UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA e INTERACCIÓN HUMANO COMPUTADORA

REPORTE DE PRÁCTICA Nº 09

NOMBRE COMPLETO: LOPEZ BETANCOURT MICHELLE

Nº de Cuenta: 318309028

GRUPO DE LABORATORIO: 02

GRUPO DE TEORÍA: 06

SEMESTRE 2025-1

FECHA DE ENTREGA LÍMITE: 19 DE OCTUBRE DE 2024

CALIFICACIÓN: _____

REPORTE DE PRÁCTICA:

1.- Ejecución de los ejercicios que se dejaron, comentar cada uno y capturas de pantalla de bloques de código generados y de ejecución del programa.

- Para esta práctica se tenían que realizar dos ejercicios, los cuales consistían en que mediante el dado de 10 caras realizado anteriormente se debía lograr que cayera al piso, girara y mostrara un numero “random”, haciendo que la tirada del dado se repitiera al presionar alguna tecla en el teclado. El segundo ejercicio era que por integrante del equipo se debía elegir un tipo de vehículo ya fuera terrestre o aéreo, cada uno de nosotros debíamos crear un recorrido en donde el vehículo elegido se moviera alrededor de nuestro tablero de monopoly, se debía tomar en cuenta que cada uno iniciaría a partir de una esquina diferente y además el modelo escogido debía tener movimiento de llantas o de hélices. En caso de elegir un vehículo terrestre este no podía volver a ser un carro o similar a uno motorizado de 4 ruedas.

Código generado

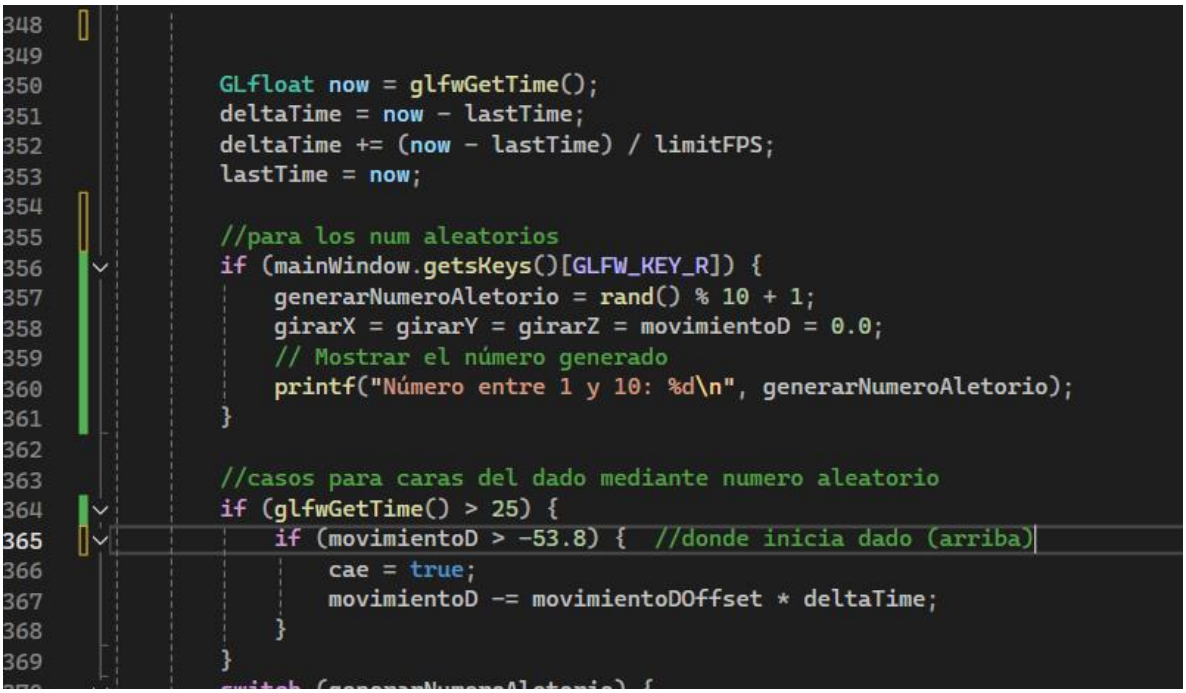
```
8     Adicional.- ,Textura Animada
9     */
10    //para cargar imagen
11    #define STB_IMAGE_IMPLEMENTATION
12
13    #include <stdio.h>
14    #include <string.h>
15    #include <cmath>
16    #include <vector>
17    #include <math.h>
18    #include <stdlib.h>
19    #include <time.h>
20
21    #include <glew.h>
22    #include <glfw3.h>
23
24    #include <glm.hpp>
```

```
57
58      //dado
59      float movimientoD;
60      float movimientoDOffset;
61      float girarX;
62      float girarY;
63      float girarZ;
64      bool cae;
65
66      //moto
67      bool avanzaMoto;
68      float movimientoMoto;
69
70      float movimientoMoto2;
71      float movimientoMotoOffset;
72      float rotllantamoto;
73      float rotllantamotoOffset;
74      float moviMoto;
75
76      int generarNumeroAletorio;
77
78      Window mainWindow;
79      std::vector<Mesh*> meshList;
```

