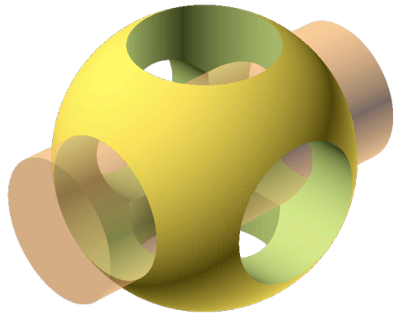


Creación de árbol mediante OpenSCAD



OpenScad

Profesora: [Nancy Hitschfeld K.](#)

Profesor Auxiliar: [Michel Llorens A.](#)

[Sebastián González M.](#)

Alumno: Jorge Gutiérrez

Fecha: 6 de Diciembre de 2015

Objetivos

- Uso de OpenSCAD y operaciones propias de un árbol CGS
- Modelar un árbol
- Modelar un bosque
- Generar un versión imprimible

Diseño

- ¿qué hay de un árbol real?



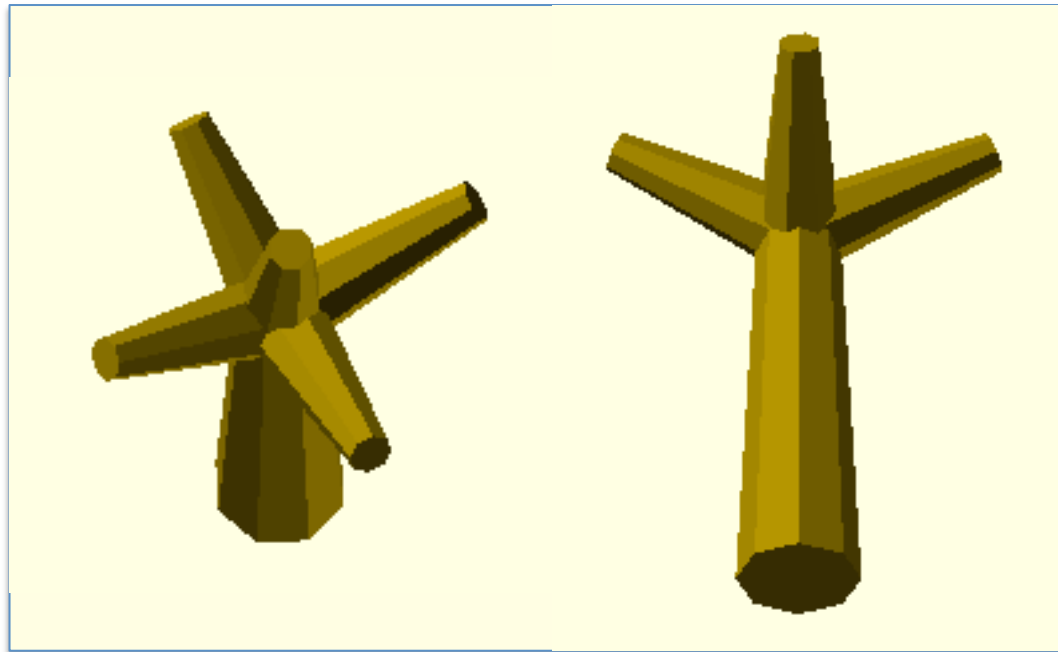
Diseño

- ¿qué hay de un árbol real?
- ¿qué cosas en común hay en su forma?



Diseño

- ¿qué hay de un árbol real?
- ¿qué cosas en común hay en su forma?
- Eureka!!



