

# examen-resuelto-2019-IG.pdf



**PruebaAlien**



**Informática Gráfica**



**3º Grado en Ingeniería Informática**

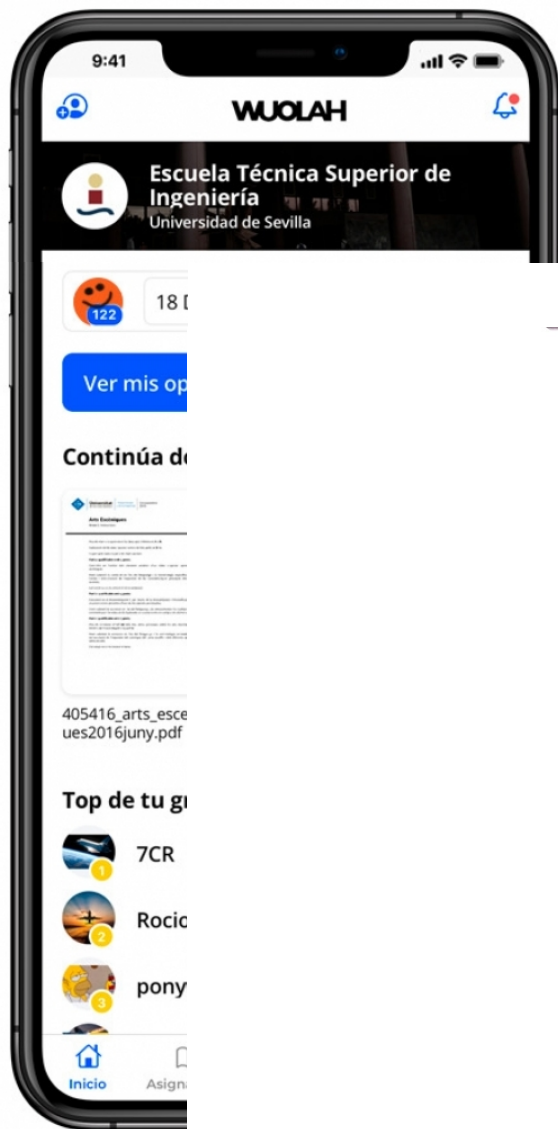


**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.





# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



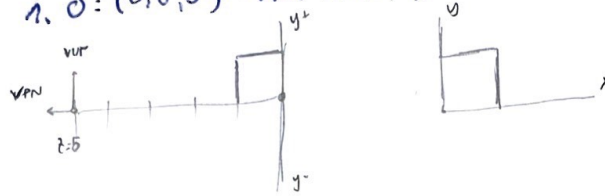
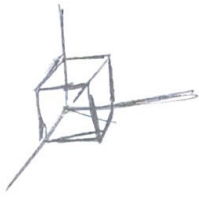
EXAMEN  
LG FEB 2015

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Febrero 2015

1. Cubo bob 1 con sus esquemas en  $(0,0,0)$  y  $(1,1,1)$

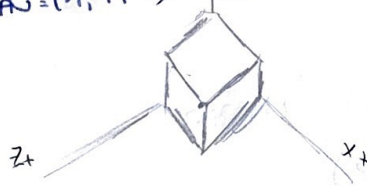
1. 0:  $(0,0,5)$   $VFN = (0,0,1)$   $VP = (0,1,0)$



2. 0:  $(0,0,5)$   $VFN = (0,0,1)$   $VP = (0,-1,0)$



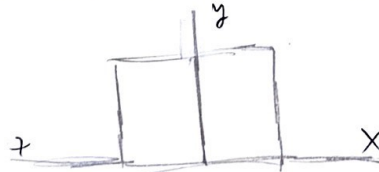
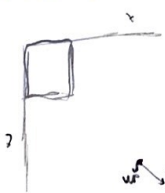
3. 0:  $(5,5,5)$   $VFN = (1,1,1)$   $VP = (0,1,0)$



4. 0:  $(0,5,0)$   $VFN = (0,1,0)$   $VP = (1,0,0)$



5. 0:  $(5,0,5)$   $VFN = (1,0,1)$   $VP = (0,1,0)$





**KEEP  
CALM  
AND  
ESTUDIA  
UN POQUITO**

2

v = tabla vertices  
t = tabla triangulos  
Area del objeto?

```

area-triangulo ( float x, float y, float z ) {
    float a = abs(x-y);
    float b = abs(x-z);
    float c = abs(y-z);

    float s = (a+b+c) / 2;

    return sqrt(s*(s-a)*(s-b)*(s-c));
}

area() {
    double total = 0;
    for (int i = 0; i < f.size(); i++) {
        total += area-triangulo(f[i](0), f[i](1), f[i](2));
    }
    return total;
}

```

3



$P_1 = (r, 0, 0)$  N pasos  
 $P_2 = (r, h, 0)$

Asignar coordenadas de textura en la generacion de la ~~topologia~~ geometria:

