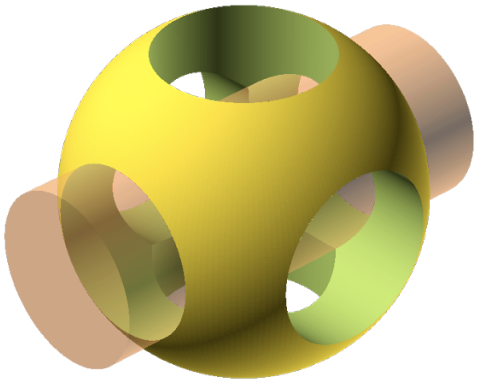


OpenScad

Sumario

Introducción a Openscad

Instalación de Openscad en GNU/Linux

Instalar Openscad en GNU/Linux es muy sencillo. En este escrito comentaremos como puedes hacerlo si usas como distro Debian o similares. También como compilarlo.

Instalar una versión stable de Openscad

```
# apt-get install openscad
```

Instalar una version nightly de Openscad

```
# apt-get remove openscad --purge
# apt autoremove
# wget -qO - http://files.openscad.org/OBS-Repository-Key.pub | sudo apt-key add
-
# echo "deb http://download.opensuse.org/repositories/home:/t-
paul/Debian_9.0/ ./" >> /etc/apt/sources.list
# apt-get update
# apt-get install openscad-nightly
```

Compilar Openscad desde el código fuente

```
# apt-get remove openscad --purge
# apt-get install git flex libharfbuzz-dev libfreetype6-dev
# apt-get install libfontconfig1-dev libqscintilla2-dev libcgal-dev libopencsg-
dev
# apt-get install libglew-dev libeigen3-dev libglib2.0-dev libglib2.0-cil-dev
libxml2-dev
$ git clone https://github.com/openscad/openscad
$ cd openscad/scripts
```

Con el siguiente script comprobamos si nos falta alguna dependencia por instalar:

bash check-dependencies.sh

Finalmente si hemos instalado todas las dependencias y check-dependencies.sh nos dice OK a todo vamos a compilar.

Si deseamos también podemos ejecutar el script uni-get-dependencies.sh para instalar las dependencias.

```
$ cd ..
$ qmake
$ make
$ ./openscad
# make install
$ openscad
```

¿Qué es Openscad y para que puedo usarlo?

OpenSCAD es una aplicación libre para crear objetos sólidos de CAD. No es un editor interactivo sino un compilador 3D basado en un lenguaje de descripción textual. Un documento de OpenSCAD especifica primitivas geométricas y define como son modificadas y manipuladas para reproducir un modelo 3D.

OpenSCAD está **disponible** para Windows, **Linux** y OS X. **El desarrollo de OpenSCAD comenzó en 2010.**

OpenSCAD permite al diseñador crear modelos 3D precisos y diseños paramétricos que pueden ser fácilmente ajustados cambiando los parámetros.

Comparado con la mayoría de otros formatos de fichero CAD, que no son fácilmente leíbles por humanos, **los documentos OpenSCAD son como el software de código abierto**.

Podemos usar Openscad para realizar piezas que luego podemos imprimir con una impresora 3D.
Esto nos permite poder diseñar algo directamente tecleando el código y pudiendo parametrizar la pieza mediante variables. Esas variables nos permitirán poder ajustar la pieza para realizar cambios de forma rapida usando cualquier editor de texto que tengamos a mano o nos guste.

Enlaces de Interés

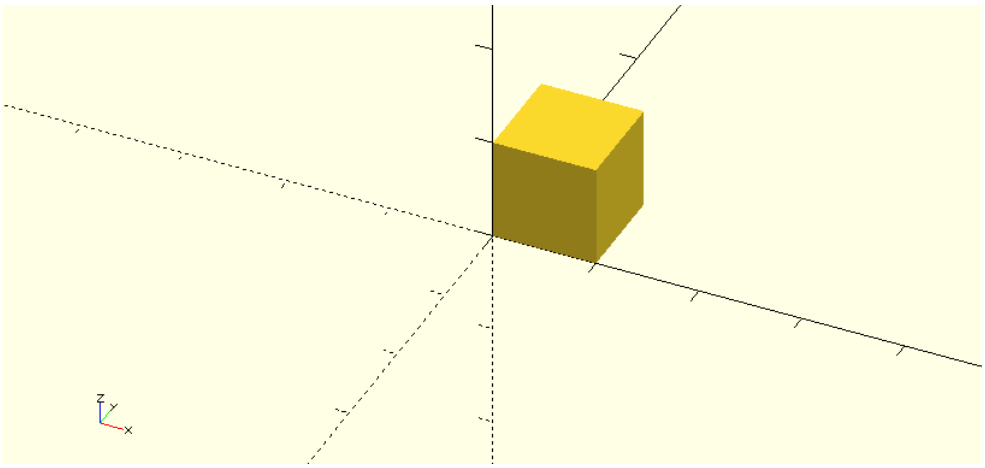
Web oficial: http://www.openscad.org/
Github: https://github.com/openscad/openscad
Cheat Sheet: http://www.openscad.org/cheatsheet/index.html
Openscad online: http://openscad.net/
Openscad Language:
https://en.wikibooks.org/wiki/OpenSCAD_User_Manual/The_OpenSCAD_Language

Diseño paramétrico de un cubo

Vamos a probar varios códigos para ir viendo como funciona el tema.