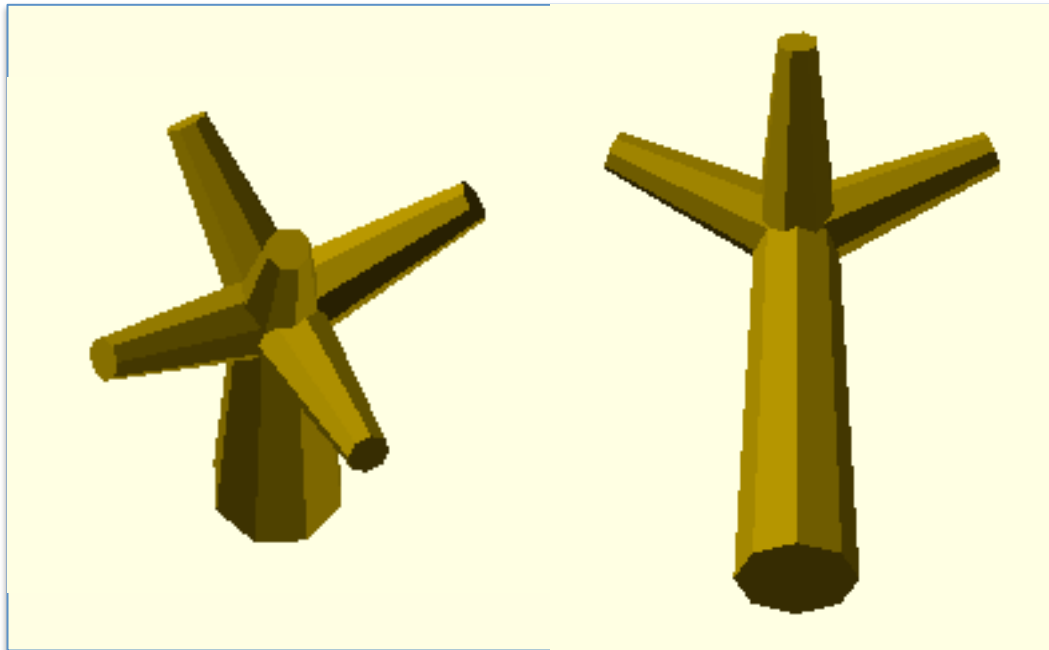
Implementación

- Lo mas importante es generar una rama...

```
color([0.6,0.5,0.1])  
rotate([180,0,0]) cylinder(r1=0.1*n, r2=0.1*(n  
+1), h=length, $fn=8);
```

Implementación

- Lo mas importante es generar una rama...



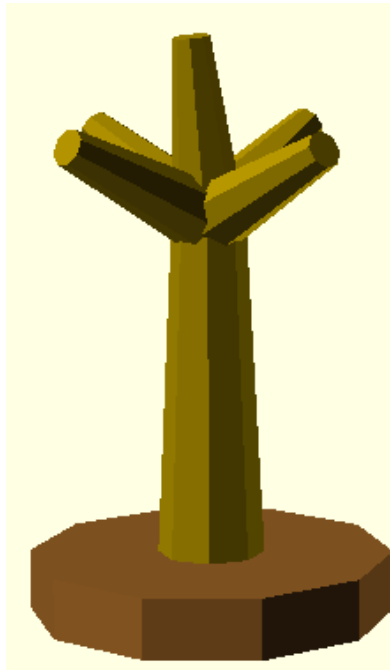
Implementación

- Lo mas importante es generar una rama...
Luego el arbol es solo muchas de ellas en una forma caracteristica a las ramificaciones.