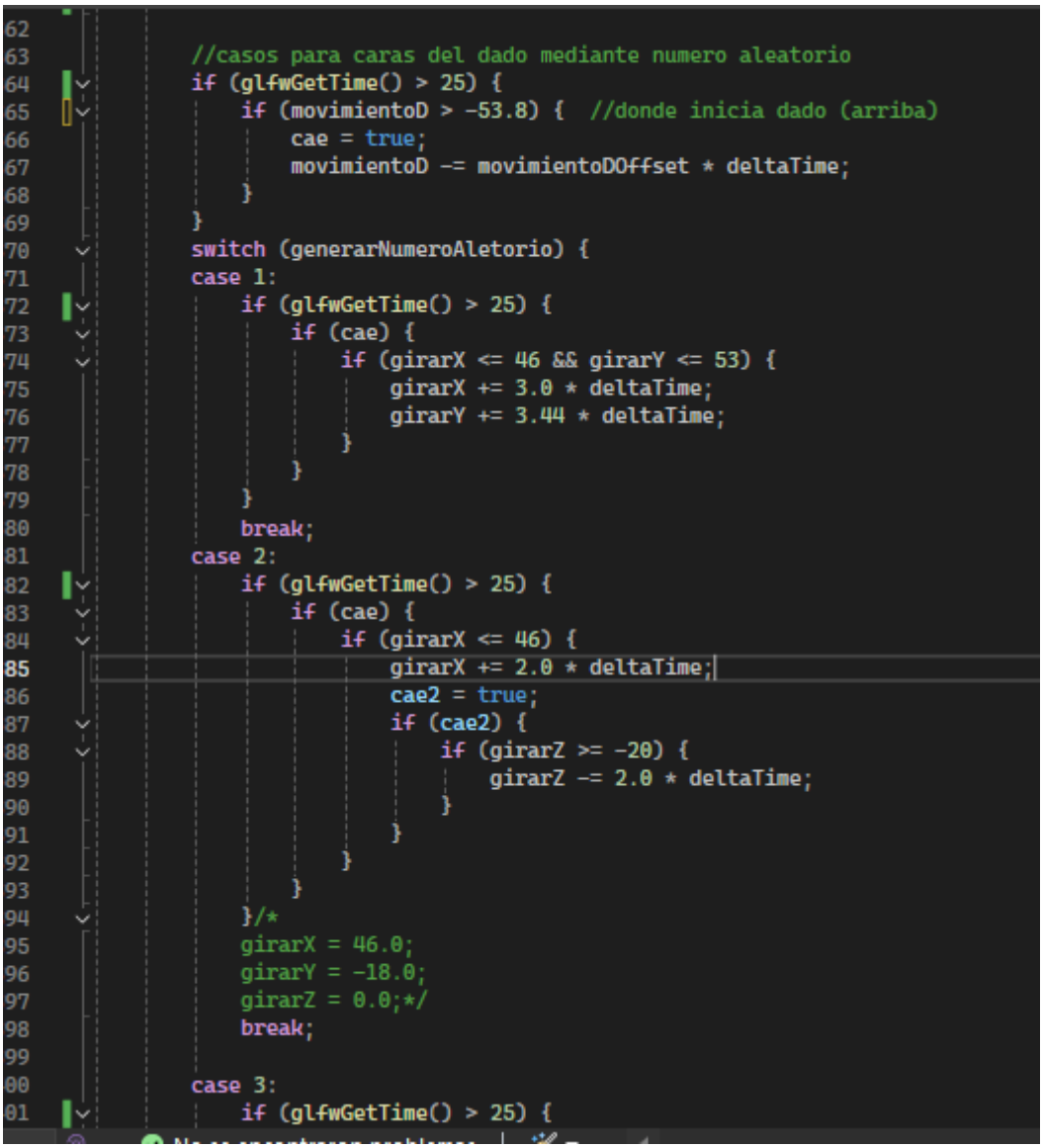
```
Texture pisoTexture;

//Model arco_M;
//Model cartel_M;
Model dado10_M;
Model cuerpomoto_M;
Model llantamoto_M;
Model mapa_M;

Skybox skybox;

//materiales
Material Material_brillant
```

```
247
248
249         /*arco_M = Model();
250         arco_M.LoadModel("Models/arco.obj");
251         cartel_M = Model();
252         cartel_M.LoadModel("Models/cartel.obj");*/
253         dado10_M = Model();
254         dado10_M.LoadModel("Models/dado10.obj");
255         cuerpomoto_M = Model();
256         cuerpomoto_M.LoadModel("Models/cuerpomoto.obj");
257         llantamoto_M = Model();
258         llantamoto_M.LoadModel("Models/llantamoto.obj");
259         mapa_M = Model();
260         mapa_M.LoadModel("Models/mapa.obj");
261
262
263         std::vector<std::string> skyboxFaces;
264         skyboxFaces.push_back("Textures/Skybox/cupertin-lake_rt.tga
265         skyboxFaces.push_back("Textures/Skybox/cupertin-lake_lf.tga
266         skyboxFaces.push_back("Textures/Skybox/cupertin-lake_dn.tga
267         skyboxFaces.push_back("Textures/Skybox/cupertin-lake_up.tga
```

