**Trabajo 3 : Lámpara Nocturna**

**Asignatura : Bitbloq y Arduino**

**Profesor : Alberto Valero**

**Curso : Experto Universitario en Robótica, Programación e Impresión 3D**

**Universidad UNIR**

**Fecha : 9 de diciembre de 2016**

**1 DESCRIPCIÓN**

La actividad consiste en realizar una lámpara nocturna que coste de los siguientes componentes :

* Led indicativo de lámpara encendida o apagada. Lo llamaremos PILOTO.
* Led iluminador . Lo llamaremos LUZ.
* Sensor de luz
* Pulsador

El funcionamiento debe ser el siguiente :

* Al pulsar el botón se activa ( se enciende el PILOTO para indicar que la lámpara está en funcionamiento ).
* Al volver a pulsar el botón se desactiva ( se apaga el PILOTO para indicar que la lámpara no está funcionando ).
* Al hacerse de noche ( poca luz ambiente ) se enciende la LUZ durante 10 segundos . A los 10 segundos se apaga y no vuelve a encender hasta que no se realiza un ciclo completo de día > noche ( es decir , queda inoperativa hasta que se vuelve a hacer de noche ) .

**2 ELEMENTOS UTILIZADOS EN LA ACTIVIDAD**

**Material :**

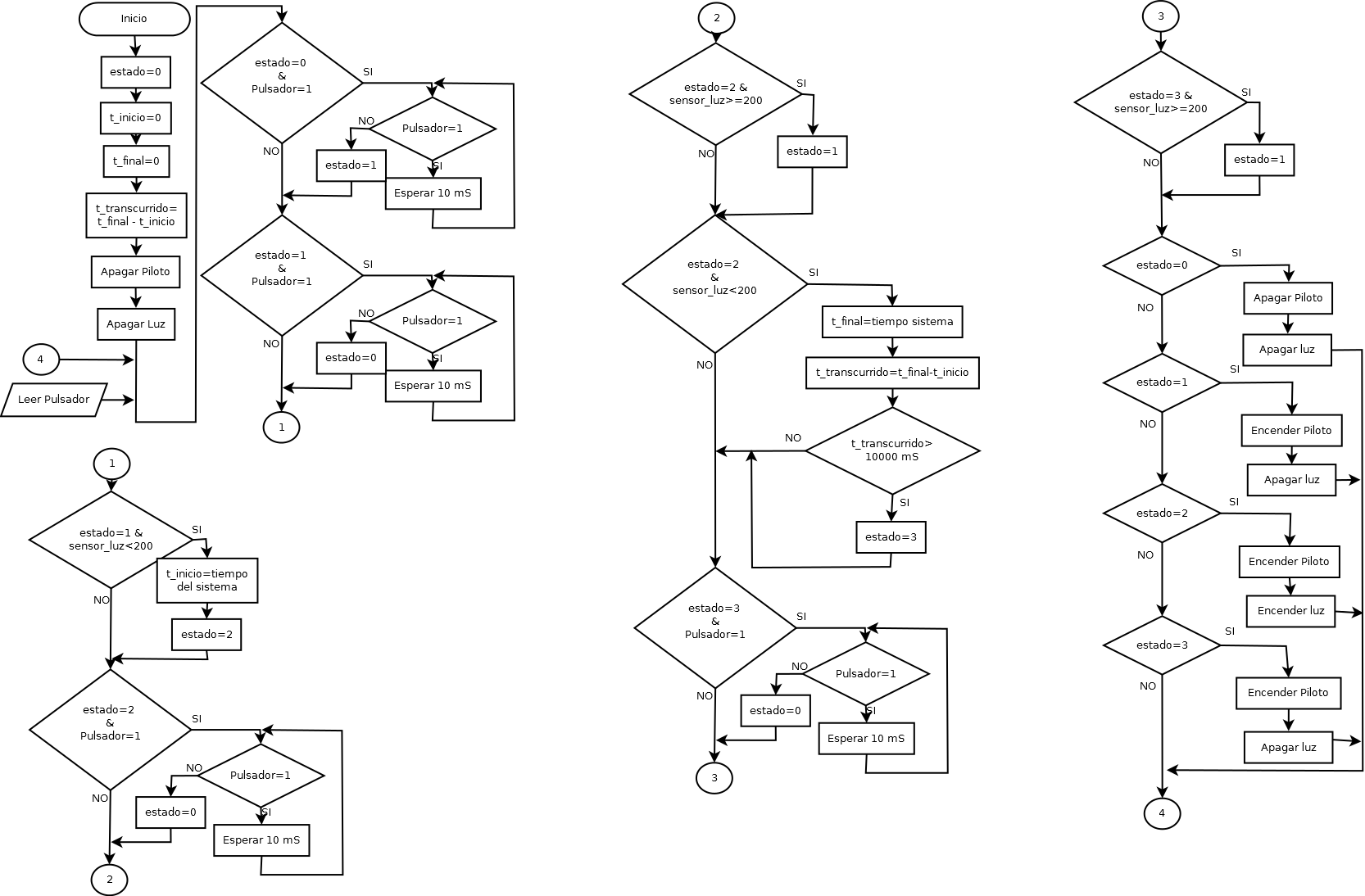
* 1 Placa Bq Zum Core
* 1 Portapilas
* 1 Zum bloq Pulsador
* 2 Zum bloq LED , verde y azul
* 1 Zum bloq Sensor de Luz
* 8 tornillos M3x10 mm rosca madera
* 2 tornillos M3x12 mm + 2 tuercas M3
* Pegamento , sierra , atornilladores plano y de estrella
* Piezas impresas 3D con PLA en color naranja para la lámpara y negro para la caja porta electrónica y pilas.

**Sofware :**

* Bitbloq para realizar la programación del semáforo
* OpenScad para diseñar las piezas impresas
* Cura 15.04.6 para convertir a gcode
* Dia para realizar el diagrama de flujo