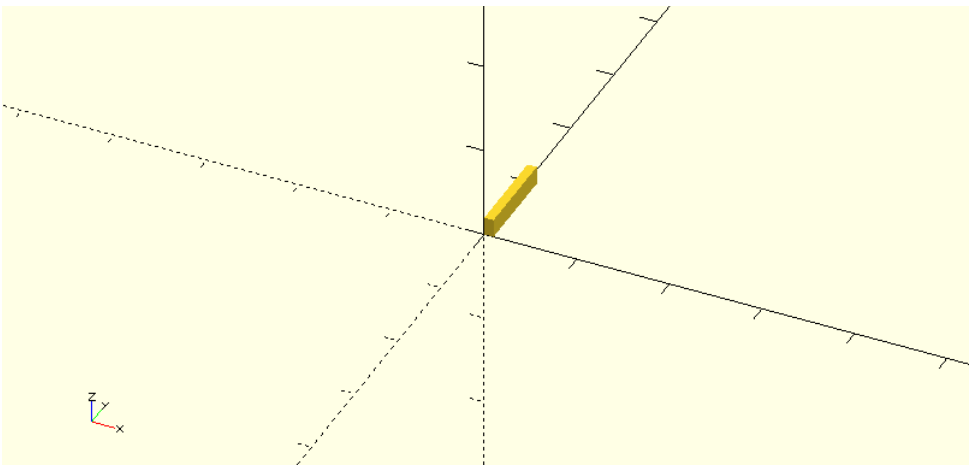
Ejemplo 000:

```
cube([100,100,100]);
```



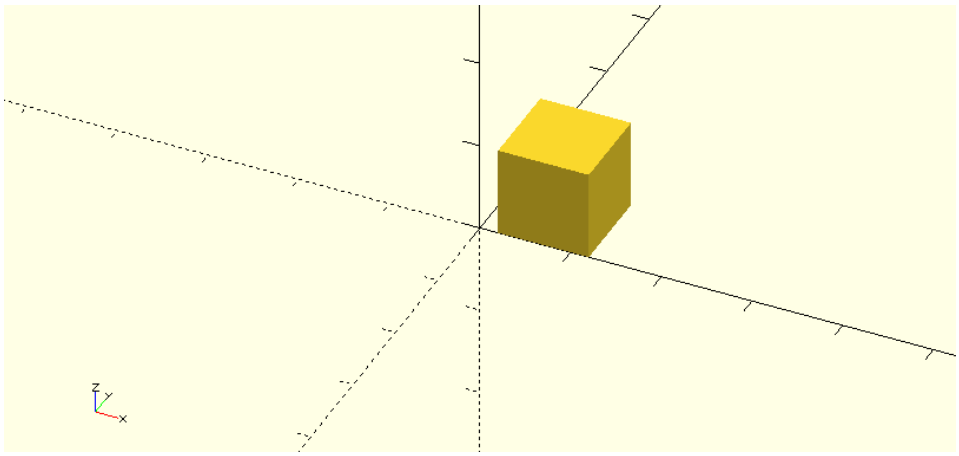
Ejemplo 001:

```
/* Código de ejemplo de un cubo */
// Parámetros del cubo
ancho=10;
fondo=100;
alto=20;
x=ancho;
y=fondo;
z=alto;

cube([x,y,z]);
```



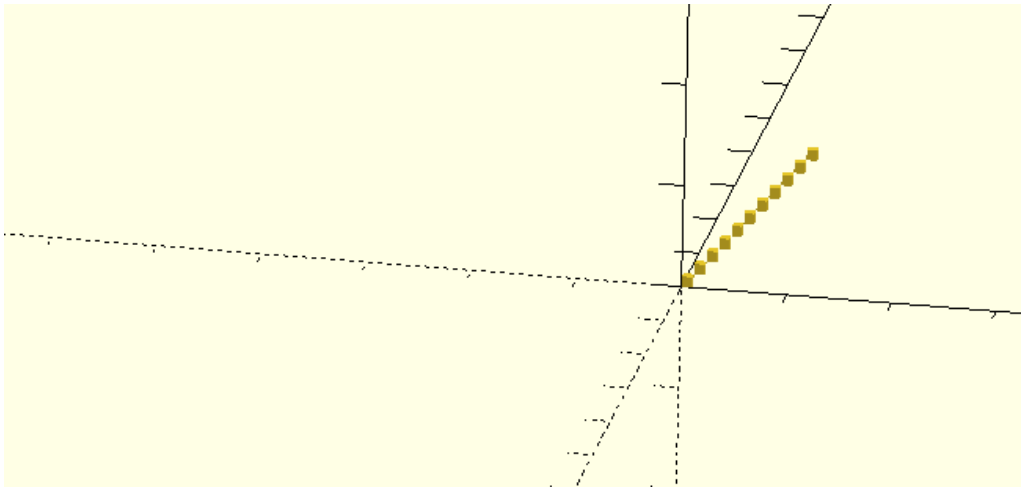
Ejemplo 002:

```
x=100;
y=100;
z=100;

translate([20,0,0]) {
    cube([x,y,z]);
};
```



