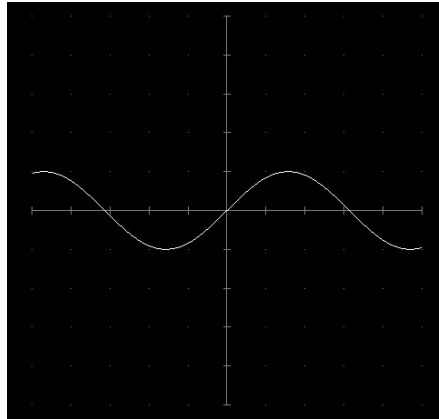

QUESTÃO - EXAME Dezembro (6 valores)

Codifique em JOGL um programa que desenha uma função sinusoidal (função seno), conforme o ilustrado na figura abaixo (eixos X e Y, marcas nos eixos, grelha e sinusoidal). É necessário **apenas codificar a função `display()`**. Assuma-se que só existe 1 visor e a projecção é ortogonal (não necessita de codificar esses elementos). Comentar o código, explicando o que ele faz.



Solução:

```
public void display(GLAutoDrawable drawable) {
    GL gl = drawable.getGL();
    gl.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
    gl.glLoadIdentity();

    // Define origem
    gl.glTranslatef(0,0,-13);
    gl.glColor3f(0.5f,0.5f,0.5f); // Define cor cinza

    // Desenha os eixos X e Y
    gl.glBegin(GL.GL_LINES);
    gl.glVertex3f(-5,0,0); // Primeiro ponto
    gl.glVertex3f(5,0,0); // Segundo ponto
    gl.glVertex3f(0,5,0); // Primeiro ponto
    gl.glVertex3f(0,-5,0); // Segundo ponto
    gl.glEnd();

    // Desenha marcas
    gl.glBegin(GL.GL_LINES);
    for (float x=-5; x<=5; x++)
    {
        gl.glVertex3f(x,0.1f,0);
        gl.glVertex3f(x,-0.1f,0);
        gl.glVertex3f(0.1f,x,0);
        gl.glVertex3f(-0.1f,x,0);
    }
    gl.glEnd();

    // Desenha grelha
    gl.glBegin(GL.GL_POINTS);
    for (float x=-5; x<=5; x++)
        for (float y=-5; y<=5; y++)
            gl.glVertex3f(x,y,0);
    gl.glEnd();

    // Desenha funcao
```

```

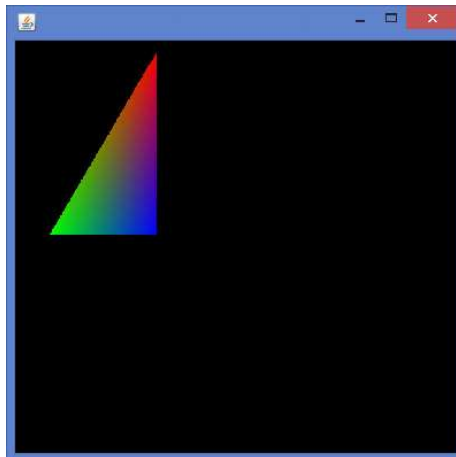
gl.glColor3f(1,1,1); // Define cor branca
gl.glBegin(GL.GL_LINE_STRIP);
    for (float x=-5; x<=5; x=x+0.1f)
    {
        float y=(float)Math.sin(x); // Funcao sinusoidal
        gl.glVertex3f(x,y,0);
    }
gl.glEnd();

gl.glFlush();
}

```

Questão P-fólio – época normal (6 valores)

Desenhe um triângulo conforme a figura abaixo, posicionando-o num visor localizado no canto superior esquerdo da janela. Os vértices do triângulo devem ser coloridos com vermelho, verde e azul, respectivamente. Utilize uma projecção ortogonal. Codifique **apenas a função *display()*** que deverá incluir a **definição do visor e da projecção**, além do desenho do triângulo. Comente o código devidamente, explicando o que ele faz.



