

# Practica-1-Resuelta.pdf



Zukii



Informática Gráfica



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación  
Universidad de Granada



**10 % dto Mac**      **5 % dto iPad**

## Superportátil. Superpotencia para tus estudios.

Precios especiales para estudiantes y profesores.

Con cualquier Mac o iPad  
Microsoft 365 incluido.



### Rossellimac

Tu especialista en Apple.

Descuento de 10% en MacBook Pro, iMac y Mac Studio y descuento de 5% en iPad para estudiantes y profesores mayores de 18 años. Oferta válida hasta el 24/9/2023. Microsoft 365 Personal (Valorado en 69€) incluido con la compra de cualquier Mac o iPad.

The ColaCao logo is written in a stylized, blue, cursive font with a white outline, set against a red and yellow background with dynamic, brushstroke-like patterns.

**PERDÓN, ¿TE LLAMAS APROBADO  
Y DE APELLIDO CONUNDIEZ?  
¡TÚ PUEDES!**

@Zukii on Wuolah

## Prácticas de IG Resueltas

(Mi profesor de prácticas es Antonio López)

En mi perfil (@Zukii) podrás encontrar todas las prácticas  
resueltas

### Práctica 1:

#### ÍNDICE:

- [objetos.cc](#)
- [objetos.h](#)
- [practica\\_objetos.cc](#)

## Objetos.cc

```
//*****  
// Práctica 1 usando objetos  
//*****  
  
#include "objetos_B.h"  
  
//*****  
// _puntos3D  
//*****  
  
_puntos3D::_puntos3D()  
{  
}  
  
//*****  
// dibujar puntos  
//*****  
  
void _puntos3D::draw_puntos(float r, float g, float b, int grosor)  
{  
    int i;  
    glPointSize(grosor);  
    glColor3f(r,g,b);  
    glBegin(GL_POINTS);  
    for (i=0;i<vertices.size();i++){  
        glVertex3fv((GLfloat *) &vertices[i]);  
    }  
    glEnd();  
}  
  
//*****  
// _triangulos3D  
//*****  
  
_triangulos3D::_triangulos3D()  
{  
}
```

# Superportátil. Superpotencia para tus estudios.

Precios especiales para estudiantes y profesores.



**10%**  
dto **Mac**

**5%**  
dto **iPad**

Con cualquier Mac o iPad  
**Microsoft 365 incluido.**



Escanea el QR  
para saber más.

## Rossellimac

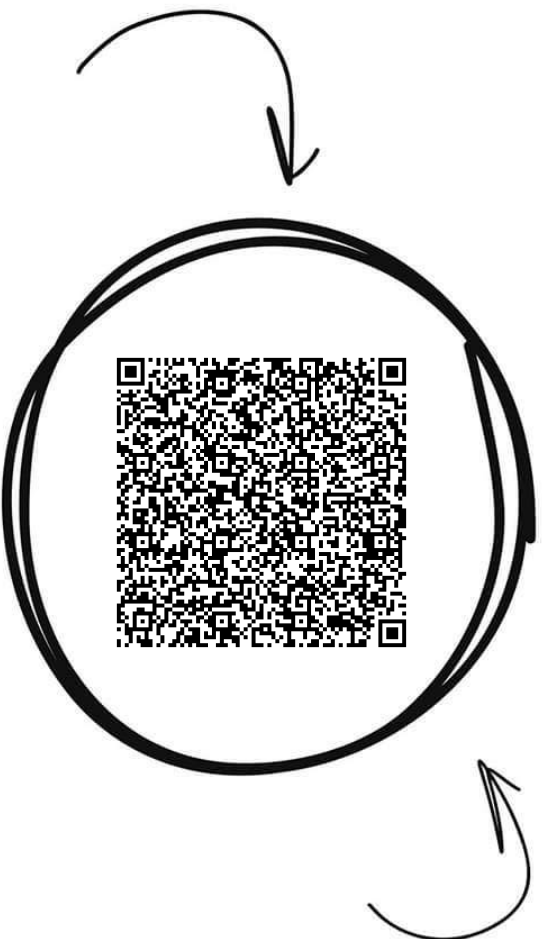
Tu especialista en Apple.