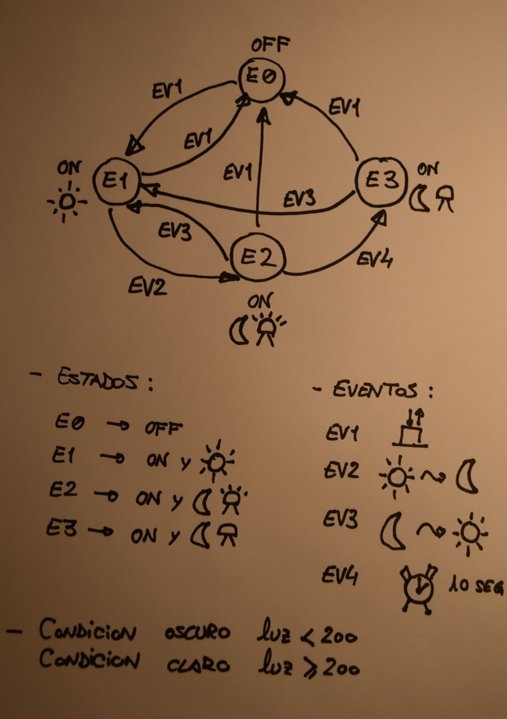
**3 DIAGRAMA DE FLUJO DEL PROGRAMA**

**TRABAJO 3 : LAMPARA NOCTURNA**



**3.1 EXPLICACION DEL DIAGRAMA DE FLUJO DEL PROGRAMA**

Este programa lo he enfocado como si fuese una **máquina de estados** , cuyos estados y eventos se recogen en la siguiente imagen :

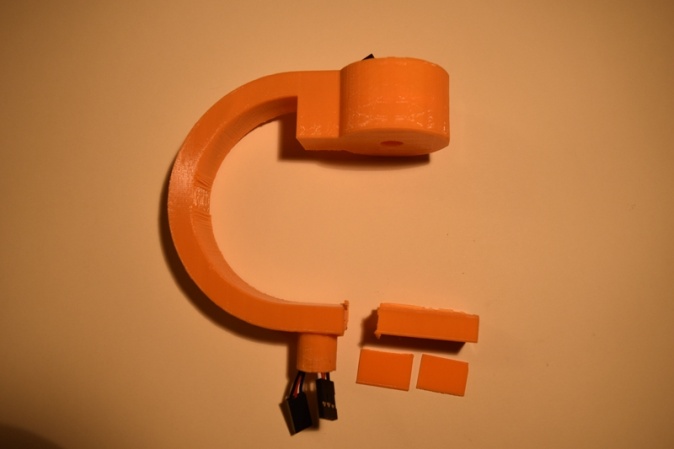


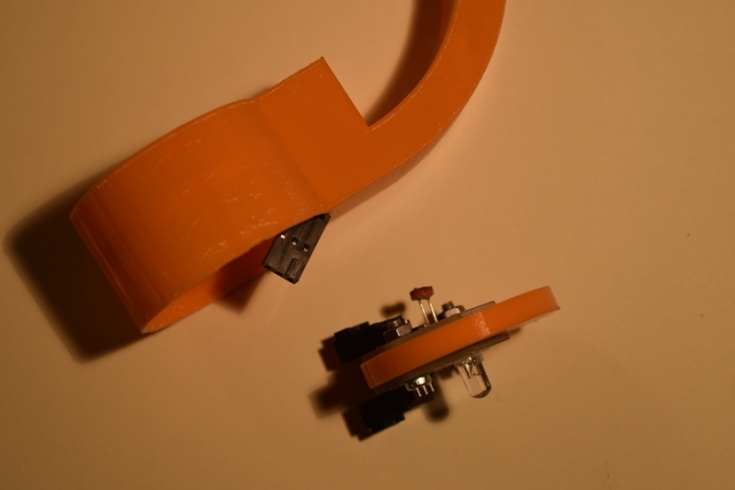
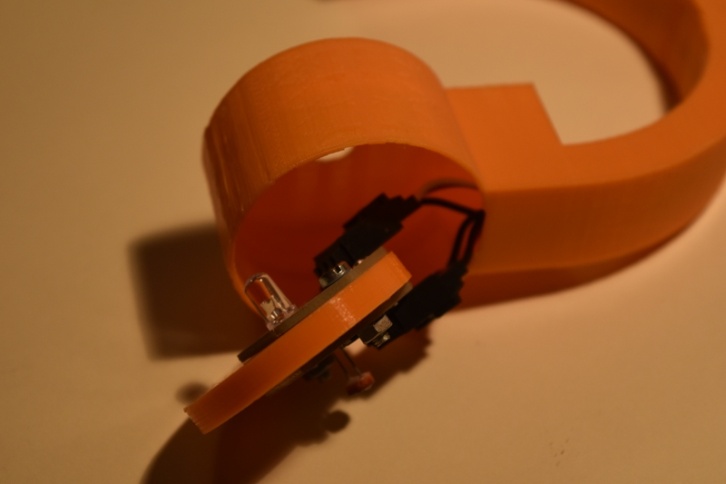
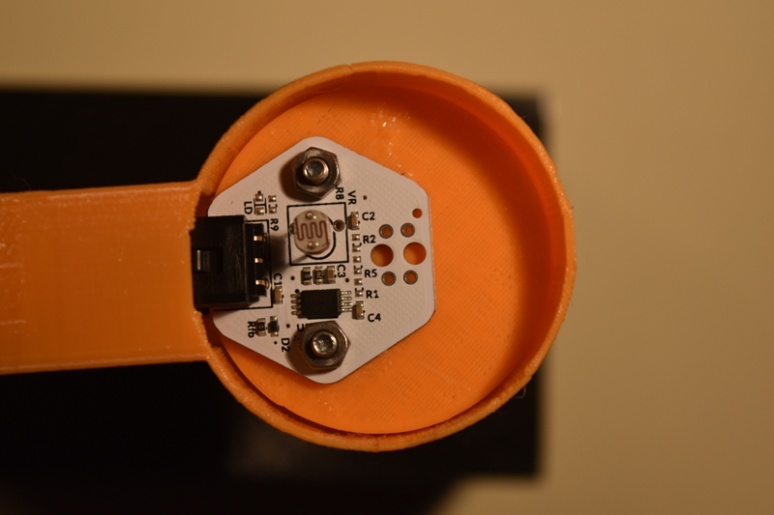
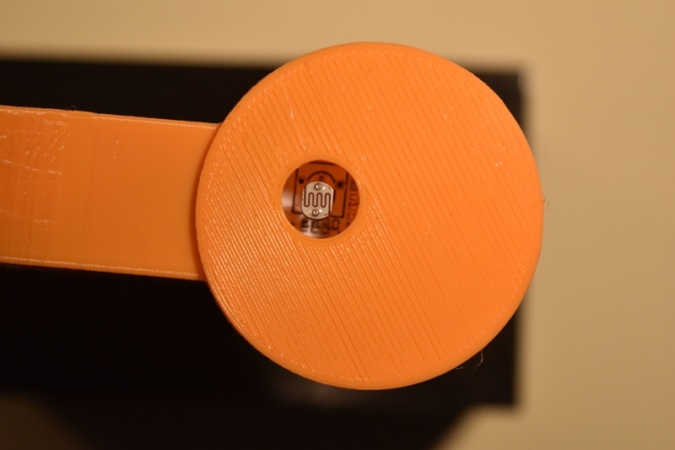
El programa comienza definiendo una variable global llamada “estado “ , que comienzo asignándole el valor 0 , y que recogerá la situación por la que vaya atravesando mi máquina de estados . Defino después tres variables denominadas t\_inicio , t\_final y t\_transcurrido , de las cuales las dos primeras van a ir recogiendo la situación en determinados momentos del valor del reloj interno del sistema , y la última si diferencia para saber el tiempo trancurrido entre dos eventos determinados y poder controlar los 10 segundos que tiene que estar la lámpara encendida.