Solução:

```

// Invocado sempre para refrescar a cena
public void display(GLAutoDrawable drawable) {
    GL gl = drawable.getGL();

    // Define o branco como cor de fundo
    gl.glClearColor (1.0f, 1.0f, 1.0f, 0.1f);
    int w = WIDTH, h = HEIGHT;

    // executa limpeza de buffer da cor
    // Se não for feito, borra o fundo...
    gl.glClear (GL.GL_COLOR_BUFFER_BIT|GL.GL_DEPTH_BUFFER_BIT);
    gl.glViewport(0, h/2, w/2, h/2);
    gl.glMatrixMode(GL.GL_PROJECTION);
    gl.glLoadIdentity();

    gl.glOrtho(300, WIDTH-300, 300, HEIGHT-300, -200.0f, 200.0f);

    gl.glBegin( GL2.GL_TRIANGLES );// Desenho com triângulos
    gl.glColor3f( 1.0f, 0.0f, 0.0f ); //vermelho

```

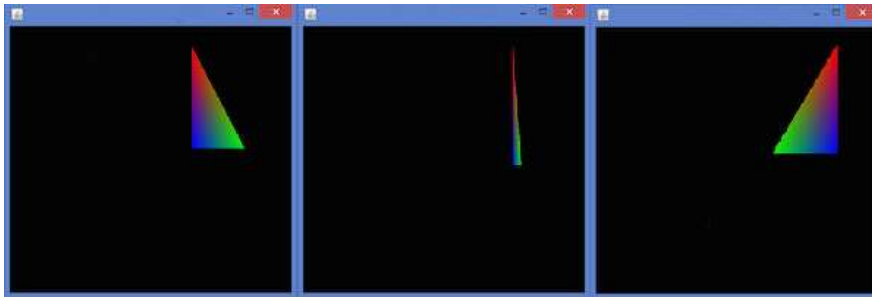
```

gl.glVertex3f( 0.5f,0.7f,0.0f ); // topo
gl.glColor3f( 0.0f,1.0f,0.0f ); //azul
gl.glVertex3f( -0.2f,-0.5f,0.0f ); // fundo a esquerda
gl.glColor3f( 0.0f,0.0f,1.0f ); //verde
gl.glVertex3f( 0.5f,-0.5f,0.0f ); //fundo a direita
gl.glEnd();
}

```

Questão P-fólio – época de recurso (6 valores)

Desenhe um triângulo conforme a figura abaixo, posicionando-o num visor localizado no canto superior direito da janela. Os vértices do triângulo devem ser coloridos com vermelho, verde e azul, respectivamente. O triângulo deverá girar em torno de X gradualmente. Utilize uma projecção ortogonal. Codifique **apenas a função `display()`** que deverá incluir a **definição do visor e da projecção**, além do desenho do triângulo em rotação. Comente o código devidamente, explicando o que ele faz.



Solução:

```

// Invocado sempre para refrescar a cena
public void display(GLAutoDrawable drawable) {
    GL gl = drawable.getGL();

    // Define o branco como cor de fundo
    gl.glClearColor (1.0f, 1.0f, 1.0f, 0.1f);
    int w = WIDTH, h = HEIGHT;

    // executa limpeza de buffer da cor
    // Se não for feito, borra o fundo...
    gl.glClear (GL.GL_COLOR_BUFFER_BIT|GL.GL_DEPTH_BUFFER_BIT);
    gl.glViewport(w/2, h/2, w/2, h/2);
    gl.glMatrixMode(GL.GL_PROJECTION);
    gl.glLoadIdentity();
    gl.glOrtho(0, WIDTH, 0, HEIGHT, -200.0f, 200.0f);

    gl.glLoadIdentity(); // Carrega a matriz identidade
    gl.glRotatef( rtri, 0.0f, 1.0f, 0.0f );// Gira o triângulo
    gl.glBegin( gl.GL_TRIANGLES );// Desenho com triângulos
    gl.glColor3f( 1.0f, 0.0f, 0.0f ); //vermelho
    gl.glVertex3f( 0.5f,0.7f,0.0f ); // topo
    gl.glColor3f( 0.0f,1.0f,0.0f ); //azul
    gl.glVertex3f( -0.2f,-0.5f,0.0f ); // fundo a esquerda
    gl.glColor3f( 0.0f,0.0f,1.0f ); //verde
    gl.glVertex3f( 0.5f,-0.5f,0.0f ); //fundo a direita
    gl.glEnd();

    gl.glFlush();
    rtri +=0.2f; //atualiza rotação
}

```