

Creando de Interfaces de Usuario

Práctica 1

Primeros pasos en las IU basadas en OpenGL

Héctor Garbisu Arocha

Curso 2015/16

Grado en Ingeniería Informática

Escuela de Ingeniería Informática

Universidad de Las Palmas de Gran Canaria

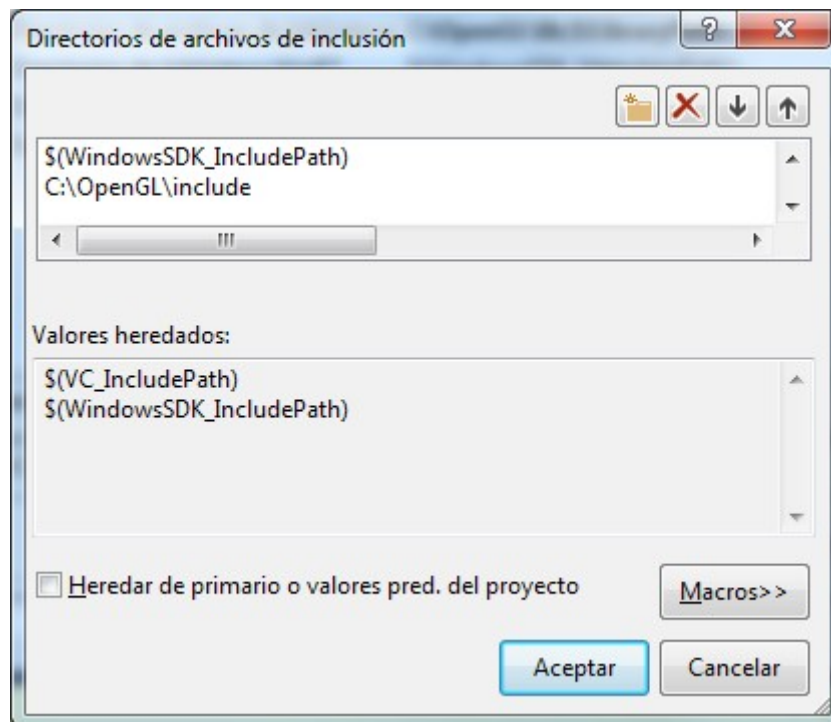
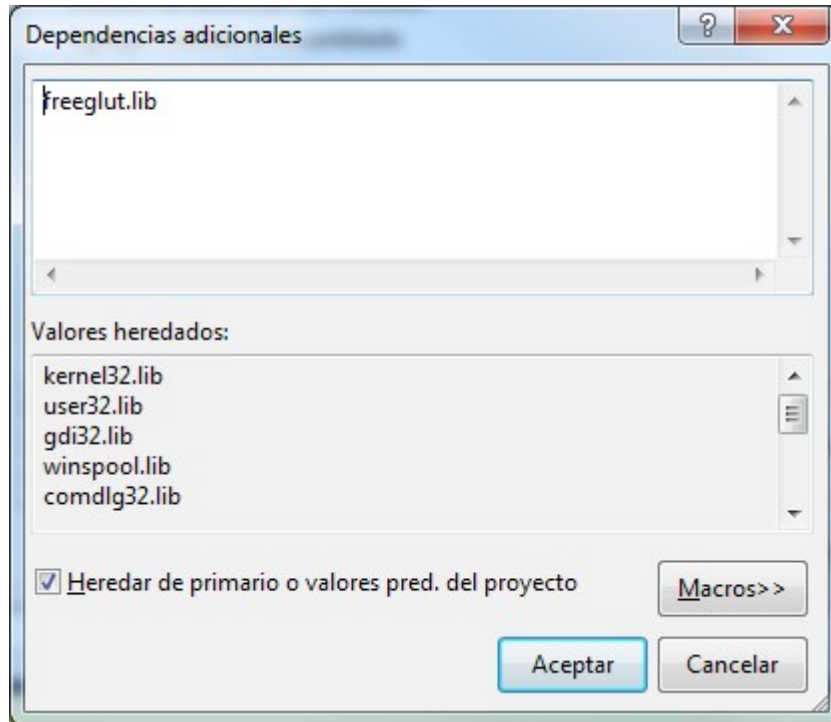
Índice