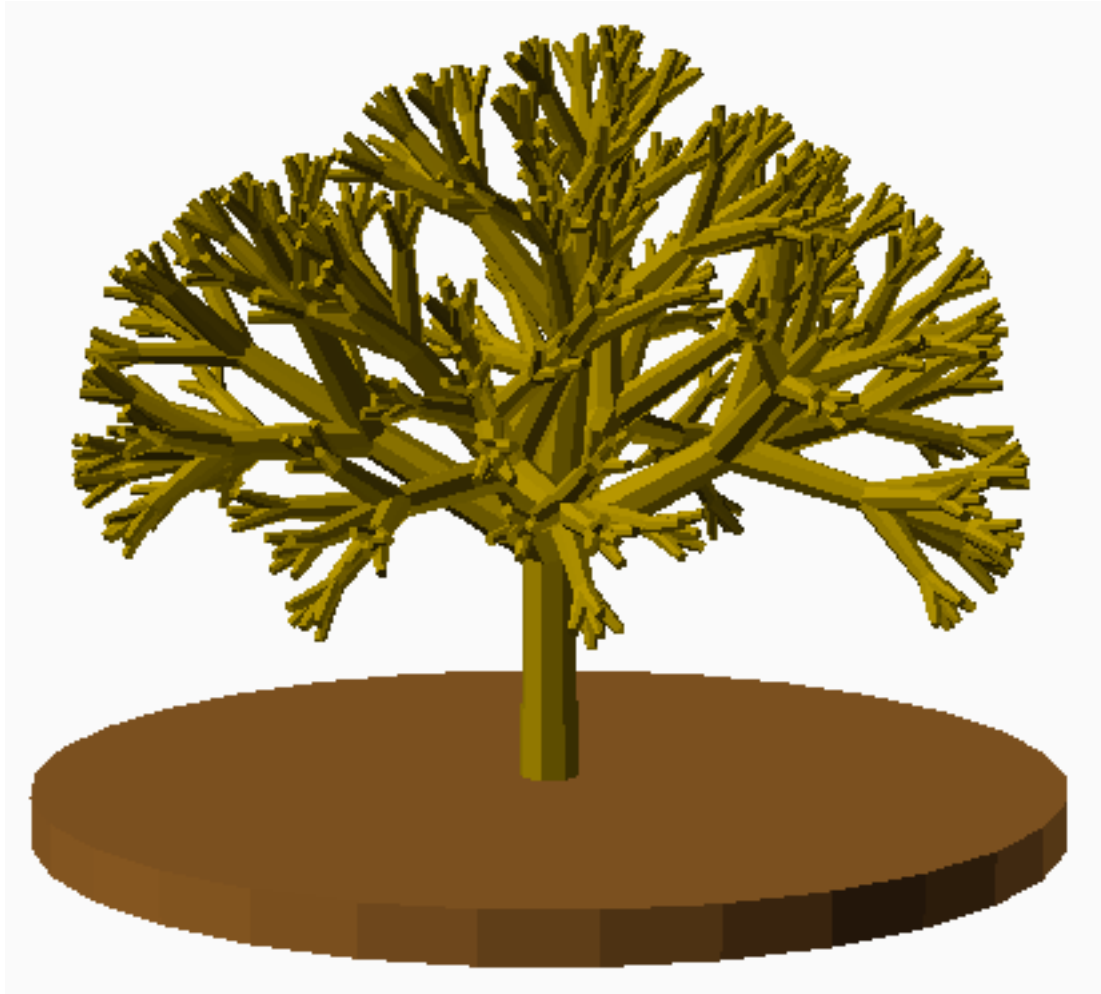
```
color([0.6,0.5,0.1])rotate([180,0,0]) cylinder(r1=0.1*n,  
r2=0.1*(n+1), h=length, $fn=8);  
rotate([angle*random[0]/5, 0, 0]) tree(n-1, ka*angle, ka,  
seeds[0]);  
rotate([angle, 0, 0+angle*random[2]/10]) tree(n-1,  
ka*angle, ka, seeds[1]);  
rotate([angle, 0, 90+angle*random[3]/10]) tree(n-1,  
ka*angle, ka, seeds[2]);  
rotate([angle, 0, 180+angle*random[4]/10]) tree(n-1,  
ka*angle, ka, seeds[3]);  
rotate([angle, 0, 270+angle*random[5]/10]) tree(n-1,  
ka*angle, ka, seeds[4]);
```


Implementación

- Lo mas importante es generar una rama...
Luego el arbol es solo muchas de ellas en una
forma caracteristica a las ramificaciones.