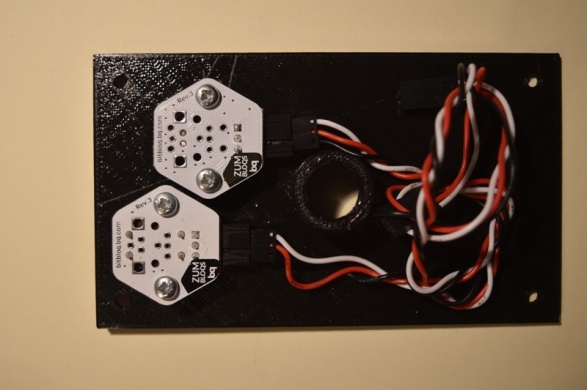
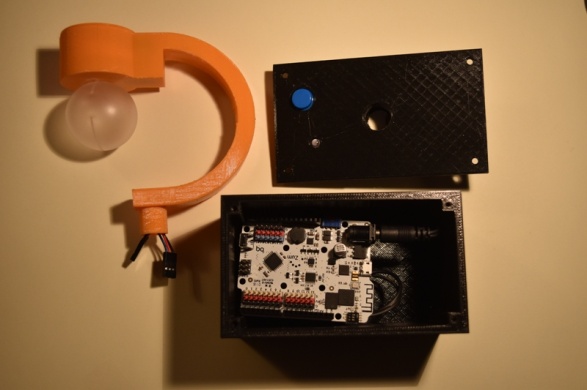
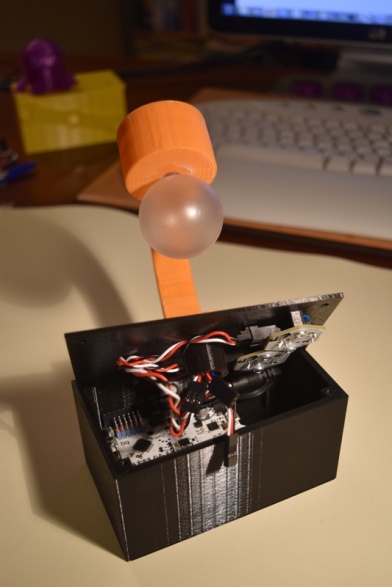
A continuación se lee el estado del pulsador y en función de ese valor “verdadero “ o “ falso “ en que se encuentre **el pulsador** , de la lectura que se recoja del **sensor\_de\_luz** ( corte establecido en 200 unidades ) y del **estado** en que se encuentre en ese momento mi sistema ( 0:3 ) , el programa va atravesando una serie de condicionales que primero establecen la **LÓGICA** de cambio de entre estados , y luego le indican que **ACCIONES** , dependiendo del estado en que se encuentre , tiene que realizar : encender / apagar los leds denominados **Luz** y **Piloto** , que inicialmente se apagan para evitar encendidos accidentales .

