Proyecto: Cnc 1313 wood machine

Universidad Central del Ecuador, Facultad de Ingeniería y Ciencias Aplicadas, Ingeniería en Sistemas de Información, Infraestructura de TI - II

23 – 07 - 2025

Elaborado por:

- Alquinga Cruz Cristopher Alejandro
- Bermeo Morrillo Dennis Alexander
- Heredia Nicolalde Vanessa Nayeli
- Narváez Rodriguez Milisen Estefanía
- Robayo Muñoz Carlos Vicente
- Sánchez Velarde Jordan Israel

I. OBJETIVOS

Objetivo General

Diseñar una réplica virtual y funcional de una máquina CNC modelo 1313 en Unity, basada en un análisis técnico profundo de su estructura mecánica, electrónica y aplicaciones reales, con el propósito de comprender su funcionamiento operativo y aportar a su uso en contextos educativos, técnicos e industriales.

Objetivos Específicos

- Analizar detalladamente la estructura mecánica de la máquina CNC, incluyendo elementos como la base, guías lineales, pórtico, husillo y el sistema de transmisión, especificando su función y disposición física dentro de la máquina real y virtual.
- Caracterizar el sistema de control electrónico, incluyendo:

  - Controladores
  - Drivers
  - Motores paso a paso
  - Sensores
  - Fuente de alimentación
  - Cableado y conexiones

  Explicando cómo se conectan entre sí para lograr el control automático del proceso de fresado.

- Documentar el sistema de sujeción de piezas, específicamente el funcionamiento de la mesa tipo “T Slot” o sistema de vacío, detallando su utilidad en el aseguramiento de materiales durante el mecanizado.
- Recrear en Unity cada uno de los componentes físicos de la máquina mediante modelado 3D y animación, asegurando que cada parte tenga una función similar a la real, incluyendo movimientos simulados y colisiones realistas.

- Desarrollar scripts funcionales en C# para controlar los movimientos simulados de los ejes X, Y y Z, emulando el comportamiento de una máquina CNC real mediante lógica programada y uso de la física de Unity (colliders, rigidbodies, etc.).
- Integrar controles de usuario en la simulación para permitir la interacción con la máquina virtual (botones, interfaz o teclado), y que sea posible comprender visualmente cómo se comporta cada componente.
- Fomentar el aprendizaje técnico y visual mediante una representación accesible, educativa y atractiva del funcionamiento de una fresadora CNC, tanto para estudiantes como para personas interesadas en automatización o diseño mecánico.

II. INTRODUCCIÓN

Las máquinas CNC (Control Numérico Computarizado) han revolucionado la industria manufacturera gracias a su precisión, repetitividad y capacidad de automatización. Entre ellas, la CNC modelo 1313 destaca por su estructura robusta y su uso en procesos de fresado, corte y grabado sobre materiales como madera, aluminio, acrílicos y plásticos industriales. Su diseño se basa en tres ejes principales (X, Y, Z) controlados mediante instrucciones codificadas (G-code), que permiten el movimiento de un husillo sobre una mesa de trabajo para esculpir piezas de forma precisa.

Esta máquina se compone de múltiples partes fundamentales, como:

- Base estructural
- Guías lineales y husillos
- Husillo o spindle
- Sistema de control electrónico
- Mesa de trabajo (tipo T Slot o vacío)

Este tipo de CNC se utiliza ampliamente en talleres de fabricación, carpintería digital, laboratorios de prototipado, y también en entornos educativos donde se enseña programación de G-code, control de máquinas y diseño asistido por computadora (CAD/CAM).

Motivación del Proyecto

Dado que el acceso a una máquina CNC real puede estar limitado por su costo o disponibilidad, el desarrollo de una simulación interactiva en Unity representa una herramienta poderosa para el aprendizaje. A través del modelado 3D, scripting y animación de componentes, es posible recrear una réplica virtual con funcionamiento similar, permitiendo visualizar cada movimiento, comprender cómo interactúan los sistemas mecánicos y electrónicos, y experimentar con el entorno sin riesgo ni gasto de materiales.

En este proyecto, se propone comparar directamente la estructura y funcionamiento de la máquina CNC 1313 real con su contraparte virtual creada en Unity, destacando tanto la fidelidad de los componentes como los desafíos técnicos implicados en la simulación. Esta comparación servirá como puente entre el mundo físico y digital, impulsando el desarrollo de habilidades técnicas aplicadas al diseño, simulación y automatización.

III. MARCO TEORICO

Una máquina CNC 1313 es una fresadora controlada por computadora que opera sobre tres ejes (X, Y, Z). Su nombre proviene de las dimensiones de trabajo: 1300 mm x 1300 mm, siendo ideal para cortar, tallar o grabar materiales como madera, MDF, acrílico, PVC, aluminio y más. Es ampliamente usada en la industria, el diseño, la ingeniería y la educación técnica.

Descripción Detallada de Componentes de la Máquina CNC 1313

Base estructural

- Descripción técnica: Fabricada generalmente en acero soldado, usando tubos de entre 6 y 12 mm de grosor. Puede estar sometida a tratamiento térmico para eliminar tensiones internas.
- Función principal: Soportar toda la estructura superior de la máquina y absorber vibraciones generadas durante el mecanizado.
- Ubicación**:** Parte inferior de la CNC.
- Importancia**:** Una base estable mejora la precisión general del sistema.
- Mantenimiento: Limpieza de polvo o residuos, verificación de anclaje al suelo si aplica.

Pórtico (Gantry)

- Descripción técnica: Estructura transversal móvil, hecha de acero o aluminio de alta resistencia. En CNCs 1313 industriales puede alcanzar longitudes mayores a 1.3 m.
- Función principal: Sostiene el eje Z y el husillo; se encarga del desplazamiento horizontal a lo largo del eje X (y también puede colaborar en el eje Y dependiendo del diseño).
- Ubicación**:** Parte superior de la estructura, apoyado sobre guías lineales del eje X.
- Importancia**:** Su rigidez afecta directamente la precisión del husillo.
- Mantenimiento**:** Lubricación de rieles de soporte, revisión de pernos de fijación.

Guías lineales

- Descripción técnica**:** Marca común HIWIN o THK, de tipo riel rectificado con patines de bolas, precisión ±0.01 mm.
- Función principal**:** Permitir movimientos suaves y precisos en los ejes X, Y y Z.
- Ubicación**:** A lo largo de cada eje, soportan al pórtico o al husillo.
- Importancia**:** Reducen fricción y mejoran la vida útil del sistema.
- Mantenimiento: Lubricación frecuente con grasa especial, limpieza de polvo o viruta.