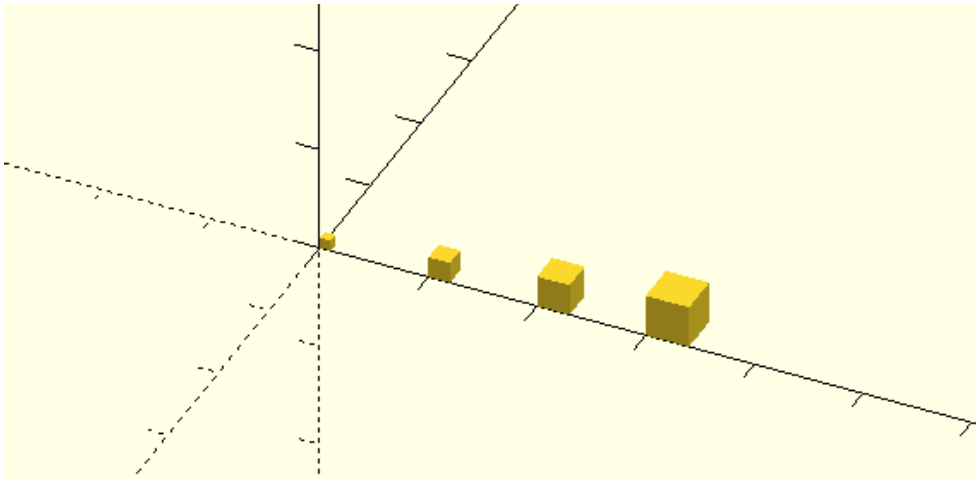
Ejemplo 003:

```
x=1;
y=1;
z=1;

for (a=[0:10]){
    translate([a,a,a]) {
            cube([x,y,z]);
    };
};
```



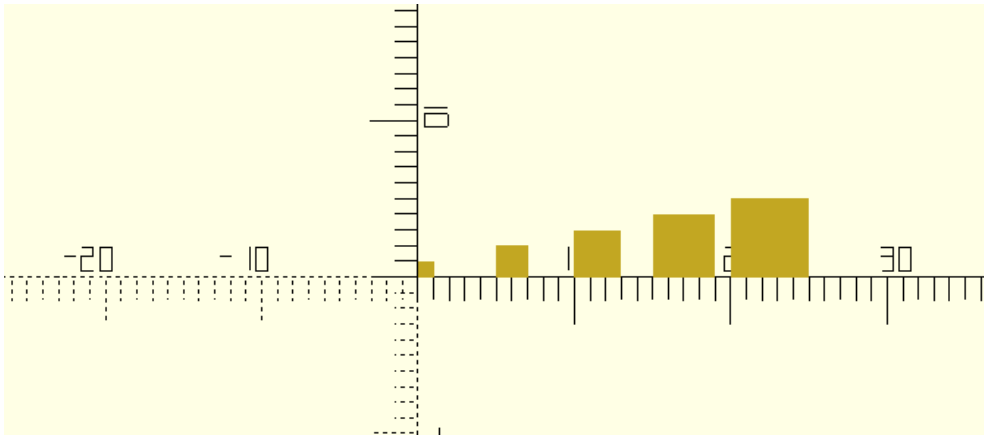
Ejemplo 004:

```
for (i=[0:3])
   translate([i*10,0,0])
     cube(i+1);
```



Ejemplo 005:

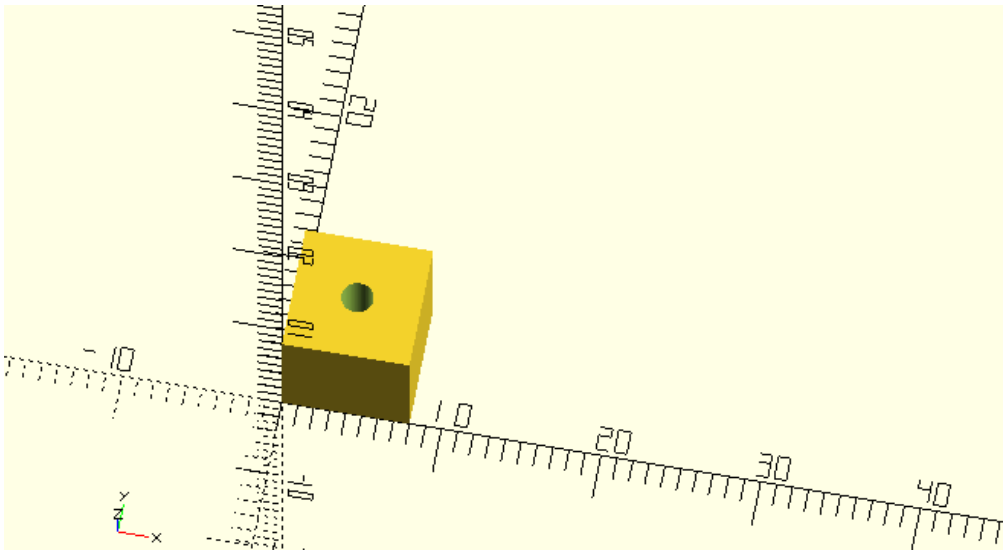
```
for (i=[0:4]){
    translate([i*5,0,0]){
        cube(i+1);
    }
}
```



Diferencia y rotación – Taladros usando cilindros

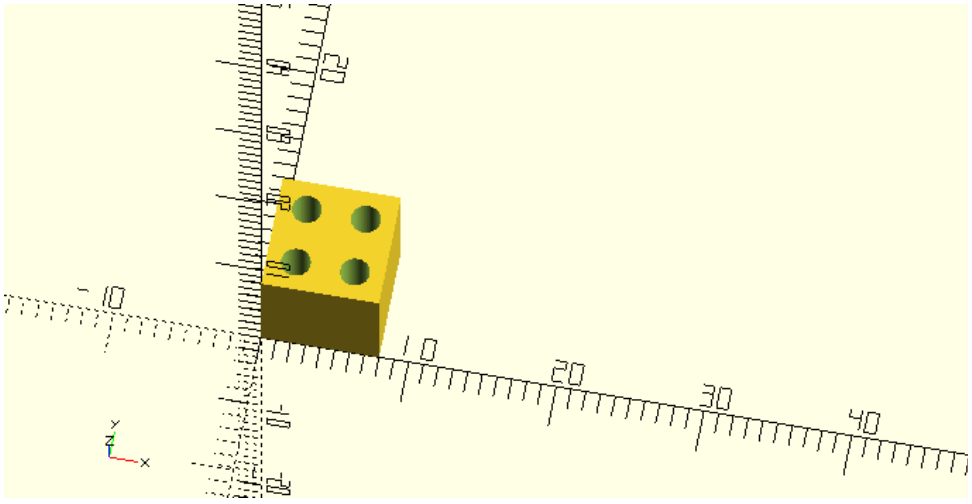
Ejemplo 006:

```
difference(){
    translate ([0,0,0]){ cube([8,8,8]); }
    translate ([4,4,-1]){ cylinder(r=1, h=10, $fn=100); }
}
```