Descuento de 10% en MacBook Pro, iMac y Mac Studio y descuento de 5% en iPad para estudiantes y profesores mayores de 18 años. Oferta válida hasta el 24/9/2023. Microsoft 365 Personal (Valorado en 69€) incluido con la compra de cualquier Mac o iPad.

# Informática Gráfica



**Comparte estos flyers en tu clase y consigue más dinero y recompensas**



**Banco de apuntes de la**

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



```
//*****  
// dibujar en modo arista  
//*****  
  
void _triangulos3D::draw_aristas(float r, float g, float b, int grosor)  
{  
    int i;  
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);  
    glLineWidth(grosor);  
    glColor3f(r,g,b);  
    glBegin(GL_TRIANGLES);  
    for (i=0;i<caras.size();i++){  
        glVertex3fv((GLfloat *) &vertices[caras[i]._0]);  
        glVertex3fv((GLfloat *) &vertices[caras[i]._1]);  
        glVertex3fv((GLfloat *) &vertices[caras[i]._2]);  
    }  
    glEnd();  
}  
  
//*****  
// dibujar en modo sólido  
//*****  
  
void _triangulos3D::draw_solido(float r, float g, float b)  
{  
    int i;  
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);  
    glColor3f(r,g,b);  
    glBegin(GL_TRIANGLES);  
    for (i=0;i<caras.size();i++){  
        glVertex3fv((GLfloat *) &vertices[caras[i]._0]);  
        glVertex3fv((GLfloat *) &vertices[caras[i]._1]);  
        glVertex3fv((GLfloat *) &vertices[caras[i]._2]);  
    }  
    glEnd();  
}  
  
//*****
```

```
// dibujar en modo sólido con apariencia de ajedrez
//*****

void _triangulos3D::draw_solido_ajedrez(float r1, float g1, float b1, float r2, float g2, float b2)
//pag 28. si es par con un color, si no, otro color
{
    int i;
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glBegin(GL_TRIANGLES);
    for (i=0; i<caras.size(); i++)
    {
        if ( i %2 == 0 )
            glColor3f(r1, g1, b1);
        else
            glColor3f(r2, g2, b2);

        glVertex3fv((GLfloat *) &vertices[caras[i]._0]);
        glVertex3fv((GLfloat *) &vertices[caras[i]._1]);
        glVertex3fv((GLfloat *) &vertices[caras[i]._2]);
    }

    glEnd();
}

//*****
// clase cubo
//*****

_cubo::_cubo(float tam)
{
    //8 puntos - 12 triangulos
    //vertices
    vertices.resize(8);
    vertices[0].x=0;vertices[0].y=0;vertices[0].z=0; //0 -> (0,0,0)
    vertices[1].x=tam;vertices[1].y=0;vertices[1].z=0; //1 -> (tam,0,0)
    vertices[2].x=0;vertices[2].y=tam;vertices[2].z=0; //2 -> (0,tam,0)
    vertices[3].x=0;vertices[3].y=0;vertices[3].z=tam; //3 -> (0,0,tam)
    vertices[4].x=tam;vertices[4].y=tam;vertices[4].z=0; //4 -> (tam,tam,0)
    vertices[5].x=tam;vertices[5].y=0;vertices[5].z=tam; //5 -> (tam,0, tam)
    vertices[6].x=0;vertices[6].y=tam;vertices[6].z=tam; //6 -> (0,tam,tam)
```



# Superportátil. Superpotencia para tus estudios.

Precios especiales para estudiantes y profesores.