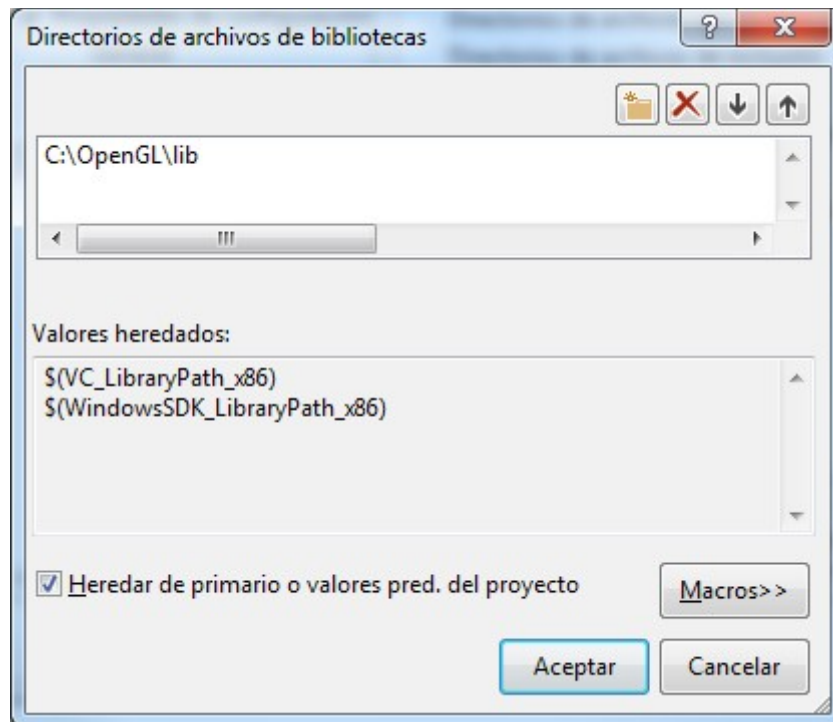
Configuración del entorno	pág. 3
Tarea	pág. 7
Anexo. Código completo	pág. 8

1. Configuración del entorno

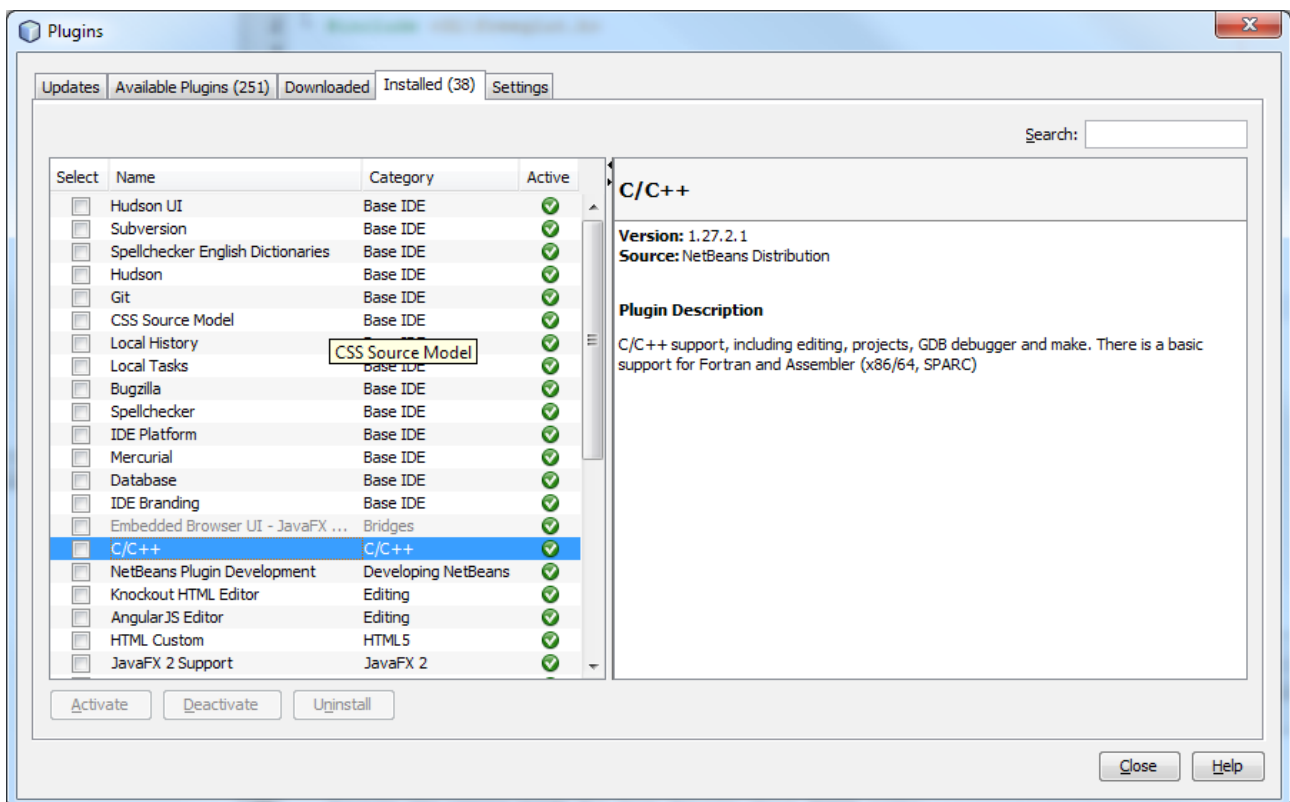
Para las prácticas usaremos tanto NetBeans como VisualStudio.