//Cilindro sin tapadera

```

v.push_back(P1)
tex.push_back({0,0})
v.push_back(P2)
tex.push_back({0,z/h})
for (int i = 0; i < N; i++) {
    v.push_back(P1.rotarY((360/N)*i));
    tex.push_back({(1+i)/N, 0});
    v.push_back(P2.rotarY((360/N)*(1+i)));
    tex.push_back({(1+i)/N, z/h});
}

```

rotar (Tupla2f p, Tupla2f o, float a)

p = punto a rotar  
o = centro rotación  
a = angulo que rota

//Tapadera: duplicar vertices y asignar textura

```

for (int i = 0; i < N; i++) {
    v.push_back(v[i+1]);
    tex.push_back(rotar({0,z/h}, {1,z/h}, i*(360/N)));
}

```

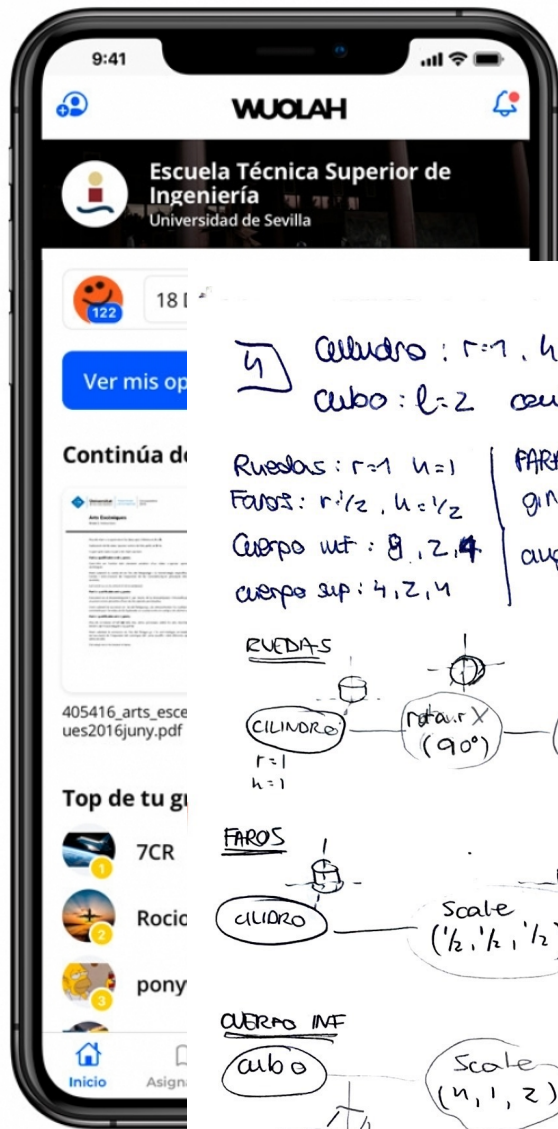
//Centro de la base

```

v.push_back({0,h,0})
tex.push_back({1,0})

