

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERÍA

INGENIERÍA EN COMPUTACIÓN

Laboratorio de Computación Grafica E Interacción Humano-Computadoras

---

Grupo: 11

Grupo de teoría: 3

Semestre 2021-2

## **Proyecto**

Fecha de entrega limite: 25 de julio de 2021

---

418046595

andrwe713@gmail.com

Gallardo Tinoco Andrés Amadeus

# 1. Desarrollo

## 1.1. Modelado



Figura 1: Modelo en software Blender



Figura 2: Modelo en software Blender

Todos los modelos son únicos y fueron realizados en el software de Blender, de igual forma fueron texturizados en dicho software, todos los modelos están en un solo archivo para manejar las dimensiones de estos, el archivo esta disponible en el repositorio de GitHub.

*[https://github.com/AndresAmadeus/Proyecto\\_compu\\_grafica](https://github.com/AndresAmadeus/Proyecto_compu_grafica)*

## 1.2. Importacion

```

mesa_centro = Model();
mesa_centro.LoadModel("Models/mesa_centro.obj");
mesa_centro_aceite = Model();
mesa_centro_aceite.LoadModel("Models/mesa_centro_aceite.obj");
mesa_centro_freidora1 = Model();
mesa_centro_freidora1.LoadModel("Models/mesa_centro_freidora1.obj");
mesa_centro_freidora2 = Model();
mesa_centro_freidora2.LoadModel("Models/mesa_centro_freidora2.obj");
mesa_centro_puertal = Model();
mesa_centro_puertal.LoadModel("Models/mesa_centro_puertal.obj");
mesa_centro_puerta2 = Model();
mesa_centro_puerta2.LoadModel("Models/mesa_centro_puerta2.obj");
mesa_centro_puerta3 = Model();
mesa_centro_puerta3.LoadModel("Models/mesa_centro_puerta3.obj");
mesa_centro_puerta4 = Model();
mesa_centro_puerta4.LoadModel("Models/mesa_centro_puerta4.obj");
estructura = Model();
estructura.LoadModel("Models/estructura.obj");
lavadero = Model();
lavadero.LoadModel("Models/lavadero.obj");
mesal = Model();
mesal.LoadModel("Models/mesal.obj");
mesa2 = Model();
mesa2.LoadModel("Models/mesa2.obj");
puerta_metal = Model();
puerta_metal.LoadModel("Models/puerta_metal.obj");
mesa_ext_1 = Model();
mesa_ext_1.LoadModel("Models/mesa_ext_1.obj");
mesa_ext_2 = Model();
mesa_ext_2.LoadModel("Models/mesa_ext_2.obj");
mesa_ext_3 = Model();
mesa_ext_3.LoadModel("Models/mesa_ext_3.obj");
silla_ext_1 = Model();
silla_ext_1.LoadModel("Models/silla_ext_1.obj");
silla_ext_2 = Model();
silla_ext_2.LoadModel("Models/silla_ext_2.obj");
silla_ext_3 = Model();
silla_ext_3.LoadModel("Models/silla_ext_3.obj");
silla_ext_4 = Model();
silla_ext_4.LoadModel("Models/silla_ext_4.obj");
silla_ext_5 = Model();
silla_ext_5.LoadModel("Models/silla_ext_5.obj");
silla_ext_6 = Model();
silla_ext_6.LoadModel("Models/silla_ext_6.obj");
silla_ext_7 = Model();
silla_ext_7.LoadModel("Models/silla_ext_7.obj");
silla_ext_8 = Model();
silla_ext_8.LoadModel("Models/silla_ext_8.obj");
silla_ext_9 = Model();
silla_ext_9.LoadModel("Models/silla_ext_9.obj");
silla_ext_10 = Model();
silla_ext_10.LoadModel("Models/silla_ext_10.obj");
silla_ext_11 = Model();
silla_ext_11.LoadModel("Models/silla_ext_11.obj");
silla_ext_12 = Model();
silla_ext_12.LoadModel("Models/silla_ext_12.obj");