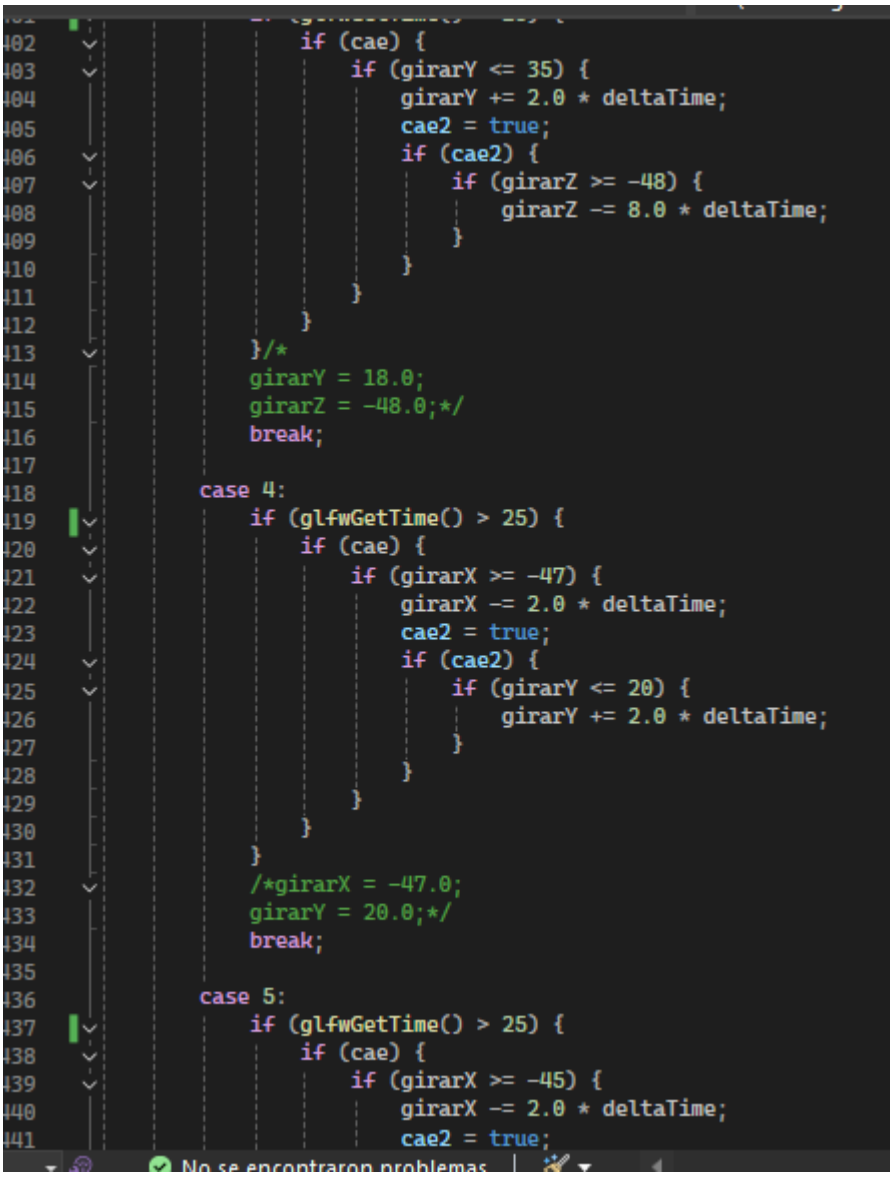
```
319         avanza = true;
320         offsetMovCartel = 0.1f;
321         offsetRotCartel = 3.0f;
322         float prevTime = glfwGetTime(); //tiempo en que se tarda el ejecutar
323
324         //dad0
325         movimientoD = 0.0;
326         movimientoDOffset = 0.4;
327         girarX = 1.5;
328         girarY = 1.5;
329         girarZ = 1.5;
330         cae = false;
331         float cae2 = false;
332
333         //moto
334         rotllantamoto = 1.0f;
335         rotllantamotoOffset = 5.0f;
336         movimientoMoto = 1.0f;
337         movimientoMotoOffset = 0.3;
338         avanzaMoto = true;
339         moviMoto = 0.3f;
340
341
342         generarNumeroAletorio = 0.0;
343         srand(time(NULL));
344         //int generarNumeroAletorio = 9;
345         ////Loop mientras no se cierra la ventana
```



```
348
349
350         GLfloat now = glfwGetTime();
351         deltaTime = now - lastTime;
352         deltaTime += (now - lastTime) / limitFPS;
353         lastTime = now;
354
355         //para los num aleatorios
356         if (mainWindow.getsKeys()[GLFW_KEY_R]) {
357             generarNumeroAletorio = rand() % 10 + 1;
358             girarX = girarY = girarZ = movimientoD = 0.0;
359             // Mostrar el número generado
360             printf("Número entre 1 y 10: %d\n", generarNumeroAletorio);
361         }
362
363         //casos para caras del dado mediante numero aleatorio
364         if (glfwGetTime() > 25) {
365             if (movimientoD > -53.8) {  //donde inicia dado (arriba)
366                 cae = true;
367                 movimientoD -= movimientoDOffset * deltaTime;
368             }
369         }
```

