,R); //Escribimos componente Roja
    outportb(0x3c9,G); //Escribimos componente Verde
    outportb(0x3c9,B); //Escribimos componente Azul
}
```

Con estas funciones podemos tener un total de $256 \times K$ combinaciones de colores en una tarjeta de video VGA.

III MANEJO DE INFORMACIÓN MASIVA

Otro de los problemas clásicos dentro de la Interactividad se presenta en el manejo masivo de información, en particular cuando se manejan imágenes o sonido se esta tratando con cantidades masivas de información, ya que por ejemplo, una sola imagen contiene alrededor de 256 mil caracteres y un segundo de animación corresponde a 30 imágenes o sea 7,500,000 caracteres, por lo que se muestran ejemplos de manejo de sonido e imagen en forma masiva dentro de una computadora, incluyendo el manejo de formatos de video y sonido, con el fin de introducir al tema y

muestra que los diferentes problemas de interactividad siguen siendo similares.

Estos dispositivos de captura cuentan con formatos propios o estandares para manejar información, entre los mas usuales para imágenes se tienen GIF, TIFF, JPG, PCX, BMP etc., y para sonido VOC, MOD, MIDI, WAV, etc. Algunos de estos formatos cuentan con cierto tipo de compactación, por lo cual es difícil manejarlos si no contamos con la información del formato. Pero otros son mas simples como es el caso del formato BMP para imagen y WAV para sonido. Por lo que a continuación se mostrara como se maneja información formateada.

1 FORMATO BMP

En la actualidad existen en el mercado diversos tipos de *formatos para almacenar imágenes*, su diversidad es muy grande. Algunos de los formatos mas populares son GIF, TIFF, JPG, BMP, etc. Todos estos formatos son comerciales y no es fácil encontrar información interna sobre ellos.

Uno de los formatos mas sencillos es el BMP (Bit MaP, Mapa de Bits). Un archivo con formato BMP consta de dos partes: cabecera y cuerpo. La cabecera esta compuesta de: la identificaron del formato del archivo, dirección de inicio del cuerpo, tamaño horizontal de la imagen, tamaño vertical de la imagen, el numero de bits por pixel y la paleta de colores.

Con la información de la cabecera obtenemos el tamaño de imagen, el tipo de resolución del archivo (si es de 2, 16 o 256 colores), y la paleta de colores propia de esa imagen.

En el cuerpo de la imagen se encuentra el valor del color de cada pixel, que se acomoda en la posición correspondiente de acuerdo al tamaño de horizontal y vertical, podemos pensar en algún momento que la imagen esta guardada de arriba hacia abajo lo cual es falso. El primer pixel del cuerpo corresponde a la parte inferior izquierda de la imagen, los demás pixeles se irán acomodando de izquierda a derecha y de abajo hacia arriba.

La ubicación de la información dentro del archivo es la siguiente:

BYTE	SIGNIFICADO	nombre de la variable
0	Identificación del archivo.	
10	Lugar donde empieza el cuerpo	CUERPO
18	Tamaño horizontal	TAM_X
22	Tamaño vertical.	TAM_Y
28	Bits por pixel.	BIT_PIXEL
30	Principio de la paleta de colores .	PALETA
XX	El principio del cuerpo se calcula por medio de los bits por pixel y la paleta de colores.	

La paleta de colores es una tabla donde se tienen los colores de la imagen, cada color esta formado de 4 componentes R, G, B, A donde R, G, B, indican la cantidad de rojo, verde y azul de cada color, A es una mascara que nos indica como se combina ese color con los demás, hay que tener cuidado ya que la paleta de colores esta escrita en el orden de B, G, R, A

A continuación se muestra un programa que despliega archivos con formato BMP en la pantalla de la PC.

/* Programa que despliega una imagen con formato BMP */

/* Realizado por: J. Antonio Jiménez Aviña */

/* 29-04-96 */

/*modificado por Fernando Galindo Soria

rutina propia para cambiar paleta

rutina propia para traer el color del siguiente pixel

8/vi/96 Cd. de México.*/

#include <conio.h> //Cabeceras

#include <stdio.h>

#include <graphics.h>

FILE *BMP;

//Declaración de variables

char identificación[118];

long int CUERPO, TAM_X, TAM_Y, BIT_PIXEL,
PIXEL_byte, PALETA, PALETA_TAM;

int mask;

int ShowBMP(char* fname,int,int);

void leecabecera(void);

void leepaleta(void);

void esrcolordac16(char color,char r,char g,char b);

void leecuerpo(int,int);

void main(void)

{

int gdriver=DETECT,gmode;

initgraph(&gdriver,&gmode,"");

ShowBMP("ARCHIVO.BMP",0,0); **//poner el nombre del archivo donde está la imagen a desplegar**

```
getch();
closegraph();
}
```

int ShowBMP(char* fname,int x0,int y0)

//Función que abre el archivo y despliega la imagen en el monitor

```
BMP= fopen(fname,"rb");//abre el archivo de imagen
if (BMP==NULL) return -1;
```

```
leecabecera();
leepaleta();
leecuerpo(x0,y0);
fclose(BMP);
return 0;
}
```

void leecabecera(void)