```



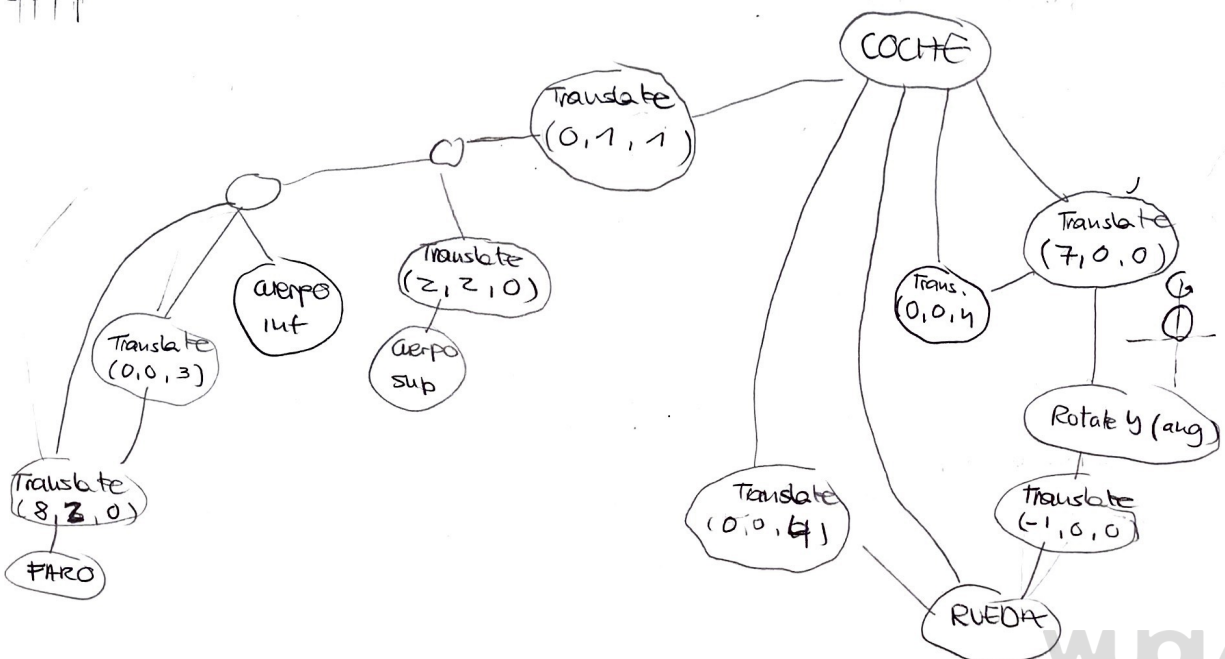
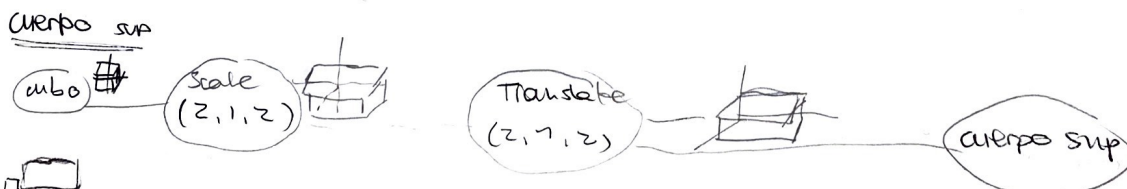
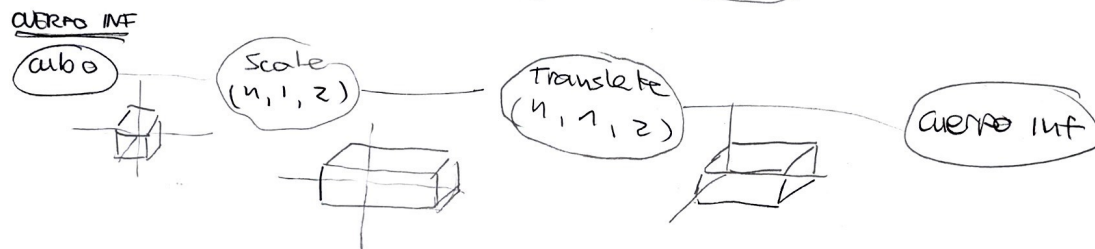
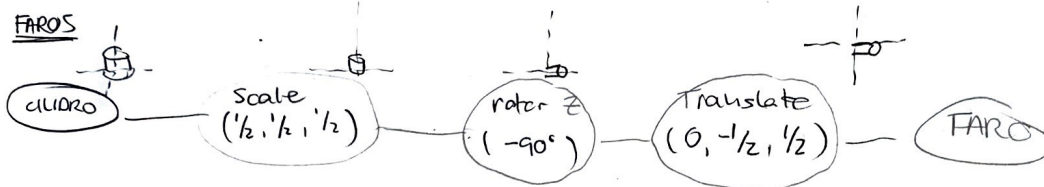
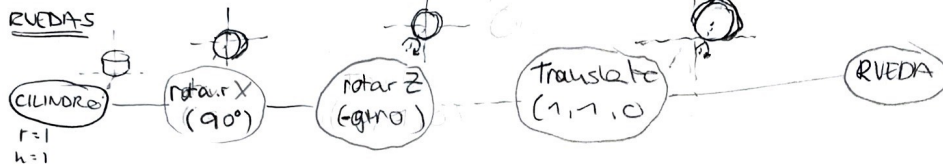
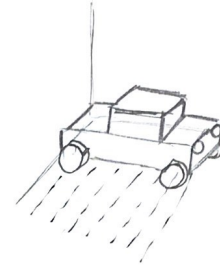


**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.



Cellulo:  $r=1, h=1$ , centro base  $(0,0,0)$   
Cubo:  $l=2$  centro  $(0,0,0)$

Ruedas:  $r=1, h=1$  | **PARAMETROS**  
Faros:  $r=1/2, h=1/2$  |  $gino = 0$   $[0-360]$   
Cuerpo inf:  $8, 2, 4$  |  $ang = 0$   $[-60, 60]$   
Cuerpo sup:  $4, 2, 4$



4.b)

dibujaCodo ( ) {

glPushMatrix ( )

glPushMatrix ( )

glTranslatef (0,1,1)

glPushMatrix ( )

glTranslatef (2,2,0)

cuerpoSuperior.dibujar ( );

glPopMatrix ( )

cuerpoInterior.dibujar ( );

glPushMatrix ( )

glTranslatef (0,0,3)

glTranslatef (8,2,0)

faro.dibujar ( );

glPopMatrix ( )

glPushMatrix ( )

glTranslatef (8,2,0);

faro.dibujar ( );

glPopMatrix ( );

glPopMatrix ( )

glPushMatrix ( )

glTranslatef (0,0,4);

rueda.dibujar ( );

glPopMatrix ( );

glPushMatrix ( );

rueda.dibujar ( );

glPopMatrix ( );

glPushMatrix ( );

glTranslatef (0,0,4)

glTranslatef (7,0,0)

glRotatef (ang, 0,1,0);

glTranslatef (-1,0,0);

rueda.dibujar ( );

glPopMatrix ( );

glPushMatrix ( )

glTranslatef (7,0,0)

glRotatef (ang, 0,1,0);

glTranslatef (-1,0,0)

rueda.dibujar ( );

glPopMatrix ( );

glPopMatrix

}