**4 SECUENCIA DE FOTOS DE LA REALIZACIÓN PRÁCTICA DE LA ACTIVIDAD**

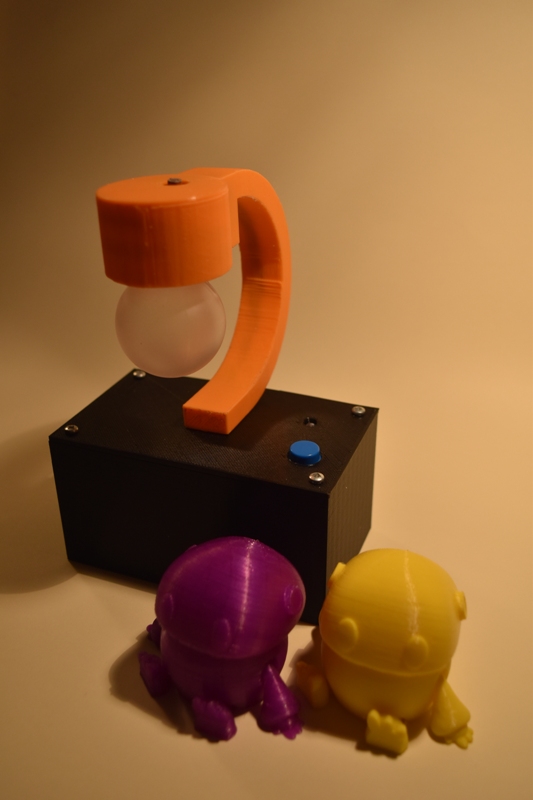
****

****

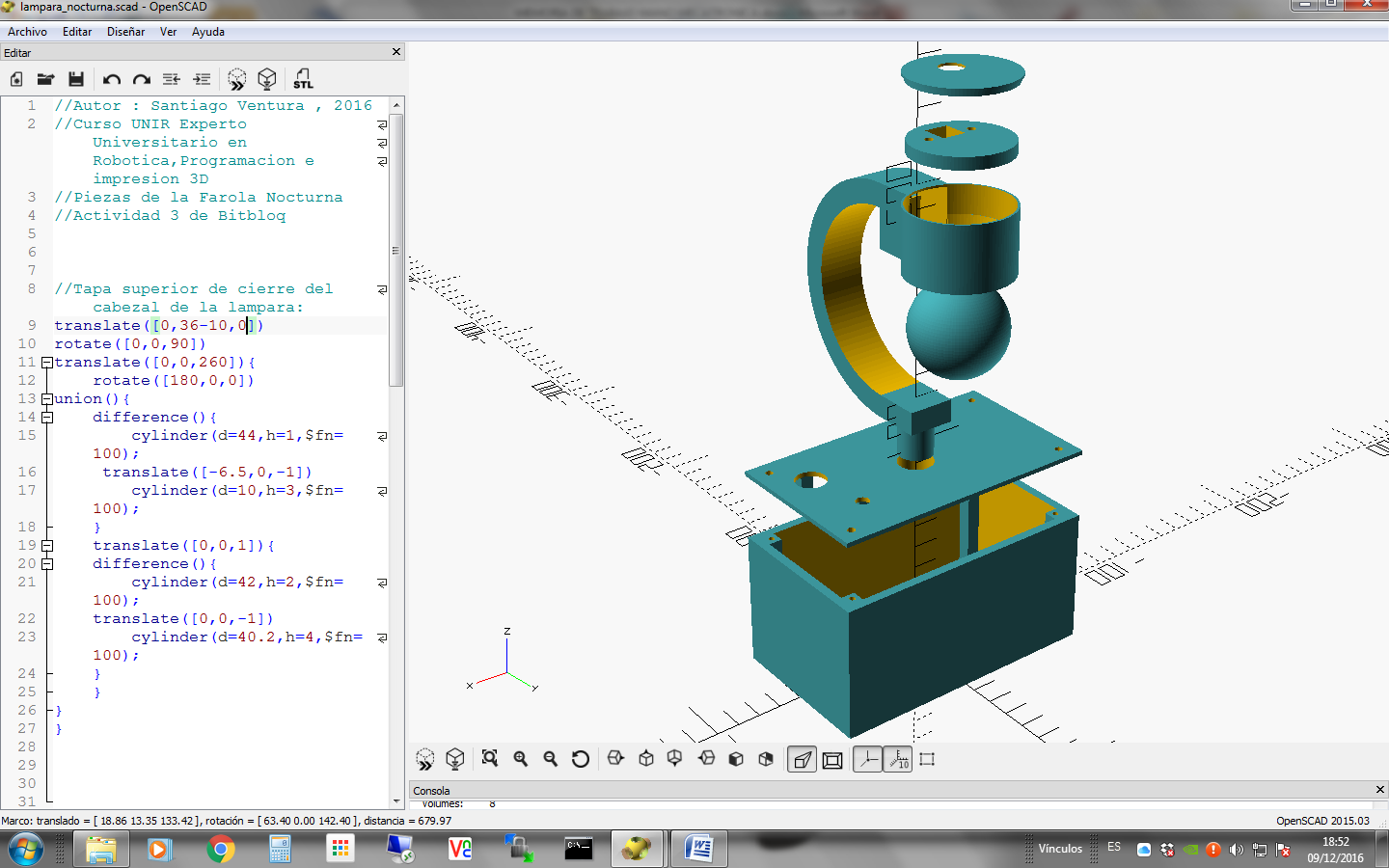