@Zukii on Wuolah

```
vertices[7].x=tam;vertices[7].y=tam;vertices[7].z=tam; //7 -> (tam,tam,tam)
```

```
// triangulos
```

```
caras.resize(12);
```

```
caras[0]._0=3;caras[0]._1=7;caras[0]._2=5; //Cara entre 3,7 y 5 //CARA FRONTAL
```

```
caras[1]._0=3;caras[1]._1=7;caras[1]._2=6; //Cara entre 3,7,6
```

```
caras[2]._0=3;caras[2]._1=0;caras[2]._2=6; //3,0,6 //CARA IZQUIERDA
```

```
caras[3]._0=0;caras[3]._1=2;caras[3]._2=6; //0,2,6
```

```
caras[4]._0=5;caras[4]._1=1;caras[4]._2=7; //5,1,7 //CARA DERECHA
```

```
caras[5]._0=4;caras[5]._1=7;caras[5]._2=1; //4,7,1
```

```
caras[6]._0=3;caras[6]._1=0;caras[6]._2=5; //3,0,5 PARTE DE ABAJO
```

```
caras[7]._0=0;caras[7]._1=1;caras[7]._2=5; //0,5,1
```

```
caras[8]._0=6;caras[8]._1=2;caras[8]._2=7; //6,2,7 PARTE DE ARRIBA
```

```
caras[9]._0=2;caras[9]._1=7;caras[9]._2=4; //2,7,4
```

```
caras[10]._0=0;caras[10]._1=1;caras[10]._2=4; //0,1,4 PARTE TRASERA
```

```
caras[11]._0=0;caras[11]._1=4;caras[11]._2=2; //0,4,2
```

```
}
```

```
//*****
```

```
// clase piramide
```

```
//*****
```

```
_piramide::_piramide(float tam, float al)
```

```
{
```

```
//vertices
```

```
vertices.resize(5);
```

```
vertices[0].x=-tam;vertices[0].y=0;vertices[0].z=tam; //0
```

```
vertices[1].x=tam;vertices[1].y=0;vertices[1].z=tam; //1
```

```
vertices[2].x=tam;vertices[2].y=0;vertices[2].z=-tam; //2
```

```
vertices[3].x=-tam;vertices[3].y=0;vertices[3].z=-tam; //3
```

```
vertices[4].x=0;vertices[4].y=al;vertices[4].z=0; //Punto de altura max - 4
```

```
caras.resize(6);
```

```
//Las caras estan hechas por triángulos
```

```
//Da igual el orden de las características
```

```
caras[0]._0=0;caras[0]._1=1;caras[0]._2=4; //Cara entre 0,1 y 4
```

```
caras[1]._0=1;caras[1]._1=2;caras[1]._2=4; //Cara entre 1,2 y 4
```

```
caras[2]._0=2;caras[2]._1=3;caras[2]._2=4;
```

```
caras[3]._0=3;caras[3]._1=0;caras[3]._2=4;
```

10 %  
dto

Mac

5 %  
dto

iPad



Con cualquier  
Mac o iPad  
Microsoft 365  
incluido.

**Rossellimac**

Tu especialista  
en Apple.

Descuento de 10% en MacBook Pro,  
iMac y Mac Studio y descuento de  
5% en iPad para estudiantes y  
profesores mayores de 18 años.  
Oferta válida hasta el 24/9/2023.  
Microsoft 365 Personal (Valorado  
en 69€) incluido con la compra de  
cualquier Mac o iPad.