```
{
//lee la cabecera
fseek(BMP,0,SEEK_SET);
fread(&identificacion,sizeof(identificación),1,BMP);//lee que sea archivo tipo BMP
fseek(BMP,10,SEEK_SET);
fread(&CUERPO,sizeof(CUERPO),1,BMP); //Lugar donde empieza el cuerpo.
fseek(BMP,18,SEEK_SET);
fread(&TAM_X,sizeof(TAM_X),1,BMP); //Tamaño horizontal.
fseek(BMP,22,SEEK_SET);
fread(&TAM_Y,sizeof(TAM_Y),1,BMP); //Tamaño vertical.
fseek(BMP,28,SEEK_SET);
fread(&BIT_PIXEL,sizeof(BIT_PIXEL),1,BMP);//Bits por pixel.
```

```
PIXEL_byte=8/BIT_PIXEL;    //numero de pixels por byte
mask=(1<<BIT_PIXEL)-1;
}
```

void leepaleta(void)

```
{
//lee del archivo la paleta de colores y la carga en la tarjeta de video
typedef struct //Estructura para el color
{
    unsigned char b, g, r, a;
} Color;
Color col;
int k;
PALETA_TAM=1<<BIT_PIXEL;
PALETA=CUERPO-4*PALETA_TAM;    //Principio de la paleta de colores.

fseek(BMP,PALETA,SEEK_SET);
for (k=0;k<PALETA_TAM;k++) //Cargando la paleta de colores
{
    fread(&col,sizeof(col),1,BMP);
    esccolordac16(k,(col.r)>>2,(col.g)>>2,(col.b)>>2);
}
```

```
}
```

```
void esccolordac16(char color,char r,char g,char b)
```

```
{ /*cambia de color paleta llamando directamente a los puertos  
del convertidor analógico digital de la tarjeta de video
```

```
utiliza initgraph y maneja paleta rgb de 16 colores b(lue) g(green) r(ed)*/
```

```
char indcol[]={0,1,2,3,4,5,20,7,56,57,58,59,60,61,62,63};
```

```
outportb(0x3C8,indcol[color]);/*inicializa el color*/
```

```
outportb(0x3C9,r);
```

```
outportb(0x3C9,g);
```

```
outportb(0x3C9,b);
```

```
}
```

```
void leecuerpo(int x0,int y0)
```

```
{ //Función que regresa el color de los pixels
```

```
// de acuerdo a sus componentes RGB y lo pone en la pantalla
```

```
int xi=0, n=0, k,Xpal,Ypal;
```

```
long int col[4];
```

```
unsigned char b,b1[4];
```

```
fseek(BMP,CUERPO,SEEK_SET);
```

```
for (Ypal=TAM_Y-1; Ypal>=0; Ypal--) //Despliega la imagen en el monitor
```

```
{xi=0; n=0;
```

```
for (Xpal=0; Xpal<TAM_X; Xpal++)
```

```
{
```

```
if (xi%PIXEL_byte==0) //desempaca los pixels empacados en un byte
```

```
{
```

```
fread(&b,sizeof(b),1,BMP); //trae un byte del archivo
```

```
k=PIXEL_byte-1;
```

```
col[k]= b & mask; //desempaca cada pixel y lo almacena en un arreglo
```

```
while (k--) col[k]=(b>>=BIT_PIXEL) & mask; //regresa el índice del color del siguiente pixel
```

```
n++; //cuenta cuantos bytes ha desempacado
```

```
}
```

```
putpixel(x0+Xpal,y0+Ypal,col[xi++%PIXEL_byte]); //despliega el siguiente pixel
```