Implementación

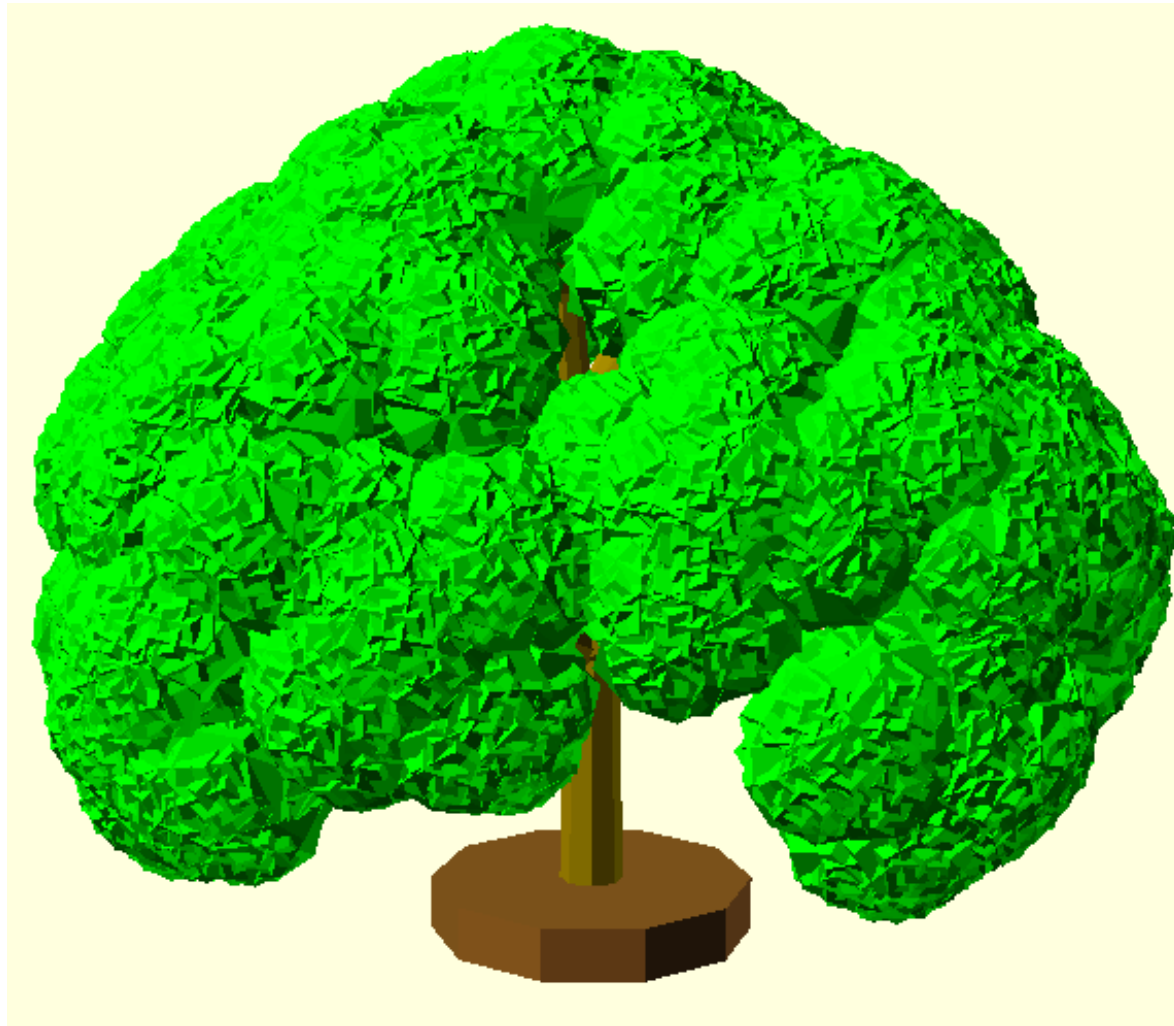


Implementación

- Lo mas importante es generar una rama...
Luego el arbol es solo muchas de ellas en una
forma caracteristica a las ramificaciones... Y
seria bonito que tuviera hojas tambien