Sistema de transmisión

- Descripción técnica**:**
  - Ejes X/Y: Piñones helicoidales acoplados a cremalleras.
  - Eje Z: Husillo de bolas de alta precisión.
- Función principal**:** Transmitir el movimiento giratorio del motor al desplazamiento lineal de los ejes.
- Ubicación: Junto a las guías, debajo o a un costado.
- Importancia: Determina la precisión de posicionamiento de la máquina.
- Mantenimiento: Ajuste de tensión, lubricación de engranajes y limpieza.

Motores

- Descripción técnica**:** Motores paso a paso (Leadshine, NEMA 34) o servomotores (Yaskawa, Delta), según el modelo.
- Función principal**:** Generar movimiento en los ejes mediante rotación controlada.
- Ubicación**:** Lateral o trasera de la base/pórtico.
- Importancia**:** Su torque y resolución influyen directamente en velocidad y precisión.
- Mantenimiento**:** Revisión de conexiones, temperatura y funcionamiento correcto.

Husillo (Spindle)

- Descripción técnica**:** Potencia entre 2.2 kW y 9 kW, hasta 24,000 RPM, refrigerado por agua o aire, con rodamientos cerámicos.
- Función principal: Hacer girar herramientas de corte para mecanizar el material.
- Ubicación: Suspendido del eje Z.
- Importancia: Es el componente que realiza el corte; su rigidez y velocidad determinan calidad de acabado.
- Mantenimiento: Revisión de rodamientos, limpieza del portaherramienta, control de temperatura.

Collet o portaherramienta

- Descripción técnica**:** Estándares como ER20 o ISO30; de acero endurecido.
- Función principal: Sujetar de forma centrada y segura la herramienta (fresa, broca, etc.).
- Ubicación: En el extremo inferior del husillo.
- Importancia: Un mal apriete puede generar vibraciones peligrosas.
- Mantenimiento: Limpieza diaria, revisión de desgaste, cambio periódico.

Mesa de trabajo

- Descripción técnica: Puede ser tipo T-slot (ranuras para pernos) o con zonas de vacío.
- Función principal: Sostener y fijar el material durante el mecanizado.
- Ubicación: Parte inferior del área de corte.
- Importancia: Si el material no está fijo, el trabajo será inexacto o fallido.
- Mantenimiento: Limpieza de residuos, revisión de ventosas/válvulas si es de vacío.

Controlador CNC

- Descripción técnica**:** Interfaces como Mach3, Syntec, NCStudio o DSP independientes.
- Función principal**:** Interpretar el archivo G-code y coordinar los movimientos de motores y husillo.
- Ubicación**:** Gabinete externo o integrado.
- Importancia**:** Es el “cerebro” del sistema.
- Mantenimiento**:** Verificación de software y firmware, revisión de entradas/salidas.

Pantalla o panel de control

- Descripción técnica: Botoneras, pantalla táctil o LCD, parada de emergencia (E-Stop), perillas.
- Función principal**:** Permitir la interacción manual con la máquina para encendido, calibraciones, monitoreo.

- Ubicación: Frontal o lateral de la estructura.
- Mantenimiento: Revisión de cables, limpieza de polvo, funcionamiento de E-Stop.

Sensores

- Descripción técnica**:** Sensores de fin de carrera, sondas de medición Z, sensores de aceite o refrigerante.
- Función principal: Detectar posiciones límites, medir altura de herramientas, mantener sistemas automáticos.
- Ubicación: En extremos de guías y cabezal.
- Mantenimiento: Verificación de funcionamiento y conexiones, limpieza regular.

Sistema eléctrico

- Descripción técnica: Incluye drivers, relés, fuentes conmutadas, cables de señal, cables de potencia, variadores (VFD).
- Función principal: Alimentar y proteger todos los componentes electrónicos.
- Ubicación: Dentro del gabinete o estructura eléctrica.
- Mantenimiento: Revisión de temperaturas, limpieza, verificar aislaciones y conexiones.

Sistema de Refrigeración del Husillo (Chiller)

- Descripción técnica: Unidad externa con bomba, depósito de agua refrigerada, sensor de temperatura y mangueras. Usado cuando el husillo es refrigerado por agua.
- Función principal: Mantener el husillo a temperaturas seguras durante el corte prolongado, evitando daños térmicos en los rodamientos.
- Ubicación: Externo, cerca de la máquina (bajo o a un costado).
- Importancia: Un sobrecalentamiento puede destruir el husillo en minutos.
- Mantenimiento: Verificación diaria del nivel de agua, cambio del líquido cada 2–4 semanas, limpieza de filtros y mangueras.

Funcionamiento básico paso a paso en la vida real:

1. Se diseña una figura en CAD y se convierte a G-code (instrucciones numéricas).
2. El controlador interpreta el G-code y envía señales a los drivers.
3. Los drivers controlan los motores para mover la herramienta a lo largo de los ejes.
4. El husillo realiza el corte mientras se mueve en trayectorias definidas.
5. Finalizado el proceso, se obtiene la figura deseada sobre el material.

Proceso simulado en Unity (equivalente al real):

1. Se carga el G-code desde un archivo.
2. Se parsean las líneas para obtener posiciones (X, Y, Z) y si es o no corte.
3. Se muestran las trayectorias con LineRenderer.
4. Al iniciar la simulación, la herramienta se mueve a cada punto:
    - Si es G1, baja para cortar (trail visible).
    - Si es G0, se mueve rápido sin cortar (trail apagado).
5. Finalizado el corte, se apaga el trail y vuelve a la posición de reposo.

Glosario Técnico

Término	Definición
CNC	Control Numérico Computarizado. Sistema automatizado para control de máquinas.
Husillo	Motor que gira la herramienta de corte a alta velocidad.
Fresa	Herramienta rotativa utilizada para cortar materiales.
G-code	Lenguaje de programación para máquinas CNC.
Eje C	Cuarto eje rotativo para piezas cilíndricas.
ATC	Auto Tool Changer, sistema de cambio automático de herramientas.
TrailRenderer	Componente de Unity que visualiza trayectorias de objetos en movimiento.
Parada de emergencia	Botón que detiene toda la operación en caso de riesgo.

¿Qué es el G-code?

El G-code es un lenguaje estándar para controlar máquinas CNC. Consiste en líneas de comandos como G0, G1, X, Y, Z, que indican movimientos:

- G0: Movimiento rápido sin corte.
- G1: Movimiento con corte (la herramienta baja).
- X, Y, Z: Coordenadas del destino.

Ejemplo:

```
G0 X-50 Y-220 Z-50    // Movimiento rápido sin cortar
G1 X50 Y-220 Z-50     // Corta en X
G1 X50 Y-220 Z50      // Sube en Z mientras sigue en X
G1 X-50 Y-220 Z50     // Regresa en X con la herramienta arriba
G1 X-50 Y-220 Z-50    // Baja la herramienta otra vez para cortar
```

Este código traza una figura rectangular en el plano, bajando la herramienta para cortar solo cuando se usa G1.

