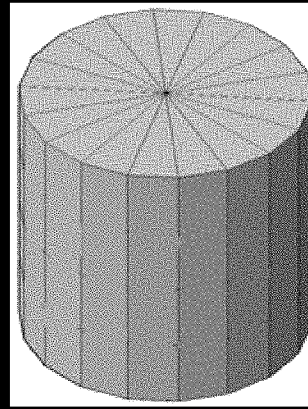


// se hace como una función que devuelve el área:

```
float Area( MallaTVT * m )
{
    float area = 0.0 ;
    // recorremos todas las caras
    for( unsigned i = 0 ; i < m->caras.size() ; i++ )
    {
        // recuperamos las posiciones en WC de los tres vértices
        Tupla3f v0 = m->vertices[m->caras[i][0]],
                v1 = m->vertices[m->caras[i][1]],
                v2 = m->vertices[m->caras[i][2]];

        // calculamos a,b,c,s
        float  a = (v1-v0).longitud(),
              b = (v2-v0).longitud(),
              c = (v1-v2).longitud(),
              s = 0.5*(a+b+c) ;

        // sumamos área del triángulo al área total
        area += sqrt( s*(s-a)*(s-b)*(s-c) ) ;
    }
    return area ;
}
```



```
// asumimos que 'n' es una constante con el número de copias del perfil
// que queremos crear (sin incluir la ultima). Se crea solo la tapa de arriba.
// asumimos que 'h' y 'r' son constantes reales predefinidas

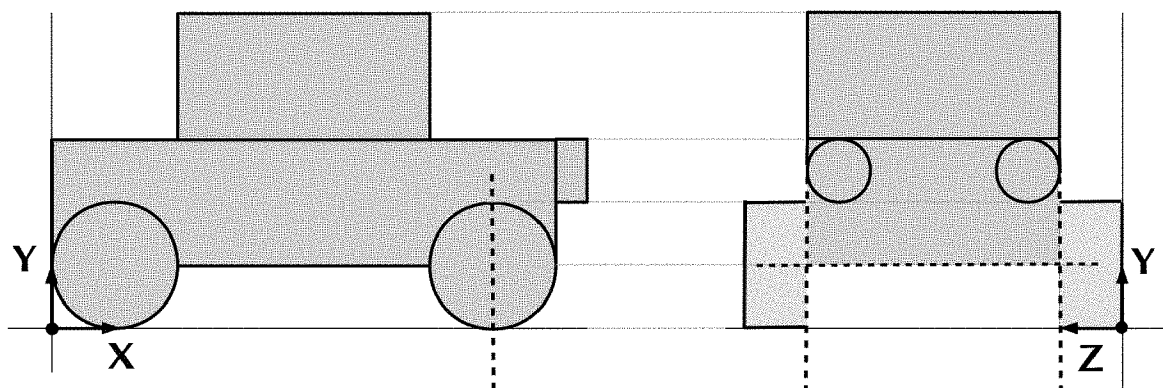
std::vector<Tupla3f> ver ;           // tabla de posiciones de vértices
std::vector<Tupla2f> cct ;           // tabla de coordenadas de textura

for( unsigned i = 0 ; i <= n ; i++ )
{
    float f = float(i)/float(n),      // fracción de perfiles ya calculados
          a = 2.0*M_PI*f,              // ángulo del perfil (radianes)
          c = cos(a), s = sin(alpha),  // seno y coseno del 'a'
          x = r*c, z = r*s ;           // coordenada x,z de los tres vértices

    // añadir los tres vértices a 'ver'
    ver.push_back( Tupla3f( x, 0, z ) ); // vértice inferior del cilindro
    ver.push_back( Tupla3f( x, h, z ) ); // vértice superior del cilindro
    ver.push_back( Tupla3f( x, h, z ) ); // vértice en la tapa (misma pos.)

    // insertar las coords. de textura a 'cct'
    cct.push_back( Tupla2f( f, 0.0 ) ); // vértice inferior del cilindro
    cct.push_back( Tupla2f( f, 0.5 ) ); // vértice superior del cilindro
    cct.push_back( Tupla2f( 0.25+0.25*c, 0.75+0.25*s ) ); // vértice en la tapa
}

// insertamos vértice en el centro de la tapa superior al final
verts.push_back( Tupla3f( 0.0, h, 0.0 ) );
cct.push_back( Tupla2f( 0.25, 0.75 ) );
```



dibujaCoche

ang\_giro

ang\_rodar

