

Introducción a la Graficación por Computadora

Dr. Roberto Núñez González¹

¹Ciencias de la Computación
Departamento de Matemáticas
Universidad de Sonora

2020



"El saber de mis hijos
hará mi grandeza"

Introducción a la Graficación por Computadora

- 1 Definición
- 2 Aplicación de la Graficación
- 3 Elementos de un Sistema Gráfico
- 4 Conceptos Varios
- 5 Dispositivos de Despliegue
- 6 Programas con Librerías Gráficas



"El saber de mis hijos
hará mi grandeza"

Definición

La **Graficación por Computadora** es uno de los muchos campos de las Ciencias de la Computación, donde se utilizan computadoras tanto para generar imágenes visuales sintéticamente como integrar o cambiar la información visual y espacial recopilada del mundo real. (Wikipedia)

No se le puede asignar un dominio específico, ya que es **multi-disciplinaria**.



"El saber de mis hijos
hará mi grandeza"

Desarrollo

El desarrollo de la Graficación se debe a dos factores principalmente:

- **Necesidad** de los usuarios
- **Avances** en Hardware y Software



"El saber de mis hijos
hará mi grandeza"

Aplicación de la Graficación

La graficación tiene aplicación en varias áreas del conocimiento:

- Interfaces de usuario
- Graficado de información en negocios, ciencia y tecnología
- Despliegue de información
- Publicación electrónica
- Diseño Asistido por Computadora (CAD)
- Simulación y animación para visualización científica.
- Entretenimiento
- Arte
- Control de procesos
- Cartografía



"El saber de mis hijos
hará mi grandeza"

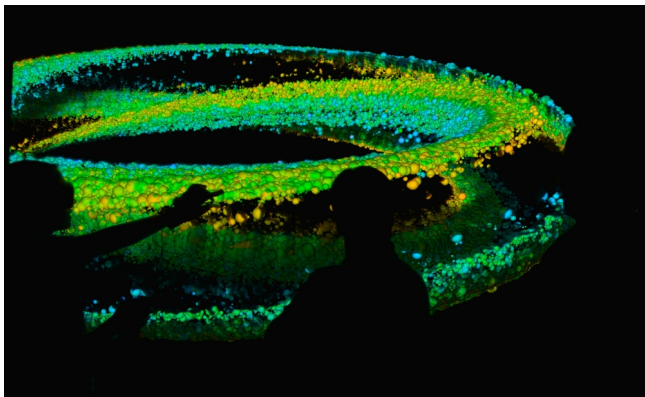


Figura: Visualización de partícula atómicas en 3D, confinadas magnéticamente en un reactor de fusión simulado.

<http://www.sci.utah.edu/>

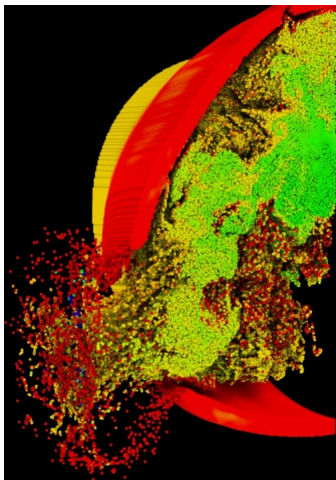


Figura: Simulación de la explosión de un contenedor.
<http://www.sci.utah.edu/>



Figura: Volume rendered temperature data from the transient simulation of the 10-m heptane pool fire. <http://www.sci.utah.edu/>

Elementos de un Sistema Gráfico

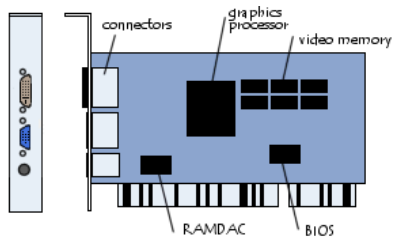
Los elementos de un sistema gráfico son los siguientes:

- Procesador (CPU)
- Procesador Gráfico (GPU)
- Memoria
- Frame-Buffer (**VRAM, DRAM**)
- Dispositivo Entrada
- Dispositivo Salida (**Raster display, Vector display**)