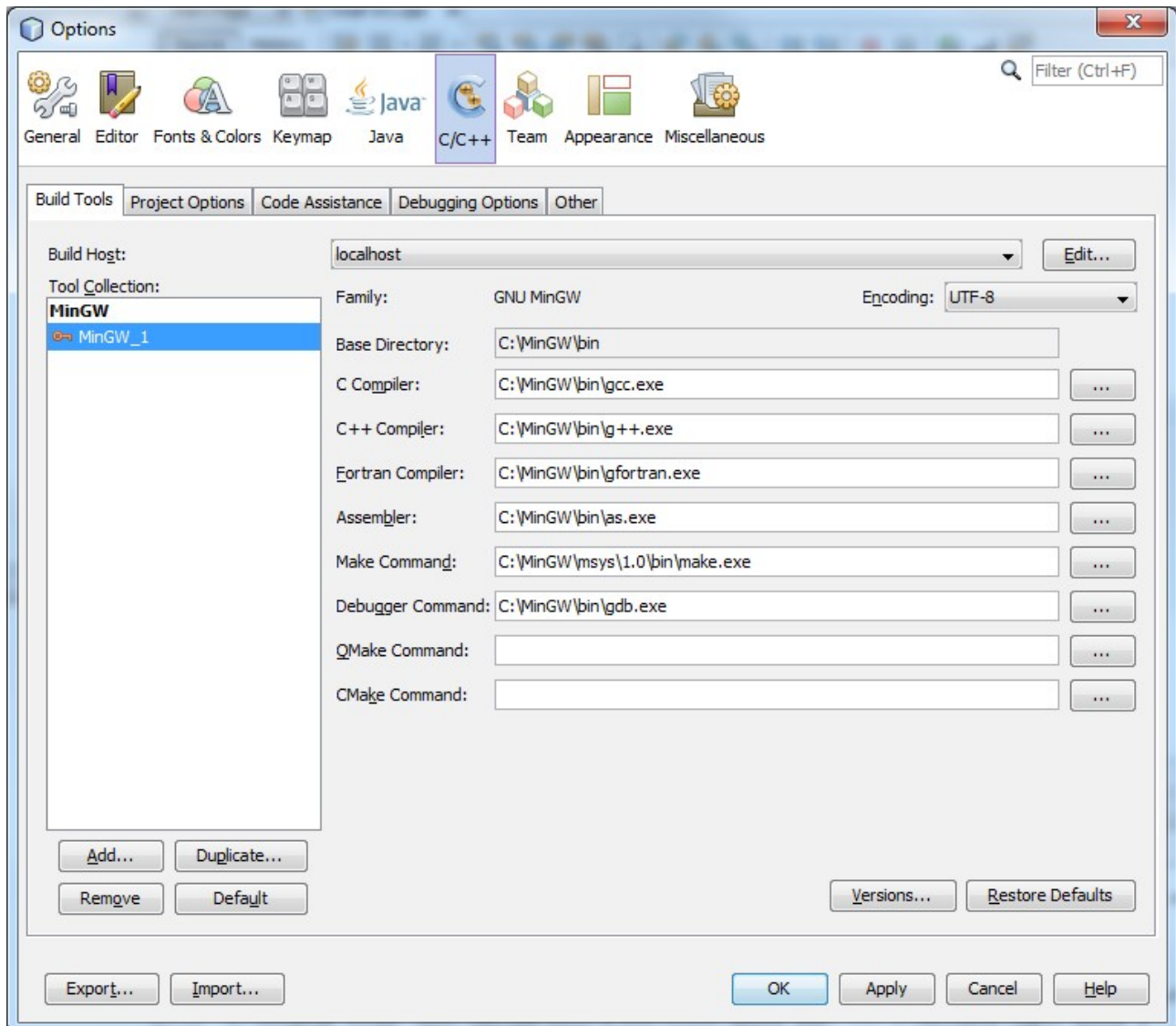
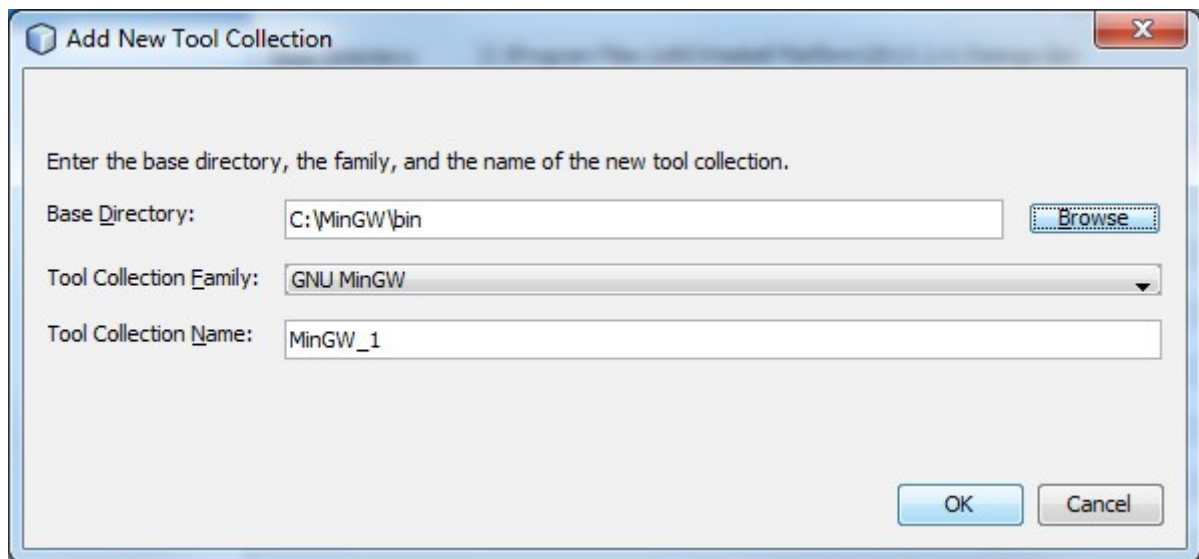
La configuración de estos entornos de programación implica añadir las librerías adecuadas, en este caso las de freeglut.

