



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

**PodaRV: Capacitación en Poda de Cerezos
mediante Realidad Virtual**

DIEGO ALBERTO SANTANDER SEPÚLVEDA

Profesor Guía: FELIPE BESOAIN PINO

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

Talca – Chile
Junio, 2025



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

**PodaRV: Capacitación en Poda de Cerezos
mediante Realidad Virtual**

DIEGO ALBERTO SANTANDER SEPÚLVEDA

Profesor Guía: FELIPE BESOAÍN PINO

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

El presente documento fue calificado con nota: _____

Talca – Chile

Junio, 2025

*En dedicatoria a mi querida prima Francisca, que siempre vivirá en mi corazón.
Que la paz y el amor te acompañen dondequiera que estés. Tu persona siempre
estará presente en nuestros recuerdos. Que descanses en paz, y que el cariño de
todos los que te amamos te rodee eternamente.*

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Tabla de Contenidos	II
Índice de Figuras	VI
Resumen	VIII
 1. Introducción	 1
1.1. Contexto del Problema	2
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. Pregunta de Investigación	3
 2. Estado del Arte y Marco Teórico	 4
2.1. Poda en la Producción de Cerezos	6
2.2. Sistema de Crecimiento(KGB) con Técnicas de Poda	10
2.3. Incorporación de Tecnologías en la Poda de Cerezos	11
2.3.1. Realidad Virtual en un Entorno Agrícola Inmersivo	11
2.3.2. Generación Procedural de Árboles	12
 3. Marco Metodológico	 15
3.1. Metodología a Utilizar	15
3.2. Herramientas y Software a Utilizar	17
3.2.1. Motor de Videojuegos: Unity	17
3.2.2. Desarrollo de la Interfaz de Usuario: Unity UI	17
3.2.3. Realidad Virtual: Oculus SDK	17
3.2.4. Otras Herramientas y Recursos Utilizados	18
3.3. Características Metodológicas del Proyecto	18
3.3.1. Enfoque de Investigación	18
3.3.2. Población y Muestra	19

3.3.3.	Procedimiento de Recolección de Datos	20
3.3.4.	Análisis de Datos	20
3.3.5.	Consideraciones Éticas	21
3.3.6.	Limitaciones	21
3.3.7.	Validación	21
4.	Diseño del Sistema	23
4.1.	Casos de Uso	23
4.2.	Diseño Funcional de los Módulos	24
5.	Implementación	26
5.1.	Estructura General del Proyecto	26
5.2.	Integración de Módulos	29
5.3.	Interacción en VR	31
5.4.	Resultados de Evaluación de Usabilidad	32
5.4.1.	Cuestionario SUS – Ronda 1	32
5.4.2.	Observaciones y Mejoras Aplicadas	33
5.4.3.	Cuestionario SUS – Ronda 2	33
5.4.4.	Comparación entre Ronda 1 y Ronda 2 del SUS	34
5.4.5.	Comentarios Abiertos de los Participantes	35
5.4.6.	Métricas de Desempeño	36
5.5.	Discusión	37
6.	Conclusiones	38
6.1.	Síntesis del Proyecto	38
6.2.	Respuesta a la Pregunta de Investigación	38
6.3.	Cumplimiento de Objetivos	39
6.4.	Lecciones Aprendidas	39
6.5.	Proyecciones y Futuras Mejoras	40
	Glosario	41
	Bibliografía	44
A:	Casos de Uso	48

B: Carta Gantt	50
C: Fruticultura e Impacto en Chile	52
C.1. Inicio de la Fruticultura	52
C.2. Fruticultura en Chile	53
C.3. Impacto Socioeconómico de la Fruticultura	53
C.4. La Cereza: Trayectoria Histórica y su Desarrollo a Nivel Nacional . .	54
C.5. Importancia del Recurso Humano en el Cultivo de Cereza	55
D: Clasificación de Tipos de Poda para la Cereza	57
D.1. Clasificación por Objetivo	57
D.1.1. Poda de Plantación	57
D.1.2. Poda de Formación	57
D.1.3. Poda de Fructificación O Producción	58
D.1.4. Poda de Rejuvenecimiento	58
D.2. Clasificación por Época	58
D.3. Clasificación según sistema de poda	59
D.4. Tipos de Entrenamiento para la Poda	59
D.4.1. Kym Green Bush (KGB)	59
D.4.2. Spanish Bush (SB)	60
D.4.3. Steep Leader (SL)	60
D.4.4. Super Slender Axe (SSA)	60
D.4.5. Tall Spindle Axe (TSA)	61
D.4.6. Upright Fruiting Offshoots (UFO) y “Y” Trellis (UFO-Y) . . .	61
D.4.7. Vogel Central Leader (VCL)	62
E: Diseño E Implementación	63
E.1. Módulo de Poda Interactiva	63
E.2. Módulo de Feedback Visual y Sonoro	64
E.3. Módulo de Evaluación y Resultados	64
E.4. Módulo de Generación Procedural de Árboles	65
E.4.1. Códigos Necesarios para el funcionamiento del Sistema	67
E.4.2. Reglas Paramétricas del Nivel 1	85
E.4.3. Parámetros Globales	87
E.4.4. Reglas Paramétricas del Nivel 2	89

E.4.5. Reglas Paramétricas del Nivel 3	91
E.5. Módulo de Interfaz de Usuario e Inmersión	92
F. Cuestionario SUS y Métricas	94
F.1. Datos del Cuestionario SUS Ronda 1	94
F.2. Datos del Cuestionario SUS Ronda 2	94
F.3. Métricas Internas de Unity	94
G. Consentimiento Informado y Cuestionario SUS con Pregunta Cua-	
litativa	95
G.1. Consentimiento Informado	95
G.2. Cuestionario SUS con Pregunta Cualitativa	95
H. Imágenes de Prototipo Final, Interfaces de Scripts y otros	96

ÍNDICE DE FIGURAS

	página
2.1. Representación de distintos sistemas de crecimiento del cerezo [1]. . .	8
2.2. Crecimiento del método KGB	11
2.3. Representación gráfica de una cadena de caracteres formando una estructura de árbol	13
5.1. Jerarquía de objetos en la escena <i>Nivel 1</i> desarrollada en Unity. . . .	27
5.2. Vista general del Nivel 1: árbol simple con interfaz mínima.	28
5.3. Vista general del Nivel 3: árbol medio complejo con interfaz mínima.	28
5.4. Vista general del Nivel 3: árbol complejo con interfaz mínima.	29
5.5. Representación de los inputs asociados al Oculus Controller	31
5.6. Resultados del cuestionario SUS en ambas rondas. Se muestran los promedios con sus respectivas barras de error ($\pm$ desviación estándar).	34
5.7. Distribución de puntajes SUS en ambas rondas mediante diagrama de caja. La línea roja indica la mediana, mientras que los bigotes y cajas muestran el rango intercuartílico y la dispersión de las respuestas. En la segunda ronda se observan valores atípicos (outliers), lo que refuerza la interpretación de mayor variabilidad en las percepciones de usabilidad.	35
A.1. Diagrama de Casos de Uso del sistema de capacitación en poda VR. .	48
B.1. Carta Gantt - Inicio de Memoria (Proyecto de Memoria de Título) . .	50
B.2. Carta Gantt - Implementación de Prototipo (Memoria de Título) . .	51
B.3. Carta Gantt - Prototipo Final y Defensa de Memoria	51
H.1. Vista general del entorno VR.	96
H.2. Herramienta de poda en acción.	97
H.3. Menú e interfaz definitiva del entorno en el Tutorial.	97
H.4. Vista en VR del Menú e interfaz.	98
H.5. Interfaz del <i>Script SessionManager</i> en Unity, mostrando las variables principales de la sesión: índice de escena del tutorial, ID del jugador, nivel actual, tiempo de inicio, cortes correctos y errores.	98

H.6. Interfaz del módulo GrammarTree en Unity, mostrando las reglas L-System paramétricas y las variables generales definidas para la generación de árboles KGB.	99
H.7. Interfaz del script TreeLSys en Unity, encargado de procesar las reglas del GrammarTree y generar la estructura del árbol en 3D.	100

RESUMEN

La industria frutícola constituye un pilar estratégico para la economía chilena, posicionando al país entre los principales exportadores de fruta fresca a nivel mundial. En particular, el cultivo de cereza dulce ha experimentado un crecimiento sostenido durante la última década, situando a Chile como el sexto productor y segundo mayor exportador global, con una participación del 20 % en el comercio internacional.

No obstante, el desarrollo sostenible de esta industria enfrenta desafíos estructurales, siendo uno de los más relevantes la optimización de prácticas agrícolas como la poda. Factores como la alta rotación de personal y la escasez de programas de capacitación especializada limitan la formación técnica del recurso humano, afectando la productividad y calidad del cultivo.

Este proyecto plantea el desarrollo de una aplicación de realidad virtual orientada a la capacitación de trabajadores agrícolas en técnicas de poda del cerezo. El entorno virtual fue diseñado mediante modelado 3D y técnicas de generación procedural, permitiendo simular condiciones de poda diversas y adaptables al nivel de experiencia del usuario.

La efectividad de la experiencia formativa será evaluada a través de simulaciones prácticas, cuestionarios de autopercepción y análisis del desempeño después del proceso de capacitación. Se espera que esta herramienta contribuya a mejorar las habilidades técnicas de los trabajadores, fortaleciendo la eficiencia operativa en el campo y potenciando la competitividad de la industria frutícola chilena en el contexto global.

1. Introducción

La fruticultura en Chile ha experimentado un crecimiento sostenido, particularmente en la producción de cerezas en la Región del Maule . Este crecimiento en el mercado ha generado una creciente demanda por cultivos de alta calidad y técnicas eficientes, como la poda, que son críticas para vincular la producción agrícola con las exigencias del mercado [2].

Sin embargo, la formación de trabajadores agrícolas enfrenta importantes desafíos, entre los cuales destacan la alta rotación de personal y la falta de programas de capacitación adecuados. Estas limitaciones subrayan la necesidad de desarrollar métodos formativos más efectivos [3].

En este contexto, la Realidad Virtual (RV) se presenta como una tecnología innovadora en el ámbito educativo y de capacitación, capaz de crear experiencias inmersivas que permiten a los usuarios interactuar con el entorno de diversas maneras, facilitando un aprendizaje más accesible y efectivo [4]. Además, la generación procedural es un enfoque que permite crear contenido de manera dinámica y adaptativa, pudiendo generar así árboles y escenarios de poda con características únicas en cada experiencia. Esto permite a los trabajadores practicar y mejorar sus habilidades en un entorno seguro y controlado, ajustando el nivel de dificultad según sus necesidades y progresos [5].

Ante este desafío en la fruticultura, surge la siguiente pregunta: ¿En qué medida una experiencia de realidad virtual, basada en generación procedural de árboles, resulta usable como herramienta de simulación para prácticas básicas de poda de cerezos?.