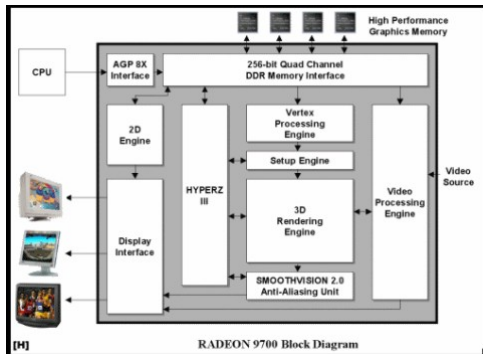


"El saber de mis hijos
hará mi grandeza"

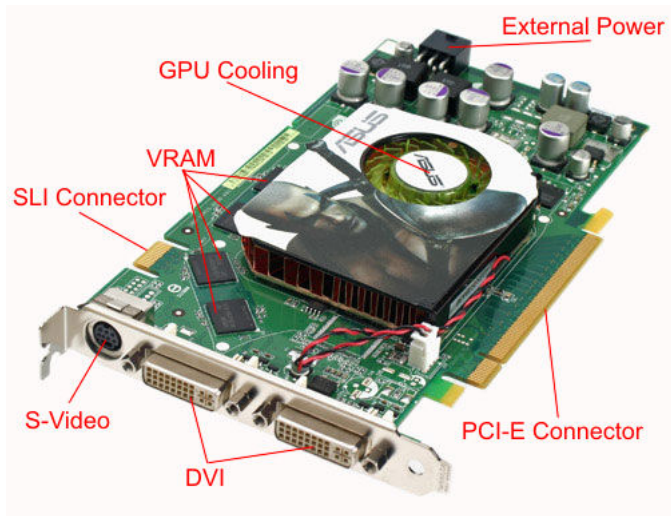
Tarjeta Gráfica o de Video



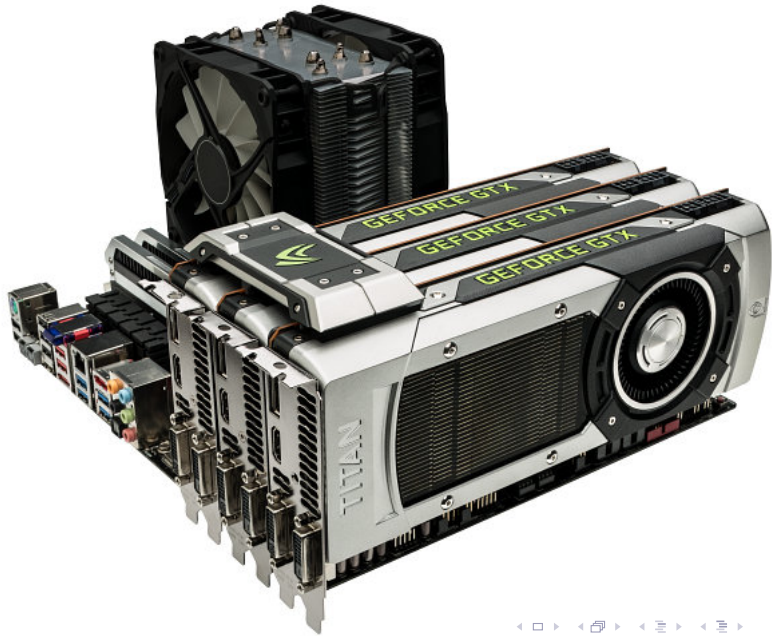
Tarjeta Gráfica o de Video



Tarjeta Gráfica o de Video



Tarjeta Gráfica o de Video



Conceptos Varios I

El **frame buffer** es la memoria de video que es utilizada para almacenar la imagen desplegada en la pantalla de la computadora.

Un **pixel** puede tener diferentes definiciones, pero para nuestro estudio podemos considerarlo como el elemento más pequeño de la pantalla de la computadora que puede ser iluminado con un cierto color e intensidad.



"El saber de mis hijos
hará mi grandeza"

Conceptos Varios II

Una **gráfica raster** es una gráfica que se almacena en la memoria de la computadora usando una malla para grabar la información (como color e intensidad) de cada pixel que forma la imagen.

La idea detras de las gráficas tipo raster es que una imagen es dividida en un mosaico regular de pequeñas celdas, llamadas pixeles, cada uno de los cuales es de un color e intensidad dados.