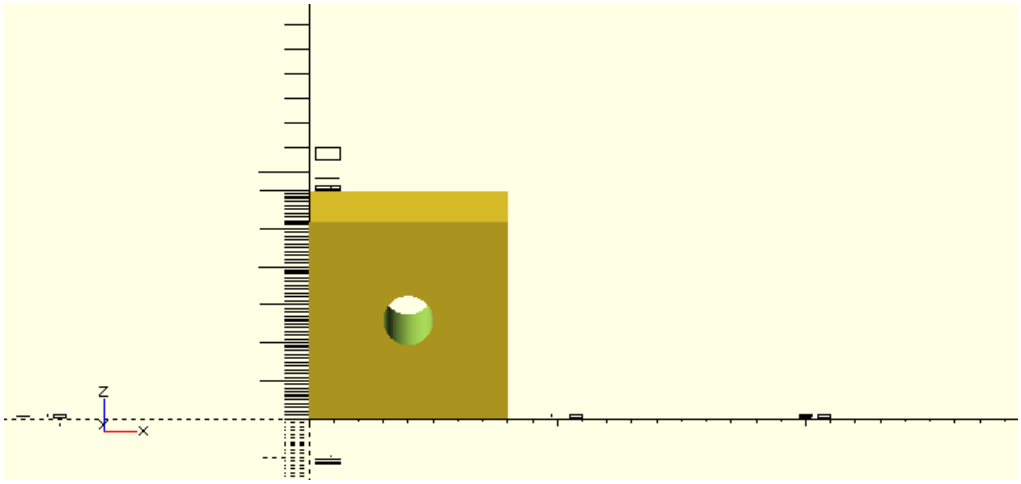
Ejemplo 007

```
difference(){
    translate ([0,0,0]){ cube([8,8,8]); }
    translate ([2,2,-1]){ cylinder(r=1, h=10, $fn=100); }
    translate ([6,6,-1]){ cylinder(r=1, h=10, $fn=100); }
    translate ([6,2,-1]){ cylinder(r=1, h=10, $fn=100); }
    translate ([2,6,-1]){ cylinder(r=1, h=10, $fn=100); }
}
```



Ejemplo 008

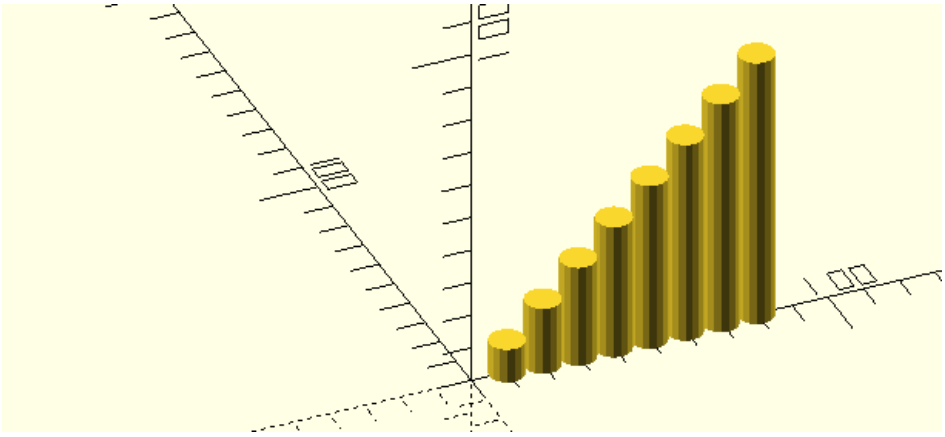
```
difference(){
   translate ([0,0,0]){ cube([8,8,8]); }
    rotate([90,0,0]) {
        translate([4,4,-9]){
            cylinder(r=1, h=10, $fn=100);
        }
   }
}
```



Extruir un circulo 2D otra forma de hacer taladros

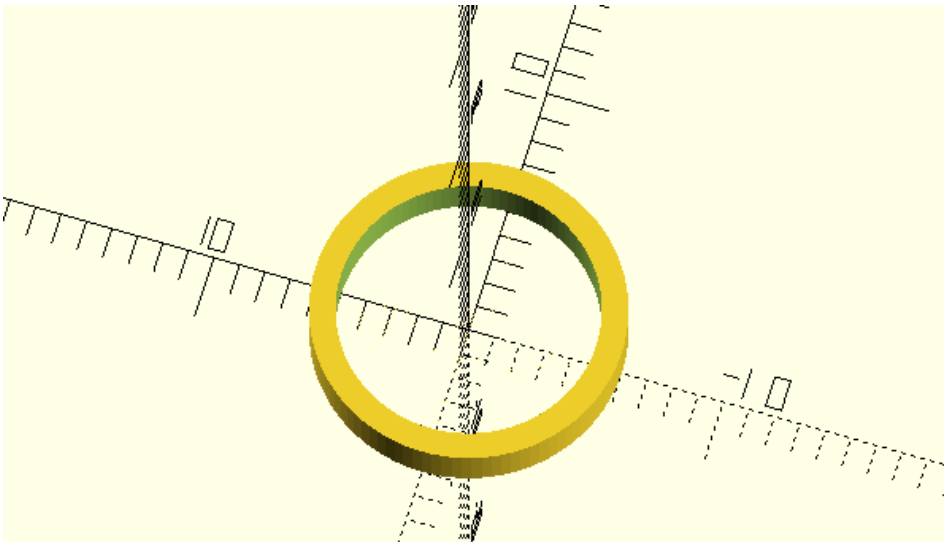
Ejemplo 009:

```
// for [inicio:incremento:final] volcado a variable a
for(a=[0:10:80]){
    translate([a,0,0]){
        linear_extrude(a) { circle (r=5); }
    }
}
```