1.1. Contexto del Problema

La fruticultura en Chile ha experimentado un notable crecimiento, especialmente en la producción de cerezas. Sin embargo, el sector enfrenta desafíos grandes en la capacitación de trabajadores agrícolas, destacando la alta rotación de personal y la falta de programas de formación adecuados. Estos problemas limitan el desarrollo de habilidades prácticas esenciales, como la poda, afectando la calidad de la producción y la competitividad en el mercado [2].

Para abordar estas dificultades, es crucial explorar alternativas innovadoras de capacitación. La Realidad Virtual (RV) se presenta como una solución potencial, ofreciendo experiencias de aprendizaje inmersivas y seguras. Además, técnicas como la generación procedural permiten crear simulaciones personalizadas que se adaptan a las necesidades de cada trabajador.

Este contexto resalta la necesidad de investigar cómo la integración de tecnologías avanzadas, como la RV, puede mejorar la formación en el sector agrícola y contribuir al crecimiento sostenible de la fruticultura en Chile.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar una experiencia de realidad virtual que permita la capacitación de trabajadores agrícolas en la poda de cerezos, adaptando la simulación a diversos niveles de experiencia mediante técnicas de generación procedural de árboles.

1.2.2. Objetivos Específicos

Objetivo específico 1: Implementar un sistema de generación procedural de árboles en el entorno virtual, con variabilidad en sus características fenotípicas.

Objetivo específico 2: Diseñar una interfaz interactiva intuitiva que permita a los usuarios practicar las técnicas correctas de poda en un entorno seguro.

Objetivo específico 3: Evaluar la efectividad de la capacitación mediante el seguimiento del progreso en cada nivel de poda, determinando el desempeño a medida que avanzan en la experiencia interactiva.

Objetivo específico 4: Integrar métricas de desempeño que permitan medir la mejora en las habilidades de poda de los usuarios tras realizar la capacitación virtual.

1.3. Pregunta de Investigación

¿En qué medida una experiencia de realidad virtual, basada en generación procedural de árboles, resulta usable como herramienta de simulación para prácticas básicas de poda de cerezos?

En la siguiente sección se presenta de manera integrada el Estado del Arte y el Marco Teórico. El Estado del Arte comprende la revisión de los principales avances y aproximaciones tecnológicas vinculadas al uso de la realidad virtual en procesos de capacitación agrícola, considerando asimismo aplicaciones y estudios científicos relevantes que han abordado esta problemática.

Por su parte, el Marco Teórico incorpora los antecedentes necesarios para situar la investigación en un contexto agrícola más amplio, permitiendo comprender la problemática de origen y su relación con la práctica de la poda de cerezos. A partir de esta base, se fundamenta la elección de tecnologías, técnicas y herramientas utilizadas en el proyecto, destacando la incorporación de la realidad virtual y la generación procedural como recursos innovadores para el fortalecimiento de los procesos de enseñanza y aprendizaje. Luego, se describe la metodología empleada en el desarrollo de la experiencia de realidad virtual, así como la implementación y evaluación de la simulación. Finalmente, se presentan los resultados obtenidos, seguidos de una discusión sobre las conclusiones y recomendaciones para futuras investigaciones en el área.

2. Estado del Arte y Marco Teórico

La realidad virtual ha demostrado ser una herramienta innovadora y efectiva en la capacitación y simulación en diversos sectores industriales, incluyendo la agricultura y el procesamiento de alimentos [4]. A través de su capacidad para replicar entornos de trabajo complejos en condiciones controladas, la realidad virtual ofrece una plataforma segura y realista para que los trabajadores desarrollen habilidades y competencias sin riesgos físicos. A continuación, se presenta el Estado del Arte en varios estudios que destacan la aplicabilidad de la realidad virtual en estos ámbitos.

- **Virtual Reality and the Application in Virtual Experiment for Agricultural Equipment** Este estudio se enfoca en el uso de la realidad virtual (VR) para crear un entorno de prueba seguro y controlado, específicamente en la simulación de tractores y otros equipos agrícolas. La investigación desarrolló una plataforma interactiva que permite a los usuarios experimentar y analizar el funcionamiento de la maquinaria en tiempo real, integrando tecnologías como MultiGen Creator y Vega Prime. Este enfoque fomenta la innovación en capacitación y diseño de maquinaria agrícola, mejorando la precisión en la simulación de condiciones de campo y reduciendo riesgos durante la capacitación de operadores y pruebas de equipos [6].
- **Procedurally Generated Virtual Reality from 3D Reconstructed Physical Space** Este artículo presenta un sistema innovador de generación de realidad virtual que utiliza el entorno físico como base, permitiendo a los usuarios desplazarse naturalmente en un entorno virtual inmersivo con retroalimentación háptica. Destaca por su facilidad de uso para crear rápidamente entornos

virtuales interactivos. Los autores sugieren futuras mejoras, como mapas más avanzados, mayor detección de objetos y avatares corporales en tiempo real, lo que podría enriquecer la experiencia del usuario y mejorar el rendimiento del sistema [7].

- **Improving the Selection of IQF Raspberries in Processing Lines: A Virtual Reality Approach for Training and Selecting Personnel** Este estudio se centra en el diseño y desarrollo de un modelo de simulación en un entorno virtual para resolver problemas en la industria, especialmente en la capacitación del personal. Este enfoque reduce costos y riesgos asociados con la formación tradicional al permitir simulaciones controladas y seguras. La implementación de gamificación y dispositivos como HTC Vive proporciona una experiencia inmersiva que mejora la motivación y el aprendizaje [8].
- **A Virtual Reality System for Fruit Processing Line Personnel Training: A Usability Study** Este artículo presenta una aplicación de realidad virtual (VR) para la simulación de la recolección, análisis y clasificación de productos en una línea de procesamiento de frutas. La prueba de usabilidad reveló que todos los participantes completaron con éxito la tarea y encontraron las interacciones muy realistas. Los resultados indican que la simulación cumple su objetivo educativo, y se sugiere realizar estudios adicionales para evaluar la percepción de profundidad y la efectividad del programa de capacitación en el personal actual y nuevo [4].

La revisión de estos estudios nos indican el potencial de la realidad virtual en la industria agrícola, permitiendo que tanto nuevos trabajadores como el personal experimentado puedan aprender y perfeccionar habilidades en un entorno simulado. Estos avances indican un camino prometedor para la implementación de soluciones virtuales en la capacitación laboral, lo cual contribuye a mejorar la eficiencia y reducir riesgos en diversas áreas industriales.

En la sección siguiente se desarrolla el Marco Teórico, donde se presentan los antecedentes agrícolas y técnicos necesarios para comprender la problemática de la poda de cerezos y la fundamentación de la incorporación de tecnologías como la realidad virtual y la generación procedural.

La fruticultura cuyos orígenes se remontan al Creciente Fértil(4000–3000 a.C.) ha sido motor del progreso social y tecnológico en múltiples civilizaciones [9]. En Chile, la apertura exportadora de la década de 1980 transformó al rubro en un pilar del PIB agrícola y del empleo rural, con ventas externas que superaron los USD3700 millones en 2024 [10]. El cerezo dulce se consolidó como la especie de mayor dinamismo: más de 60 cultivares permiten escalonar cosechas y abastecer mercados lejanos, siempre que la **poda** mantenga fruta de calibre y firmeza óptimos [11]. Asimismo, la creciente mecanización del riego y la cosecha ha disminuido la mano de obra permanente y elevado la demanda de podadores temporales (85 % en 2019), generando un cuello de botella formativo [12].

Para una síntesis detallada de estos antecedentes véase Apéndice C:

- *C.1 Inicio de la fruticultura* – evolución histórica global
- *C.2Fruticultura en Chile* – políticas, superficie y exportaciones
- *C.3 Impacto socioeconómico* – aporte al PIB y empleo rural
- *C.4Trayectoria del cerezo en Chile* – diversificación varietal
- *C.5 Recurso humano* – rotación laboral y necesidades de capacitación

A partir de este marco, el capítulo se concentra en el **papel crítico de la poda** y en las tecnologías que potencian su enseñanza:

2.1. Poda en la Producción de Cerezos

La poda de los cerezos es esencial para optimizar la calidad y cantidad de frutos. Por otro lado, el manejo de la estructura y el crecimiento natural del árbol, es decir, la forma en que se organiza y se potencia su desarrollo, impacta directamente en la eficiencia productiva y en la facilidad de cosecha. La adecuada configuración del árbol influye en aspectos críticos como la iniciación de yemas florales, el crecimiento vegetativo y la renovación de la madera. [13, 14].

- **Importancia de la Poda y el Manejo del Crecimiento en Cerezos:** La poda de los cerezos es fundamental debido a las características de crecimiento de estos árboles, los cuales suelen alcanzar grandes alturas y tener ramas que

pueden ser frágiles. Esto hace imprescindible su manejo controlado para maximizar la producción y evitar problemas estructurales en el árbol [14]. Los sistemas modernos de poda se han desarrollado para responder a los desafíos de los altos costos de producción y la escasez de mano de obra capacitada, permitiendo [14]:

- **Producción temprana:** Permite que los cerezos den frutos más rápidamente, ayudando a los agricultores a recuperar su inversión en menor tiempo.
 - **Frutas de alta calidad:** Asegura el crecimiento de frutos en ramas bien iluminadas y saludables.
 - **Renovación de ramas:** Facilita la planificación de la sustitución de ramas viejas y no productivas.
 - **Simplicidad:** Hace más sencillo el proceso de poda y manejo de los árboles.
- **Clasificaciones por Tipos de Poda:** La poda en cerezos puede clasificarse de diversas maneras según el objetivo y el momento de su realización. A continuación, se detallan los tipos principales:
- **Por Objetivo:**
 - **Poda de Plantación:** Se aplica al momento de plantar, buscando equilibrar la parte aérea y el sistema radicular. Esta poda inicial permite un crecimiento fuerte y estructurado del tronco y ramas principales [13].
 - **Poda de Formación:** Orientada a establecer la estructura básica del árbol y controlar su tamaño, permitiendo una óptima producción de madera y una organización adecuada para la fructificación [15, 13] (ver apéndice D.1.2).
 - **Poda de Fructificación:** Realizada después de la cosecha, ayuda a preparar el árbol para la siguiente temporada productiva, asegurando una buena entrada de luz y ventilación en la copa, así como el mantenimiento de una altura adecuada [15] (ver apéndice D.1.3).

- **Poda de Rejuvenecimiento:** Ejecutada en verano, redirige el crecimiento a ramas jóvenes cercanas al tronco, promoviendo la renovación de madera y un desarrollo equilibrado [16] (ver apéndice D.1.4).
- **Otros Criterios de Clasificación:** Existen otras formas de clasificar la poda, como:
 - Según el momento: **Poda de Invierno** (ver apéndice D.2) y **Poda de Verano** (ver apéndice D.2).
 - Según el sistema de poda: **Poda Larga** (ver apéndice D.3), **Poda Corta** (ver apéndice D.3) y **Poda Mixta** (ver apéndice D.3).
- **Sistemas de Crecimiento del Cerezo:** En la producción de cerezas, existen múltiples sistemas de crecimiento del cerezo que se adaptan a las necesidades y características del huerto. Entre los métodos de crecimiento más relevantes se encuentran el sistema KGB (Kym Green Bush), SB (Spanish Bush), SL (Steep Leader), SSA (Super Slender Axe), TSA (Tall Spindle Axe), UFO (Upright Fruiting Offshoots), "Y" Trellis y VCL (Vogel Central Leader) (ver apéndice D.4) [14].