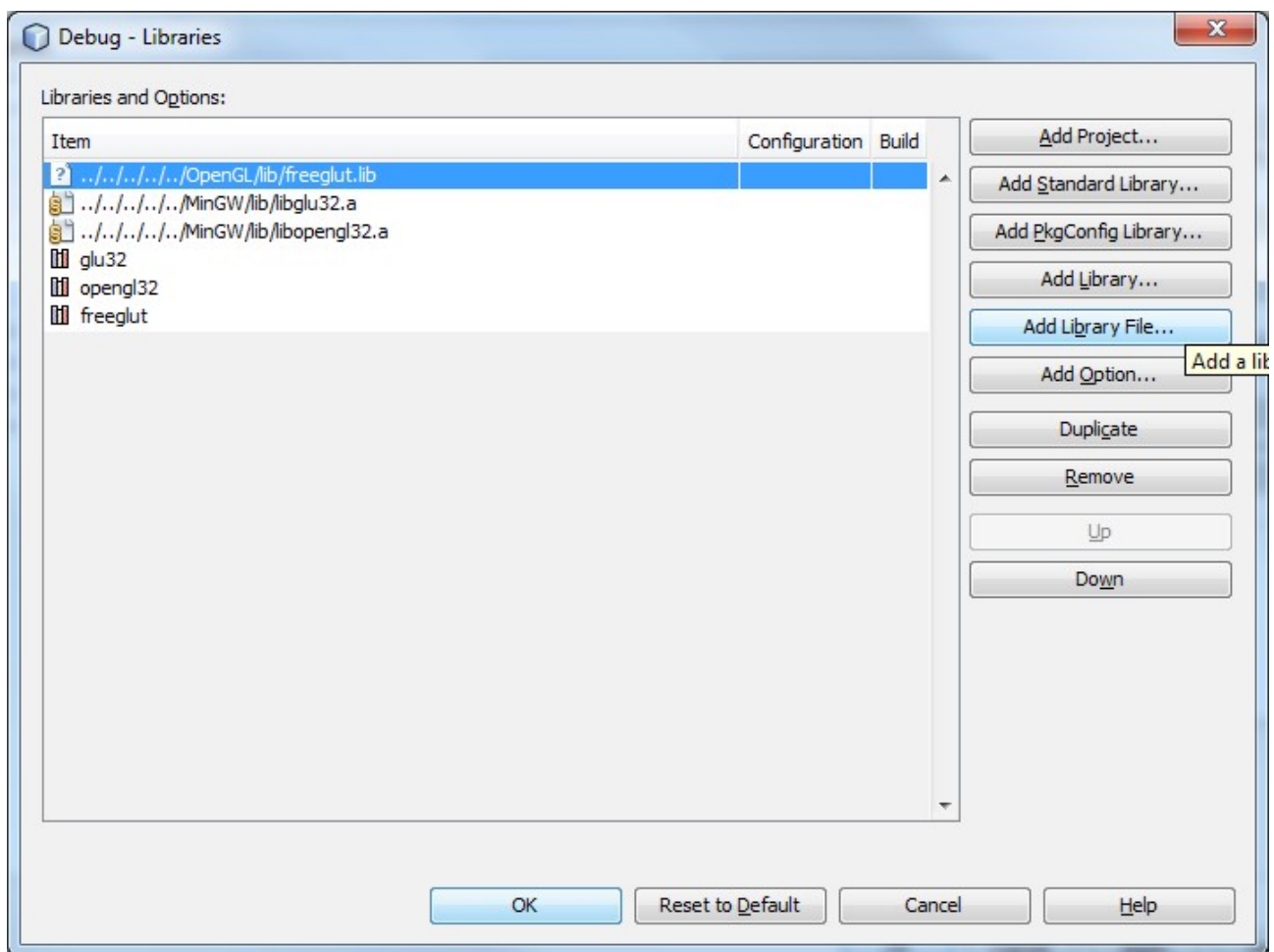




Para NetBeans, además, añadiremos un compilador de C++ (Mingw) ya que por defecto no tiene.