```

Figura 3: Codigo

```

1569 model = glm::mat4(1.0);
1570 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1571 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1572 papa1_hoja_1.RenderModel();
1573
1574 model = glm::mat4(1.0);
1575 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1576 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1577 papa2_hoja_2.RenderModel();
1578
1579 model = glm::mat4(1.0);
1580 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1581 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1582 papa3_hoja_3.RenderModel();
1583
1584 model = glm::mat4(1.0);
1585 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1586 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1587 mesa_centro.RenderModel();
1588
1589 model = glm::mat4(1.0);
1590 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1591 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1592 mesa_centro_aceite.RenderModel();
1593
1594 model = glm::mat4(1.0);
1595 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1596 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1597 mesa_centro_freidora1.RenderModel();
1598
1599 model = glm::mat4(1.0);
1600 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1601 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1602 mesa_centro_freidora2.RenderModel();
1603
1604 model = glm::mat4(1.0);
1605 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1606 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1607 mesa1.RenderModel();
1608
1609 model = glm::mat4(1.0);
1610 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1611 Material_brillante.UseMaterial(uniformSpecularIntensity, uniformShininess);
1612 mesa2.RenderModel();
1613
1614 model = glm::mat4(1.0);
1615 glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
1616 Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
1617 mesa_ext_1.RenderModel();
1618
1619 model = glm::mat4(1.0);

```

Figura 4: Codigo

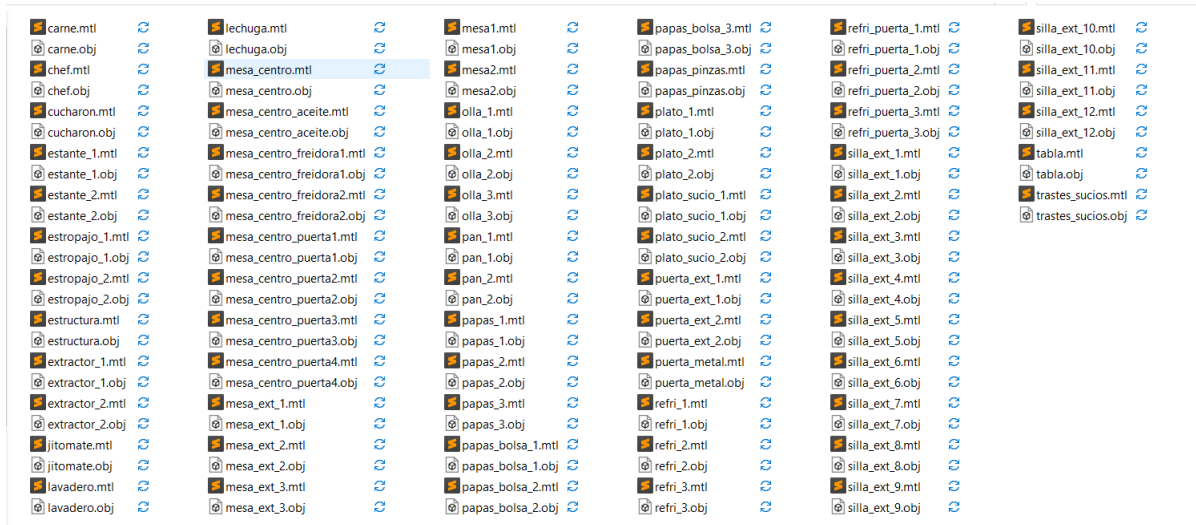


Figura 5: Modelos

Carga de todos los modelos, con un total de 65 modelos, (varios son copias entre sí, por ejemplo, ollas, puertas, sillas, mesas, etc.), cada modelo se cargo al programa, mediante un archivo .obj exportado desde el software Blender, se les asigno un material, no se realizó ningún movimiento ya que todos se encontraban en su lugar, excepto los objetos que formarían parte de las animaciones.