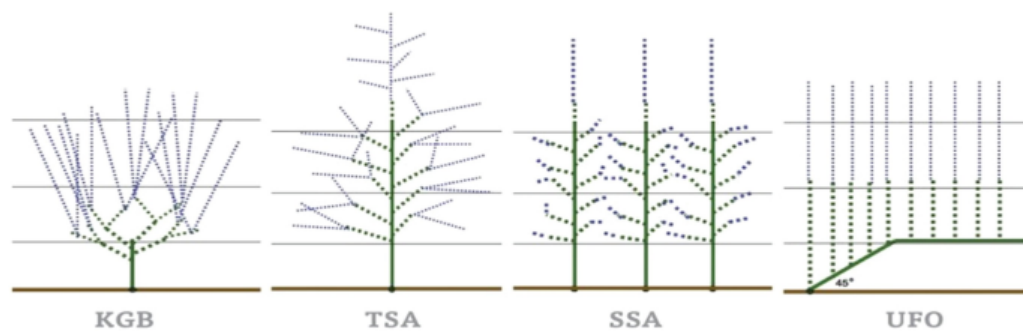


Figura 2.1: Representación de distintos sistemas de crecimiento del cerezo [1].

Para el presente marco, se ha decidido profundizar en el sistema KGB debido a sus ventajas en eficiencia de poda y calidad de producción [14]. El sistema KGB es valorado por su estructura autoportante, facilitando un mantenimiento sencillo en huertos peatonales y una fructificación estable. Además, destaca por su accesibilidad, ya que no requiere el uso de escaleras para realizar labores de poda o cosecha, lo cual lo convierte en una opción práctica y segura para los trabajadores agrícolas (ver apéndices D.4.1).

- **Consecuencias de la ausencia o mala ejecución de la poda en cerezos**

La poda es una práctica fundamental para asegurar una producción equilibrada, de calidad y sostenible en frutales como el cerezo. La falta de poda conlleva consecuencias agronómicas importantes, como la sobrecarga de frutos pequeños, pérdida de vigor vegetativo y desarrollo de madera envejecida. En Chile, estudios han evidenciado que huertos sin poda adecuada presentan una disminución progresiva del calibre y firmeza de la fruta, comprometiendo su valor comercial y generando árboles con estructura densa y poco productiva [17].

Desde el punto de vista estructural, la falta de poda favorece el desarrollo de copas densas y sombreadas, lo que reduce la producción de madera fructífera en el interior del árbol. Este fenómeno, conocido como *madera ciega*, limita el potencial productivo al restringir la renovación de ramas y la penetración de luz en la copa. Investigaciones en Europa sobre cerezos ‘Regina’ y ‘Kordia’ demostraron que los árboles no podados desarrollaban menos brotes útiles y producían frutas de menor tamaño en comparación con árboles podados anualmente [18].

Asimismo, prácticas de poda inadecuadas, como cortes en épocas lluviosas o sin las medidas de higiene necesarias, pueden incrementar el riesgo de infecciones en la madera por patógenos como *Pseudomonas syringae* o *Chondrostereum purpureum*. Estas enfermedades afectan la longevidad del árbol y su capacidad productiva, lo que refuerza la necesidad de realizar una poda técnica, considerando la época y la forma adecuada [19].

Finalmente, la ausencia de poda o su mala ejecución reduce la eficiencia operativa del huerto. Árboles demasiado altos o con ramas entrecruzadas dificultan labores como la cosecha, aplicación de agroquímicos y futuras podas, incrementando costos y riesgos laborales. Estudios recientes sobre sistemas de conducción en cerezos han destacado que una estructura bien formada y manejada facilita significativamente el trabajo agrícola y mejora el rendimiento global del cultivo [20].

2.2. Sistema de Crecimiento(KGB) con Técnicas de Poda

El crecimiento del árbol(cerezo) es fundamental para optimizar la productividad y eficiencia en la producción de cerezos, asegurando que los trabajadores estén preparados para utilizar sistemas de crecimiento modernos. A continuación, se detalla el sistema de crecimiento *Kym Green Bush* (KGB), el cual destaca por su enfoque en la accesibilidad y mecanización en la cosecha, como se muestra en la Figura 2.2 [14]

- **Características del Sistema KGB:** El sistema KGB facilita una estructura arbórea accesible desde el suelo, reduciendo el tiempo y esfuerzo en la cosecha. Es un sistema sencillo que requiere menos poda, ideal para trabajadores con poca experiencia.
 1. **Crecimiento Inicial:** Desde el primer año, los trabajadores aprenden técnicas de poda para promover un sistema radicular fuerte y un desarrollo equilibrado. Esto incluye la poda de cabeza a 18 pulgadas sobre el suelo y la eliminación de ramas débiles [14].
 2. **Poda de Formación:** En años posteriores, se enseña a los trabajadores a podar para asegurar un crecimiento controlado, eliminando líderes débiles y promoviendo una estructura uniforme. Esto ayuda a mantener la salud del árbol y previene problemas como el cancro bacteriano [14].
 3. **Manejo de la Carga de Fruto:** A medida que el árbol madura, los trabajadores aprenden a gestionar la carga de frutos y renovar ramas frutales, equilibrando producción y luz para una cosecha más eficiente [14].

En conclusión, se utilizará el sistema *Kym Green Bush* (KGB) y las técnicas de poda de formación en la primera etapa para el desarrollo e implementación de una experiencia de capacitación en realidad virtual. La elección del sistema KGB se basa en su accesibilidad y simplicidad, lo que lo convierte en una opción ideal para trabajadores con poca experiencia, permitiéndoles familiarizarse rápidamente con las técnicas básicas de poda [14]. Este sistema, al requerir menos poda y ser más fácil de manejar, facilita el aprendizaje inicial sin sobrecargar al usuario con la complejidad de otros sistemas más avanzados. Esta experiencia virtual integrará la generación procedural de árboles, lo que permitirá crear entornos de práctica que simulen distintos tipos de condiciones para el crecimiento y estructuras de árboles en tiempo

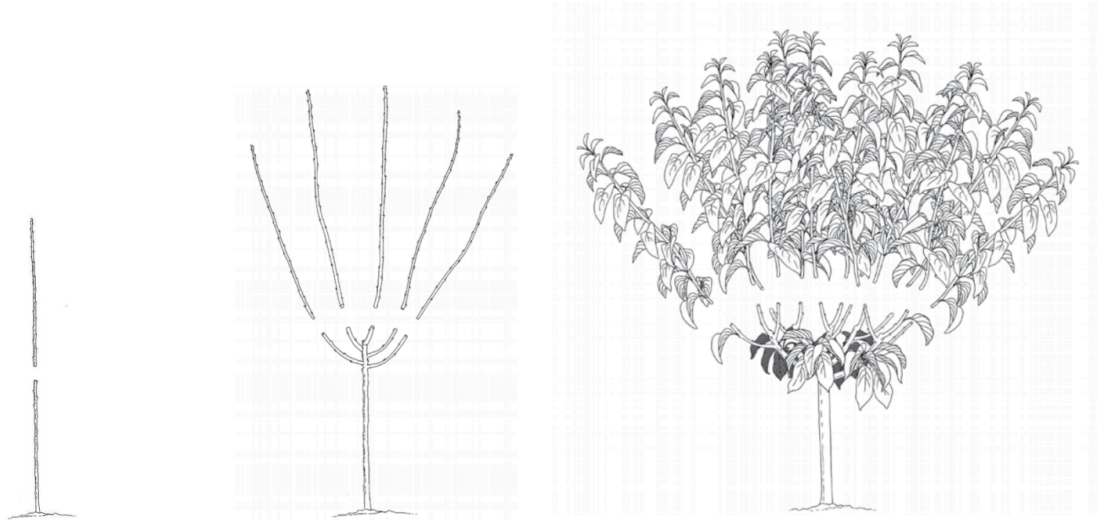


Figura 2.2: 3 cerezos de distinto nivel de desarrollo formado mediante KGB y sus respectivas podas Fuente: [14].

real. Con este enfoque, se busca ofrecer a los usuarios una plataforma inmersiva y adaptativa que facilite el aprendizaje y la práctica de técnicas de poda, promoviendo una comprensión más profunda de los principios estructurales y prácticos del sistema KGB.

2.3. Incorporación de Tecnologías en la Poda de Cerezos

2.3.1. Realidad Virtual en un Entorno Agrícola Inmersivo

La realidad virtual (RV) se ha convertido en una herramienta útil en diversos campos, especialmente en la educación y la capacitación. En el contexto agrícola, su aplicación ofrece ventajas significativas, como la posibilidad de crear entornos inmersivos y simulados donde los trabajadores puedan adquirir habilidades prácticas sin riesgos asociados [21]. Esto es particularmente relevante en actividades como la poda de cerezos, donde la precisión y el conocimiento técnico son esenciales para garantizar la calidad del producto final.

Los entornos virtuales permiten replicar escenarios reales con un nivel de detalle que facilita la transferencia de conocimiento [22]. Además, proporcionan un espacio controlado y seguro para cometer errores, algo esencial en el proceso de aprendizaje

[21]. La RV fomenta la interacción activa y el aprendizaje basado en la experiencia, aspectos que se han demostrado efectivos en la retención del conocimiento y el desarrollo de habilidades prácticas [4].

Un componente clave de los entornos virtuales es su capacidad de adaptarse a diferentes niveles de experiencia [23]. Esto es particularmente útil en la capacitación agrícola, ya que permite personalizar los desafíos presentados a cada trabajador, asegurando que la experiencia sea relevante y efectiva para todos. Por ejemplo, los trabajadores principiantes pueden practicar tareas básicas, mientras que los más avanzados enfrentan escenarios más complejos.

En el caso de la capacitación en poda, la RV no solo mejora el aprendizaje práctico, sino que también contribuye a resolver problemas estructurales del sector agrícola, como la alta rotación laboral y la falta de personal capacitado [3].

2.3.2. Generación Procedural de Árboles

La generación procedural es una técnica que utiliza algoritmos para crear contenido complejo de forma automatizada [24, 25]. A continuación, se detalla una estructura organizada para comprender sus fundamentos, enfoques y aplicaciones.

1. Fundamentos y Orígenes

Aplicación en Naturaleza y Entornos Virtuales: Con el tiempo, estas técnicas se han extendido al modelado de estructuras naturales, como el crecimiento de plantas, y a la simulación de entornos urbanos, generando edificios, calles y áreas verdes que combinan variación y cohesión estilística [24, 25].