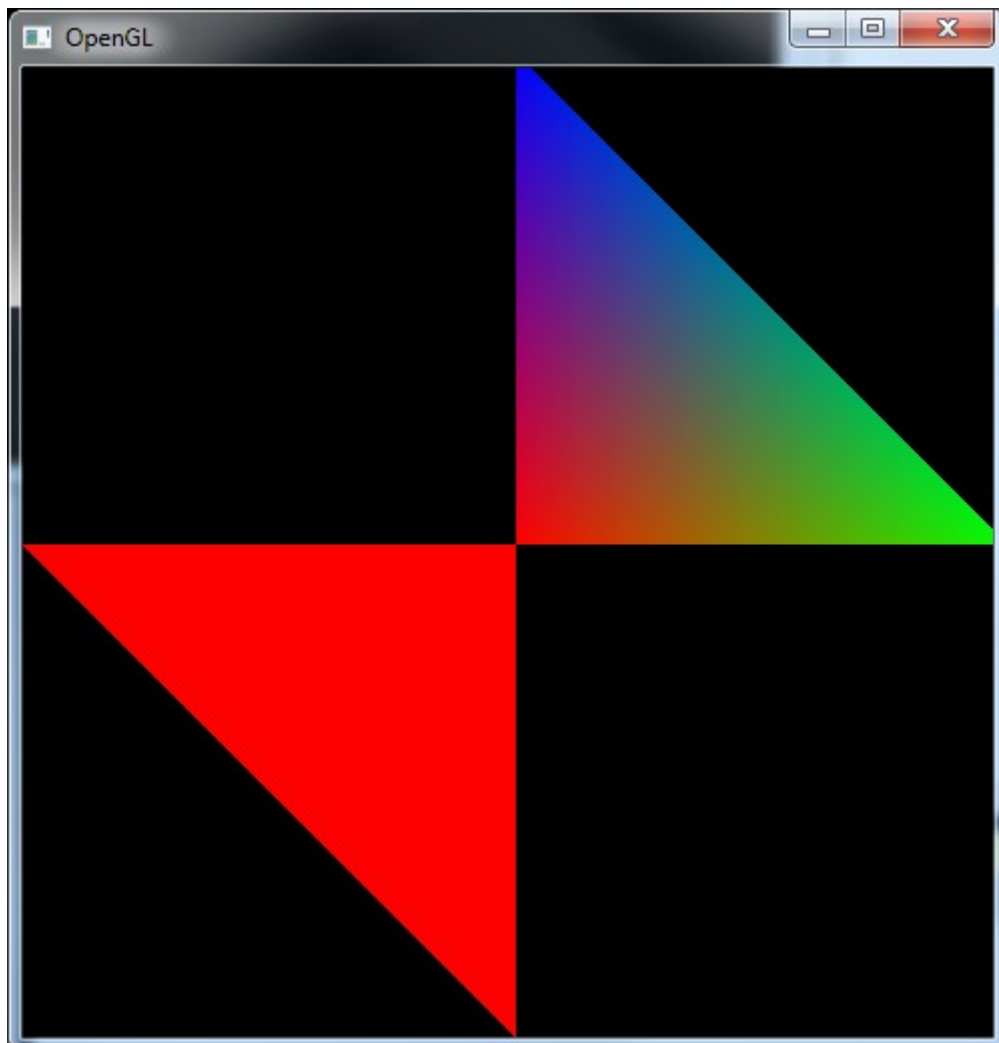


2. Tarea1

La tarea propuesta consiste en dibujar dos triángulos la forma en que GLUT nos permite dibujar un sólido es mediante la definición de sus vértices. Esta vez dejaremos que sea el programa el que haga la interpolación de los colores, haciendo que nosotros solo tengamos que definir el color de los vértices.

El primer triángulo tiene sus respectivos vértices de color rojo, verde y azul, mientras que el segundo triángulo se indica la posición de los vértices sin cambiar de color, lo que hace que sean iguales.

```
glBegin(GL_TRIANGLES);  
    glColor3f(1.0f, 0.0f, 0.0f);  
    glVertex2f(0.0f, 0.0f);  
    glColor3f(0.0f, 1.0f, 0.0f);  
    glVertex2f(1.0f, 0.0f);  
    glColor3f(0.0f, 0.0f, 1.0f);  
    glVertex2f(0.0f, 1.0f);  
  
    glColor3f(0.60f, 0.10f, 0.50f);  
    glVertex2f(0.0f, 0.0f);  
    glVertex2f(-1.0f, 0.0f);  
    glVertex2f(0.0f, -1.0f);  
  
glEnd();
```



3. Código completo

```
#include <stdio.h>
#include <GL\freeglut.h>

void Display(){

    glClear(GL_COLOR_BUFFER_BIT);

    glBegin(GL_TRIANGLES);
        glColor3f(1.0f, 0.0f, 0.0f);
        glVertex2f(0.0f, 0.0f);
        glColor3f(0.0f, 1.0f, 0.0f);
        glVertex2f(1.0f, 0.0f);
        glColor3f(0.0f, 0.0f, 1.0f);
        glVertex2f(0.0f, 1.0f);

        glColor3f(0.60f, 0.10f, 0.50f);
        glVertex2f(0.0f, 0.0f);
        glVertex2f(-1.0f, 0.0f);
        glVertex2f(0.0f, -1.0f);

    glEnd();
    glFlush();
}

void Init(){
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-1.0f, 1.0f, -1.0f, 1.0f, 1.0f, 0.0f);
}

int main(int argc, char *argv[]){
    glutInit(&argc, argv);
    glutInitWindowPosition(300, 300);
    glutInitWindowSize(500, 500);
    glutInitDisplayMode(GLUT_RGBA | GLUT_SINGLE);
    glutCreateWindow("NAME");
    Init();
    glutDisplayFunc(Display);
    glutMainLoop();
    return 0;
}
```