"El saber de mis hijos
hará mi grandeza"

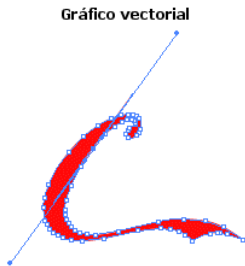
Conceptos Varios III



"El saber de mis hijos
hará mi grandeza"

Conceptos Varios IV

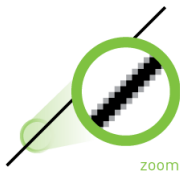
Una **gráfica vectorial** es una gráfica de computadora que es almacenada en la memoria de la computadora usando fórmulas matemáticas.



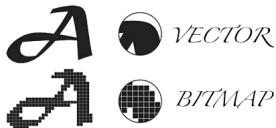
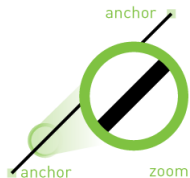
"El saber de mis hijos
hará mi grandeza"

Conceptos Varios V

Raster/bitmap



Vector



"El saber de mis hijos
hará mi grandeza"

Conceptos Varios VI

bit-mapped graphics — raster graphics

voxel — pixel en 3D



"El saber de mis hijos
hará mi grandeza"

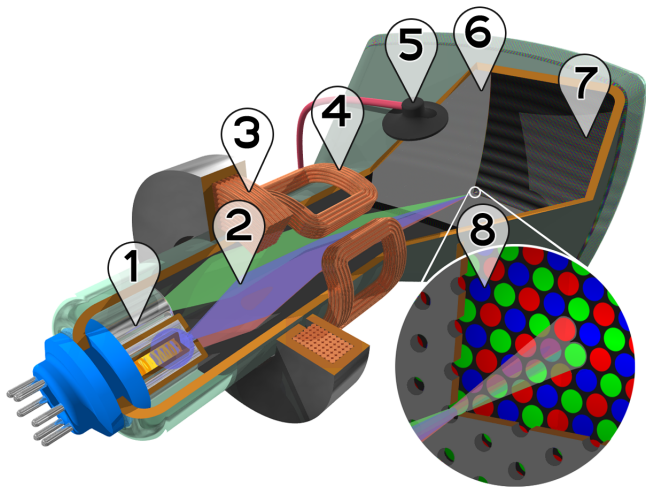
Dispositivos de Despliegue

Los **dispositivos de despliegue tipo raster** almacenan las primitivas de despliegue (lineas, caracteres, áreas sólidas, etc) en un memoria especial llamada *buffer de refresco*. Estas primitivas que forman una imagen que va a ser desplegada, se almacenan como una matriz de pixeles (columnas y renglones de pixeles).

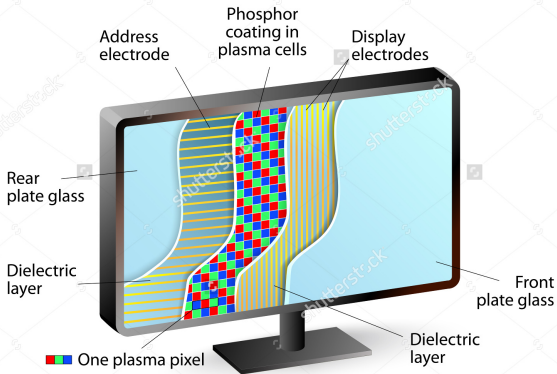


"El saber de mis hijos
hará mi grandeza"

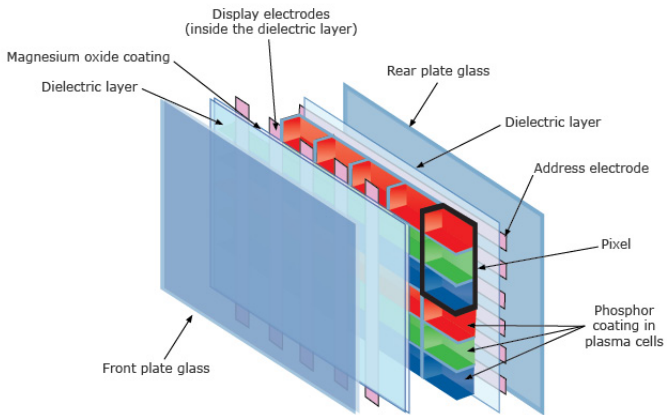
Monitor CRT



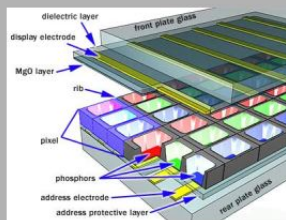
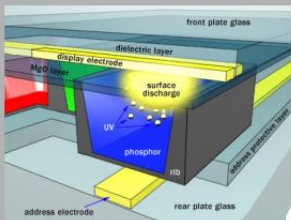
PLASMA DISPLAY