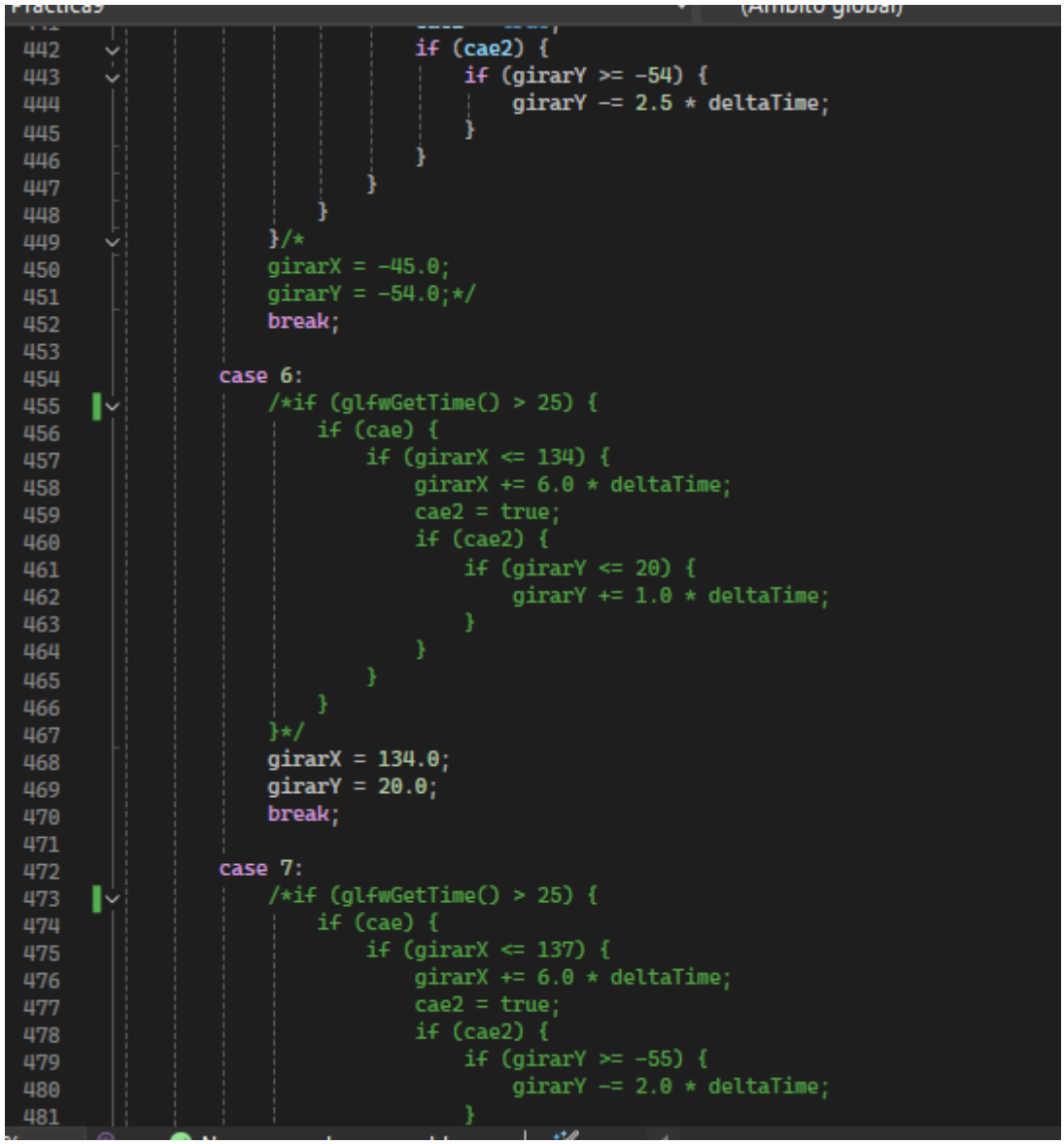


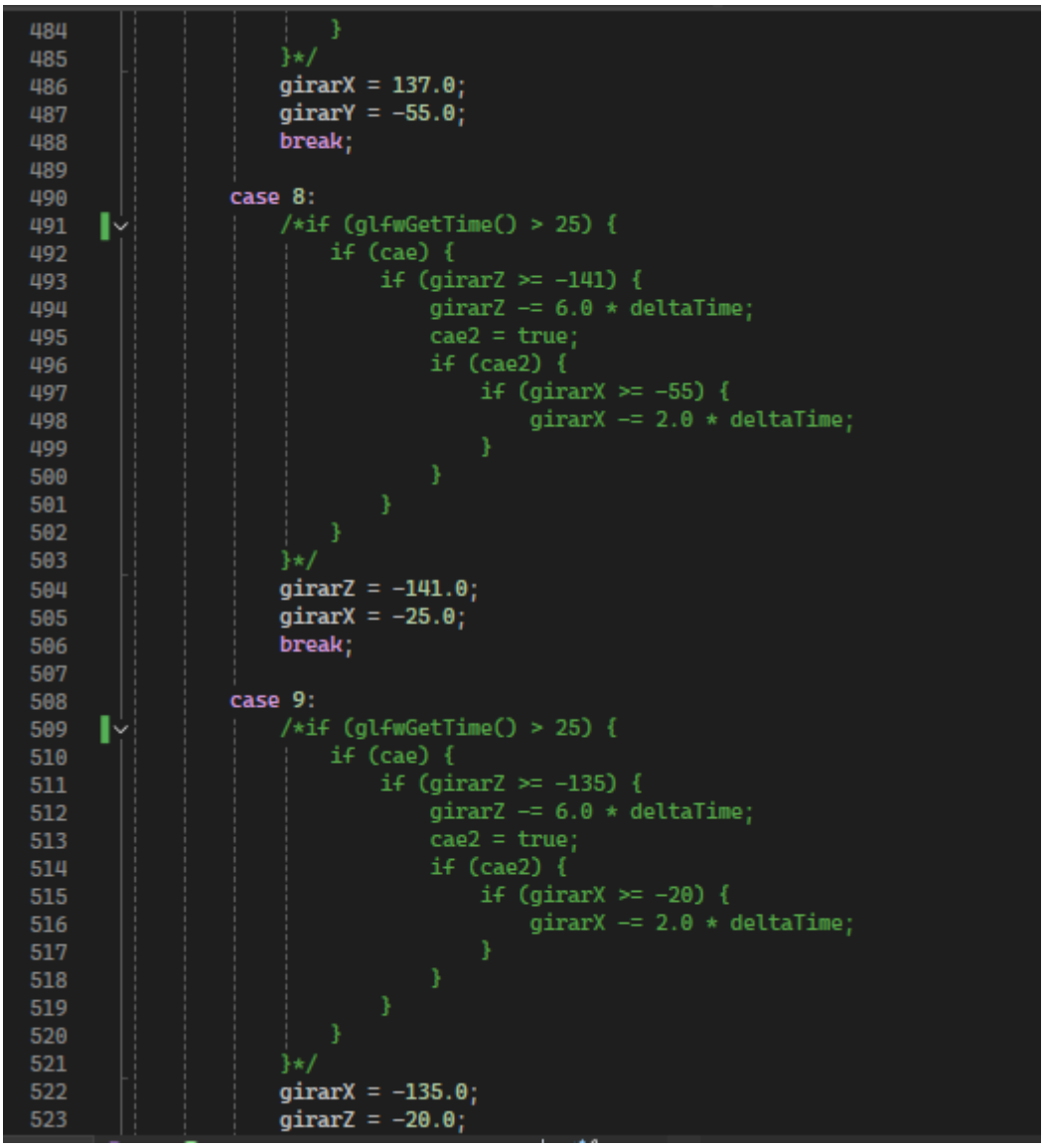
```
62
63          //casos para caras del dado mediante numero aleatorio
64          if (glfwGetTime() > 25) {
65              if (movimientoD > -53.8) {  //donde inicia dado (arriba)
66                  cae = true;
67                  movimientoD -= movimientoDOffset * deltaTime;
68              }
69          }
70          switch (generarNumeroAletorio) {
71          case 1:
72              if (glfwGetTime() > 25) {
73                  if (cae) {
74                      if (girarX <= 46 && girarY <= 53) {
75                          girarX += 3.0 * deltaTime;
76                          girarY += 3.44 * deltaTime;
77                      }
78                  }
79              }
80              break;
81          case 2:
82              if (glfwGetTime() > 25) {
83                  if (cae) {
84                      if (girarX <= 46) {
85                          girarX += 2.0 * deltaTime;
86                          cae2 = true;
87                          if (cae2) {
88                              if (girarZ >= -20) {
89                                  girarZ -= 2.0 * deltaTime;
90                              }
91                          }
92                      }
93                  }
94              }/*
95              girarX = 46.0;
96              girarY = -18.0;
97              girarZ = 0.0;*/
98              break;
99
00          case 3:
01              if (glfwGetTime() > 25) {
```



```
            if (cae) {
                if (girarY <= 35) {
                    girarY += 2.0 * deltaTime;
                    cae2 = true;
                    if (cae2) {
                        if (girarZ >= -48) {
                            girarZ -= 8.0 * deltaTime;
                        }
                    }
                }
            }
        }/*
        girarY = 18.0;
        girarZ = -48.0;*/
        break;

    case 4:
        if (glfwGetTime() > 25) {
            if (cae) {
                if (girarX >= -47) {
                    girarX -= 2.0 * deltaTime;
                    cae2 = true;
                    if (cae2) {
                        if (girarY <= 20) {
                            girarY += 2.0 * deltaTime;
                        }
                    }
                }
            }
        }
        /*girarX = -47.0;
        girarY = 20.0;*/
        break;

    case 5:
        if (glfwGetTime() > 25) {
            if (cae) {
                if (girarX >= -45) {
                    girarX -= 2.0 * deltaTime;
                    cae2 = true;
```