### 1.3. Animacion

```
//puerta metal
GLfloat ani_puerta_mtl_tm = 0.0f;
bool ani_puerta_mtl = false;
GLfloat giro_puerta_mtl = 0.0f;
GLfloat ani_puerta_mtl_tm01 = 2.0f;
GLfloat ani_puerta_mtl_tm01_std = true;

//papas fritas
GLfloat ani_papas_tm = 0.0f;
GLfloat ani_papas_posX = 0.0f, ani_papas_posY = 0.0f;
GLfloat ani_papas_posX_P1 = 0.0f, ani_papas_posY_P1 = 0.0f;
GLfloat ani_papas_posX_P2 = 0.0f, ani_papas_posY_P2 = 0.0f;
GLfloat ani_papas_posX_P3 = 0.0f, ani_papas_posY_P3 = 0.0f;
bool ani_papas = false;
GLfloat ani_papas_tm01 = 2.0f;
GLfloat ani_papas_tm02 = 2*ani_papas_tm01;
GLfloat ani_papas_tm03 = 3*ani_papas_tm01;
GLfloat ani_papas_tm04 = 4*ani_papas_tm01;
GLfloat ani_papas_tm05 = 5*ani_papas_tm01;

//Hamburguesa
GLfloat ani_ham_tm = 0.0f;
bool ani_ham = false;
GLfloat ani_ham_tm01 = 2.0f;
GLfloat ani_ham_tm02 = 2 * ani_ham_tm01;
GLfloat ani_ham_tm03 = 3 * ani_ham_tm01;
GLfloat ani_ham_tm04 = 4 * ani_ham_tm01;
GLfloat ani_ham_tm05 = 5 * ani_ham_tm01;
GLfloat ani_ham_posX_P1 = 0.0f, ani_ham_posY_P1 = 0.0f, ani_ham_posZ_P1 = 0.0f;
GLfloat ani_ham_posX_P2 = 0.0f, ani_ham_posY_P2 = 0.0f, ani_ham_posZ_P2 = 0.0f;
GLfloat ani_ham_posX_P3 = 0.0f, ani_ham_posY_P3 = 0.0f, ani_ham_posZ_P3 = 0.0f;
GLfloat ani_ham_posX_P4 = 0.0f, ani_ham_posY_P4 = 0.0f, ani_ham_posZ_P4 = 0.0f;
GLfloat ani_ham_posX_P5 = 0.0f, ani_ham_posY_P5 = 0.0f, ani_ham_posZ_P5 = 0.0f;

//Trastes sucios
GLfloat ani_tra_tm = 0.0f;
bool ani_tra = false;
GLfloat ani_tra_posZ = 0.0f, ani_tra_posY = 0.0f;
GLfloat ani_tra_posY_P1 = 0.0f, ani_tra_posY_P2 = 0.0f;
GLfloat ani_tra_tm01 = 2.0f;
GLfloat ani_tra_tm02 = 2 * ani_tra_tm01;
GLfloat ani_tra_tm03 = 3 * ani_tra_tm01;
GLfloat ani_tra_tm04 = 4 * ani_tra_tm01;
GLfloat ani_tra_tm05 = 5 * ani_tra_tm01;
```

Figura 6: Codigo

```

//papas fritas
if (mainWindow.get_btn_p()) {
    mainWindow.res_btn_p();
    ani_papas = true;
    ani_papas_tm = 0.0f;
    ani_papas_posX_P1 = 0;
    ani_papas_posX_P2 = 0;
    ani_papas_posX_P3 = 0;
    ani_papas_posY_P1 = 0;
    ani_papas_posY_P2 = 0;
    ani_papas_posY_P3 = 0;
}

if (ani_papas) {
    ani_papas_tm += deltaTime / del_mod;
    if (ani_papas_tm < ani_papas_tm01) {
        ani_papas_posX = 0.66 * ani_papas_tm / ani_papas_tm01;
        ani_papas_posY = -4.014*pow((ani_papas_posX - 0.386),2) + 0.6;

        ani_papas_posX_P1 = ani_papas_posX;
        ani_papas_posY_P1 = ani_papas_posY;
    }
    else if (ani_papas_tm < ani_papas_tm02){
        ani_papas_posX = 0.66 - 0.78 * (ani_papas_tm - ani_papas_tm01) / ani_papas_tm01;
        ani_papas_posY = -2.873 * pow((ani_papas_posX - 0.336), 2) + 0.6;
    }
    else if (ani_papas_tm < ani_papas_tm03) {
        ani_papas_posX = -0.12 + 0.93 * (ani_papas_tm - ani_papas_tm02) / ani_papas_tm01;
        ani_papas_posY = -2.021 * pow((ani_papas_posX - 0.424), 2) + 0.6;

        ani_papas_posX_P2 = ani_papas_posX + 0.12;
        ani_papas_posY_P2 = ani_papas_posY;
    }
    else if (ani_papas_tm < ani_papas_tm04) {
        ani_papas_posX = 0.81 - 1.06 * (ani_papas_tm - ani_papas_tm03) / ani_papas_tm01;
        ani_papas_posY = -1.556 * pow((ani_papas_posX - 0.370), 2) + 0.6;
    }
    else if (ani_papas_tm < ani_papas_tm05) {
        ani_papas_posX = -0.25 + 1.18 * (ani_papas_tm - ani_papas_tm04) / ani_papas_tm01;
        ani_papas_posY = -1.255 * pow((ani_papas_posX - 0.441), 2) + 0.6;

        ani_papas_posX_P3 = ani_papas_posX + 0.25;
        ani_papas_posY_P3 = ani_papas_posY;
    }
    else {
        ani_papas = false;
    }
}

model = glm::mat4(1.0);
model = glm::translate(model, glm::vec3(0.0f + ani_papas_posX_P3, 0.0f + ani_papas_posY_P3, 0.0f));
glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
Material_opaco.UseMaterial(uniformSpecularIntensity, uniformShininess);
papas_1.RenderModel();

```

Figura 7: Código

Cada animación se realizó de manera independiente, es decir se pueden ejecutar todas las animaciones al mismo tiempo, para cada animación, se crearon funciones específicas para el

movimiento espacial de los objetos, y es posible modificar los intervalos de cada animación o velocidad, aunque no es recomendable ya que los sonidos de algunas animaciones están al tiempo en que están corren.