Ejemplo 010:

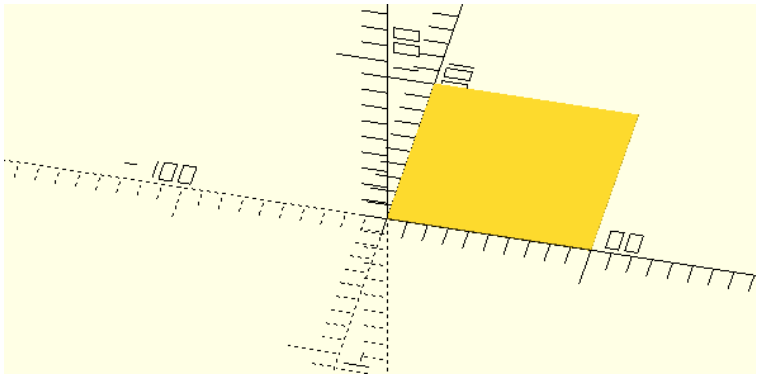
```
difference(){
    cylinder(r=6, h=2, $fn=100);
    translate([0,0,-1]){ linear_extrude(10) { circle (r=5, $fn=100); } }
}
```



Polígonos mediante coordenadas de puntos

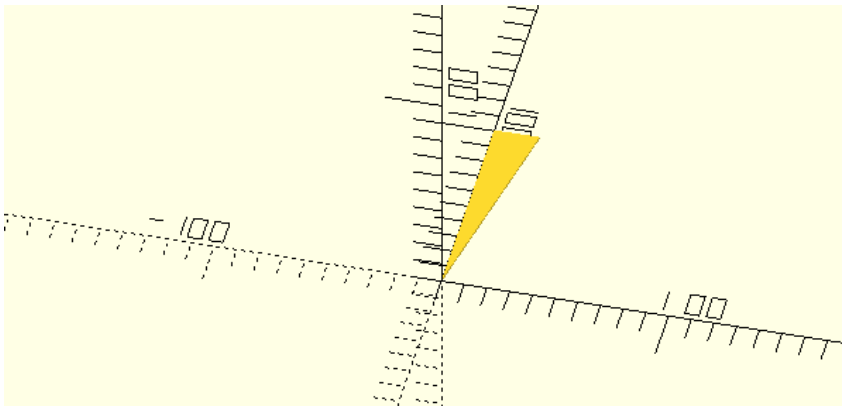
Ejemplo 11:

```
polygon(points=[[0,0],[100,0],[100,100],[0,100]]);
```



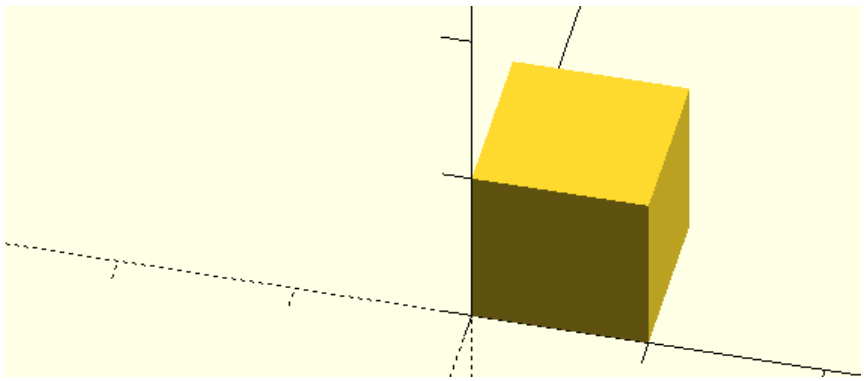
Ejemplo 12:

```
polygon(points=[[0,0],[20,100],[0,100]]);
```



Ejemplo 13:

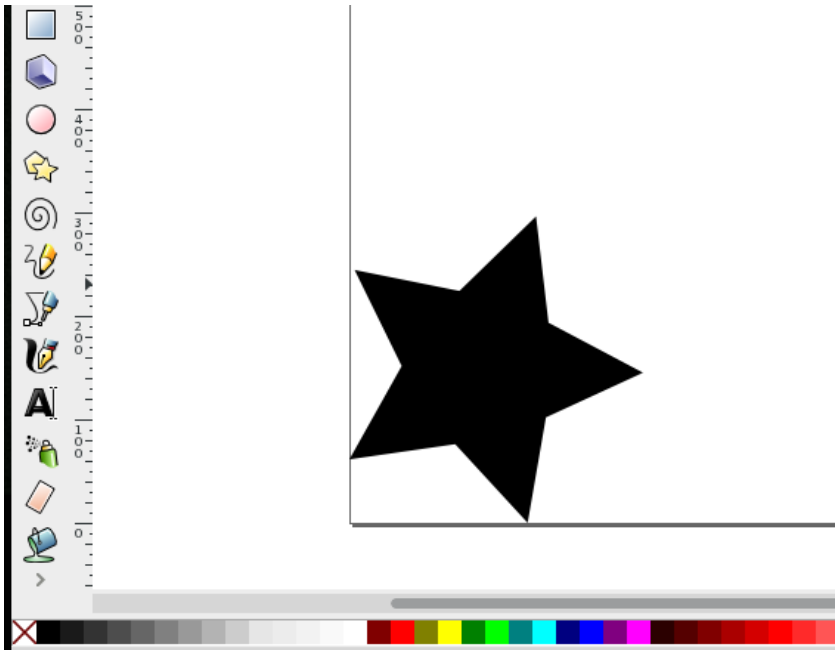
```
linear_extrude(100) {
    polygon(points=[[0,0],[100,0],[100,100],[0,100]]);
}
```