No se encontraron problemas


Practica9
(Ámbito global)
442 if (cae2) {
443 if (girarY >= -54) {
444 girarY -= 2.5 * deltaTime;
445 }
446 }
447 }
448 }
449 }/*
450 girarX = -45.0;
451 girarY = -54.0;*/
452 break;
453
454 case 6:
455 /*if (glfwGetTime() > 25) {
456 if (cae) {
457 if (girarX <= 134) {
458 girarX += 6.0 * deltaTime;
459 cae2 = true;
460 if (cae2) {
461 if (girarY <= 20) {
462 girarY += 1.0 * deltaTime;
463 }
464 }
465 }
466 }
467 }*/
468 girarX = 134.0;
469 girarY = 20.0;
470 break;
471
472 case 7:
473 /*if (glfwGetTime() > 25) {
474 if (cae) {
475 if (girarX <= 137) {
476 girarX += 6.0 * deltaTime;
477 cae2 = true;
478 if (cae2) {
479 if (girarY >= -55) {
480 girarY -= 2.0 * deltaTime;
481 }



```
484                 }
485             }*/
486             girarX = 137.0;
487             girarY = -55.0;
488             break;
489
490         case 8:
491             /*if (glfwGetTime() > 25) {
492                 if (cae) {
493                     if (girarZ >= -141) {
494                         girarZ -= 6.0 * deltaTime;
495                         cae2 = true;
496                         if (cae2) {
497                             if (girarX >= -55) {
498                                 girarX -= 2.0 * deltaTime;
499                             }
500                         }
501                     }
502                 }
503             }*/
504             girarZ = -141.0;
505             girarX = -25.0;
506             break;
507
508         case 9:
509             /*if (glfwGetTime() > 25) {
510                 if (cae) {
511                     if (girarZ >= -135) {
512                         girarZ -= 6.0 * deltaTime;
513                         cae2 = true;
514                         if (cae2) {
515                             if (girarX >= -20) {
516                                 girarX -= 2.0 * deltaTime;
517                             }
518                         }
519                     }
520                 }
521             }*/
522             girarX = -135.0;
523             girarZ = -20.0;
```

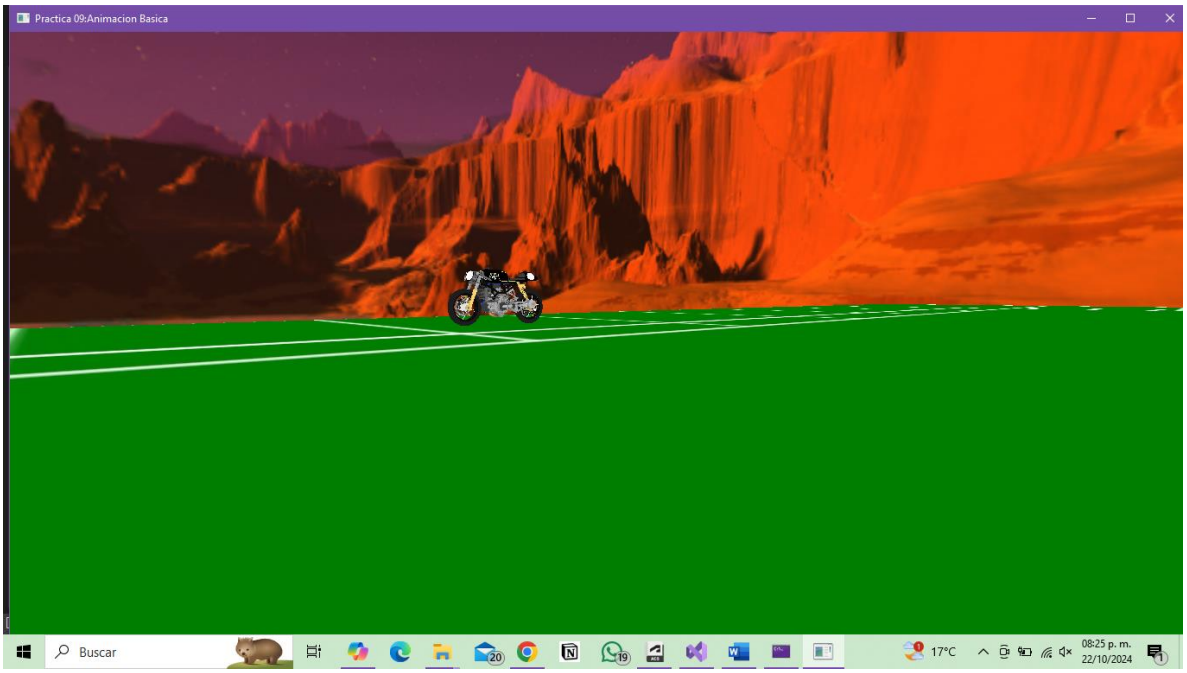
```
                    }
                }
            }
        }*/
        girarX = -135.0;
        girarZ = -20.0;
        break;

    case 10:
        if (glfwGetTime() > 25) {
            if (cae) {
                if (girarZ <= 134) {
                    girarZ += 6.0 * deltaTime;
                    cae2 = true;
                    if (cae2) {
                        if (girarY <= 25) {
                            girarY += 2.0 * deltaTime;
                        }
                    }
                }
            }
        }
        /*girarZ = 134.0;
        girarY = 25.0;*/
        break;
    }
```