****

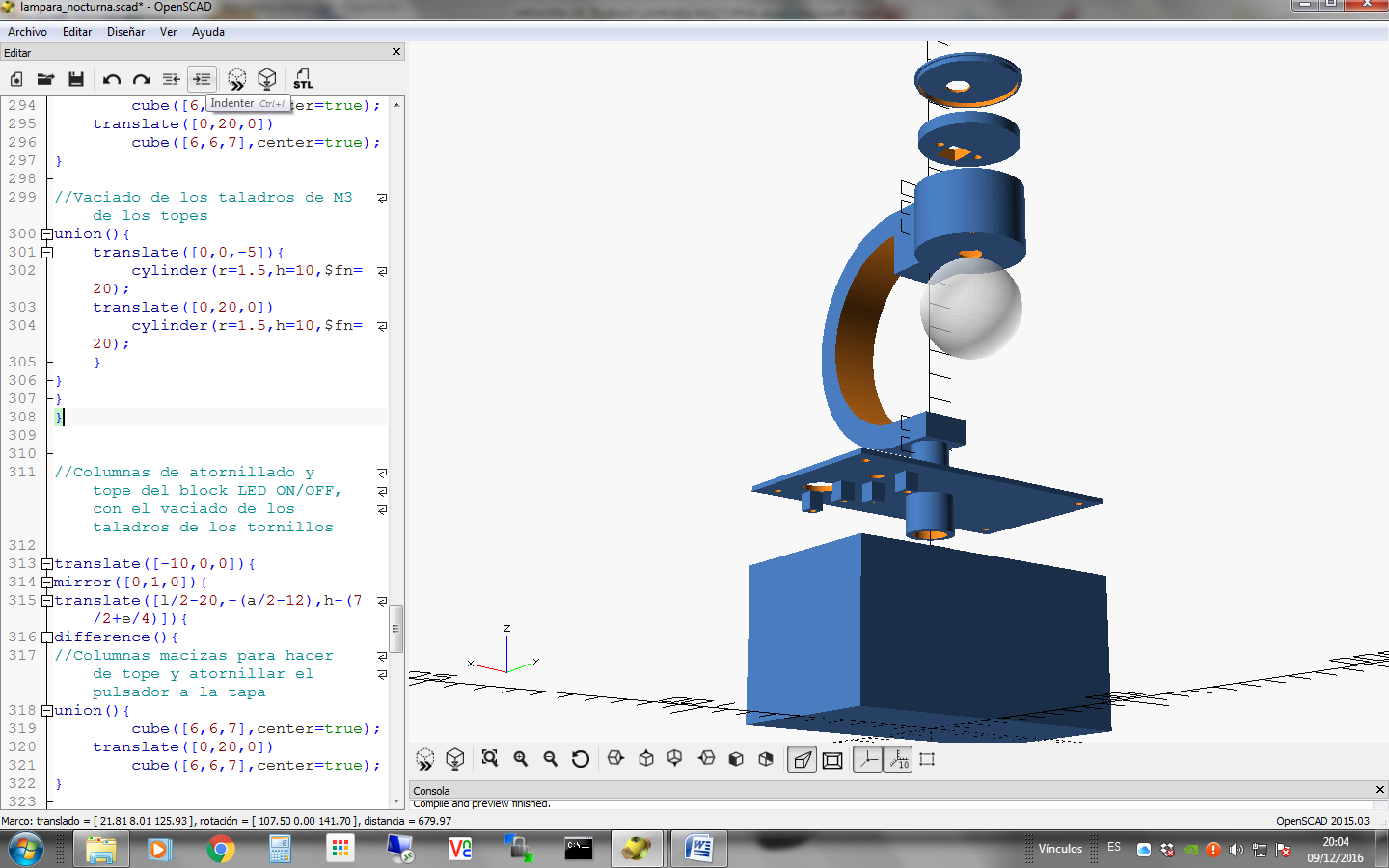
****

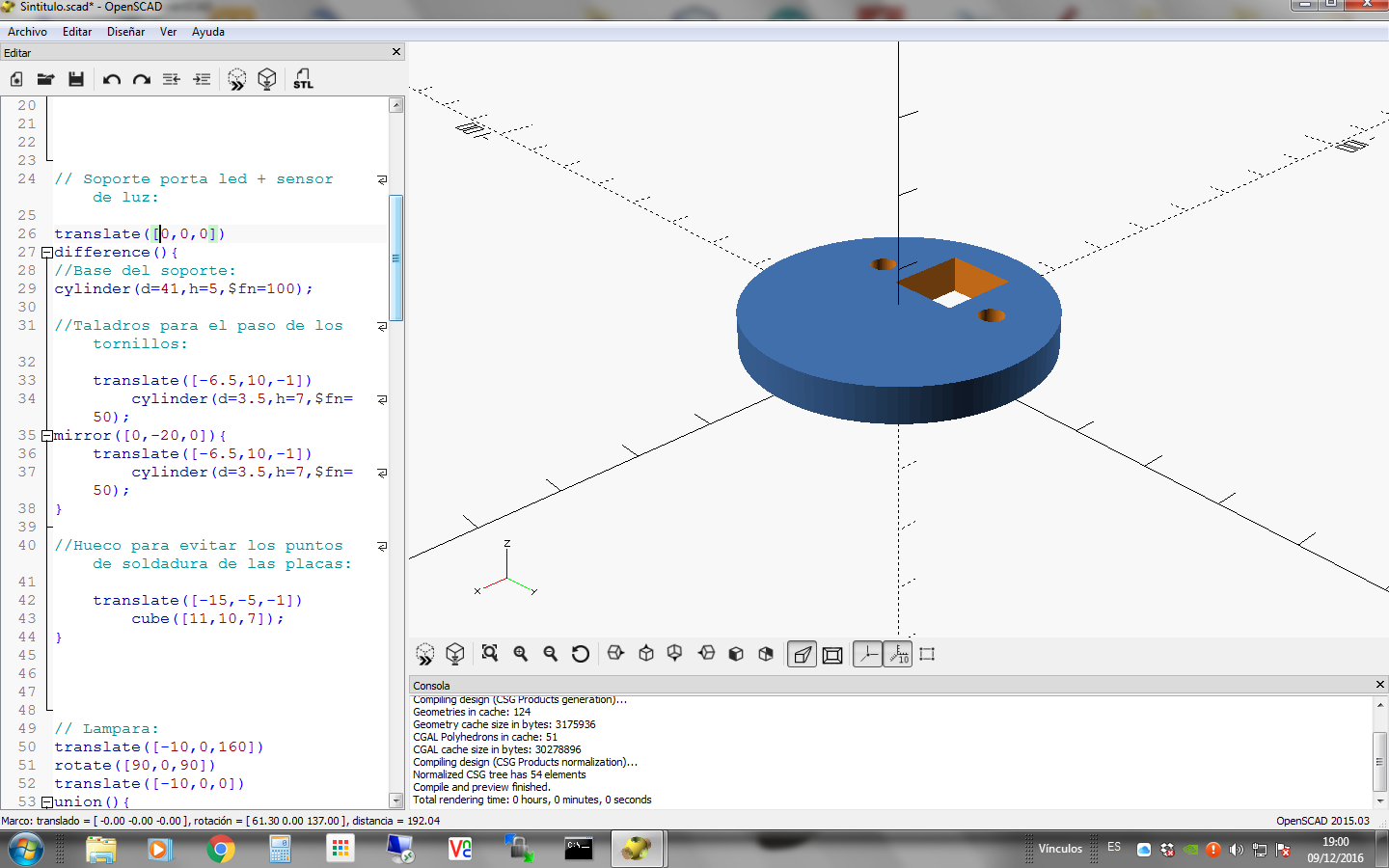
****

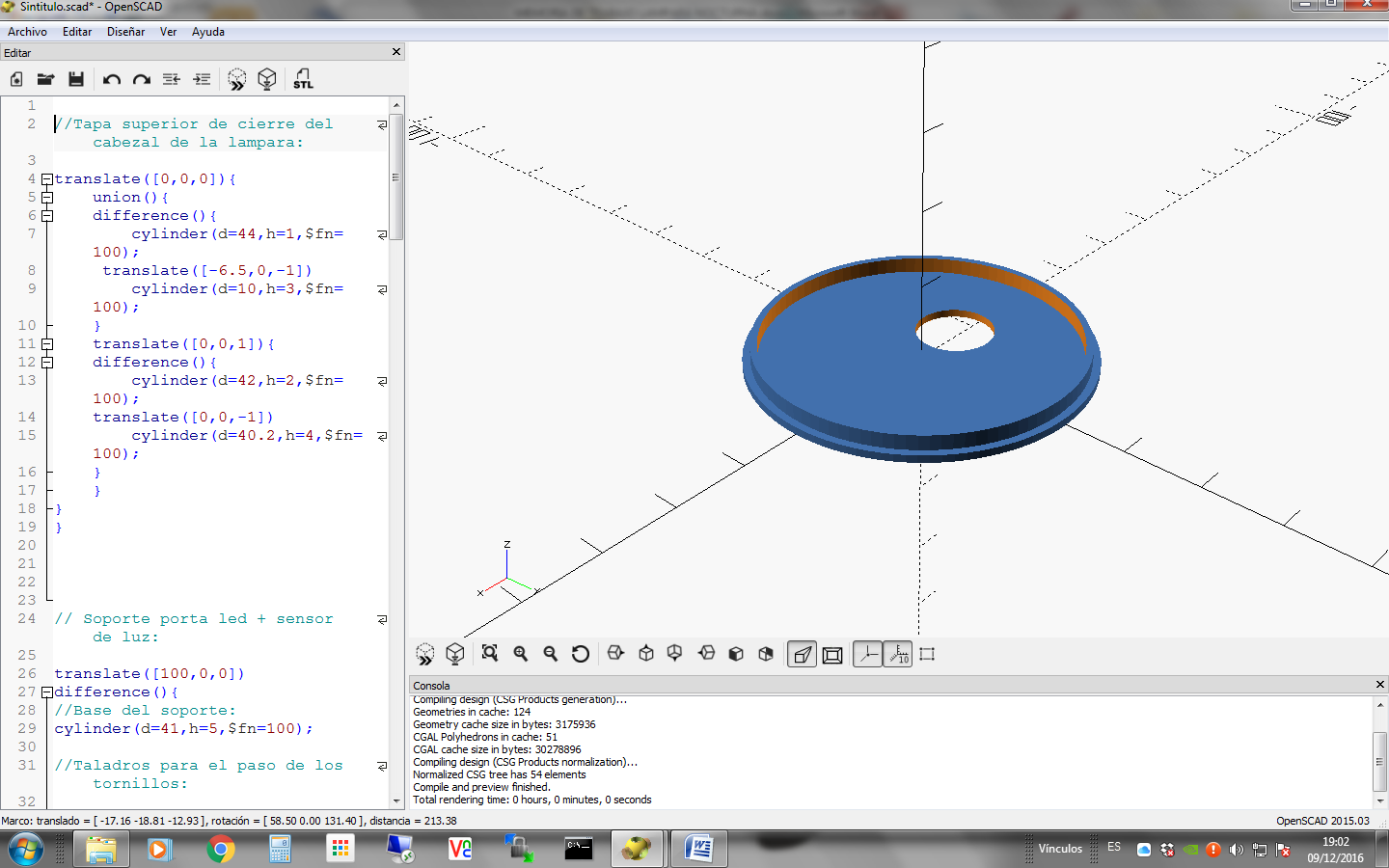
****

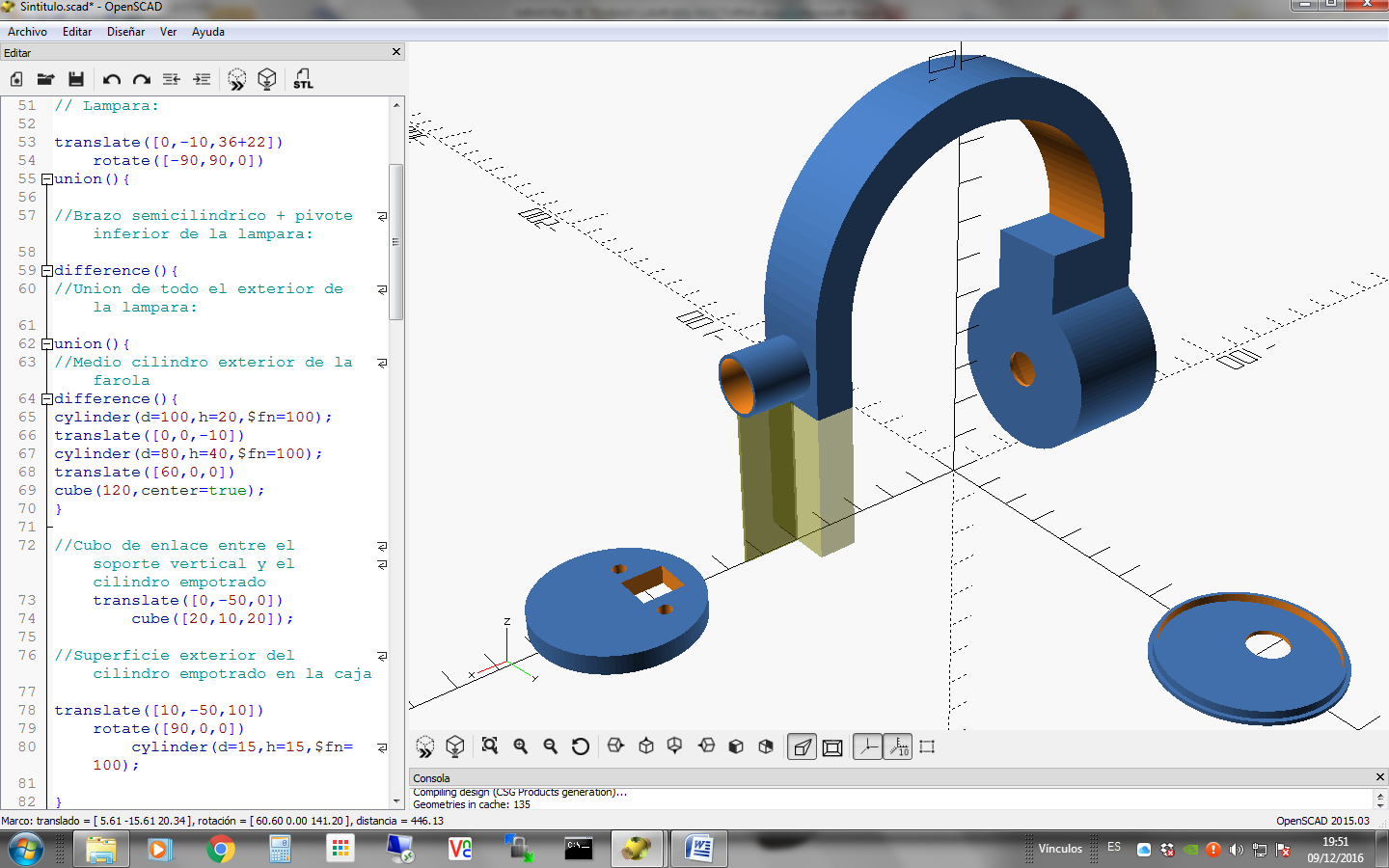
**5. ALGUNOS DETALLES DE LAS PIEZA DISEÑADAS EN OPENSCAD Y LOS ARCHIVOS GENERADOS PARA PODER IMPRIMIR EN MI IMPRESORA 3D , CON DETALLE DE LOS SOOPRTES AÑADIDOS AL ARCO EN VOLADIZO**