Simulación de la CNC en Unity

Con el objetivo de representar fielmente esta máquina en un entorno digital, se implementó un modelo funcional en Unity, donde se simula tanto el movimiento como la ejecución del G-code.

Componentes simulados:

- mov_izq_der, mov_arrib_abajo, movimiento_adel_atr: representan los movimientos en los ejes X, Y y Z respectivamente.
- TrailRenderer: dibuja una línea visible cuando la herramienta está cortando.
- InputFields: permiten ingresar coordenadas manualmente para simular cortes personalizados.

Scripts principales usados:

- **GcodeParser.cs**: interpreta el archivo .gcode, identifica si es corte (G1) o solo movimiento (G0) y lo convierte en coordenadas que puede entender Unity.
- **CncSimulator.cs**: mueve los ejes uno por uno hacia cada punto, bajando el eje Z si hay corte y subiéndolo al finalizar.
- **CoordinateInputManager.cs**: gestiona el ingreso manual de coordenadas y carga el G-code para iniciar la simulación.

Seguridad y Condiciones de Operación

El uso de una máquina CNC requiere ciertas condiciones para asegurar una operación segura y eficiente. Las siguientes normas deben observarse tanto en la máquina real como en simulaciones con propósitos educativos:

- Usar gafas de protección, orejeras y guantes antideslizantes cuando se opere la máquina.
- Verificar el correcto funcionamiento de los sensores de fin de carrera antes de iniciar un trabajo.
- Activar el botón de parada de emergencia en caso de desplazamientos inesperados.
- Mantener despejada el área de corte y revisar que el material esté correctamente fijado.
- No tocar la herramienta ni el material mientras el husillo esté en rotación.
- Apagar la máquina antes de realizar ajustes o mantenimiento.

Mantenimiento Preventivo de la CNC

El mantenimiento periódico garantiza el rendimiento y prolonga la vida útil de la máquina CNC 1313:

- Lubricar guías lineales, husillo y cremallera cada 40 horas de trabajo.
- Revisar tensión de bandas o piñones semanalmente.
- Limpiar residuos del sistema de vacío y mesa de trabajo después de cada uso.
- Verificar temperatura del husillo durante trabajos prolongados.
- Sustituir herramientas desgastadas para mantener la calidad de corte.

Análisis de Materiales Compatibles

Material	Dureza relativa	Velocidad recomendada (RPM)	Fresa sugerida
MDF	Baja	16,000 – 20,000	Carburo de 2 filos rectos
Acrílico	Media	18,000 – 22,000	Punta pulida o de un solo filo
Aluminio	Alta	12,000 – 16,000	Carburo de un solo filo
Plástico	Baja	14,000 – 20,000	Fresa en espiral
Compósitos	Media	15,000 – 18,000	Punta de diamante policristalino

Integración del Proyecto CNC en el Aula Virtual

Como parte de una propuesta educativa innovadora, la simulación de la máquina CNC 1313 fue integrada dentro de un entorno de aula virtual interactiva, con el objetivo de complementar la formación técnica de los estudiantes mediante el uso de herramientas digitales modernas.

Esta aula virtual no solo sirve como un repositorio estático de información, sino como una plataforma dinámica y estructurada, diseñada para que el estudiante explore la máquina desde tres enfoques clave:

1. Comprensión teórica profunda, a través de contenidos técnicos y visuales sobre la máquina real.
2. Exploración visual guiada, mediante recursos interactivos que permiten identificar partes y funciones.
3. Experimentación práctica, gracias a una simulación en 3D que emula el funcionamiento real de la fresadora CNC dentro de Unity.

IV. METODOLOGIA

Metodología técnica

Estructura general del proyecto

1. Carpeta carcasa

Esta carpeta contiene archivos .dae (Digital Asset Exchange) que representan elementos visuales estáticos, utilizados para simular visualmente la carcasa externa de la máquina CNC. Estos objetos no forman parte de la lógica de movimiento, sino que sirven como elementos de ambientación y contexto. Es común importar modelos 3D externos en formato .dae cuando se quiere preservar la geometría original exportada desde software CAD como Blender, Fusion 360 o SketchUp.

2. Carpeta materials

Esta carpeta contiene los materiales utilizados en el proyecto. Los materiales definen el aspecto visual de los objetos, como colores, texturas, transparencia y brillo. Por ejemplo, un material puede definir que una pieza tenga apariencia metálica o plástica.

3. Carpeta modelo

Aquí se encuentran los modelos 3D base de la máquina CNC. Estos fueron descargados desde el sitio cults3d.com y convertidos a un formato compatible con Unity (como .fbx, .obj o .dae). Posteriormente, estos modelos fueron ensamblados a mano en Unity para representar fielmente la estructura de una máquina CNC real. El ensamble virtual conserva la lógica estructural realista de una máquina con perfiles de aluminio y partes impresas en 3D.

4. Carpeta Resources/gcodes

La carpeta Resources es una carpeta especial en Unity que permite acceso dinámico a archivos durante la ejecución mediante el método Resources.Load<T>(). Es por esta razón que el archivo ejemplo.gcode debe estar dentro de Resources/gcodes. Los archivos fuera de Resources no pueden ser cargados dinámicamente en tiempo de ejecución con este método.

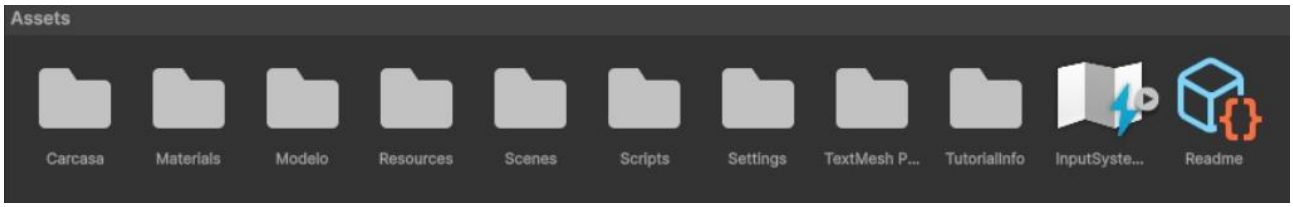
Dentro de gcodes, se encuentra el archivo de texto ejemplo.gcode, que contiene líneas de comandos G-code utilizados por el simulador para ejecutar trayectorias de corte. Este archivo puede ser sobrescrito o simulado con código G ingresado manualmente.