2. Principales Enfoques en la Generación Procedural

Sistemas-L y Fundamentos en Reescritura: Los Sistemas-L (L-systems) se derivan de los sistemas de reescritura de cadenas, cuya primera definición formal fue establecida por Thue, y su estudio ganó relevancia gracias al trabajo de Chomsky en gramáticas formales y a la notación de Backus-Naur para definir lenguajes de programación; en este contexto, en 1968, el biólogo Aristid Lindenmayer introdujo los Sistemas-L, caracterizados por aplicar sus reglas de producción de forma *paralela* (reemplazando simultáneamente todas las letras de una cadena) en contraste con la aplicación *secuencial* de las gramáticas de Chomsky [26]. Esta característica,

inspirada en procesos biológicos como la división celular en organismos multicelulares, permite simular el crecimiento orgánico de plantas y árboles, haciéndolos ideales para entornos interactivos y simulaciones en tiempo real [24, 26].

Otros Enfoques: Existen métodos alternativos para modelar el crecimiento de árboles, tales como:

- **Particle Flow:** Utiliza partículas para simular dinámicas de crecimiento [24, 25].
- **Space Colonization:** Emula el proceso de colonización del espacio por las ramas, generando resultados realistas, aunque con mayor demanda computacional y ajustes complejos [24, 25].

Estos métodos requieren una mayor capacidad computacional y calibraciones más detalladas, lo que puede limitar su uso en aplicaciones en tiempo real [24, 25].

Sistemas OL de Árboles y con Paréntesis Los sistemas OL de árboles aplican reglas de reescritura directamente sobre la estructura arbórea. Cada rama se reemplaza simultáneamente, generando nuevas conexiones. Pueden representarse como listas topológicas o mediante cadenas con paréntesis, extendiendo la metáfora de la *tortuga* gráfica con los símbolos [y] para manejar el estado [26].

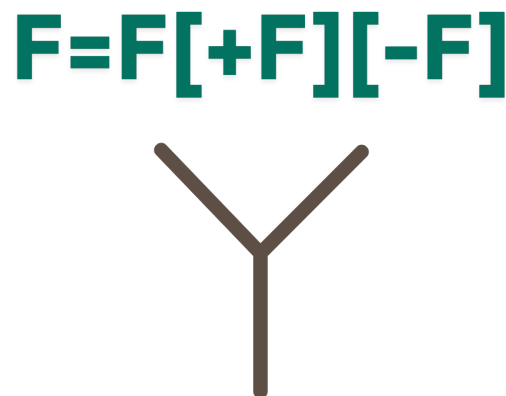


Figura 2.3: Ejemplo de árbol generado con reglas y delimitadores de estructura [26].

Sistemas L Estocásticos A diferencia de los sistemas deterministas, los L-systems estocásticos introducen variabilidad mediante reglas con probabilidades asignadas, evitando la repetición de estructuras idénticas. Esto permite simular paisajes más diversos y naturales [26].

Modelado Avanzado y Control de Parámetros

L-systems Paramétricos Estos extienden los L-systems tradicionales al incorporar valores numéricos en los módulos, permitiendo representar propiedades geométricas como ángulos, longitudes y condiciones lógicas. Esto mejora la flexibilidad y precisión en la simulación del crecimiento vegetal [26].

Interpretación con Tortuga En estos sistemas, cada símbolo puede incluir parámetros que controlan la distancia recorrida o la rotación de la tortuga virtual ($F(a)$, $+(a)$, etc.). Esto permite representar estructuras complejas con mayor fidelidad geométrica, incluso con valores irracionales [26].

En conjunto, estos modelos permiten construir simulaciones vegetales complejas y variadas que refuerzan el aprendizaje inmersivo en entornos de realidad virtual orientados a la poda técnica.

3. Marco Metodológico

La metodología utilizada en este proyecto se seleccionó con el propósito de garantizar un desarrollo estructurado, eficiente y enfocado en los objetivos específicos de la investigación. Dado el carácter innovador de integrar tecnologías de realidad virtual en la capacitación agrícola, se adoptó un enfoque basado en el diseño y la creación, complementado con técnicas de análisis cuantitativo y cualitativo. Este marco metodológico asegura que las decisiones tomadas durante el desarrollo y evaluación de la experiencia de realidad virtual estén fundamentadas en un proceso iterativo y validado por datos, maximizando la efectividad de la solución propuesta.

3.1. Metodología a Utilizar

El presente proyecto se enmarca metodológicamente en el enfoque *Design & Creation*, también conocido como *Design Science Research* (DSR), el cual se orienta al diseño y desarrollo de soluciones tecnológicas funcionales que resuelvan problemas específicos en contextos reales. Este enfoque resulta adecuado dado que permite articular la fase creativa del diseño con procesos sistemáticos de desarrollo, evaluación y mejora iterativa [27, 28].

En esta investigación, se adopta una estructura metodológica basada en cinco etapas propuestas por el modelo de Design and Creation descrito por *Oates* y aplicado en proyectos tecnológicos con enfoque educativo por autores como *Besoain* [29, 28]. Estas etapas son: *Problemática*, *Resolución*, *Desarrollo*, *Evaluación* y *Conclusión*. A continuación, se describe cómo se aplica cada una en el contexto del presente proyecto:

- **Problemática (Awareness):** Se identifica una problemática relevante: la necesidad de mejorar las técnicas de capacitación en poda de cerezos mediante tecnologías emergentes. La alta rotación de personal y la carencia de formación práctica efectiva son factores clave que motivan el desarrollo de este sistema de capacitación en VR [3].
- **Resolución (Suggestion):** Se plantea una solución tentativa a través del diseño conceptual de una experiencia en VR que simule escenarios de poda en un entorno inmersivo y controlado. Durante esta etapa se definen los casos de uso, los objetivos de aprendizaje y la narrativa que estructurará la experiencia del usuario [28].
- **Desarrollo (Development):** Se implementa la solución propuesta mediante el desarrollo del prototipo funcional. Esto incluye el modelado 3D de los árboles, la generación procedural de estructuras vegetales, la programación de mecánicas interactivas y la integración de una interfaz gráfica intuitiva, todo dentro de un motor de desarrollo compatible con VR.
- **Evaluación (Evaluation):** Se validará la efectividad del sistema a través de pruebas de usabilidad iniciales, seguidas de evaluaciones de aprendizaje e inmersión. Los instrumentos incluirán cuestionarios post experiencia y métricas de desempeño dentro del entorno VR. Estudios previos han demostrado que la realidad virtual puede ser una plataforma efectiva para realizar pruebas de usabilidad moderadas, permitiendo una experiencia inmersiva y comparable a los entornos presenciales [30]. Esta fase será iterativa, permitiendo realizar ajustes progresivos en base al feedback obtenido.
- **Conclusión (Conclusion) :** Finalmente, en esta etapa se recogen los aprendizajes y se analizan los resultados obtenidos. Se identifican las áreas que necesitan más investigación y desarrollo, y se sugieren posibles líneas de mejora que podrían ser abordadas en investigaciones futuras. Esta conclusión también ayuda a orientar la implementación de la solución en contextos reales de capacitación [28].

3.2. Herramientas y Software a Utilizar

Para llevar a cabo la implementación de la aplicación (PodaRV), es fundamental seleccionar las herramientas y el software más efectivos y eficientes que permitan un desarrollo ágil y cómodo. Para esto, se realizó una comparación de distintas opciones, evaluando su eficacia, facilidad de uso y potencial a largo plazo. A continuación, se detallan las herramientas seleccionadas, explicando las razones detrás de cada elección y su contribución esperada al éxito del proyecto.

3.2.1. Motor de Videojuegos: Unity

Unity será la plataforma principal de desarrollo, elegida por su versatilidad en la creación de experiencias inmersivas y su capacidad de integrar herramientas de realidad virtual. Unity permite la implementación de modelos 3D, interacciones dinámicas y una interfaz intuitiva que facilita el desarrollo ágil en entornos de simulación [31, 32].

3.2.2. Desarrollo de la Interfaz de Usuario: Unity UI

La interfaz de usuario (UI) de la aplicación será desarrollada utilizando las herramientas de Unity UI, las cuales permiten crear interfaces visuales e interactivas de forma sencilla. La UI permitirá a los usuarios navegar entre las opciones del programa, acceder a tutoriales y ajustar configuraciones. Además, la interfaz se diseñará de manera que no interrumpa la experiencia inmersiva en VR, siendo accesible y funcional durante la capacitación [31, 32].

3.2.3. Realidad Virtual: Oculus SDK

Para la experiencia de realidad virtual, se utilizará el Oculus SDK (Software Development Kit) para la integración con dispositivos de realidad virtual, como los visores Oculus Quest. Esta herramienta proporcionará las interfaces necesarias para interactuar de manera natural con el entorno virtual, permitiendo a los usuarios realizar la poda de cerezos de manera precisa y realista. Gracias a la compatibilidad con Unity, el SDK permitirá una integración fluida de los controles y las interacciones dentro de la experiencia VR, brindando una inmersión completa para el usuario [31] [32].

3.2.4. Otras Herramientas y Recursos Utilizados

Para el desarrollo de la aplicación, también se emplearán otras herramientas y recursos, tales como:

- **Visual Studio** para la programación y depuración del código en C#.
- **Git-hub** para el control de versiones, asegurando que el proyecto pueda ser gestionado de manera eficiente.
- **Postman u otra Aplicación** para realizar pruebas de APIs (si es necesario integrar funcionalidades externas).
- **Trello** para la gestión de tareas y organización del trabajo durante el proceso de desarrollo.
- **Unity Asset Store** para obtener activos y herramientas adicionales que faciliten el desarrollo, como modelos 3D, texturas, scripts o paquetes especializados en VR.
- **Figma o Sketch** para el diseño de interfaces de usuario, permitiendo crear maquetas y prototipos visuales antes de implementarlas en Unity.

Estas herramientas adicionales facilitarán la gestión del proyecto y garantizarán que se cumplan los objetivos del desarrollo de la aplicación PodaRV en tiempo y forma.

3.3. Características Metodológicas del Proyecto

3.3.1. Enfoque de Investigación

El presente proyecto se enmarca metodológicamente en el enfoque *Design & Creation*, también conocido como *Design Science Research* (DSR), el cual se orienta al diseño y desarrollo de soluciones tecnológicas funcionales que respondan a necesidades específicas en contextos reales. Este enfoque resulta pertinente ya que articula la fase creativa del diseño con procesos sistemáticos de desarrollo, evaluación y mejora iterativa [27, 28].

La metodología contempla dos grandes etapas: (1) diseño y desarrollo del sistema y (2) evaluación del prototipo funcional, las cuales se retroalimentan mediante ciclos de mejora continua. En la primera etapa, se aplicaron principios de diseño instruccional y técnicas como modelado 3D, generación procedural y programación en motores de realidad virtual para construir la experiencia interactiva. En la segunda etapa, se aplicaron estrategias de validación con usuarios, considerando variables de desempeño técnico y percepción de usabilidad.