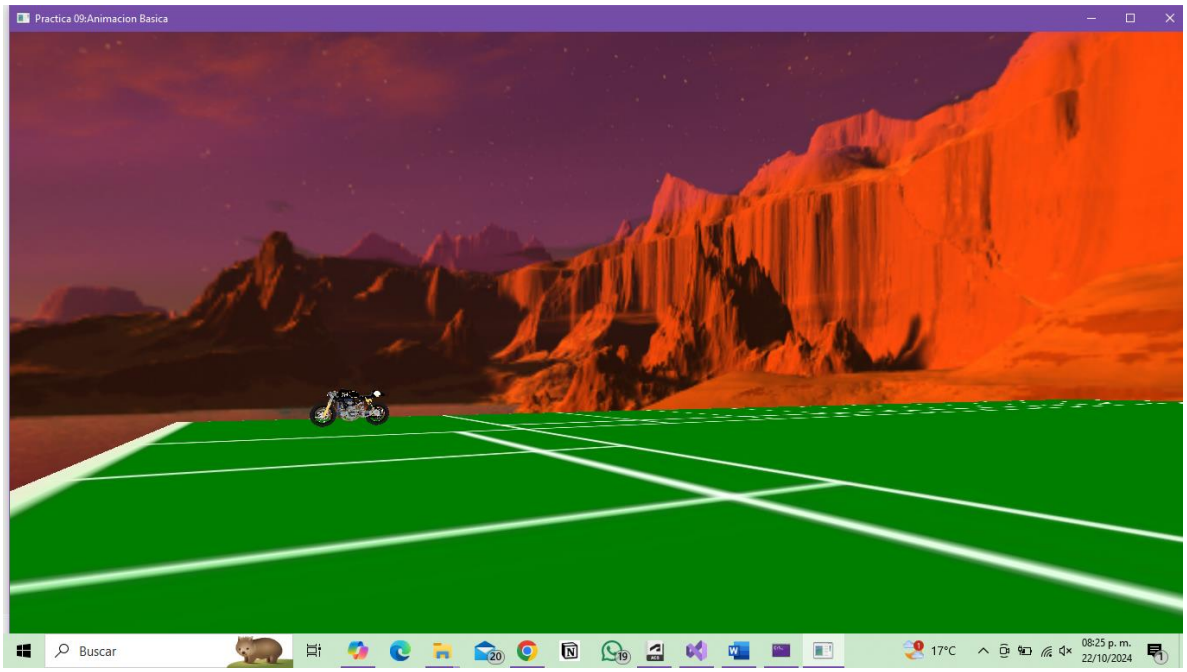
```
//para movimiento de la moto en el tablero empezando desde -x, -z
if (glfwGetTime() > 20) {
    if (avanzaMoto)
    {
        if (movimientoMoto < 140.0f)
        {
            movimientoMoto += movimientoMotoOffset * deltaTime;
            rotllantamoto += rotllantamotoOffset * deltaTime;


        }

        else
        {
            avanzaMoto = !avanzaMoto;
        }
    }


    else
    {

        avanzaMoto = !avanzaMoto;
    }
}
```

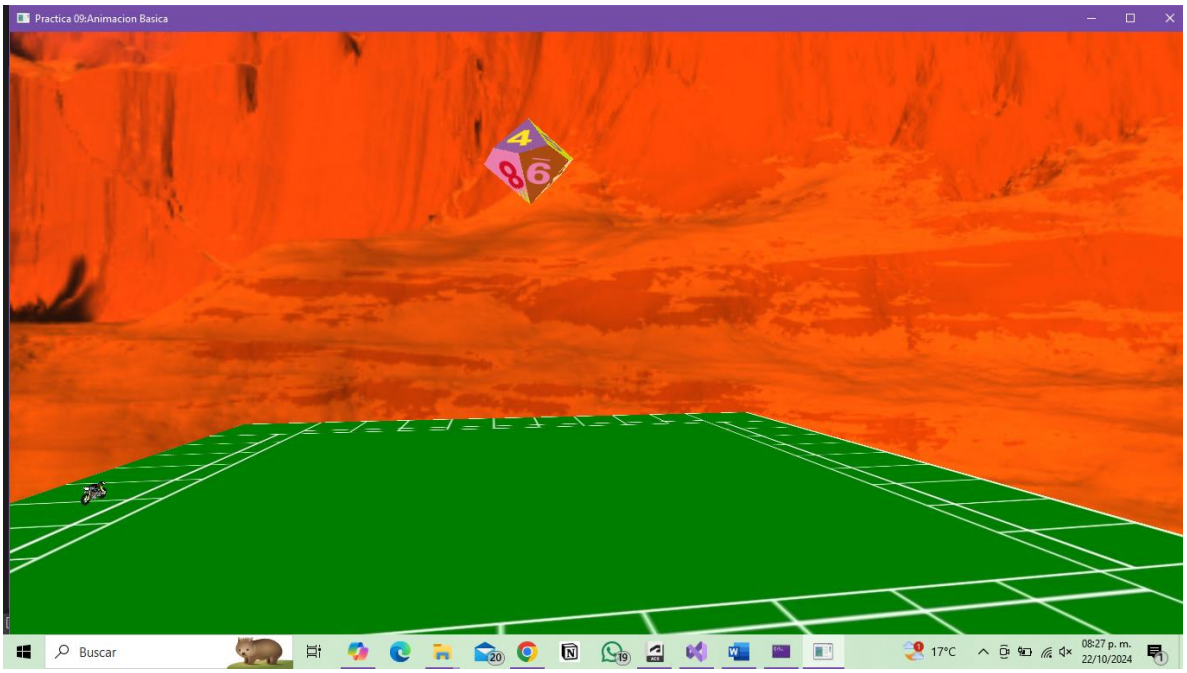
```
656
657         //Practica 9
658         //1.  Su dado de 10 caras cae sobre el piso, gira y cae en un número "random", se repite la tirada al presionar una tecla
659         model = glm::mat4(1.0);
660         model = glm::translate(model, glm::vec3(0.0f, movimientoD + 60.5f, 0.0f));
661         model = glm::rotate(model, girarX * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
662         model = glm::rotate(model, girarY * toRadians, glm::vec3(0.0f, 1.0f, 0.0f));
663         model = glm::rotate(model, girarZ * toRadians, glm::vec3(0.0f, 0.0f, 1.0f));
664         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
665
666         dado10_M.RenderModel();
667
668
669
670         /*2. Por integrante del equipo elegirán un tipo de vehículo: terrestre o aéreo . Cada integrante del equipo creará un
671         recorrido en el cual el vehículo se moverá alrededor de su tablero de Monopoly. Cada vehículo iniciará su recorrido a
672         partir de una esquina diferente. (el vehículo terrestre no puede ser un carro o vehículo similar
673         motorizado de 4 ruedas, se debe de tener movimiento de llantas o de hélices en sus vehiculos.)
674         */
675
676         //tablero monopoly
677         model = glm::mat4(1.0);
678         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 0.0f));
679         modelaux = model;
680         model = glm::scale(model, glm::vec3(5.5f, 5.5f, 5.5f));
681         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
682         mapa_M.RenderModel();
683
684         //moto
685         model = glm::mat4(1.0);
686         model = glm::translate(model, glm::vec3(movimientoMoto2 - 70.0f, 1.5f, movimientoMoto - 70.0f));
687         modelaux = model;
688         model = glm::scale(model, glm::vec3(0.9f, 0.9f, 0.9f));
```

```
683
684         //moto
685         model = glm::mat4(1.0);
686         model = glm::translate(model, glm::vec3(movimientoMoto2 - 70.0f, 1.5f, movimientoMoto - 70.0
687         modelaux = model;
688         model = glm::scale(model, glm::vec3(0.9f, 0.9f, 0.9f));
689         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
690         cuerpomoto_M.RenderModel();
691
692         //llanta moto delante
693         model = modelaux;
694         model = glm::translate(model, glm::vec3(0.0f, 0.0f, 2.0f));
695         model = glm::rotate(model, rotllantamoto * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
696         model = glm::scale(model, glm::vec3(0.9f, 0.9f, 0.9f));
697         //color = glm::vec3(0.5f, 0.5f, 0.5f);//llanta con color gris
698         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
699         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
700         llantamoto_M.RenderModel();
701
702         //llanta moto delante
703         model = modelaux;
704         model = glm::translate(model, glm::vec3(0.0f, 0.0f, -1.9f));
705         model = glm::rotate(model, rotllantamoto * toRadians, glm::vec3(1.0f, 0.0f, 0.0f));
706         model = glm::scale(model, glm::vec3(0.9f, 0.9f, 0.9f));
707         //color = glm::vec3(0.5f, 0.5f, 0.5f);//llanta con color gris
708         glUniform3fv(uniformColor, 1, glm::value_ptr(color));
709         glUniformMatrix4fv(uniformModel, 1, GL_FALSE, glm::value_ptr(model));
710         llantamoto_M.RenderModel();
711
712
713
714         //blending: transparencia o traslucidez
715         glEnable(GL_BLEND);
```