5. Carpeta Scenes

Contiene la escena principal del proyecto Unity, donde se ensamblan todos los componentes visuales, interactivos y de lógica. Esta escena es la que se carga al inicio y contiene todo el entorno del simulador CNC.

6. Carpeta Scripts

Contiene todos los scripts en C# que gestionan el comportamiento del simulador. Se describen a continuación con profundidad.



Funcionamiento detallado de los scripts

- *CncSimulator.cs*

Este script controla el **movimiento del cabezal CNC en los ejes X, Y y Z** y gestiona cuándo debe activarse o desactivarse el rastro de corte (TrailRenderer).

- mov_izq_der: movimiento horizontal (eje X)
- mov_arrib_abajo: movimiento vertical del cabezal (eje Y)
- movimiento_adel_atr: desplazamiento sobre la mesa (eje Z)

El movimiento está limitado mediante valores limiteX, limiteY y limiteZ para evitar salidas del área de trabajo.

El método MoverAHacia inicia el desplazamiento a un nuevo punto, y determina si se debe bajar el cabezal (esCorte) para cortar o no.

El TrailRenderer simula el trazo del corte solo si esCorte == true. Una vez se alcanza el destino, si hubo corte, el cabezal se eleva automáticamente para evitar marcas no deseadas.

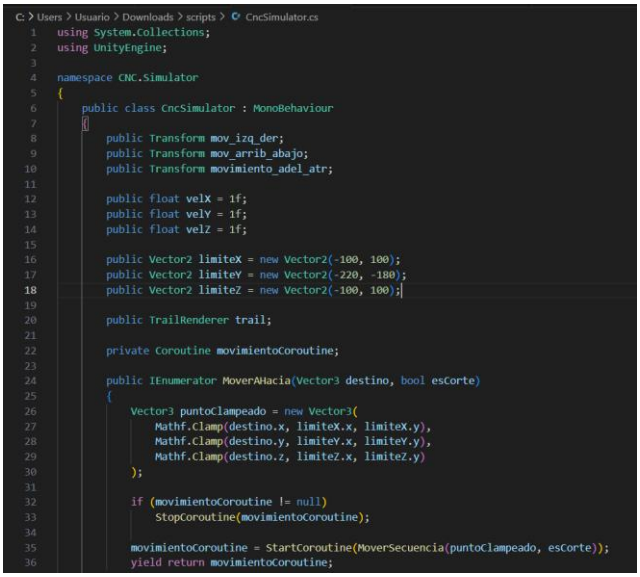
Un ejemplo parctico es este script mueve el cabezal de la máquina sobre los tres ejes (X, Y, Z) respetando límites físicos como:

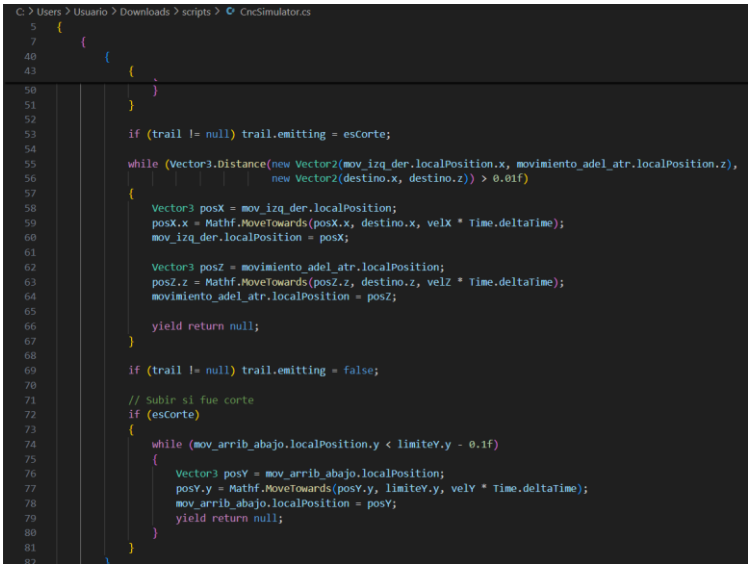
public Vector2 limiteX = new Vector2(-100, 100);

Su función principal MoverAHacia(Vector3 destino, bool esCorte) ejecuta la secuencia completa de mecanizado:

- Si esCorte == true, baja el cabezal hasta -220 (posición de corte).
- Emite un **TrailRenderer** (línea visible) durante el desplazamiento.

- Finaliza subiendo el cabezal para evitar marcas indeseadas





- *CoordinateInputManager.cs*

Este script gestiona:

- la entrada manual de coordenadas
- la carga del archivo G-code desde Resources
- la ejecución del proceso de corte
- el reinicio de la escena

Gestiona la carga y ejecución del G-code. Tiene dos formas:

- **Automática**: lee el archivo ejemplo.gcode desde Resources.
- **Manual**: interpreta líneas pegadas por el usuario como G0 X20 Y-190 Z0.

Función clave:

void CargarYCortarDesdeGcode() // Ejecuta la simulación desde archivo

Al finalizar el proceso, oculta la interfaz y eleva los ejes para mejorar la visualización .

Funciones destacadas:

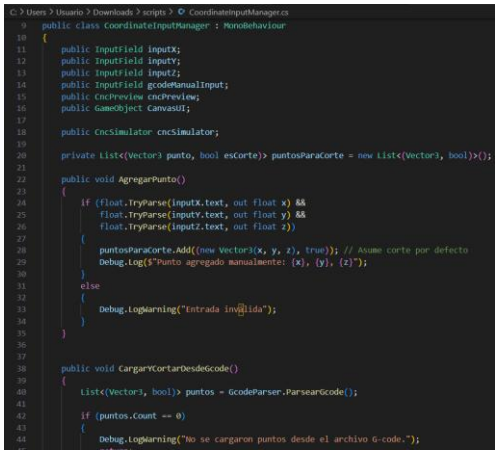
- CargarYCortarDesdeGcode(): lee y ejecuta el G-code del archivo ejemplo.gcode
- CargarDesdeInputManual(): permite al usuario pegar texto G-code manualmente, el cual es interpretado línea por línea
- ProcesarCorte(): recorre la lista de puntos obtenida desde el G-code e inicia la simulación

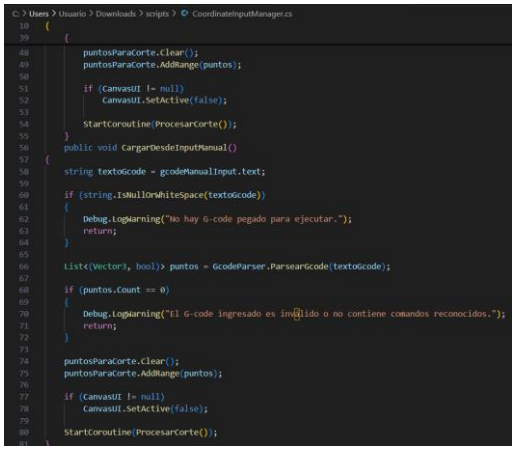
Al finalizar el corte:

1. Se eleva el eje Y a -190 para asegurar que no se siga cortando

2. Se mueve el eje Z a 200 para alejar visualmente la herramienta
3. Se espera 5 segundos y se reactiva el CanvasUI

También provee un método ReiniciarEscena() que recarga la escena principal completamente, borrando el TrailRenderer y reiniciando todo el estado de la máquina.