Para la evaluación del sistema se utilizaron los siguientes instrumentos y métodos:

- **Cuestionario post experiencia:** La evaluación final se centró en la aplicación del cuestionario *System Usability Scale* (SUS) en dos rondas, complementado con una pregunta abierta para captar observaciones cualitativas sobre la experiencia. Inicialmente se contempló incluir otros instrumentos como UMUX o encuestas específicas para trabajadores agrícolas, pero se descartaron debido a ajustes en la planificación y el alcance.
- **Pruebas prácticas en el entorno VR:** Cada sesión generó automáticamente archivos CSV con métricas clave como tiempo total de ejecución, número de cortes, precisión técnica y cantidad de errores, permitiendo un seguimiento detallado del desempeño del usuario en la simulación.

Este esquema metodológico permite abordar la validación desde una perspectiva principalmente cuantitativa, complementada con insumos cualitativos, para analizar de forma integral tanto el rendimiento técnico del prototipo como la percepción de usabilidad por parte de los participantes. Los datos obtenidos sirven como base para futuras iteraciones y ajustes.

3.3.2. Población y Muestra

La población objetivo corresponde a personas interesadas en la capacitación en poda de cerezos mediante tecnología de realidad virtual. Para esta etapa de desarrollo se realizó una prueba de usabilidad preliminar mediante muestreo por conveniencia, seleccionando estudiantes de la carrera de Ingeniería en Desarrollo de Videojuegos y Realidad Virtual, quienes cuentan con conocimientos básicos en tecnología inmersiva.

Esta prueba permitió detectar problemas de usabilidad y recopilar retroalimentación temprana sobre la experiencia de uso, sirviendo como insumo para ajustes del

prototipo. Dado su carácter exploratorio, no se estableció una estratificación específica por edad o experiencia en fruticultura. Se definió un tamaño muestral orientativo de entre 8 y 15 participantes por ronda, considerando la recomendación de Nielsen [33] de que entre 5 y 10 usuarios son suficientes para identificar la mayoría de los problemas de usabilidad.

3.3.3. Procedimiento de Recolección de Datos

Cada sesión de uso del sistema generó un archivo CSV que registró automáticamente las métricas de interacción del usuario, incluyendo tiempo total, número de cortes realizados, errores y precisión técnica de poda calculada en función de las reglas definidas por el L-System.

Adicionalmente, se administró el cuestionario *System Usability Scale (SUS)* de forma presencial e individual, en formato físico (papel y lápiz), inmediatamente después de la experiencia inmersiva. A cada participante se le entregó una hoja con las 10 afirmaciones estándar y la escala de Likert de cinco puntos. Como complemento, se incluyó una pregunta abierta para recoger comentarios cualitativos sobre la experiencia y sugerencias de mejora.

El consentimiento informado fue aplicado formalmente en la segunda ronda del cuestionario SUS (ver Apéndice G.1). En la primera ronda, se comunicaron verbalmente los objetivos del proyecto y la naturaleza voluntaria de la participación, asegurando la comprensión de los participantes y su derecho a retirarse en cualquier momento.

3.3.4. Análisis de Datos

El análisis se realizó con enfoque cuantitativo, considerando los registros de Unity y los resultados de los cuestionarios. Se calcularon promedios, desviaciones estándar y correlaciones básicas para describir tendencias y relaciones entre desempeño técnico y percepción de usabilidad.

- **Métricas del sistema (Ver CSV Apéndice F.3):** tiempo total de ejecución, número de cortes totales y correctos, errores cometidos y precisión técnica.
- **Datos subjetivos (Ver Apéndice G.2):** puntaje SUS (escala 0–100) y análisis cualitativo básico de las respuestas abiertas.

Los cálculos y gráficos se realizaron mediante herramientas como Excel y Google Colab, facilitando la visualización de resultados y su interpretación.

3.3.5. Consideraciones Éticas

El presente estudio se desarrolló respetando principios éticos fundamentales en investigación con personas, priorizando la autonomía, la privacidad y el bienestar de los participantes. Se solicitó el consentimiento informado de forma documentada para la segunda ronda del SUS, y de forma verbal y explícita para la primera. Se informó sobre los objetivos del proyecto, el carácter voluntario de la participación y la posibilidad de retirarse en cualquier momento.

Todos los datos recolectados se manejaron de forma anónima y con fines exclusivamente académicos, resguardando la confidencialidad mediante identificadores y almacenamiento seguro. Se tomaron medidas para evitar riesgos físicos o malestares derivados del uso de la tecnología de realidad virtual, como sesiones cortas, pausas y supervisión constante.

3.3.6. Limitaciones

Algunos usuarios pueden experimentar incomodidad o menor familiaridad con la realidad virtual, lo que podría influir en el tiempo de interacción y la cantidad de errores. También se reconoce que el tamaño reducido de la muestra limita la generalización estadística de los resultados. No obstante, los datos obtenidos sirven como línea base para orientar mejoras futuras y planificar evaluaciones a mayor escala con trabajadores agrícolas en contextos reales. Aunque inicialmente se contemplaba desarrollar desde cero un sistema de generación procedural, la incorporación de un módulo preexistente basado en L-System permitió enfocar el desarrollo en la creación de gramáticas especializadas para simular árboles con patrón KGB, lo cual se considera una optimización en el uso de recursos. Esta incorporación no alteró los objetivos del proyecto, sino que facilitó un avance más eficiente en aspectos visuales, mecánicas y narrativa interactiva.

3.3.7. Validación

La validación del sistema se llevó a cabo en dos niveles: técnico y perceptual.

A nivel técnico, se recopilaron datos de desempeño del usuario directamente desde el sistema VR mediante archivos CSV generados por sesión. Estos incluyeron métricas como el tiempo total de ejecución, la precisión de los cortes y la cantidad de errores cometidos durante la tarea.

A nivel perceptual, se aplicó el cuestionario SUS, que permitió evaluar la experiencia del usuario en términos de usabilidad, facilidad de uso e integración de la tecnología en un contexto agrícola. Además, se incorporó una pregunta abierta complementaria para recoger observaciones cualitativas que aportaron matices adicionales sobre la percepción del sistema y posibles mejoras.

Esta validación, abordada desde una perspectiva principalmente cuantitativa pero enriquecida con insumos cualitativos, permitió analizar de forma integral tanto el desempeño objetivo como la percepción subjetiva de los usuarios.

A partir de esta base, en la sección siguiente se presenta el diseño conceptual del sistema, donde se detallan los casos de uso, los módulos principales y las interacciones que estructuran la experiencia de realidad virtual orientada a la capacitación en poda de cerezos.

4. Diseño del Sistema

El presente apartado detalla el diseño conceptual del sistema de capacitación en poda de cerezos mediante realidad virtual. Esta etapa precede a la implementación técnica y tiene como objetivo definir la estructura funcional del prototipo, sus componentes principales y las interacciones que permitirán una experiencia de aprendizaje efectiva, inmersiva y evaluable.

El diseño se basa en los requerimientos pedagógicos definidos en la etapa inicial del proyecto, así como en principios de interacción para entornos virtuales. A través de este diseño se busca garantizar una navegación intuitiva, una retroalimentación clara y una simulación coherente con las prácticas reales de poda.

A continuación, se presentan los casos de uso definidos para el usuario final y el diseño funcional de cada uno de los módulos principales del sistema.

4.1. Casos de Uso

Los casos de uso representan las interacciones que el usuario puede realizar dentro del sistema, junto con las respuestas esperadas por parte del sistema de realidad virtual. Cada caso refleja una acción importante para el proceso de capacitación, alineada con los objetivos pedagógicos definidos.

A continuación se describen los casos de uso principales identificados:

- **CU1 – Iniciar simulación:** El usuario accede al sistema, selecciona la opción de comenzar la experiencia y es transportado al entorno virtual del huerto de cerezos.

- **CU2 – Explorar entorno:** El usuario se desplaza libremente por el espacio virtual utilizando controles de movimiento (joystick), con la posibilidad de observar árboles desde distintos ángulos.
- **CU3 – Seleccionar rama para poda:** Al posicionar el controlador sobre una rama, esta se resalta indicando que puede ser seleccionada. El sistema registra la intención de poda.
- **CU4 – Ejecutar corte de rama:** Mediante un botón, el usuario simula el uso de la herramienta de poda y corta la rama seleccionada. El sistema evalúa si el corte es correcto según reglas de poda predefinidas.
- **CU5 – Recibir retroalimentación:** Una vez realizado el corte, el sistema entrega retroalimentación visual (colores, mensajes) y sonido, indicando la acción.
- **CU6 – Finalizar experiencia y recibir evaluación:** Al completar la poda de todos los árboles asignados, el sistema entrega una evaluación general que resume el desempeño del usuario.

La Tabla A.1, incluida en el Anexo A, presenta una síntesis estructurada de los casos de uso identificados en el sistema, incluyendo los actores principales, la descripción funcional de cada escenario y los resultados esperados en términos de interacción y respuesta del sistema. Complementariamente, la Figura A.1, también disponible en dicho anexo, muestra una representación visual de estos casos de uso mediante un diagrama UML, lo que permite observar gráficamente la relación entre el actor principal (usuario) y las distintas funcionalidades del sistema.

4.2. Diseño Funcional de los Módulos

El sistema de capacitación en poda VR se compone de diversos módulos funcionales que operan de manera integrada. A continuación, se describen brevemente los principales módulos diseñados, junto con una referencia a su descripción detallada, la cual se encuentra disponible en el Apéndice E.

- **Módulo de Poda Interactiva:** Permite al usuario seleccionar y cortar ramas dentro del entorno virtual mediante controladores VR. Detecta colisiones,

ejecuta la acción de corte y comunica los datos al módulo de evaluación. Su diseño funcional completo se presenta en el Apéndice E.1.

- **Módulo de Feedback Visual y Sonoro:** Proporciona retroalimentación inmediata sobre la calidad del corte mediante señales visuales (colores) y auditivas (sonidos positivos o negativos), mejorando la percepción del desempeño. El desarrollo técnico de este módulo puede consultarse en el Apéndice E.2.
- **Módulo de Evaluación y Resultados:** Analiza los cortes realizados por el usuario y genera una evaluación final. Considera criterios técnicos de poda, como tipo de rama, ubicación del corte y secuencia de acciones. Los detalles específicos de este componente se encuentran en el Apéndice E.3.
- **Módulo de Generación Procedural de Árboles:** Utiliza gramáticas L-System personalizadas para crear árboles virtuales con estructuras variables según el patrón de poda KGB, permitiendo diversidad y realismo en la práctica. La estructura interna de este módulo está descrita en el Apéndice E.4.
- **Módulo de Interfaz de Usuario e Inmersión:** Gestiona el HUD, los menús y la navegación dentro de la simulación. Facilita una experiencia fluida e intuitiva en el uso de la aplicación, adaptada al entorno de realidad virtual. Una explicación detallada de su funcionamiento se encuentra disponible en el Apéndice E.5.