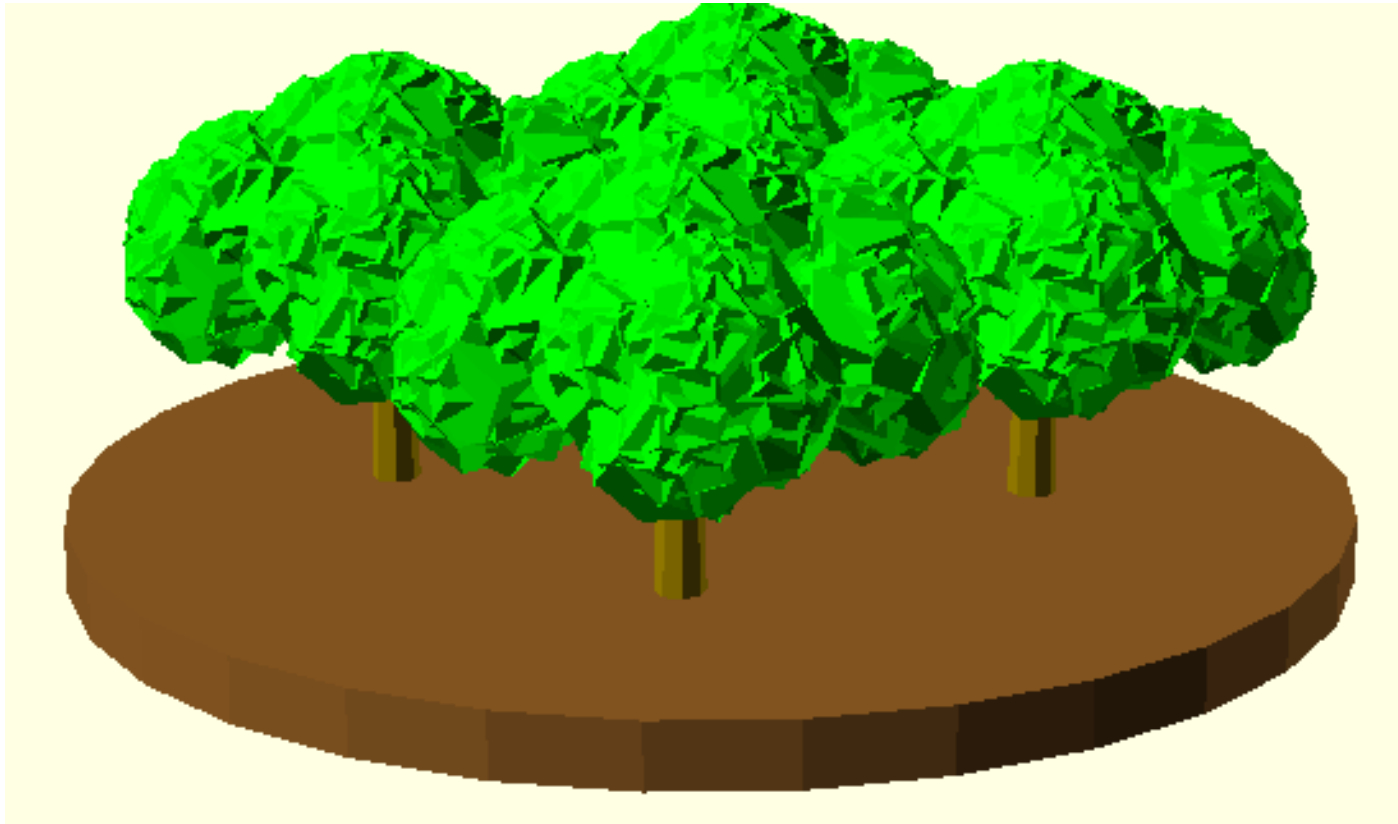


Implementación



Implementación

- Y ahora simplemente un bosque son varios arboles...