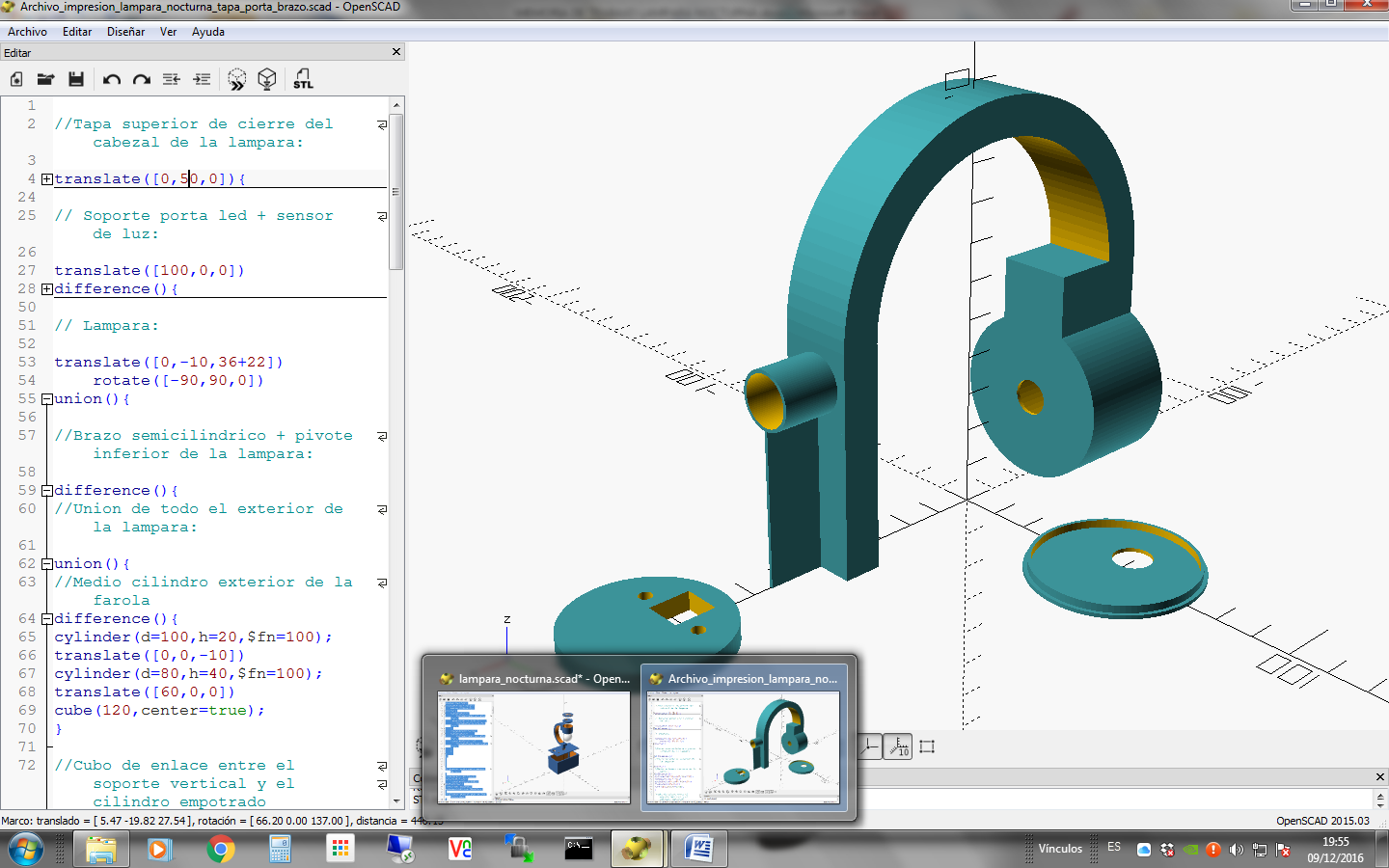
****

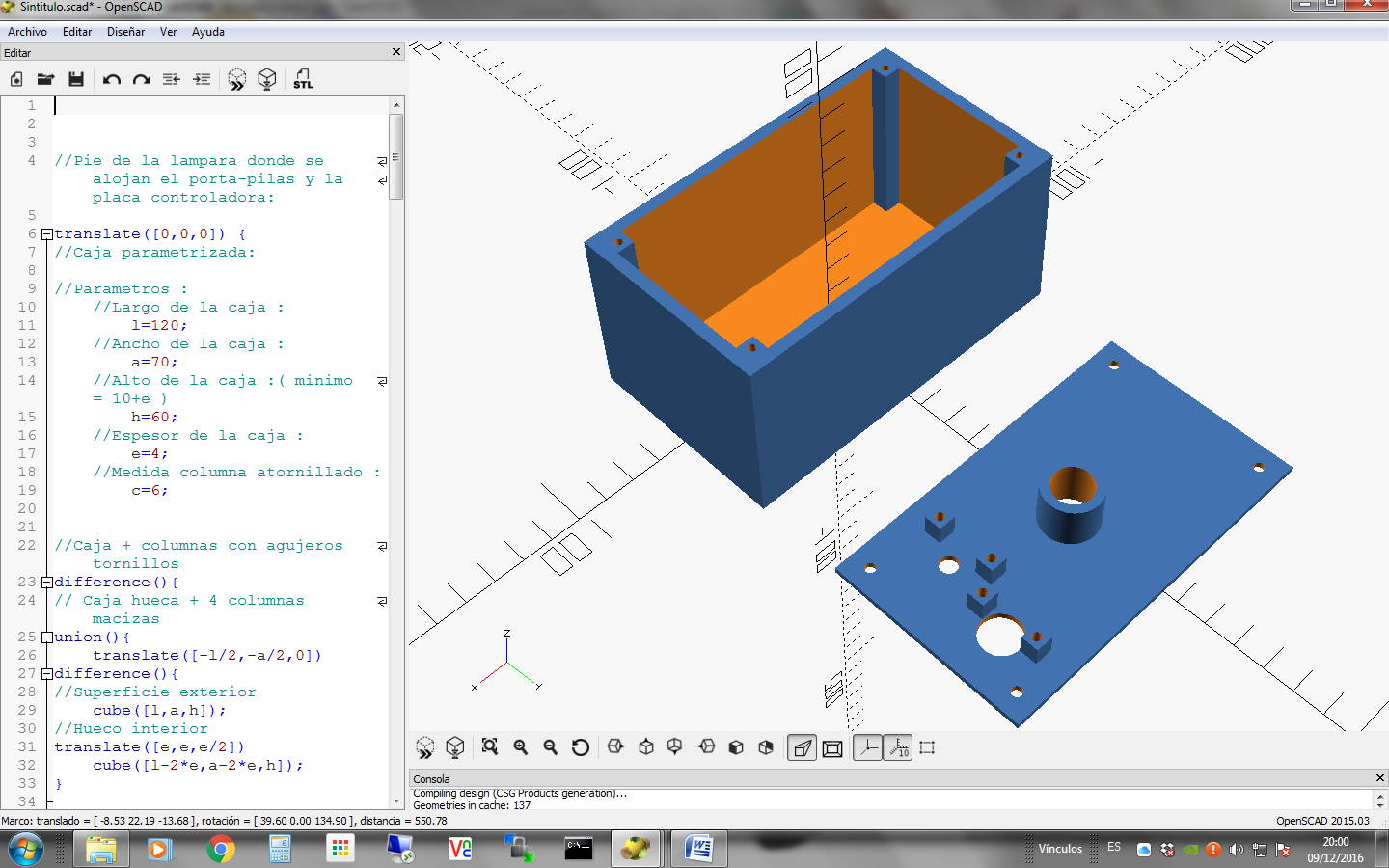
****