@Zukii on Wuolah

```
caras[4]._0=3;caras[4]._1=1;caras[4]._2=0;
caras[5]._0=3;caras[5]._1=2;caras[5]._2=1;
}

//*****
// clase icosahedro
//*****

_icosahedro::_icosahedro(float tam, float al)
{
//vertices
vertices.resize(6);
vertices[0].x=0;vertices[0].y=0;vertices[0].z=0; //0 -> (0,0,0)
vertices[1].x=tam;vertices[1].y=al/2;vertices[1].z=tam; //1 -> (tam,al/2,tam)
vertices[2].x=tam;vertices[2].y=al/2;vertices[2].z=-tam; //2 -> (tam,al/2,-tam)
vertices[3].x=-tam;vertices[3].y=al/2;vertices[3].z=-tam; //3 -> (-tam,al/2,-tam)
vertices[4].x=-tam;vertices[4].y=al/2;vertices[4].z=tam; //4 -> (-tam,al/2,tam)
vertices[5].x=0;vertices[5].y=al;vertices[5].z=0; //5 -> (0,al,0)

caras.resize(8);
//Las caras estan hechas por triángulos
//OJO NO DEBE HABER CARA ENTRE 1,2,3 Y 2,3,4 (cara oculta)
caras[0]._0=0;caras[0]._1=1;caras[0]._2=2; //Cara entre 0,1 y 2
caras[1]._0=0;caras[1]._1=2;caras[1]._2=3;
caras[2]._0=0;caras[2]._1=3;caras[2]._2=4;
caras[3]._0=0;caras[3]._1=4;caras[3]._2=1;
caras[4]._0=2;caras[4]._1=3;caras[4]._2=5; //Superiores
caras[5]._0=3;caras[5]._1=4;caras[5]._2=5;
caras[6]._0=4;caras[6]._1=1;caras[6]._2=5;
caras[7]._0=1;caras[7]._1=2;caras[7]._2=5;
}
```

## Objetos.h

```
//*****  
// Práctica 1 usando objetos  
//*****  
  
#include <vector>  
#include <GL/gl.h>  
#include "vertex.h"  
#include <stdlib.h>  
  
using namespace std;  
  
const float AXIS_SIZE=5000;  
  
//*****  
// clase punto  
//*****  
  
class _puntos3D  
{  
public:  
  
    _puntos3D();  
void    draw_puntos(float r, float g, float b, int grosor);  
  
vector<_vertex3f> vertices;  
//vector<_vertex3f> colores;  
//int color;  
//Métodos para cambiar los valores de estos  
  
};  
  
//*****  
// clase triángulo  
//*****  
  
class _triangulos3D: public _puntos3D  
{  
public:  
  
    _triangulos3D();
```

@Zukii on Wuolah

```
void draw_aristas(float r, float g, float b, int grosor);  
void draw_solido(float r, float g, float b);  
void draw_solido_ajedrez(float r1, float g1, float b1, float r2, float g2, float b2);
```

```
vector<_vertex3i> caras;  
};
```

```
//*****  
// clase cubo  
//*****
```

```
class _cubo: public _triangulos3D  
{  
public:  
  
    _cubo(float tam=0.5);  
};
```

```
//*****  
// clase piramide  
//*****
```

```
class _piramide: public _triangulos3D  
{  
public:  
  
    _piramide(float tam=0.5, float al=0.75);  
};
```

//Crear sólido básico. Vértices y caras. Diamante por ejemplo, icosaedro

```
//*****  
// clase icosaedro - HECHA POR MI  
//*****
```

```
class _icosaedro: public _triangulos3D  
{  
public:  
    _icosaedro(float tam=0.5, float al=1.5); //LA altura es la altura entre el vertice mas  
bajo y mas alto
```

The ColaCao logo is written in a stylized, blue, cursive font with a white outline, set against a red and yellow background with dynamic, brushstroke-like patterns.

# PERDÓN, ¿TE LLAMAS APROBADO Y DE APELLIDO CONUNDIEZ? ¡TÚ PUEDES!

@Zukii on Wuolah

};