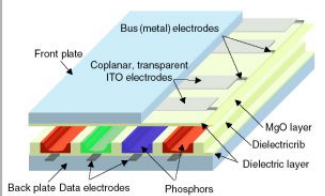
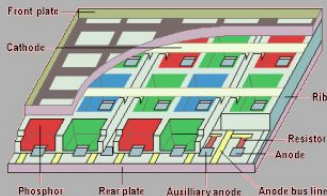
Monitor PLASMA



Schematic Views of Plasma Display Panels



Source: <http://electronics.howstuffworks.com/plasma-display3.htm>



Source: <http://www.plasmatechscience.org/theinnerworkings2.html>

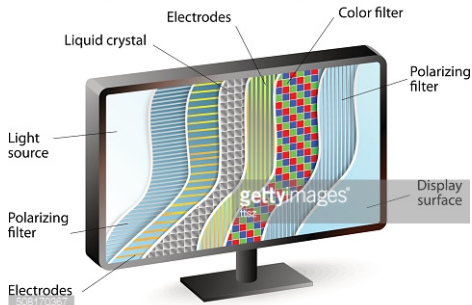
Source: J.P. Boeuf; J. Phys. D, 36 (2003) R53



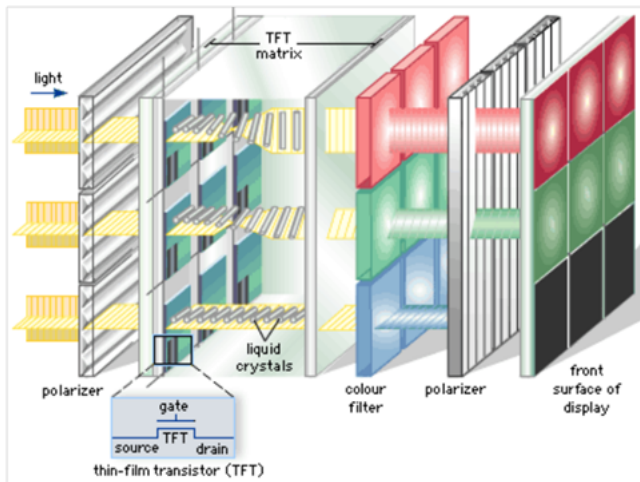
Copyright © 2005 CLARYCON, all rights reserved

Monitor LCD (Liquid Crystal Display)

LIQUID CRYSTAL DISPLAY



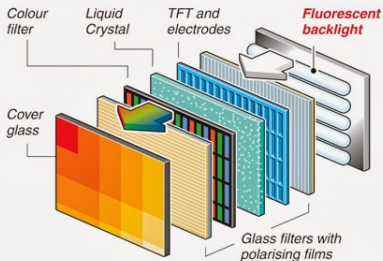
Monitor LCD (Liquid Crystal Display)



Monitor LED (Light Emitting Diode)

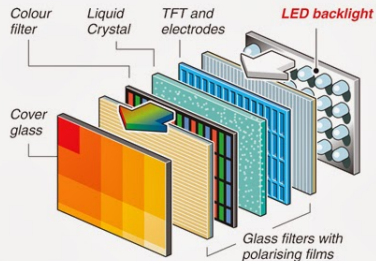
LCD - Liquid Crystal Display


Today's dominant flat display technology produces images by blocking or allowing light to pass from the light source behind the LCD display