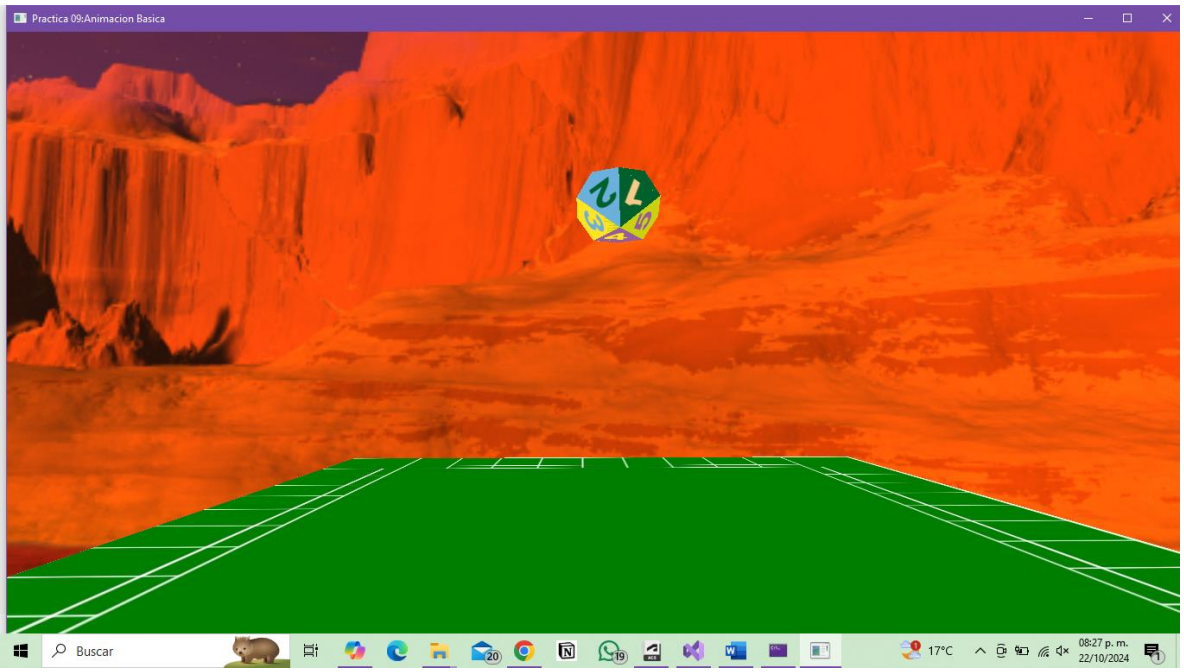
Ejecución del programa

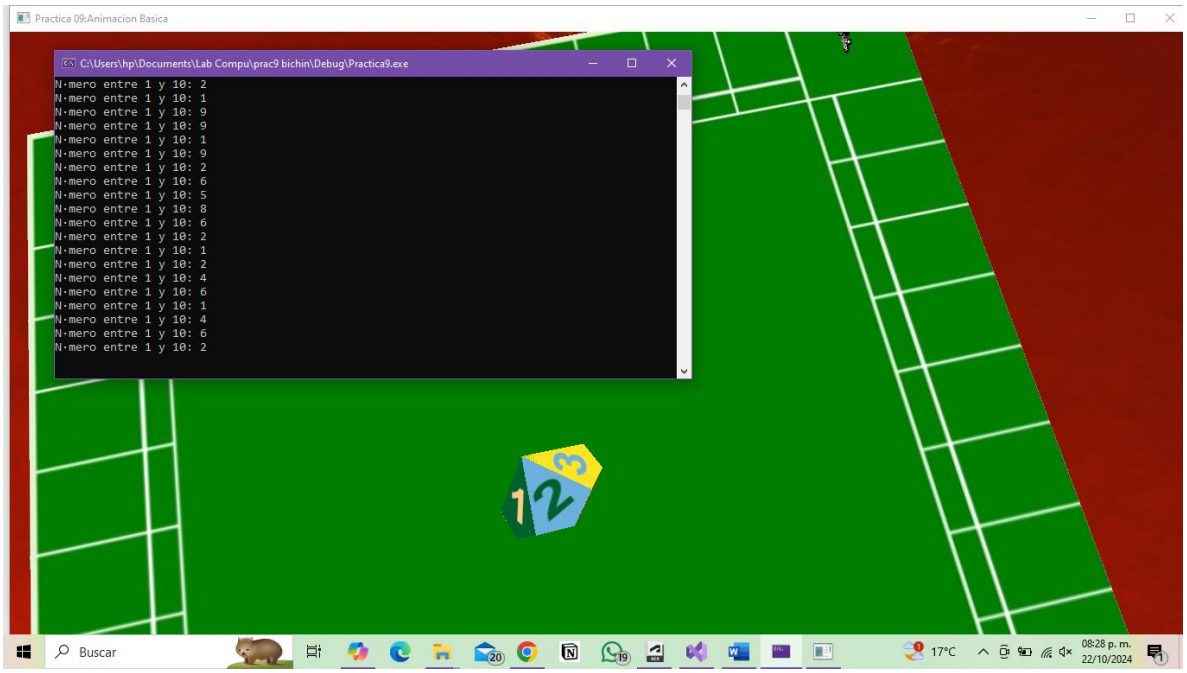

Practica 09:Animacion Basica
Buscar
17°C
08:25 p. m.
22/10/2024


Practica 09:Animacion Basica
Buscar
17°C
08:25 p. m.
22/10/2024



Mediante la letra R se generan los números aleatorios y el dado gira





2.- Liste los problemas que tuvo a la hora de hacer estos ejercicios y si los resolvió explicar cómo fue, en caso de error adjuntar captura de pantalla.

- Uno de los problemas que tuve al inicio fue con el dado de 10 caras ya que el que había realizado no estaba bien hecho para que en este ejercicio al momento de caer solo se viera una cara, por lo que tuve que descargar un modelo y agregarle una textura que elegí para que se viera en el dado.

- El segundo problema que tuve fue al momento de crear los casos para cada una de las caras del dado, ya que no sabia bien como delimitarlas por lo que estuve jugando con los valores hasta que logré encontrar cada uno correctamente.

- El tercer problema que tuve fue con el modelo del vehículo que elegí, ya que no se exportaba de manera correcta la textura y tenia que separar cada una de sus partes para poder tener por separado las llantas del modelo y asi poderle dar movimiento.

- El cuarto problema que tuve fue al momento de hacer el movimiento de las llantas de la moto, ya que, si giraban, pero no de manera correcta, por lo que tuve que volver a abrirlas en 3DMAX y me di cuenta que no se había exportado bien con el pivote en donde iba, asi que volví a modificar el pivote y volverlo a abrir en OpenGL y ahora si ya se movían de la manera correcta.

- El quinto problema que tuve fue al momento de hacer las líneas para el movimiento de la moto dentro del tablero de monopoly, ya que no supe como

hacer para que la moto siguiera todo el recorrido de cada una de las esquinas y tuve que agregarle un valor un poco alto en la variable del tiempo ya que si lo dejaba mas pequeño la moto se desplazaba de lugar.

- El sexto problema que presente fue con el dado, ya que al inicio no se genera el numero aleatorio y cae el dado de manera vertical, pero cuando se presiona la letra R ahora si ya genera el numero aleatorio correspondiente a la cara que mostrara y gira, aquí también le tuve que agregar un valor un poco alto en la variable del tiempo cuando ejecuta el código ya que si no no se visualizaba la primera vez que caía el dado.

3.- Conclusión:

a. Los ejercicios del reporte: Complejidad, Explicación.
   Después de haber realizado los ejercicios propuestos para este practica puedo concluir que no logre realizarlos todos con éxito, ya que aunque comprendí la manera de darle animación a los objetos mediante la respectiva implementación del código para el caso de estos ejercicios no supe muy bien como realizarlos, ya que me confunde un poco como manejar los valores, lo que hicimos durante estos ejercicios fue primero para el dado de 10 caras descargar un modelo y darle su correspondiente textura para después cargarlo en OpenGL y con esto realizar los casos para cada una de las caras del dado y la función de random que nos ayudaba con el número que debía salir al tirar el dado, así como también la implementación de una tecla para que esta acción se volviera a repetir. Para el caso del segundo ejercicio se buscó un modelo de una moto y se separaron el cuerpo y una de sus llantas para poder darles posteriormente el movimiento, se agrego el tablero de monopoly usado para nuestro proyecto y en una esquina diferente cada uno de los integrantes de nuestro equipo situó su modelo.

   La complejidad de los ejercicios para el movimiento del dado y de la moto puedo decir que si fue mas complicado que cosas realizadas anteriormente ya que no me quedo bien claro y no supe bien cómo hacerlo mediante la forma correcta y como saber cada valor.

b. Comentarios generales: Faltó explicar a detalle, ir más lento en alguna explicación, otros comentarios y sugerencias para mejorar desarrollo de la práctica.
   Por mi parte puedo concluir que en el caso de esta practica me quedaron algunas dudas, quizá falto comentar un poco mas acerca de como realizar cada ejercicio correctamente o como se podía implementar en el código.

c. Conclusión

Después de haber concluido con la practica puedo mencionar que se cumplió no al 100% los objetivos propuestos, debido a que, aunque comprendí el uso correcto de la animación básica en los modelos me falto haber realizado de manera correcta los ejercicios, aun asi mediante las actividades propuestas pude poner en practica los conocimientos que he ido aprendiendo durante las sesiones de laboratorio y asi cumplir con la mayoría.