- *GcodeParser.cs*

Este script se encarga de leer y traducir líneas de G-code en coordenadas 3D que el simulador puede interpretar.

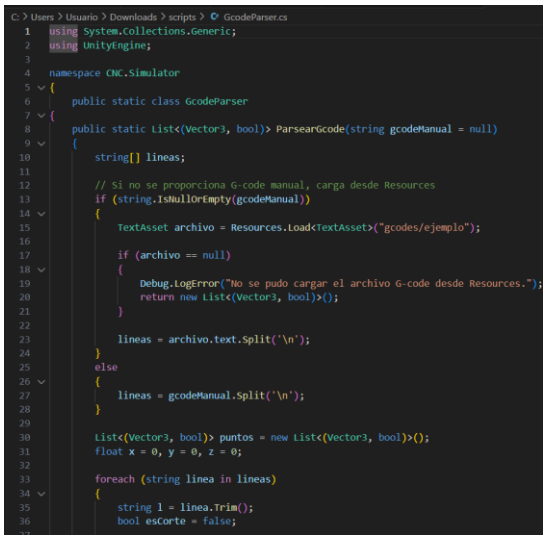
Cada línea G0 (movimiento sin corte) o G1 (movimiento con corte) es transformada en una tupla (Vector3 punto, bool esCorte).

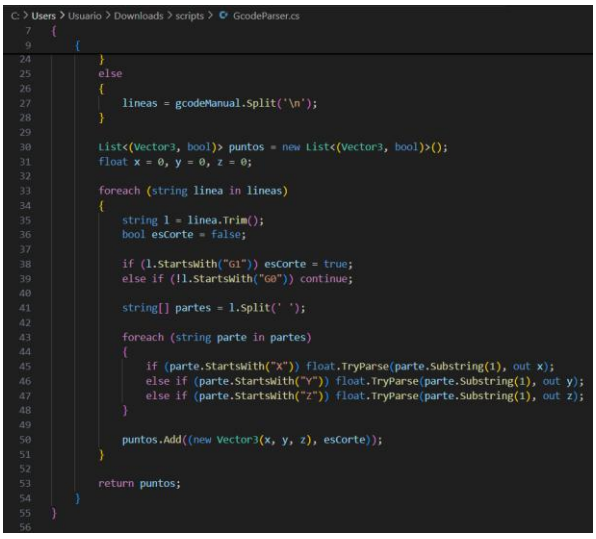
El parser reconoce los comandos G0 y G1 y extrae los valores de X, Y y Z desde el texto. También permite parsear tanto desde archivos como desde texto pegado por el usuario.

Es decir, traduce instrucciones como G1 X10 Y-200 Z50 a coordenadas:

(Vector3 punto, bool esCorte) = (new Vector3(10, -200, 50), true);

Distingue entre G0 (movimiento sin corte) y G1 (corte real) y devuelve una lista ordenada para ejecutar.





- *Rotador.cs*

Script simple que rota el objeto al que está asignado, simulando la herramienta giratoria del cabezal CNC.

transform.Rotate(new Vector3(0,0,1) * velocidadRotacion * Time.deltaTime);

Esto permite una visualización más realista del cabezal en movimiento.

C: > Users > Usuario > Downloads > scripts > Rotador.cs

```
using UnityEngine;

public class Rotador : MonoBehaviour
{
    public float velocidadRotacion = 360f; // grados por segundo

    void Update()
    {
        transform.Rotate(new Vector3(0,0,1) * velocidadRotacion * Time.deltaTime);
    }
}

```

- *CameraPanning.cs*

Este script ejecuta un movimiento suave de cámara desde un punto inicial a uno final (startPoint → targetPoint) al iniciar la escena. Es útil para presentar la máquina de forma dinámica al usuario.

C: > Users > Usuario > Downloads > scripts > CameraPanning.cs

```
using UnityEngine;

public class CameraPanning : MonoBehaviour
{
    public Transform startPoint;
    public Transform targetPoint;
    public float duration = 3.0f; // Tiempo del paneo
    private float elapsedTime = 0f;
    private bool isPanning = true;

    void Start()
    {
        // Posiciona la cámara en el punto inicial
        transform.position = startPoint.position;
        transform.rotation = startPoint.rotation;
    }

    void Update()
    {
        if (!isPanning) return;

        elapsedTime += Time.deltaTime;
        float t = Mathf.Clamp01(elapsedTime / duration);

        // Movimiento suave de posición y rotación
        transform.position = Vector3.Lerp(startPoint.position, targetPoint.position, t);
        transform.rotation = Quaternion.Slerp(startPoint.rotation, targetPoint.rotation, t);

        if (t >= 1.0f)
        {
            isPanning = false; // Detener el movimiento
        }
    }
}

```

- *AbrirPDF.cs*

Permite abrir un documento PDF con instrucciones o información adicional. Detecta automáticamente la plataforma (Windows, Linux, macOS) y abre el archivo ubicado en StreamingAssets.

Funcionamiento general del sistema

1. Al iniciar la escena, la cámara hace un paneo hasta enfocar la máquina.
2. El usuario puede:
    - Ingresar coordenadas manuales (X, Y, Z)
    - Cargar un archivo G-code
    - Pegar texto G-code manualmente
3. Se genera una lista de puntos que el simulador CNC debe seguir.
4. Al comenzar el corte:
    - El CanvasUI desaparece
    - La máquina recorre los puntos, bajando el cabezal solo si el punto requiere corte (G1)
    - El TrailRenderer simula el trazo del corte
5. Al finalizar:
    - El eje Y se eleva (cabezal arriba)
    - El eje Z sube completamente (visión clara)
    - El CanvasUI reaparece tras 5 segundos
6. El botón “Reiniciar” permite volver al estado inicial.

Exportación a webgl

Al exportar a WebGL, se deben tener en cuenta:

- Compresión de archivos**:** WebGL usa compresión gzip o Brotli. El servidor debe estar configurado para enviar la cabecera Content-Encoding: gzip o br si los archivos están comprimidos.
- StreamingAssets no funciona igual: Los archivos como PDF deben cargarse desde rutas accesibles vía HTTP.
- Uso de Resources**:** Es compatible con WebGL y permite cargar gcodes/ejemplo.gcode directamente.