[Practica\\_objetos.cc](#)

```
#include "stdlib.h"
```

```
#include "stdio.h"
```

```
#include <GL/glut.h>
```

```
#include <ctype.h>
```

```
#include "objetos_B.h"
```

```
// tamaño de los ejes
```

```
//const int AXIS_SIZE=5000;
```

```
// variables que definen la posicion de la camara en coordenadas polares
```

```
GLfloat Observer_distance;
```

```
GLfloat Observer_angle_x;
```

```
GLfloat Observer_angle_y;
```

```
// variables que controlan la ventana y la transformacion de perspectiva
```

```
GLfloat Window_width,Window_height,Front_plane,Back_plane;
```

```
// variables que determinan la posicion y tamaño de la ventana X
```

```
int
```

```
UI_window_pos_x=50,UI_window_pos_y=50,UI_window_width=450,UI_window_height=450;
```

```
_piramide piramide(0.85,1.3);
```

```
_cubo cubo(0.5);
```

```
_icosahedro icosahedro(0.4,1); //Icosahedro para la clase
```

```
//*****  
//  
//*****  
  
void clear_window()  
{  
  
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );  
}  
  
//*****  
// Funcion para definir la transformación de proyeccion  
//*****  
  
void change_projection()  
{  
  
glMatrixMode(GL_PROJECTION);  
glLoadIdentity();  
  
// formato(x_minimo,x_maximo, y_minimo, y_maximo,Front_plane, plano_traser)  
// Front_plane>0 Back_plane>PlanoDelantero)  
glFrustum(-Window_width,Window_width,-  
Window_height,Window_height,Front_plane,Back_plane);  
}  
  
//*****  
// Funcion para definir la transformación de vista (posicionar la camara)  
//*****
```

@Zukii on Wuolah

```
void change_observer()
{

// posicion del observador
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0,0,-Observer_distance);
glRotatef(Observer_angle_x,1,0,0);
glRotatef(Observer_angle_y,0,1,0);
}

//*****
// Funcion que dibuja los ejes utilizando la primitiva grafica de lineas
//*****

void draw_axis()
{
glBegin(GL_LINES);
// eje X, color rojo
glColor3f(1,0,0);
glVertex3f(-AXIS_SIZE,0,0);
glVertex3f(AXIS_SIZE,0,0);
// eje Y, color verde
glColor3f(0,1,0);
glVertex3f(0,-AXIS_SIZE,0);
glVertex3f(0,AXIS_SIZE,0);
// eje Z, color azul
glColor3f(0,0,1);
glVertex3f(0,0,-AXIS_SIZE);
glVertex3f(0,0,AXIS_SIZE);
glEnd();
```

@Zukii on Wuolah

```
}
```

```
//*****  
// Funcion que dibuja los objetos  
//*****
```

```
void draw_objects()
```

```
{
```

```
//piramide.draw_aristas(1.0,0.5,0.0,3); //color 1,0.5,0 y grosor 3  
//cubo.draw_aristas(1.0,0.5,0.0,3);  
//cubo.draw_solido(1.0,0.5,0);  
//cubo.draw_solido_ajedrez(0,0.5,0.25,1,0,0);  
//icosahedro.draw_aristas(1.0,0.5,0.0,3); //color 1,0.5,0 y grosor 3  
icosahedro.draw_solido_ajedrez(0,0.5,0.25,1,0,0);
```

```
}
```

```
//*****  
//  
//*****
```

```
void draw_scene(void)
```

```
{
```

```
clear_window();  
change_observer();  
draw_axis();
```



ColaCao

APROBASTE LA COURSE NAVETTE,  
SUPERASTE A TU EX E HICISTE  
NUEVOS AMIGOS. ESTE EXAMEN NO  
ES NADA PARA TI. **TÚ PUEDES.**

@Zukii on Wuolah

```
draw_objects();  
glutSwapBuffers();  
}
```

```
//*****
```

```
// Funcion llamada cuando se produce un cambio en el tamaño de la ventana
```

```
//
```

```
// el evento manda a la funcion:
```

```
// nuevo ancho
```

```
// nuevo alto
```

```
//*****
```

```
void change_window_size(int Ancho1,int Alto1)
```

```
{
```

```
change_projection();
```

```
glViewport(0,0,Ancho1,Alto1);
```

```
glutPostRedisplay();
```

```
}
```

```
//*****
```

```
// Funcion llamada cuando se produce aprieta una tecla normal
```

```
//
```

```
// el evento manda a la funcion:
```

```
// codigo de la tecla
```

```
// posicion x del raton
```

```
// posicion y del raton
```

```
//*****
```