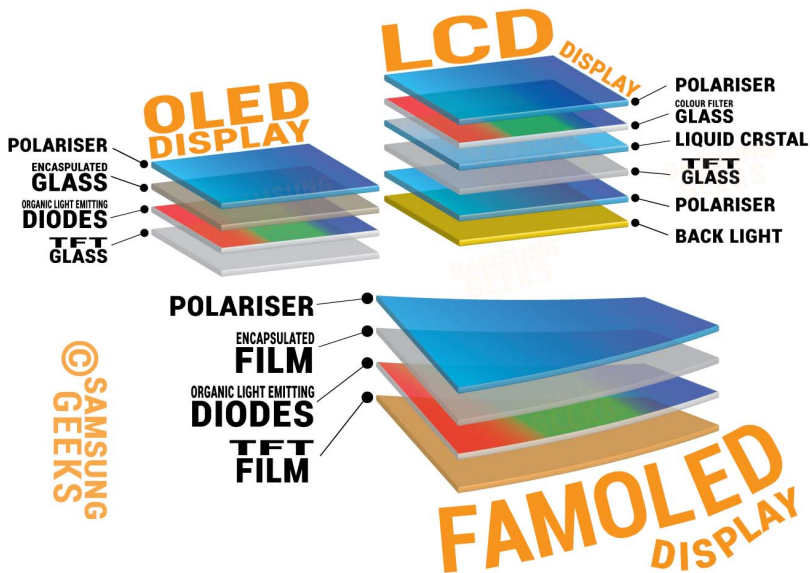
LED - Light Emitting Diode

LEDs are LCD TVs that replace the cold cathode fluorescent lamps (CCFL) used in conventional LCD displays

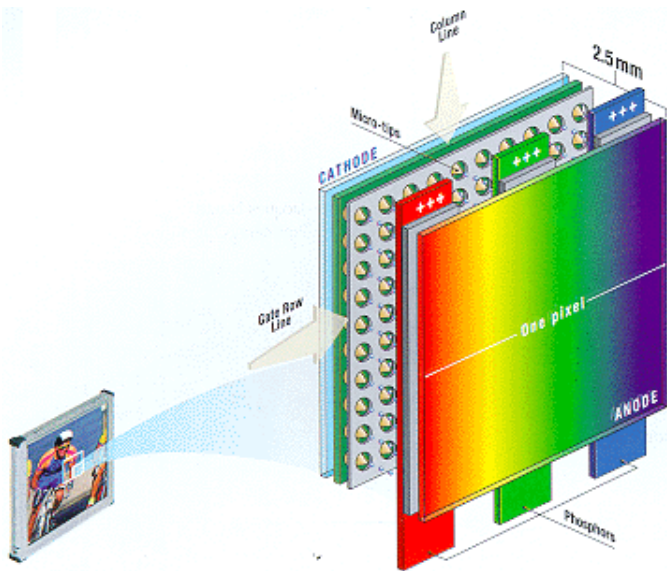


Sources: Reuters, HowStuffWorks.com, Discovery Channel, CNET, OSRAM  REUTERS

Monitor OLED (Organic Light Emitting Diode)



Monitor FED



Programa para graficar con Turbo C o GCC/WinBGI

```
#include<stdio.h>
#include<stdlib.h>
#include<graphics.h>

int main(void)
{
    int gdriver=DETECT,gmode,errorcode;
    initgraph(&gdriver,&gmode," ");
    errorcode=graphresult();

    if(errorcode != grOK)
    {
        printf("ERROR GRAFICO: %s\n",grapherrormsg(errorcode));
        printf("Presionar alguna tecla para salir: ");
        getch();
        exit(1);
    }

    line(0,0,getmaxx(),getmaxy());
    getch();
    closegraph();
    return(0);
}
```



"El saber de mis hijos
hará mi grandeza"

Programa para graficar con libreria SDL_bgi (Linux) I

```
/* Programa linea_dda.c  -*- C -*-
 *
 * Roberto Nuñez Gonzalez
 * Ciencias de la Computacion
 * Universidad de Sonora
 * 2016
 *
 * Basado en el programa simple.c, distribuido con la libreria
 * SDL_bgi, y elaborado por Guido Gonzato (May 2015)
 *
 * To compile:
 * gcc -o simple simple.c -lSDL_bgi -lSDL2
 */

#include <stdio.h>
#include <stdlib.h>
// #include <graphics.h> /* ESTA LINEA PARA WINDOWS */
#include <SDL2/SDL_bgi.h>

// *****
void message (char *str, int maxx, int maxy)
{
    cleardevice ();
    settxtstyle (DEFAULT_FONT, HORIZ_DIR, 2);
    settxtjustify (CENTER_TEXT, CENTER_TEXT);
    setcolor (RED);
    outtextxy (maxx / 2, maxy / 2, str);
    settxtstyle (DEFAULT_FONT, HORIZ_DIR, 1);
    setcolor (YELLOW);
    outtextxy (maxx / 2, maxy - 20, "Press a key to continue");
    getch ();
}
```