Código

- Hablemos de lo que realmente NO importa

```
module tree(n, angle, ka=1/2, seed=rands(-1000000,+1000000,1)) {  
  if (n>0) translate([0,0,length]) rotate([0,0,random[0]*180]) {  
    color([0.6,0.5,0.1]) rotate([180,0,0]) cylinder(r1=0.1*n,  
r2=0.1*(n+1), h=length, $fn=8);  
    rotate([angle*rs[0]/5, 0, 0]) tree(n-1, ka*angle, ka, seeds[0]);  
    rotate([angle, 0, 0+angle*random[2]/10]) tree(n-1, ka*angle,  
ka, seeds[1]);  
    rotate([angle, 0, 90+angle*random[3]/10]) tree(n-1, ka*angle,  
ka, seeds[2]);  
    rotate([angle, 0, 180+angle*random[4]/10]) tree(n-1, ka*angle,  
ka, seeds[3]);  
    rotate([angle, 0, 270+angle*random[5]/10]) tree(n-1, ka*angle,  
ka, seeds[4]); }  
  else {  
    color([0,1,0]) sphere(1.5);  
  }  
}
```

Código

- Hablemos de lo que realmente NO importa

```
module forest() {  
  translate([0,5,0]) tree(3, 55, ka=0.75, seed=2);  
  translate([0,-5,0]) tree(3, 55, ka=0.75, seed=2);  
  translate([5,0,0]) tree(3, 55, ka=0.75, seed=2);  
  translate([-5,0,0]) tree(3, 55, ka=0.75, seed=2);  
}
```

Resultados y Observaciones

- Arboles mediante recursividad son fáciles y buenos
- El uso de fractales puede ayudar



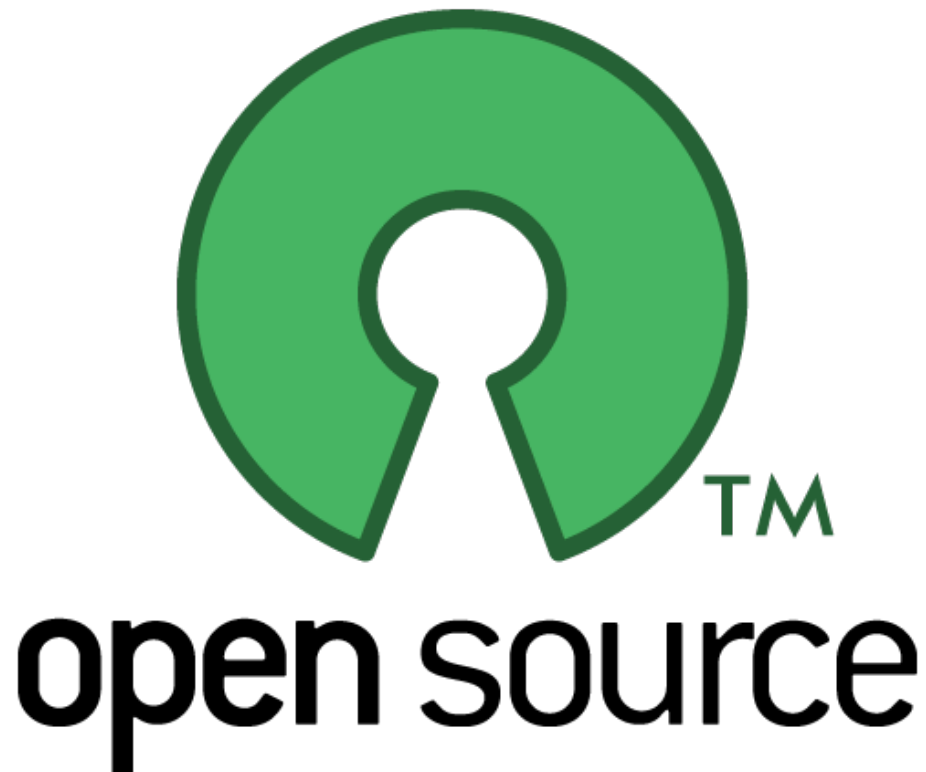
Resultados y Observaciones

- Tiempos y uso de CPU para reenderizado



Resultados y Observaciones

- OpenSCAD muy amigable con el programador



**KEEP
CALM
AND PUT ON
A HAPPY
FACE**