En la siguiente sección se aborda la implementación del sistema, describiendo la organización general del proyecto, la integración de los módulos definidos en el diseño y el desarrollo de las interacciones en realidad virtual que dan forma al prototipo final.

5. Implementación

5.1. Estructura General del Proyecto

El sistema fue desarrollado en el motor de videojuegos Unity, versión 2022.3.15f1 LTS, y se encuentra gestionado mediante un repositorio en GitHub. El proyecto está estructurado en diversas escenas que representan las etapas clave del proceso de capacitación en poda de cerezos, desde la introducción hasta la evaluación final del desempeño.

Organización de la Escena *Nivel 1* como ejemplo

La Figura 5.1 muestra la jerarquía de objetos en la escena *Nivel 1*, donde se destacan los principales componentes del sistema: entorno virtual, árbol de práctica, controladores XR, sistema de interfaz de usuario y control de flujo de la simulación. Esta organización modular favorece el mantenimiento, reutilización y expansión del sistema.

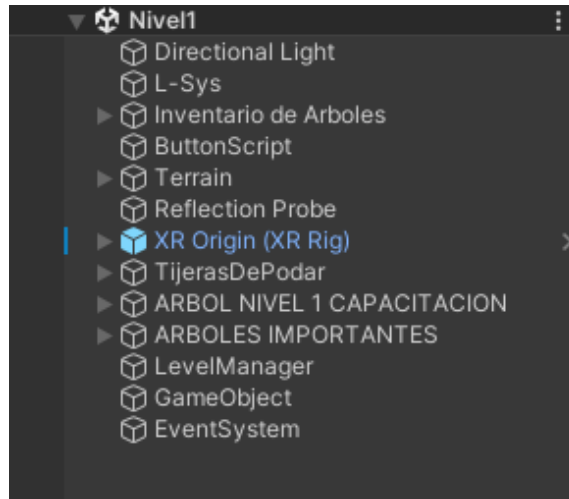


Figura 5.1: Jerarquía de objetos en la escena *Nivel 1* desarrollada en Unity.

Escenas del Sistema de Capacitación

A continuación, se describen las principales escenas que conforman la experiencia formativa:

- **Introducción o Menú:** Escena inicial donde el usuario se identifica mediante un sistema de ingreso de datos (ID). Incluye un tutorial interactivo sobre los controles y la interfaz.
- **Nivel 1 – Poda Básica:** Primer acercamiento práctico a la poda. Presenta un árbol de estructura simple, con el objetivo de que el usuario se familiarice con las herramientas, los controles y la lógica de la tarea.

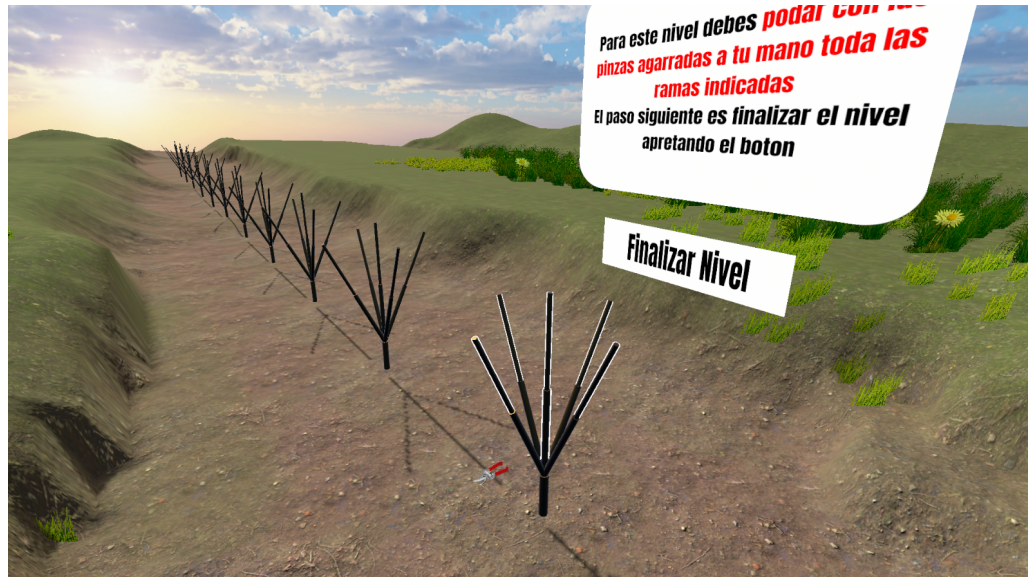


Figura 5.2: Vista general del Nivel 1: árbol simple con interfaz mínima.

- **Nivel 2 – Poda Intermedia:** Escenario con un árbol de estructura más compleja. Introduce mayor dificultad, cortes más técnicos y retroalimentación activa durante la ejecución.

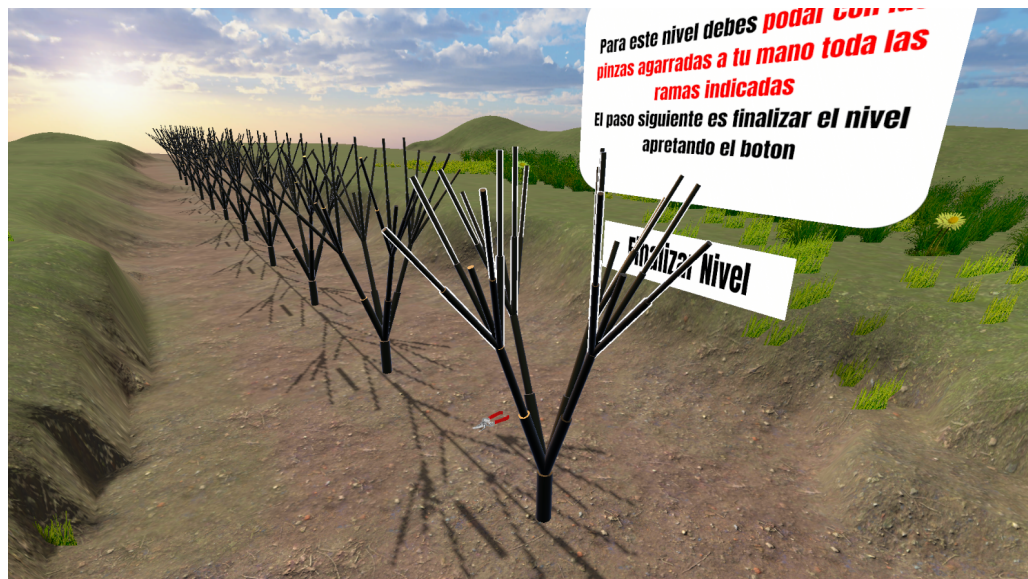


Figura 5.3: Vista general del Nivel 3: árbol medio complejo con interfaz mínima.

- **Nivel 3 – Poda Avanzada:** Escena final de práctica. El árbol, generado

proceduralmente, requiere una poda precisa bajo condiciones similares a las del trabajo real. Este nivel evalúa el dominio técnico adquirido.

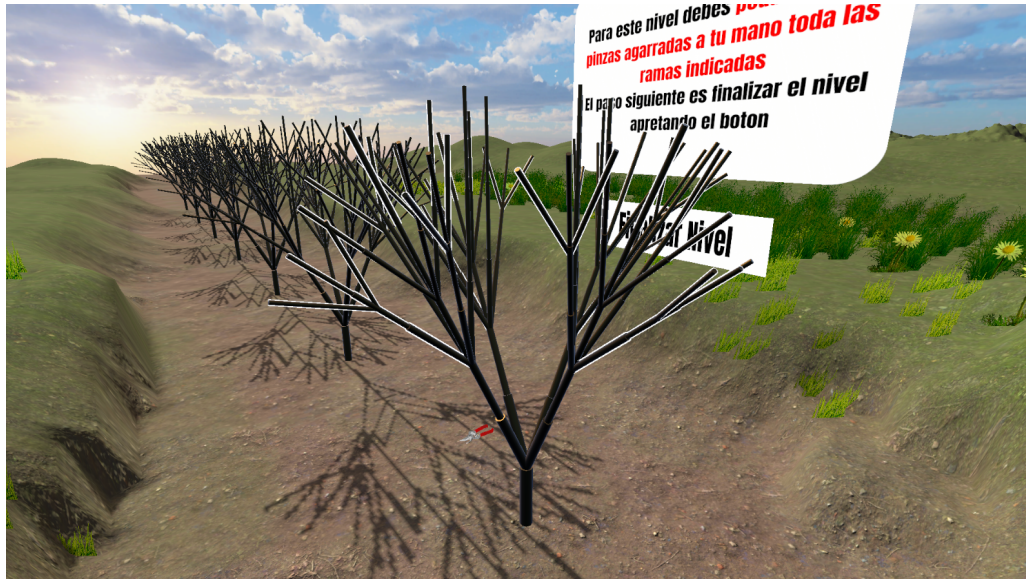


Figura 5.4: Vista general del Nivel 3: árbol complejo con interfaz mínima.

- **Escena Final:** Resume el desempeño del usuario, mostrando resultados cuantitativos y cualitativos de la experiencia de poda.

Cada escena fue diseñada con un enfoque pedagógico progresivo, permitiendo al usuario avanzar desde niveles introductorios hasta desafíos técnicos reales. Esta estructura contribuye al aprendizaje gradual, asegurando la consolidación de habilidades prácticas mediante simulaciones seguras e inmersivas.

5.2. Integración de Módulos

El sistema implementado en Unity integra diversos módulos que trabajan en conjunto para ofrecer una experiencia inmersiva y realista de capacitación en poda de cerezos. A continuación, se describen los principales módulos desarrollados:

- **Módulo de Poda Interactiva:** Simula el proceso de corte mediante unas pinzas virtuales controladas por el botón del dedo medio del mando derecho en el dispositivo VR. Las pinzas utilizan un *raycast* para detectar ramas, y al

presionar el gatillo, se activa el corte y la rama cae utilizando las físicas del motor de juego. La simulación actual aplica cortes sobre ramas completas. No se implementó aún la detección y segmentación exacta en el punto de interacción del usuario, pues requeriría un sistema más complejo de particionado de mallas. Se deja como línea de mejora para alcanzar mayor realismo en la poda.

- **Módulo de Feedback Visual y Sonoro:** Al ejecutar un corte, el sistema entrega retroalimentación inmediata mediante un mensaje visual que indica si el corte fue correcto o no, acompañado por un efecto de sonido distintivo según el tipo de corte realizado. Esto refuerza el aprendizaje del usuario en tiempo real.
- **Módulo de Evaluación y Resultados:** Registra métricas clave del desempeño del usuario en un archivo CSV, como cantidad de cortes correctos, tiempo y errores. Estos datos se sobrescriben por sesión y son utilizados para análisis posteriores, aunque no se presentan directamente al usuario.
- **Módulo de Generación Procedural de Árboles:** Emplea un sistema basado en L-Systems mediante *Scriptable Objects*, que definen reglas y parámetros del crecimiento arbóreo. Se destacan los scripts **Deriver.cs**, encargado de generar las cadenas de crecimiento, y **Generation.cs**, responsable de construir visualmente el árbol usando prefabs con rotaciones y posiciones definidas.