Programa para graficar con libreria SDL_bgi (Linux) II

```
cleardevice ();
settextjustify (LEFT_TEXT, TOP_TEXT);

}

//*****
int main (void)
{
int maxx,maxy;
int gd, gm;
int x0,y0,xf,yf;
int x;
double y,m;

gd = SDL;
gm = SDL_800x600;
initgraph (&gd, &gm, "");
//gd = DETECT;
//initgraph (&gd, &gm, "");
maxx = getmaxx ();
maxy = getmaxy ();
setbkcolor (BLACK);
cleardevice ();

//message ("Drawing in SLOW mode:",maxx,maxy);
//sdlbgislow ();

//message ("Drawing in FAST mode:",maxx,maxy);
//sdlbgifast ();

x0=100;
```

Programa para graficar con libreria SDL_bgi (Linux) III

```
y0=100;
xf=500;
yf=500;

m=1.0*(yf-y0)/(xf-x0);

setcolor(WHITE);
x=x0;
y=y0;

while(x<=xf){
  putpixel(x,y,WHITE);
  x=x+1;
  y=y+m;
}
getch ();
closegraph ();
}
```

Programa para graficar en OpenGL/Mesa3D(linux, windows) I

```
/* demo.c */
/* Para compilar gcc demo.c -o demo -lglut -lGL */

#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include <GL/gl.h>
#include <GL/glut.h>
#include <GL/glu.h>

void setup() {   glClearColor(1.0f, 1.0f, 1.0f, 1.0f); }

void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(0.5f, 0.0f, 0.0f);
    glRectf(-0.75f,0.75f, 0.75f, -0.75f);
    glutSwapBuffers();
}

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);
    glutInitWindowSize(800,600);
    glutCreateWindow("Hello World");
```

Programa para graficar en OpenGL/Mesa3D(linux, windows) II

```
    setup();  
    glutDisplayFunc(display);  
    glutMainLoop();  
    return 0;  
}
```

Programa para graficar en OpenGL/Mesa3D(linux, windows) I

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<GL/glut.h>

// Prototipos de funciones
void disp(void);
void keyb(unsigned char key, int x, int y);

// Identificador de la ventana
static int win;

int main(int argc, char **argv)
{
    ////////////
    // INIT //
    ////////////

    // Inicia GLUT
    glutInit(&argc, argv);

    // Se especifica el modo de despliegue a tipo RGB y
    // con buffer sencillo (no animacion)
    glutInitDisplayMode(GLUT_RGBA | GLUT_SINGLE);

    // Se define el tamaño de la ventana
    glutInitWindowSize(500,500);
```

Programa para graficar en OpenGL/Mesa3D(linux, windows) II

```
// Se define la posicion donde la ventana se dibujara
glutInitWindowPosition(100,100);

// Si queremos despliegue de pantalla completa:
// glutFullScreen();

// Se crea la ventana, se asigna el nombre y se guarda
// el identificador de la ventana
win = glutCreateWindow("Primera ventana");

//////////
// CALLBACK //
//////////

glutDisplayFunc(displ);
glutKeyboardFunc(keyb);

//////////
// OPENGL //
//////////

// SE DEFINE EL COLOR QUE SE UTILIZA PARA LIMPIAR LA PANTALLA
glClearColor(0.0,0.0,0.0,0.0);

// ENTRA EN EL CICLO PRINCIPAL
glutMainLoop();

return 0;
}
```

Programa para graficar en OpenGL/Mesa3D(linux, windows) III

```
////////////////////////////////////  
////////////////////////////////////  
void disp(void)  
{  
    // LIMPIA LA PANTALLA  
    glClear(GL_COLOR_BUFFER_BIT);  
  
    // DIBUJAR ALGO  
    glutWireTeapot(0.5);  
  
}  
////////////////////////////////////  
////////////////////////////////////  
void keyb(unsigned char key, int x, int y)  
{  
    printf("Se presiono %c en las coordenadas (%d,%d)\n",key,x,y);  
    //cout << "Se presiono " << key << " en las coordenadas(" << x << ", " << y << " );  
    //cout << endl;  
    if(key == 'q'){  
        printf("Salir\n");  
        //cout << "Salir" << endl;  
        glutDestroyWindow(win);  
        exit(0);  
    }  
}
```