Diseñar en Inkscape e importar y extruir en openscad

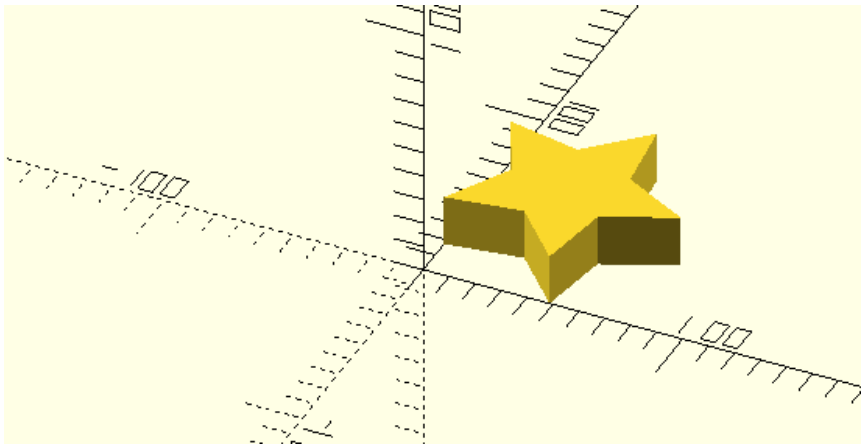
Ejemplo 012:

Primero lo diseñamos en Inkscape y lo exportamos como dxf.



En openscad lo importamos así:

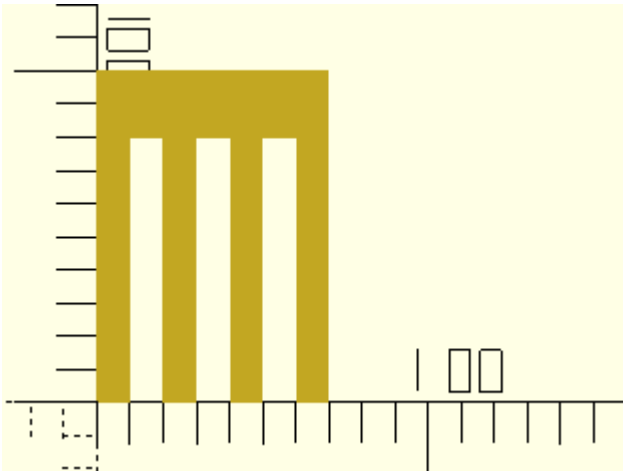
```
translate([0,0,0]){
    rotate([0,0,0]) {
        linear_extrude(height = 20) import("/home/fanta/dibujo.dxf");
    }
 }
```



Ejercicios prácticos

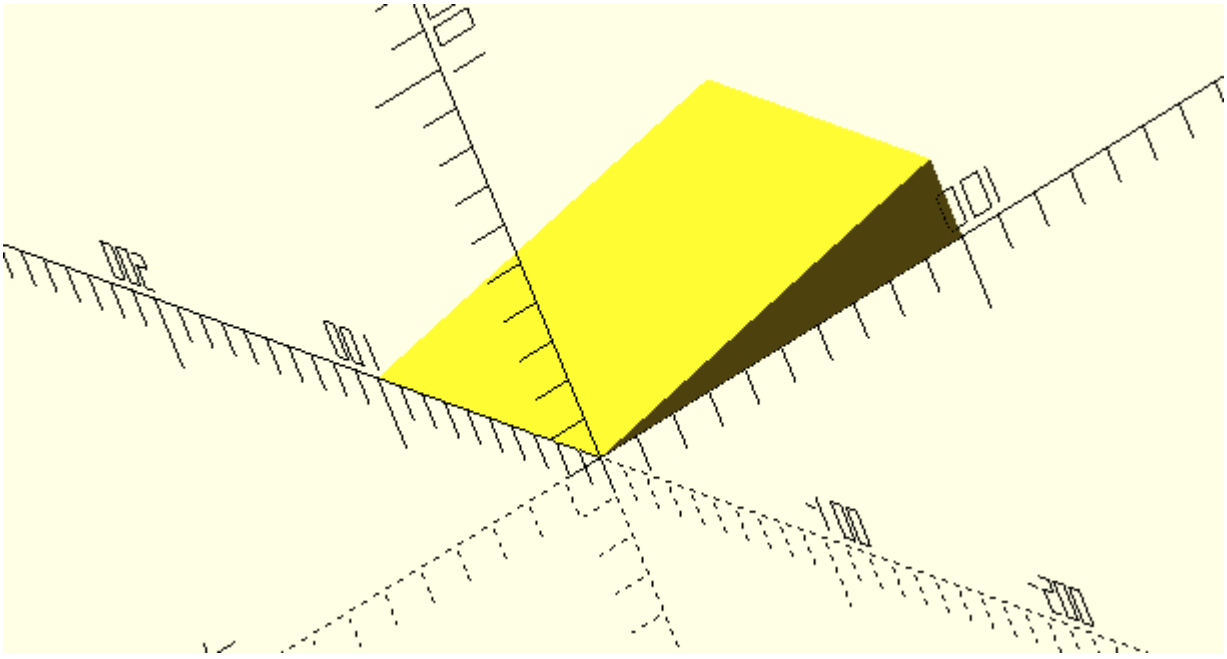
Ejercicio 1:

Realizar el siguiente dibujo usando polygon. Altura 100mm anchura 60mm. Cada pata 10mm



Ejercicio 2:

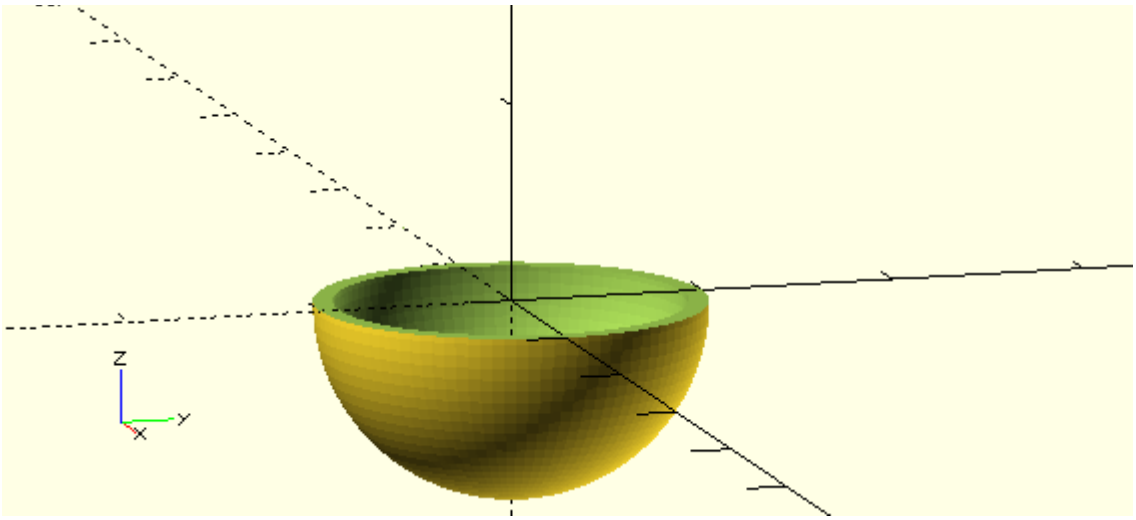
Realizar un calzador para puertas del tamaño que consideremos.

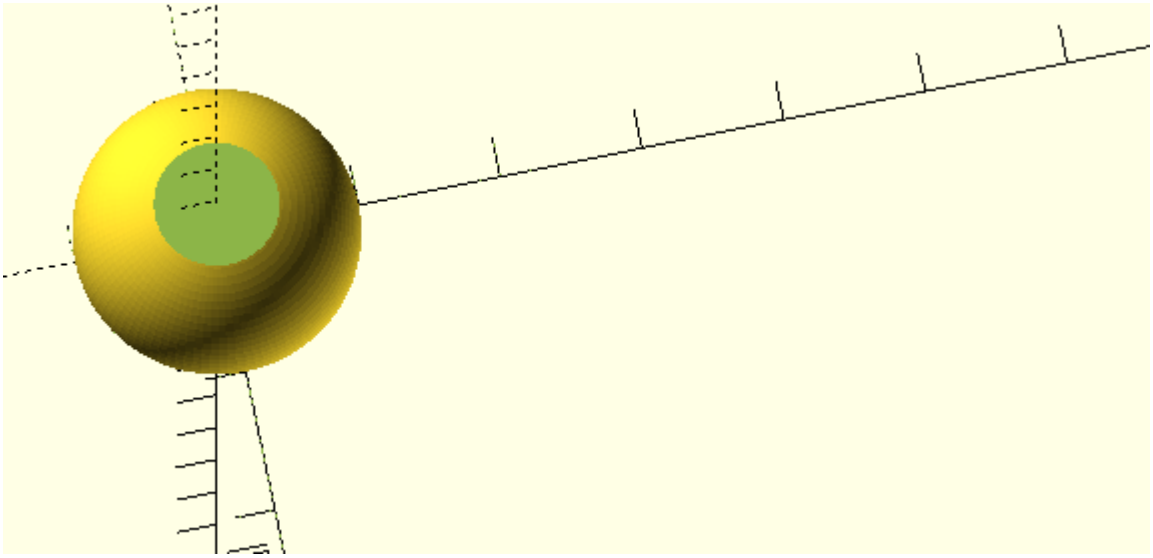


Ejercicio 3:

Realizar un cuenco hueco en forma de semiesfera que se sostenga sobre una mesa.

La parte de abajo ha de poder apoyar y no ha de salirse el liquido. El tamaño que se quiera.