Metodología general

Análisis técnico de la máquina real

Se inició con un estudio detallado de la estructura mecánica y electrónica de la CNC 1313 real, recopilando información sobre sus:

- Componentes físicos (base, pórtico, husillo, motores, guías, etc.).
- Especificaciones técnicas (materiales, potencias, tipos de movimiento).
- Principios de funcionamiento y secuencia operativa.

Este análisis permitió definir qué elementos serían representados gráficamente y qué acciones debían ser simuladas mediante scripts en Unity.

Modelado 3D de componentes en Unity

Se creó una nueva escena en Unity 3D para comenzar con el modelado estructural.
El diseño se dividió en partes funcionales que reflejan la disposición de una CNC real:

Elemento	Descripción en Unity
Base	Cubo escalado como plataforma estática
Pórtico (gantry)	Objeto móvil sobre eje X
Husillo (spindle)	Cilindro o bloque que se desplaza en eje Z
Mesa de trabajo	Superficie con textura metálica, bajo el recorrido
Cabezal de corte	Componente visual con TrailRenderer para simular el corte

Todos los objetos fueron organizados jerárquicamente en la escena para imitar la relación física entre partes (pórtico soporta eje Z, husillo cuelga del eje Z, etc.).

Estructura del proyecto en Unity
El proyecto se organizó dentro del motor Unity en una estructura modular que incluye carpetas como Assets, Scenes, Scripts, Prefabs, Resources y Materials.

- Dentro de Assets se concentraron los elementos visuales y lógicos.
- En Scenes se almacenó la escena principal del simulador CNC.
- La carpeta Scripts contiene los controladores funcionales como GcodeParser.cs, CncSimulator.cs y CoordinateInputManager.cs.
- En Prefabs se definieron elementos reutilizables como el husillo, la base, el cabezal o el sistema de ejes.
- Resources incluye el archivo de ejemplo .gcode cargado por el parser, y Materials contiene los shaders y colores básicos usados en la simulación.

Versión de Unity y herramientas utilizadas
Para el desarrollo se utilizó Unity versión 2022.3.x LTS, que ofrece estabilidad y soporte a largo plazo. Se trabajó con el pipeline Built-in, aunque se considera migrar a URP (Universal Render Pipeline) para mejorar la calidad visual.
Además, se usaron herramientas y paquetes como:

- **TextMeshPro** para los textos de la interfaz
- **Cinemachine** (opcional) para controlar cámaras dinámicas
- **TrailRenderer** como componente para representar el corte en tiempo real
- **Input System** nativo de Unity para manejar interacción con el teclado o UI

Asignación de componentes móviles

Cada parte fue asignada a una variable transform en el script CncSimulator.cs:

```
public Transform mov_izq_der;      // Eje X
public Transform mov_arrib_abajo;  // Eje Y
public Transform movimiento_adel_atr; // Eje Z
```

Esto permitió controlar de forma separada el desplazamiento en cada eje, simulando el comportamiento real de la máquina. Además, se definieron **límites de movimiento** para cada eje mediante Vector2 limite X/Y/Z.

Lectura e interpretación de G-code

Para convertir comandos numéricos en movimientos reales, se creó el script GcodeParser.cs, que:

- Carga un archivo de texto (.gcode) desde la carpeta Resources.
- Interpreta cada línea con comandos como G0 o G1.
- Extrae coordenadas X, Y, Z.
- Identifica si es corte (G1) o solo movimiento sin cortar (G0).

El resultado es una lista de puntos con bandera esCorte, como este ejemplo:

List<(Vector3, bool)> puntos = [(X, Y, Z), esCorte];

Simulación del corte

Al ejecutar el corte, el script CncSimulator.cs realiza:

- Movimiento hacia el punto deseado en X y Z**.**
- Si el punto es de corte (G1):
    - Baja el eje Y para simular que la herramienta toca el material.
    - Activa el TrailRenderer para dejar una línea visible.
- Al finalizar:
    - Sube el eje Y.
    - Desactiva el trail.

Este comportamiento refleja el proceso real de fresado, bajando la herramienta únicamente cuando hay material que cortar.

Scripts principales y componentes usados
La lógica principal de la simulación se basa en tres scripts clave:

- GcodeParser.cs: interpreta comandos G0 y G1, extrae coordenadas y genera una lista estructurada de puntos.
- CncSimulator.cs: controla los movimientos por ejes simulados (X, Y, Z), activa el TrailRenderer y simula el comportamiento del cabezal.
- CoordinateInputManager.cs: permite ingresar coordenadas manualmente y ejecutar recorridos personalizados.
  Además, se utilizaron componentes como TrailRenderer (para el trazo de corte), Transform (para desplazamientos), InputField (para capturas de usuario), Button (para disparar acciones) y elementos de UI nativa de Unity.

Entrada manual de coordenadas

Además del G-code, se programó una interfaz para permitir ingresar coordenadas manualmente mediante campos InputField (inputX, inputY, inputZ) en el script CoordinateInputManager.cs.

- El botón Agregar Punto guarda los valores en una lista de corte.
- El botón Cortar ejecuta la secuencia sobre los puntos ingresados.

Esto permite simular trayectorias simples sin necesidad de cargar archivos.

Exportación WebGL e integración al aula virtual

Una vez completada la simulación, se exportó el proyecto a formato WebGL, para integrarlo a una página HTML interactiva accesible desde el aula virtual.

La exportación se configuró con:

- Resolución adaptable a navegador.
- Menor uso de recursos para compatibilidad con equipos educativos.

En la página HTML se añadieron:

- Un botón para lanzar la simulación.
- Información contextual.
- Documentación y exploración visual.

Flujo de trabajo en Unity (Editor, Game/Scene, iluminación)

Durante el desarrollo se trabajó simultáneamente en las pestañas Scene y Game para ajustar visualmente los movimientos, trayectoria y espacio de trabajo.
Se utilizó una cámara principal en vista isométrica, que permite observar la máquina desde una perspectiva cómoda para análisis educativo.
La iluminación fue básica, con una fuente direccional blanca simulando luz de taller, aunque se deja abierta la posibilidad de implementar iluminación industrial y HDR para futuras versiones.
Todos los objetos fueron organizados jerárquicamente en la escena para reflejar la cinemática real (por ejemplo, el husillo cuelga del eje Z, que a su vez es arrastrado por el pórtico en eje X).

Validación y mejoras

Se realizaron pruebas funcionales para verificar:

- Precisión de trayectorias.
- Comportamiento esperado del movimiento y corte.
- Respuesta a errores (valores fuera de límites, entrada vacía, etc.).