Con el fin de controlar la complejidad estructural y simular el proceso de poda progresiva, se definieron tres conjuntos de reglas paramétricas, cada uno asociado a un nivel de dificultad (Nivel 1, 2 y 3). Estas reglas fueron implementadas como gramáticas L-System paramétricas utilizando *ScriptableObjects* en Unity, y están documentadas en detalle en los Apéndices E.4.2, E.4.4 y E.4.5, incluyendo tanto sus producciones como los parámetros globales utilizados. Esta separación por niveles permite replicar con fidelidad los distintos estados de desarrollo de un cerezo bajo el patrón KGB, desde un árbol joven hasta uno completamente desarrollado y con alta densidad de ramas. Cabe destacar que el módulo de generación procedural de árboles no incorpora un modelo estocástico, es decir, las estructuras se generan de manera determinista a partir de las reglas paramétricas definidas, sin variaciones aleatorias en la producción de ramas o hojas.

- **Módulo de Interfaz de Usuario e Inmersión:** Integra paneles y elementos UI en cada nivel, mediante Canvas, para guiar al usuario durante la capacitación. Incluye botones de navegación (inicio y fin), textos explicativos y mensajes contextuales visibles sin interferir en la experiencia inmersiva.

5.3. Interacción en VR

Durante la experiencia inmersiva, el usuario interactúa con el entorno virtual utilizando los controles del dispositivo Oculus Quest. Cada botón del mando cumple una función específica que facilita la navegación, manipulación de objetos y ejecución de cortes de poda. Las acciones asignadas son las siguientes:

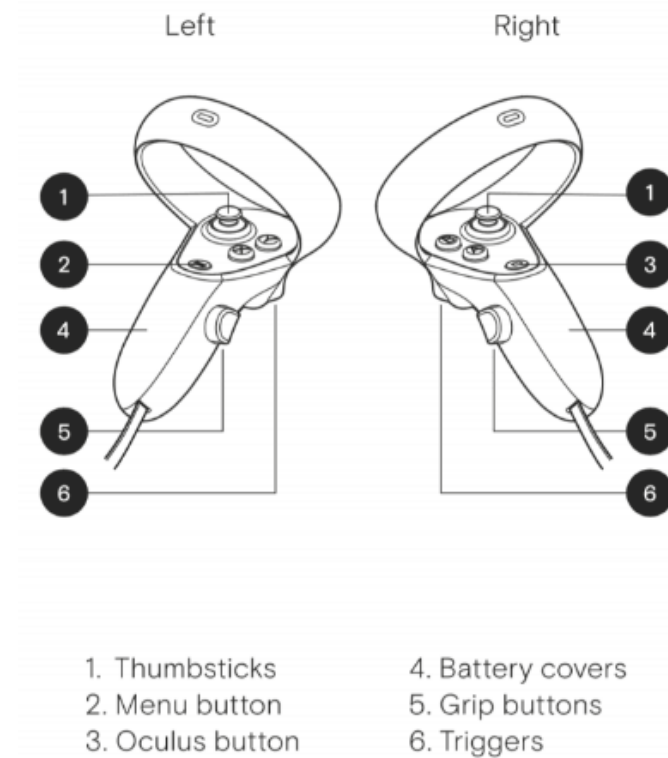


Figura 5.5: Inputs o entradas de los distintos botones para el Oculus Controller

- **Joystick izquierdo:** Permite desplazarse por el entorno virtual, simulando el movimiento libre del usuario dentro del huerto.

- **Grip buttons (ambas manos):** Se utilizan para tomar o agarrar objetos dentro del entorno, como las tijeras de poda.
- **Trigger buttons (ambas manos):** Al presionar el gatillo, se ejecuta la acción de corte sobre la rama seleccionada, activando además el sistema de físicas para simular su caída.
- **Joystick derecho:** Permite realizar giros rápidos de 360 grados sobre el eje vertical, facilitando el cambio de dirección y el reconocimiento del espacio sin necesidad de girar físicamente.

Esta configuración ha sido diseñada para proporcionar una experiencia accesible e intuitiva, permitiendo al usuario realizar tareas de poda de forma realista y ergonómica.

5.4. Resultados de Evaluación de Usabilidad

En esta sección se presentan los resultados obtenidos a partir de la evaluación de usabilidad del sistema mediante el cuestionario *System Usability Scale* (SUS) en dos rondas aplicadas en distintos momentos del desarrollo. Para consultar los datos completos de cada ronda, véanse los Apéndices F.1 y F.2. Las métricas objetivas extraídas desde Unity se detallan en el Apéndice F.3.

Para su análisis y representación gráfica, todos los datos fueron procesados en un cuaderno de Google Colab, lo que permitió calcular estadísticas descriptivas, verificar correlaciones y generar visualizaciones que complementan la interpretación de los resultados. Además del promedio, se calcularon métricas de dispersión (desviación estándar) y se representaron los resultados en gráficas con barras de error y boxplots, con el fin de identificar la variabilidad entre participantes.

5.4.1. Cuestionario SUS – Ronda 1

Para evaluar la usabilidad inicial del prototipo, se aplicó el cuestionario SUS a un grupo de diez participantes (8 hombres y 2 mujer) con distintos niveles de familiaridad en VR, cuyo rango etario se ubicó entre 20 y 26 años. Este instrumento consta de 10 afirmaciones valoradas en una escala de Likert de cinco puntos, lo que permite obtener un puntaje total de usabilidad entre 0 y 100.

Resultados cuantitativos: El puntaje promedio obtenido en esta primera ronda fue de **73 puntos** ($n = 10$), un valor que ya supera el umbral de referencia de 68 establecido como indicador de “buena usabilidad”. Esto indica que, pese a tratarse de una versión preliminar, la experiencia inmersiva resultó razonablemente satisfactoria para la mayoría de los usuarios. La desviación estándar fue de 11.2, lo que indica que los puntajes estuvieron relativamente concentrados en torno al promedio.

Observaciones cualitativas: Los comentarios de los participantes apuntaron a varios aspectos a mejorar:

- La interfaz, aunque clara en lo esencial, requería accesos más directos a ciertas funciones.
- La fluidez de respuesta en algunos controles VR podía optimizarse para incrementar la sensación de inmersión.

5.4.2. Observaciones y Mejoras Aplicadas

A partir de esta primera evaluación, se implementaron las siguientes correcciones:

- **Interfaz de usuario:** Se reestructuraron los menús para agrupar acciones afines y se añadió un HUD contextual que muestra atajos en cada nivel.
- **Optimización de rendimiento:** Se afinó la lógica de colisiones y se redujo la carga de prefabs en escena, logrando una tasa de frames más estable.
- **Variabilidad de escenarios:** Se refinaron las reglas paramétricas del L-System para incrementar la diversidad de formas y densidades de rama.

5.4.3. Cuestionario SUS – Ronda 2

Con el prototipo revisado, se aplicó una segunda ronda del SUS a diecinueve participantes (15 hombres y 4 mujeres, incluyendo nuevos usuarios y algunos veteranos de la primera fase) con un rango etario de entre 20 y 35 años, para medir el impacto de los ajustes realizados.

Resultados cuantitativos: El puntaje promedio ascendió a **74.44 puntos** ($n = 19$), lo que confirma una mejora leve pero constante respecto a la primera ronda. En este caso, la desviación estándar fue de 13.1, reflejando una mayor dispersión en las

respuestas, es decir, algunos usuarios evaluaron el sistema mucho mejor o peor que la media.

Observaciones cualitativas: Se destacó positivamente:

- Una mayor claridad en la navegación entre niveles.
- Una sensación de control más firme gracias a la optimización de la respuesta de los mandos.

Aún se sugirió:

- Incrementar la retroalimentación visual al cortar, añadiendo partículas y marcas temporales.
- Incorporar un breve tutorial guiado en el arranque para nuevos usuarios.

5.4.4. Comparación entre Ronda 1 y Ronda 2 del SUS

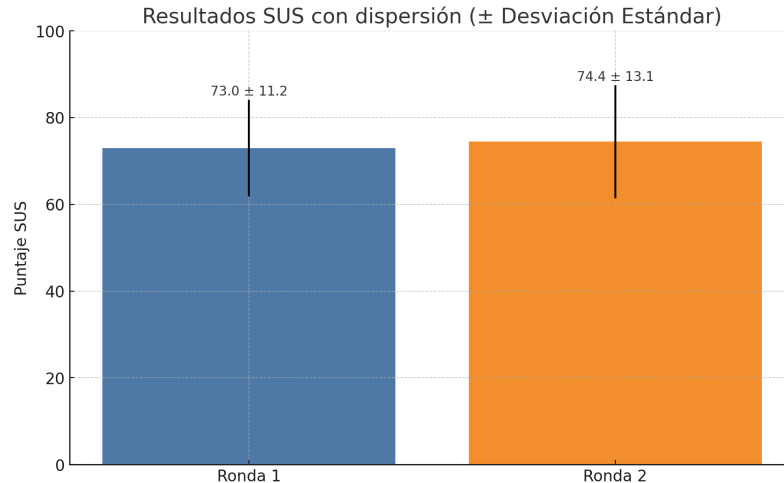


Figura 5.6: Resultados del cuestionario SUS en ambas rondas. Se muestran los promedios con sus respectivas barras de error ($\pm$ desviación estándar).

En la **Ronda 1** se obtuvo un puntaje promedio de **73.0 puntos** ($\sigma = 11,2$, $n = 10$), mientras que en la **Ronda 2** el promedio ascendió a **74.4 puntos** ($\sigma = 13,1$, $n = 19$). Ambos valores superan el umbral de referencia de 68, considerado como indicador de “buena usabilidad”.

La diferencia de +1.4 puntos entre rondas indica una leve mejora en la percepción de usabilidad, aunque lo más relevante es el incremento en la **dispersión de las respuestas** en la segunda ronda. Esto sugiere que, si bien la experiencia general se mantiene positiva, algunos participantes evaluaron el sistema de manera mucho más crítica o más favorable que la media.