Soluciones ejercicios prácticos

Ejercicio 1:

```
linear_extrude(10) {
    polygon(points=[[0,0],[10,0],[10,80],[20,80],[20,0],[30,0],[30,80],[40,80],
[40,0],[50,0],[50,80],[60,80],[60,0],[70,0],[70,100],[0,100]]);
}
```

Ejercicio 2:

```
linear_extrude(100) {
    polygon(points=[[0,0],[20,100],[0,100]]);
}
```

Ejercicio 3:

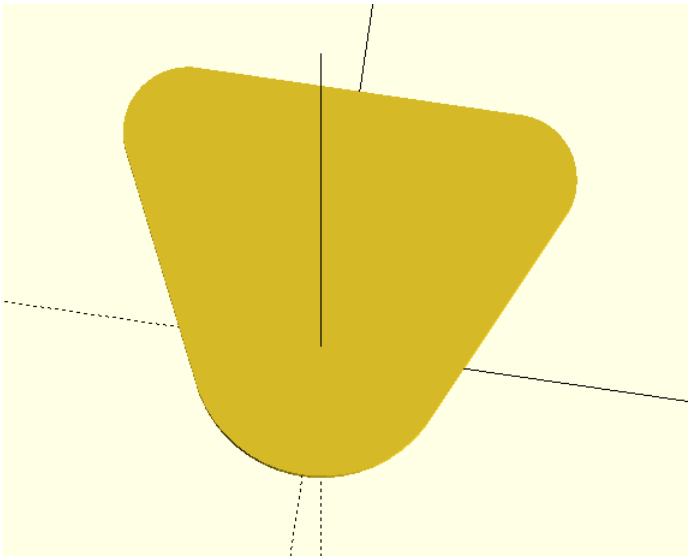
```
 difference(){
   difference(){
        difference(){
            sphere(d=20, $fn=100);
            sphere(d=18, $fn=100);
        }
        translate([-25,-25,0]){ cube([50,50,20]); }
    }
    translate([-25,-25,-29]){ cube([50,50,20]); }
}
```

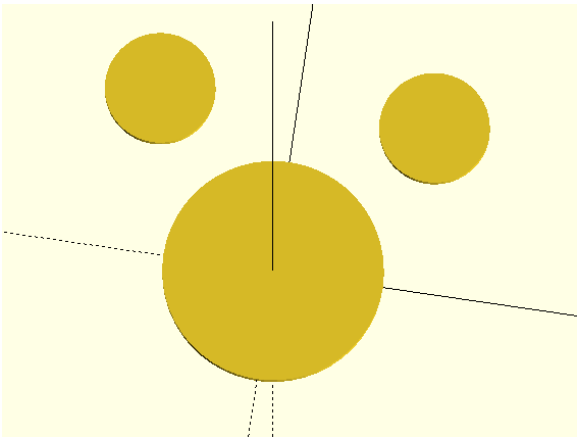
Hacer un HULL entre cilindros

Es posible disponer varios cilindros y que mediante la función hull se unan las tangentes entre estos.

Un ejemplo con 3 cilindros puede ser este:

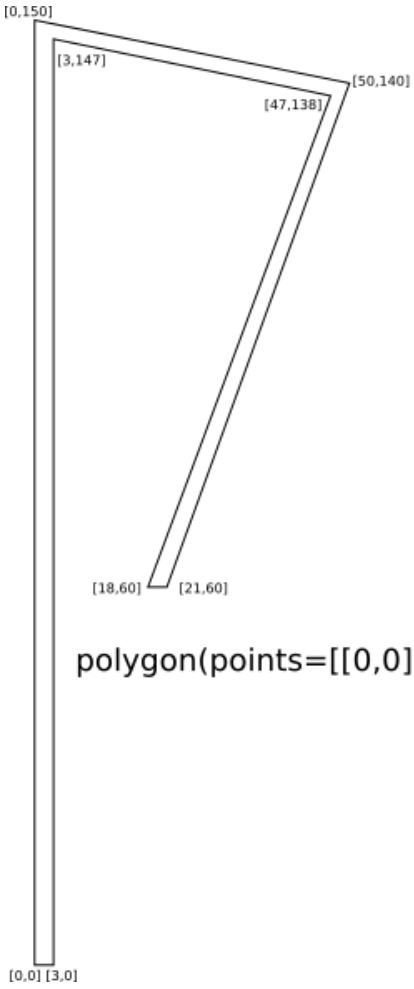
```
r1 =20;    // Radio cilindro 1
r2 = 10;   // Radio cilindro 2
r3 = 10;   // Radio cilindro 2
th = 2;    // Grosor

hull() {
    translate([0, 0, 0]){
          cylinder(r = r1, h = th, center = true, $fn = 50);
    }
    translate([25, 30, 0]){
        cylinder(r = r2, h = th, center = true, $fn = 50);
    }
    translate([-25, 30, 0]){
        cylinder(r = r3, h = th, center = true, $fn = 50);
    }
}
```





Diseñar un tapador de webcam para notebooks

Es posible hacerlo de muchas formas. Una posible es usando inkscape para sacar las coordenadas de los puntos que vamos a necesitar para generar un poligono extruible.




```
polygon(points=[[0,0],[0,150],[50,140],[21,60],[18,60],[47,138],[3,147],[3,0]]);
```

En openscad el código que se ve en esa imagen sirve pero no es parametrico. Para hacerlo parametrico hemos de currrarlo un poco:

```
altura=150;
anchura=40;
grosor=3;
extrusion=100;
extruir="no"; // valores posibles si o no

if (extruir=="no"){
    polygon(points=[
    [0,0],
    [0,altura],
    [anchura,altura-(altura*15/100)],
    [anchura/2,altura/2],
    [(anchura/2)-grosor,(altura/2)],
```

```
    [(anchura)-grosor,((altura-(altura*15/100))-grosor)+1],
    [grosor,(altura-grosor)-1],
    [grosor,0]
    ]);
}else{
    linear_extrude(extrusion){
        polygon(points=[
        [0,0],
        [0,altura],
        [anchura,altura-(altura*15/100)],
        [anchura/2,altura/2],
        [(anchura/2)-grosor,(altura/2)],
        [(anchura)-grosor,((altura-(altura*15/100))-grosor)+1],
        [grosor,(altura-grosor)-1],
        [grosor,0]
        ]);
    }
}
```