```
}
```

```
fread(&b1,1,(4-n%4)%4,BMP); //ignora bytes para quedar en múltiplos de 4
```

```
}
```

```
}
```

2 MANEJO DE SONIDO

Como se puede ver en un problema tan simple como el desplegar una imagen en pantalla se requieren tener una serie de conocimientos que normalmente pasan transparentes a los usuarios normales de la computación, pero que sin embargo, son cotidianos en el manejo de sistemas interactivos, por otro lado el manejo masivo de información requiere un tratamiento similar en el caso de archivos de voz, ya que también se requiere obtener la información del encabezado y usarla para desempacar los valores de los sonidos, por lo que, el desplegar imágenes es un buen ejercicio para manejo de información masiva, que se puede aplicar directamente al manejo de otros tipos de datos como es el caso del sonido, por lo que a continuación se comentara sin entrar a detalle y con el fin de que se vea la semejanza con el manejo de imágenes unos ejemplos de manejo masivo de sonido.

Dentro de los *archivos de sonido*, unos de los mas importantes son los que tienen formatos tipo *WAV* y *VOC*. En estos formatos podemos grabar tanto música como voz.

El formato de Microsoft Waveform (WAV), es uno de los mas populares actualmente gracias a Windows y sus múltiples aplicaciones en Multimedia, ya que su calidad de sonido es lo bastante buena al ser grabado en 8 o 16 BYTES. Lo malo es que ocupa demasiado espacio para unos cuantos segundos de sonido.

El formato WAV esta organizado de acuerdo a la estructura RIFF (*formato de archivo de intercambio de recurso*). Esta estructura fue diseñada hace varios años para los *archivos de recursos multimedia*, Un archivo de Formato WAV esta compuesto de una cabecera y un cuerpo.

Los elementos de la *Cabecera* son:

- 1) *Identificación del formato RIFF* (4 BYTES). Regresa la cadena "RIFF"
- 2) *Longitud de datos del formato RIFF* (4 BYTES)
- 3) *Identificación del formato como formato de sonido (WAV)* (4 BYTES). Regresa la cadena "WAVE"

- 4) *Datos del formato RIFF* (puede o no existir) La longitud se da en longitud de datos del formato RIFF
- 5) *Identifica al bloque como un fragmento de formatos* (4 BYTES). Regresa la cadena "fmt"
- 6) *Longitud de los datos del formato que viene a continuación* (4 BYTES)
- 7) *Categoría al que pertenece el formato WAV* (2 BYTES). No esta documentada pero si regresa un 01 indica que es un formato *Pulse Code Modulation* (PCM)
- 8) *Canales de salida.* (2 BYTES). Indica si es un archivo Mono o Estéreo (1 para Mono y 2 para Estéreo)
- 9) *Frecuencia de muestreo*, muestras por segundo a la que debe de reproducir cada canal (2 BYTES). Este valor esta dado en Hertz
- 10) *Número medio de BYTES que deben de transmitirse por segundo* (2 BYTES). También se puede calcular por la siguiente formula: $\text{Número medio de BYTES por segundo} = \text{Canales} * \text{Número de muestras por segundo} * (\text{Número de Bits por muestra} / 8)$, donde el Numero de Bits por Muestra es igual al Numero de Muestras por segundo entre ocho.
- 11) *Alineamiento del bloque* (2 BYTES). Ya que la computadora puede procesar solo un bloque a la vez. Este dato se utiliza para alinear los bufers de cada canal
- 12) *Especifico del formato* (2 BYTES). No se encuentra documentado
- 13) *Data* Identifica al siguiente bloque como fragmento de datos (4 BYTES). Regresa la cadena "fmt"
- 14) *Longitud del bloque* (4 BYTES).Indica la longitud del cuerpo del archivo de sonido

Los elementos del *Cuerpo* son:

- 1) *Bloque de datos.* Es donde se encuentra grabada la información a reproducir.

Como se observa cada uno de los datos tiene una longitud específica la cual esta dada por el fabricante o bien dentro del formato.

Ahora bien, existen muchas otras formas para almacenar y manejar datos masivos, por lo que, finalmente comentaremos otra de las formas de almacenamiento masivo de sonidos conocida como formato VOC. Este tipo de formato es solo para *archivos que hayan sido grabados por medio de un código de modulación de pulsos (PCM) de 8 bits*.

El Formato VOC consta en su cabecera de un bloque de encabezamiento y un bloque de datos, dentro del bloque de datos puede haber a su vez diferentes tipos de bloques, los cuales proporcionan una manera elegante de programar pausas, repeticiones etc. Pero comúnmente solo encontramos un tipo de bloque, el correspondiente a los datos de voz.

Como se puede observar el problema principal que se tiene para manejo de mucha de la información interactiva se reduce al conocimiento o desconocimiento de los formatos con que se almacena o transmite.

CONCLUSIÓN

La interactividad es una de los campos mas importantes a nivel mundial e incluye áreas aparentemente tan disimolas como la graficación, robotica, realidad virtual, proceso distribuido y muchas otras que si se analizan independientemente del problema específico tienen practicante la misma arquitectura y los mismos componente. Por lo que, cada vez es mas importante que nuestros alumnos se introduzcan en esta área.

Sin embargo, prácticamente no se desarrollan actividades académicas sobre este tema, porque se cree que se requiere contar con grandes infraestructuras para desarrollarlas. Es por lo anterior, que el objetivo de este trabajo es mostrar que nos podemos introducir en la interactividad sin necesidad de contar con grandes recursos, ya que prácticamente todos los problemas del desarrollo

de un sistema interactivo están presentes en el manejo de una tarjeta de video.

Esta idea ya se ha comenzado a aplicar involucrando a estudiantes desde secundaria hasta posgrado, siendo muy gratificante observar como estos estudiantes cada vez han profundizando mas sobre este tema y lo fácil que ha sido aplicar la experiencia obtenida en la programación de las tarjetas de video a la programación de otros tipos de dispositivos interactivos.

Con lo cual, también se ha logrado romper un bloqueo tradicional ya que mucha gente no se ha dado cuenta que si son capaces de manejar un dispositivo pueden manejar con relativa facilidad muchos otros y que es necesario romper con los feudos de conocimiento, ya que *los problemas que se atacan son similares y corresponden a una gran área conocida como interactividad.*

BIBLIOGRAFÍA

[1] Peter M. Ridge, "Guía oficial de Sound Blaster", Osborne McGraw-Hill España 1994.

[2] Boris Bertelson & Mathias Rasch, "PC al limite", Data Beck Colombia 1995.

[3] "PC Interno", Data Beck.