## 1.4. Luces

```

/*
//Declaración de primer luz puntual
pointLights[0] = PointLight(1.0f, 0.0f, 0.0f,
    0.0f, 1.0f,
    2.0f, 1.5f, 1.5f,
    0.3f, 0.2f, 0.1f);
pointLightCount++;
*/
unsigned int spotlightCount = 0;
//luces del interior
spotlights[0] = Spotlight(1.0f, 1.0f, 1.0f, //color
    0.0f, 2.0f, //aln dln
    0.0f, 0.0f, 0.0f, //pos
    0.0f, -1.0f, 0.0f, //direc
    1.0f, 0.0f, 0.0f, //poli
    15.0f); //tam
spotlightCount++;

/*
spotlights[1] = Spotlight(1.0f, 1.0f, 1.0f, //color blanco
    0.0f, 2.0f, //aln dln
    -3.5f, 4.5f, -4.7f, //pos
    0.0f, -1.0f, 0.0f, //direc
    3.0f, 0.0f, 0.0f, //poli
    80.0f); //tam
spotlightCount++;

spotlights[2] = Spotlight(1.0f, 1.0f, 1.0f, //color blanco
    0.0f, 2.0f, //aln dln
    -3.5f, 4.5f, 2.7f, //pos
    0.0f, -1.0f, 0.0f, //direc
    3.0f, 0.0f, 0.0f, //poli
    80.0f); //tam
spotlightCount++;
**/

spotlights[1] = Spotlight(1.0f, 1.0f, 1.0f, //color blanco
    0.0f, 2.0f, //aln dln
    4.7f, 4.5f, -4.6f, //pos
    0.0f, -1.0f, 0.0f, //direc
    2.0f, 0.0f, 0.0f, //poli
    80.0f); //tam
spotlightCount++;

spotlights[2] = Spotlight(1.0f, 1.0f, 1.0f, //color blanco
    0.0f, 2.0f, //aln dln
    4.7f, 4.5f, -1.0f, //pos
    0.0f, -1.0f, 0.0f, //direc
    2.0f, 0.0f, 0.0f, //poli
    80.0f); //tam
spotlightCount++;
*/

```

Figura 8: Código

Todas las luces son Spotlight, todas tiene dirección hacia abajo  $-Y$ , se cambiaron algunos parámetros para que funcionaran como focos, y se posicionaron el lugar correcto, para ello se crearon bases color negro el techo de la estructura donde están las luces.



## 1.5. Sonido

```
//Sonidos
void play_music(int n) {
    switch (n) {
        case 1:
            PlaySound(TEXT("Sounds/07019076.wav"), NULL, SND_ASYNC | SND_NOSTOP | SND_LOOP);
            break; //fondo
        case 2:
            PlaySound(TEXT("Sounds/bbc_household _07002109.wav"), NULL, SND_ASYNC);
            break; //servir
        case 3:
            PlaySound(TEXT("Sounds/bbc_household _07067001.wav"), NULL, SND_ASYNC);
            break; //lavar
        case 4:
            PlaySound(TEXT("Sounds/bbc_kitchen ap_07011060.wav"), NULL, SND_ASYNC);
            break; //puerta pequeña
        case 5:
            PlaySound(TEXT("Sounds/bbc_steel door_07037627.wav"), NULL, SND_ASYNC );
            break; //metal abrir
    }
}
```

Figura 9: Código

```
ani_sopa_cm = 0.0f;

thread th_02(play_music, 2);
if (th_02.joinable()) {
    th_02.join();
}
```

Figura 10: Código

Para el sonido se usó una biblioteca para reproducir sonido, no fue posible lograr que varios sonidos se reproduzcan al mismo tiempo por lo que cuando se ejecuta una animación con sonido el sonido de fondo se detiene y al terminar vuelve a comenzar.

Por último, también fueron usados hilos, ya que ahora existe un hilo independiente del principal que ejecuta el sonido.

## 2. Resultado