Se anotaron posibles mejoras para versiones futuras:

- Incluir más figuras.
- Añadir botón de emergencia.
- Mejorar animación del husillo.

Acceso al manual de ejemplos de G-code

Como complemento a la simulación, se habilitó una opción dentro del aula virtual que dirige al estudiante a un manual práctico con 30 ejemplos de G-code listos para usar. Este recurso tiene como objetivo fomentar la experimentación y el aprendizaje autónomo.

Cada ejemplo contiene:

- Código G-code preconfigurado (formas simples, figuras geométricas, letras, trayectorias personalizadas).
- Breve descripción del diseño.
- Captura del resultado esperado.
- Instrucciones para copiar y pegar el código en el archivo ejemplo.gcode.

El acceso a este manual se encuentra en la misma página HTML del paisaje interactivo, y permite que los estudiantes:

- Practiquen distintos recorridos de corte sin necesidad de escribir el código desde cero.
- Visualicen cómo cada instrucción se traduce en movimiento real dentro de la simulación.
- Comprendan la lógica detrás de la programación CNC mediante ensayo y error.

Este recurso convierte la simulación en una herramienta altamente didáctica y flexible, ideal para cursos de automatización, manufactura digital o programación de máquinas CNC.

V. RESULTADOS

Como resultado de este proyecto, se logró desarrollar e integrar una simulación funcional de la máquina CNC 1313 utilizando el motor gráfico Unity, con representación precisa de componentes, movimientos y ejecución real de comandos G-code, todo dentro de un entorno de aula virtual interactiva. A continuación se detallan los principales resultados alcanzados:

Simulación 3D realista de la CNC 1313

- Se recrearon los movimientos independientes de los tres ejes (X, Y, Z) respetando sus límites reales.
- El sistema replica de forma visual el funcionamiento del husillo, con descenso cuando corta y ascenso cuando no.
- Se incorporó un efecto de trazo mediante TrailRenderer, que deja una línea visible solo cuando hay corte, simulando el mecanizado real.

Interpretación de G-code funcional

- Se logró desarrollar un parser de G-code que identifica instrucciones tipo G0 (movimiento rápido sin corte) y G1 (movimiento con corte).
- La simulación responde correctamente a la secuencia de coordenadas cargadas desde el archivo .gcode, ejecutando trayectorias precisas.
- Se probó exitosamente con múltiples figuras de prueba (cuadrados, triángulos, líneas, patrones de elevación).

Interfaz interactiva para ingresar coordenadas

- Se creó una UI amigable y funcional, que permite al estudiante:
    - Ingresar manualmente coordenadas (X, Y, Z).
    - Visualizar una vista previa del recorrido antes de cortar.
    - Ejecutar el movimiento paso a paso con un solo clic.

Acceso directo a 30 ejemplos de G-code

- Se integró un manual interactivo con 30 diseños prediseñados en G-code, accesible desde la misma aula virtual.
- Esto permite a los estudiantes practicar sin necesidad de escribir código desde cero, fomentando el aprendizaje por observación y prueba.

Exportación WebGL e integración a la plataforma educativa

- El proyecto fue exportado en formato WebGL, permitiendo su ejecución desde cualquier navegador, sin necesidad de instalar software adicional.
- La simulación fue integrada al aula virtual, acompañada de documentación técnica, exploración visual y otros recursos formativos (PDF, panel UI, ficha técnica, etc.).

Comprobación de fidelidad funcional

- Se logró una correspondencia lógica y visual entre el modelo real y el digital, cumpliendo con los principios de movimiento y corte explicados en el marco teórico.
- Se validó la utilidad del sistema como herramienta de apoyo educativo en temas de manufactura digital y automatización.

VI. DISCUCIONES

El desarrollo e implementación de la simulación 3D de la máquina CNC 1313 en Unity permitió comprender y representar digitalmente los principales procesos de una fresadora industrial de forma funcional, interactiva y educativa. A través de esta experiencia, se obtuvieron aprendizajes importantes tanto en lo técnico como en lo pedagógico.

1. Ventajas de la simulación desarrollada

- La simulación replicó los movimientos independientes de los tres ejes, el comportamiento del husillo y la ejecución precisa de instrucciones G-code, generando una experiencia cercana a la operación real de una CNC.
- El usuario puede interactuar con la simulación tanto cargando archivos .gcode como ingresando coordenadas manualmente, lo cual amplía su utilidad didáctica.
- Al exportarse en formato WebGL, la simulación no requiere instalación, lo que la hace ideal para entornos educativos con equipos limitados.
- La inclusión de un manual con 30 ejemplos de G-code ofrece al estudiante una herramienta directa para practicar y visualizar cómo se comporta el código en una máquina virtual.

2. Limitaciones identificadas

- En una CNC real, el husillo puede realizar cambio automático de herramientas (conos ISO30, ER20, etc.). Esta funcionalidad no fue incluida en la simulación actual.
- Aunque funcional, la apariencia visual de la simulación puede mejorar. Actualmente no se utilizan texturas metálicas realistas ni efectos de iluminación que imiten acero, aluminio cepillado o pintura industrial.
- El efecto de corte está representado solo por un trazo visual (trail), pero no hay interacción física real con el material ni una animación de viruta o proceso de desbaste.
- Aunque funcional, la UI puede expandirse con más botones, leyendas dinámicas, controles de velocidad, y paneles informativos.

3. Propuestas de mejoras futuras

Área	Mejora propuesta
Visual	Aplicar texturas metálicas realistas (URP o HDRP), iluminación industrial, sombreado.
Funcionalidad	Simular cambio de herramienta (ATC), añadir más tipos de herramientas.
Corte realista	Animación de material eliminado, capas de profundidad, partículas.
UI	Panel lateral con etiquetas de cada eje, botón de reinicio, parada de emergencia virtual.
Control manual	Agregar sliders o joysticks virtuales para mover la máquina en tiempo real sin G-code.
Compatibilidad	Subir múltiples archivos .gcode, o permitir arrastrar/soltar desde el navegador.
Evaluación	Agregar un sistema que indique si el G-code fue bien ejecutado, o con errores.

4. Impacto educativo del proyecto

La simulación desarrollada no solo replica una máquina CNC, sino que transforma su complejidad en una experiencia comprensible y visual. Esto representa un gran avance para la enseñanza de manufactura digital, ya que:

- Permite a los estudiantes experimentar sin miedo a dañar una máquina real.
- Democratiza el acceso al conocimiento, ya que funciona en línea y es gratuita.
- Estimula el pensamiento computacional, al conectar código (G-code) con movimientos físicos.