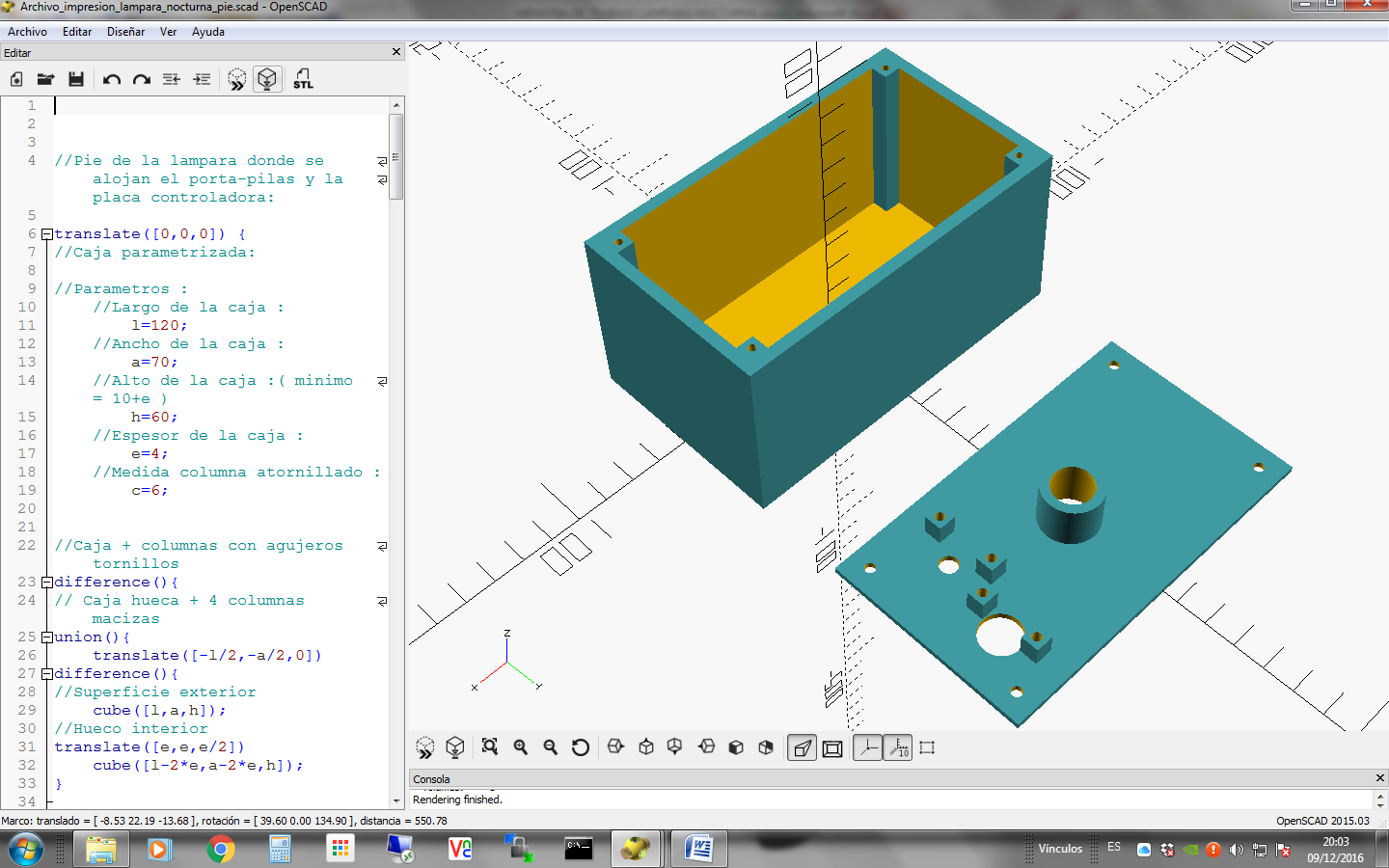
****

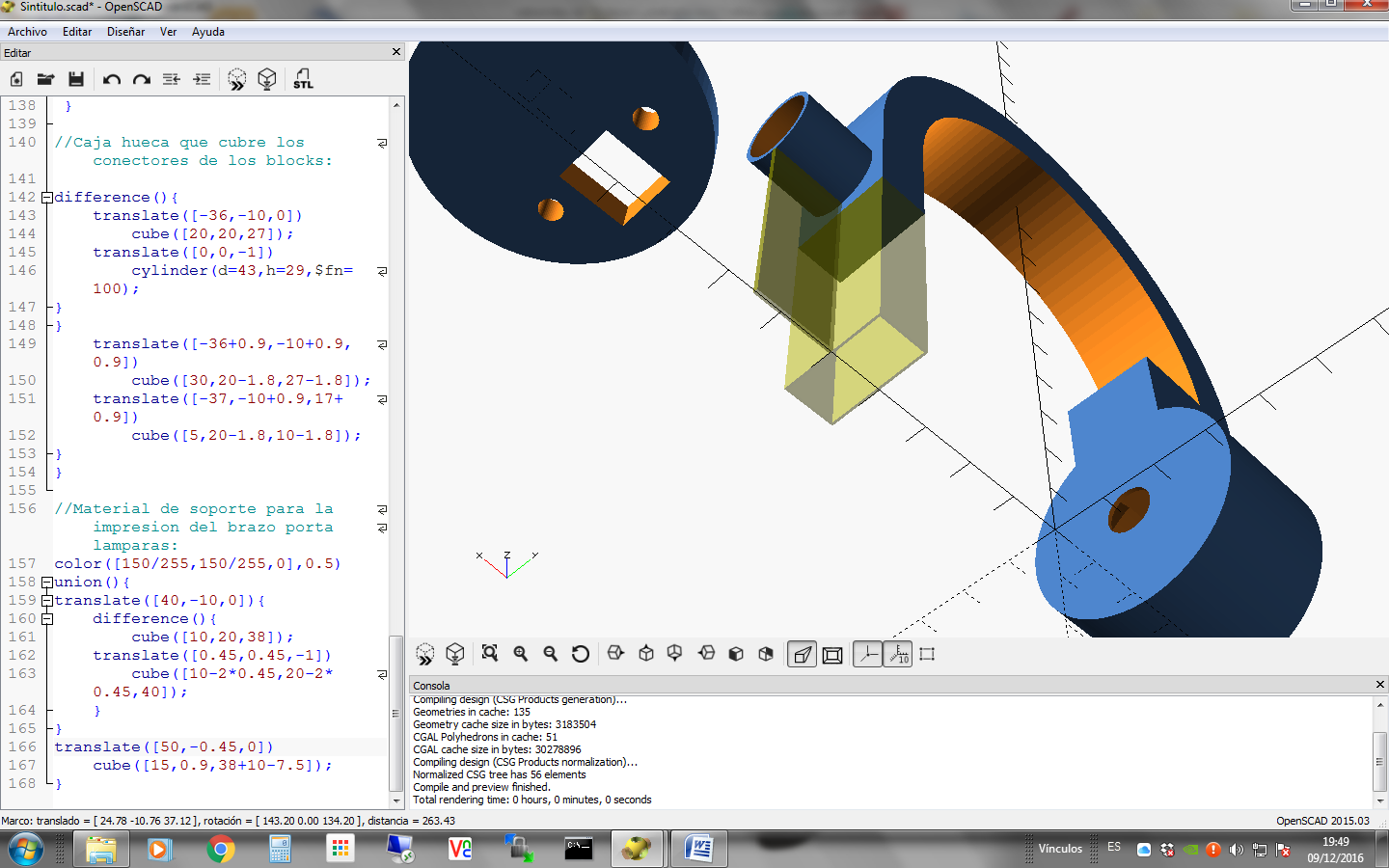
****

****

****

****

****

****