@Zukii on Wuolah

```
void normal_keys(unsigned char Tecla1,int x,int y)
{

if (toupper(Tecla1)=='Q') exit(0);
}

//*****
// Funcion llamada cuando se produce aprieta una tecla especial
//
// el evento manda a la funcion:
// codigo de la tecla
// posicion x del raton
// posicion y del raton

//*****

void special_keys(int Tecla1,int x,int y)
{

switch (Tecla1){
    case GLUT_KEY_LEFT:Observer_angle_y--;break;
    case GLUT_KEY_RIGHT:Observer_angle_y++;break;
    case GLUT_KEY_UP:Observer_angle_x--;break;
    case GLUT_KEY_DOWN:Observer_angle_x++;break;
    case GLUT_KEY_PAGE_UP:Observer_distance*=1.2;break;
    case GLUT_KEY_PAGE_DOWN:Observer_distance/=1.2;break;
}

glutPostRedisplay();
}
```

```
//*****  
// Funcion de inicializacion  
//*****  
  
void initialize(void)  
{  
    // se inicializan la ventana y los planos de corte  
    Window_width=.5;  
    Window_height=.5;  
    Front_plane=1;  
    Back_plane=1000;  
  
    // se inicia la posicion del observador, en el eje z  
    Observer_distance=3*Front_plane;  
    Observer_angle_x=0;  
    Observer_angle_y=0;  
  
    // se indica cual sera el color para limpiar la ventana (r,v,a,al)  
    // blanco=(1,1,1,1) rojo=(1,0,0,1), ...  
    glClearColor(1,1,1,1);  
  
    // se habilita el z-bufer  
    glEnable(GL_DEPTH_TEST);  
    change_projection();  
    glViewport(0,0,UI_window_width,UI_window_height);  
}  
  
//*****  
// Programa principal
```

@Zukii on Wuolah

```
//  
// Se encarga de iniciar la ventana, asignar las funciones e comenzar el  
// bucle de eventos  
//*****  
  
int main(int argc, char **argv)  
{  
    // se llama a la inicialización de glut  
  
    glutInit(&argc, argv);  
  
    // se indica las características que se desean para la visualización con OpenGL  
    // Las posibilidades son:  
    // GLUT_SIMPLE -> memoria de imagen simple  
    // GLUT_DOUBLE -> memoria de imagen doble  
    // GLUT_INDEX -> memoria de imagen con color indizado  
    // GLUT_RGB -> memoria de imagen con componentes rojo, verde y azul para cada pixel  
    // GLUT_RGBA -> memoria de imagen con componentes rojo, verde, azul y alfa para cada  
    pixel  
    // GLUT_DEPTH -> memoria de profundidad o z-bufer  
    // GLUT_STENCIL -> memoria de estarcido  
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH);  
  
    // posicion de la esquina inferior izquierda de la ventana  
    glutInitWindowPosition(UI_window_pos_x, UI_window_pos_y);  
  
    // tamaño de la ventana (ancho y alto)  
    glutInitWindowSize(UI_window_width, UI_window_height);  
  
    // llamada para crear la ventana, indicando el titulo (no se visualiza hasta que se llama  
    // al bucle de eventos)
```

The ColaCao logo is written in a stylized, blue, cursive font with a white outline, set against a red background with yellow streaks.

# PERDÓN, ¿TE LLAMAS APROBADO Y DE APELLIDO CONUNDIEZ? ¡TÚ PUEDES!

@Zukii on Wuolah

```
glutCreateWindow("Práctica 1");

// asignación de la funcion llamada "dibujar" al evento de dibujo
glutDisplayFunc(draw_scene);

// asignación de la funcion llamada "cambiar_tamano_ventana" al evento correspondiente
glutReshapeFunc(change_window_size);

// asignación de la funcion llamada "tecla_normal" al evento correspondiente
glutKeyboardFunc(normal_keys);

// asignación de la funcion llamada "tecla_Especial" al evento correspondiente
glutSpecialFunc(special_keys);

// funcion de inicialización
initialize();

// inicio del bucle de eventos
glutMainLoop();
return 0;
}
```

Zukii