VII. CONCLUSIONES

Se logró diseñar una réplica virtual y funcional de la máquina CNC 1313, respetando tanto su estructura mecánica como su comportamiento operativo. La simulación desarrollada en Unity permitió representar fielmente los movimientos en los ejes X, Y y Z, así como la ejecución de trayectorias definidas mediante instrucciones G-code, cumpliendo con el objetivo general del proyecto.

Se analizaron y modelaron los componentes fundamentales de la máquina real**,** como la base estructural, el pórtico, el husillo, el sistema de transmisión, los motores paso a paso y la mesa de trabajo, asegurando una correspondencia visual y funcional precisa en el entorno virtual.

Se caracterizó adecuadamente el sistema de control electrónico, replicando mediante scripts la lógica de interpretación de comandos, el movimiento del husillo según los ejes, y la activación del corte solo cuando es necesario, tal como ocurre en una CNC real.

Se documentó y simuló el sistema de sujeción de piezas, incorporando una mesa representativa sobre la cual se realizan los cortes en la simulación, reforzando el realismo del proceso de mecanizado.

Se desarrollaron scripts personalizados en Unity (como GcodeParser.cs**,** CncSimulator.cs**,** y CoordinateInputManager.cs) que permiten cargar archivos .gcode, visualizar trayectorias, ejecutar cortes paso a paso e ingresar coordenadas manualmente, logrando una interacción funcional completa.

Se integró la simulación en un aula virtual interactiva, accesible desde navegador mediante exportación WebGL. Esta aula incluye marco teórico, exploración por partes, panel de controles, documentación técnica, video explicativo y un manual con 30 ejemplos de G-code, fortaleciendo el aprendizaje visual y práctico del estudiante.

El proyecto cumplió con los estándares técnicos esperados y demostró su utilidad como recurso educativo, permitiendo al usuario comprender el funcionamiento de una máquina CNC real sin necesidad de contacto físico con una, reduciendo riesgos, costos y barreras de acceso.

A pesar de sus limitaciones visuales o funcionales menores, la simulación representa un avance significativo en la enseñanza de manufactura digital, y puede seguir evolucionando con la inclusión de texturas avanzadas, efectos de corte más realistas, cambio de herramientas virtual, y controles manuales interactivos.

VIII. RECOMENDACIONES

Profundizar en el desarrollo visual del modelo 3D
Se recomienda implementar texturas realistas (metal, aluminio cepillado, pintura industrial) y mejorar la iluminación general del entorno para incrementar la inmersión y el atractivo visual de la simulación. Esto puede realizarse con URP o HDRP en Unity.

Agregar simulación de cambio de herramientas (ATC)
Para enriquecer el realismo del modelo, futuros equipos pueden desarrollar un sistema de cambio automático de herramientas, que incluya animaciones y selección virtual de fresas según el tipo de operación.

Expandir la UI con controles avanzados
Se sugiere mejorar la interfaz gráfica añadiendo:

- Botón de parada de emergencia.
- Indicadores en pantalla de posición actual del cabezal.
- Velocidad ajustable del movimiento.
- Modo paso a paso para observación detallada.

Incluir simulación física del corte
Se recomienda trabajar en la simulación del efecto real del mecanizado, agregando animaciones de viruta, profundidad de corte y capas de material para representar mejor el trabajo sobre madera, acrílico o aluminio.

Desarrollar un módulo de evaluación automática
Futuros estudiantes pueden crear un sistema que valide el G-code cargado y notifique errores de sintaxis, coordenadas fuera de rango, o incluso “evaluar” si el diseño fue ejecutado correctamente según un patrón.

Ampliar la biblioteca de ejemplos G-code
Es recomendable extender el manual con más ejemplos avanzados (curvas Bézier, nombres personalizados, figuras 3D simples), y permitir su carga directa desde el aula virtual sin modificar archivos locales.

Optimizar para dispositivos móviles o de bajos recursos
Se sugiere explorar la compatibilidad con tablets o Chromebooks, reduciendo el uso de polígonos y ajustando la exportación WebGL para garantizar un mejor rendimiento.

Documentar cada cambio realizado
Para mantener la continuidad del proyecto, todo nuevo aporte debe ser documentado en un repositorio, con versiones de Unity usadas, estructura de carpetas, scripts modificados y mejoras aplicadas.

Mantener el enfoque educativo y colaborativo del proyecto
Finalmente, se invita a los futuros grupos a seguir trabajando con la misma visión formativa, compartiendo conocimiento, documentando avances y mejorando lo que ya se ha construido con esfuerzo y dedicación.

IX. REFERENCIAS

Fuentes técnicas sobre maquinaria CNC

- Wattsan. (s.f.). *Máquina CNC modelo M1 1313*. Recuperado de: https://wattsan.com/product/wattsan-m1-1313/
- Virmer. (s.f.). *Wattsan A1 1313 Router CNC*. Recuperado de: https://virmer.com/catalog/cnc-milling-machines/cnc-router-machine-wattsan-a1-1313/
- Marathon OS. (s.f.). *Standard Mechanical Components of CNC Routers*. Recuperado de: https://marathon-os.com/library/cnc-router-standard-mechanical-component-6816586e8cf03b81bb7ccc73

Catálogos y proveedores de routers CNC

- iGolden CNC. (s.f.). *CNC Wood Router Machine 1313*. Recuperado de: https://www.igolden-cnc.com/product/1313-small-cnc-wood-router-machine/
- Forsun CNC. (s.f.). *CNC Router con ATC y mesa de vacío*. Recuperado de: https://forsuncnc.com/

Repositorios de modelos 3D para simulación

- 3D CAD Browser. (s.f.). *CNC Router 3D Models*. Recuperado de:
  - https://www.3dcadbrowser.com/3d-models/cnc
  - https://www.3dcadbrowser.com/3d-model/cnc-machine-router-from-wood
- Free3D. (s.f.). *Modelos 3D de máquinas CNC*. Recuperado de: https://free3d.com/es/modelos-3d-premium/máquina-cnc
- GrabCAD. (s.f.). *Biblioteca de diseños CNC*. Recuperado de:
  - https://grabcad.com/library/tag/cnc%20router
  - https://grabcad.com/library/cnc-router-best-cnc-router-for-furniture-1
- MyMiniFactory. (s.f.). *Triple CNC Machine – Proyecto imprimible en 3D*. Recuperado de: https://www.myminifactory.com/object/3d-print-triple-cnc-machine-78412

X. ANEXOS

- Presentación de la exposición
  https://gamma.app/docs/Proyecto-Replica-Virtual-de-Maquina-CNC-1313-9dwljqft2sucoul

- Maquina cnc 1313 gemelo digital y real

- Video de funcionamiento de la maquina cnc 1313 del gemelo digital
  https://drive.google.com/file/d/1YUHjWetcHADeOruL-JSTJarrYCXJUiX3/view?usp=sharing