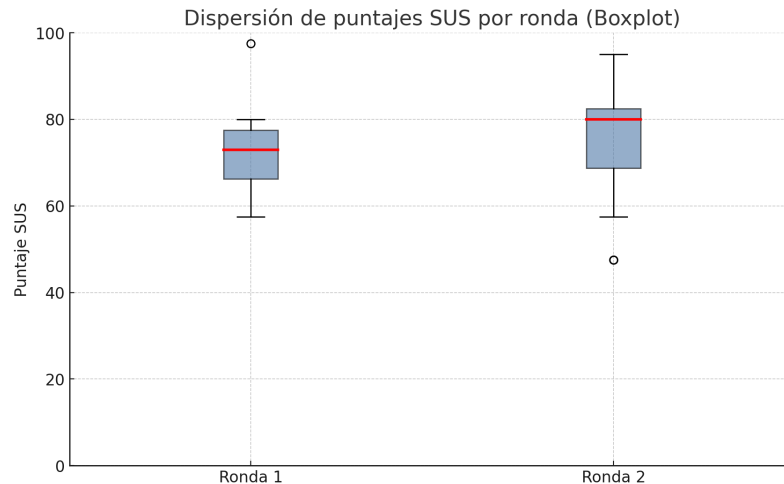


Figura 5.7: Distribución de puntajes SUS en ambas rondas mediante diagrama de caja. La línea roja indica la mediana, mientras que los bigotes y cajas muestran el rango intercuartílico y la dispersión de las respuestas. En la segunda ronda se observan valores atípicos (outliers), lo que refuerza la interpretación de mayor variabilidad en las percepciones de usabilidad.

En la Figura 5.6 se presenta un gráfico de barras con error que resume los resultados de ambas rondas, mostrando los promedios y su desviación estándar. Complementariamente, la Figura 5.7 ilustra la distribución completa de los puntajes mediante un diagrama de caja, en el cual se observa la presencia de valores extremos en la segunda ronda.

5.4.5. Comentarios Abiertos de los Participantes

A continuación se presentan literalmente algunos de los comentarios más representativos recogidos en la segunda ronda del SUS (para visualización de las opiniones Apéndice F.1 y F.2:

- “Faltaría un efecto más notorio cuando fallo el corte.”
- “Un pequeño paso a paso al inicio ayudaría a arrancar sin dudas.”

5.4.6. Métricas de Desempeño

Además del SUS, se registraron métricas objetivas desde Unity en cada sesión (para visualización de los datos ver Apéndice F.3):

Tabla 5.1: Métricas de desempeño por sesión ($n = 19$ participantes)

Métrica	Media	Desviación Estándar
Tiempo total (segundos)	284.19	75.36
Cortes realizados	62.63	16.75
Cortes correctos	50.00	15.32
Errores	12.63	13.94
Precisión (%)	79.53	19.99

Interpretación de las Métricas de Desempeño

A partir de la tabla 5.1 se observa:

- **Tiempo total promedio ($284s \pm 75s$):** La desviación estándar elevada sugiere variabilidad en la experiencia de los usuarios, posiblemente debida a diferencias en habilidades motrices o exploración del entorno.
- **Total de cortes ($62,6 \pm 16,8$) vs. Cortes correctos ($50,0 \pm 15,3$):** La diferencia de aproximadamente 12,6 cortes representa el promedio de errores cometidos por sesión.
- **Total de errores ($12,6 \pm 13,9$):** Confirma que la mayoría de los fallos están agrupados alrededor de la media, aunque algunos usuarios presentan muy pocos errores y otros llegan a duplicar la media.
- **Precisión promedio ($79,5\% \pm 20\%$):** A pesar de la complejidad del entorno, los usuarios alcanzan casi un 80 % de éxito, si bien la dispersión indica margen de mejora para reducir la variabilidad.

En conjunto, estas observaciones confirman que el prototipo alcanza niveles de precisión cercanos al 80 %, aunque se identifican diferencias entre usuarios en cuanto al tiempo de uso y la cantidad de errores cometidos. Para próximas iteraciones, se sugiere incorporar un tutorial guiado al inicio que oriente a los nuevos usuarios en el uso de los controles y la mecánica de poda, así como añadir retroalimentación

visual en tiempo real que permita minimizar errores durante la ejecución. Asimismo, se recomienda optimizar la interfaz para favorecer una operación más eficiente y acortar la duración promedio de cada sesión.

Para complementar la evidencia de la experiencia inmersiva y facilitar la validación del prototipo, se incluye en los Apéndices H el enlace de descarga de la build final en formato APK, junto con un video demostrativo grabado directamente desde el visor Oculus Quest 2. Estos recursos permiten observar en detalle la interacción real del usuario con el sistema, la generación procedural de árboles y la ejecución de cortes de poda.

5.5. Discusión

El análisis de los cuestionarios SUS muestra que el prototipo mantuvo buena usabilidad en ambas rondas (73.0 y 74.4 puntos, sobre el umbral de 68). No obstante, en la segunda ronda la desviación fue mayor ($\sigma = 13,1$), con respuestas más dispersas y outliers, lo que evidencia percepciones más heterogéneas. El gráfico con barras de error confirma que, pese a la leve mejora en la media, la variabilidad se incrementó. En cuanto a las métricas internas (Tabla 5.1), la elevada desviación estándar en el tiempo total (75s) refleja diferencias en la curva de aprendizaje según la familiaridad previa con VR. Aun así, la precisión promedio se mantuvo cercana al 80 %, aunque con un rango de dispersión (± 20 %) que sugiere dificultades en parte de los usuarios. Asimismo, los 12,6 errores promedio por sesión evidencian la necesidad de reforzar la retroalimentación inmediata para disminuir la tasa de fallos. De estos hallazgos se desprenden mejoras clave: incluir un tutorial inicial para uniformar la curva de aprendizaje, añadir retroalimentación visual y auditiva en tiempo real, y realizar sesiones de seguimiento para evaluar la retención de habilidades. Estas medidas fortalecerían la coherencia entre la usabilidad percibida (SUS) y el desempeño objetivo, consolidando un prototipo más robusto para la capacitación en poda de cerezos en VR.

6. Conclusiones

6.1. Síntesis del Proyecto

En este trabajo se diseñó, desarrolló y evaluó PodaRV, una aplicación de realidad virtual para capacitar en técnicas de poda de cerezos, basada en generación procedural de árboles KGB mediante L-Systems paramétricos. El prototipo, implementado en Unity con Oculus SDK, integró un módulo de poda interactiva, retroalimentación visual y sonora, y un sistema de registro automático de métricas en CSV. Se realizaron dos rondas de evaluación SUS, obteniendo 73,0 puntos ($\sigma = 11,2$, $n = 10$) en la primera y 74,4 puntos ($\sigma = 13,1$, $n = 19$) en la segunda, complementadas con métricas objetivas: tiempo total promedio de 284,2 s (± 75 s), 62,6 cortes por sesión ($\pm 16,8$), 50,0 cortes correctos ($\pm 15,3$), 12,6 errores ($\pm 13,9$) y una precisión media de 79,5 % (± 20 %).

6.2. Respuesta a la Pregunta de Investigación

¿En qué medida una experiencia de realidad virtual, basada en generación procedural de árboles, resulta usable como herramienta de simulación para prácticas básicas de poda de cerezos?

Los resultados obtenidos confirman que **PodaRV** alcanza niveles de usabilidad satisfactorios y ofrece una funcionalidad robusta como simulador de poda. El puntaje SUS registrado en la primera ronda (73) y en la segunda (74,4) supera ampliamente el umbral de referencia (68), lo que indica que los usuarios perciben la experiencia inmersiva como intuitiva, coherente y fácil de usar. Asimismo, la precisión promedio, cercana al 80 %, y la proporción de cortes correctos frente a errores demuestran que

el sistema responde de forma estable y fiable a las acciones del usuario, sin presentar obstáculos técnicos relevantes. Aunque se observa cierta variabilidad en los tiempos de interacción y en los niveles de precisión, estos desvíos se atribuyen principalmente al grado de familiaridad de los participantes con la realidad virtual, más que a deficiencias del prototipo. En conjunto, estos indicadores corroboran que **PodaRV** cumple con los objetivos funcionales y de usabilidad establecidos, consolidándose como una herramienta efectiva para la práctica básica de poda de cerezos.

6.3. Cumplimiento de Objetivos

1. **Generación procedural de árboles:** Se implementó con éxito un sistema L-Systems paramétrico que produce ejemplares KGB con variabilidad fenotípica.
2. **Interfaz intuitiva en VR:** La UI y los controles de Oculus fueron refinados tras retroalimentación, logrando navegación fluida y aumentada inmersión.
3. **Evaluación de usabilidad:** Se aplicaron dos rondas de SUS, mejorando de 73,0 a 74,4 puntos tras ajustes en la interfaz y el rendimiento.
4. **Métricas objetivas integradas:** El sistema registró automáticamente tiempo, cortes, errores y precisión, permitiendo un análisis cuantitativo riguroso de cada sesión.

6.4. Lecciones Aprendidas

A lo largo del desarrollo de PodaRV se identificaron varias lecciones clave. En primer lugar, la importancia de la iteración temprana quedó demostrada, ya que las pruebas de usabilidad realizadas en fases iniciales permitieron detectar problemas de navegación y respuesta del sistema, los cuales se corrigieron antes de llegar a la versión final. Además, se constató el rol fundamental de los sistemas L-Systems paramétricos, pues la implementación de reglas gramaticales permitió generar árboles con variabilidad controlada, facilitando ajustes rápidos en la estructura y densidad del árbol sin necesidad de modelado manual de cada ejemplar. Por último, se comprobó que la experiencia inmersiva en VR constituye un factor motivador para los usuarios: el entorno tridimensional, junto con la retroalimentación visual y sonora,

enriqueció la información disponible sobre ángulos de corte, posición de la herramienta, permitiendo una práctica más precisa y comprensible de la técnica de poda.

6.5. Proyecciones y Futuras Mejoras

En términos de proyección, se propone integrar dispositivos hápticos para mejorar la sensación de corte y aumentar la fidelidad de la experiencia. Asimismo, se contempla extender el prototipo hacia la capacitación en otros cultivos frutales, como manzanos y vides, adaptando las reglas paramétricas a cada especie y sus requerimientos agronómicos. Otra línea de mejora consiste en refinar las gramáticas L-System, de forma que representen con mayor realismo los patrones de ramificación de los cerezos en distintas etapas de crecimiento. Finalmente, se proyecta la incorporación de ejemplos y escenarios con distintos niveles de complejidad, casos de poda formativa, de renovación o mantenimiento, e incluso variables de clima y suelo, enriqueciendo así el contexto formativo.

Con estas proyecciones, **PodaRV** se perfila como una plataforma versátil y escalable para la formación agrícola en VR, con potencial de expandirse a nuevos cultivos y escenarios, y de complementar de manera significativa los métodos formativos tradicionales.

Con miras a replicar o escalar la experiencia desarrollada, se adjunta la build final y un video de la experiencia en funcionamiento, disponibles en los Apéndices H. Estos insumos pueden servir de base para iteraciones futuras o pruebas piloto.

Glosario

Caducifolio m. adj. Se refiere a las plantas que pierden sus hojas de manera estacional, generalmente durante el otoño o el invierno, como estrategia para adaptarse a condiciones climáticas adversas.

Cereza (***Prunus avium L.***) m. Especie de árbol frutal originario de la región entre el Mar Negro y el Mar Caspio, conocido por su fruto y su adaptabilidad a distintas regiones productoras.

Creciente Fértil f. Región geográfica que abarca partes de Mesopotamia y el valle del Nilo, cuna de las primeras prácticas de agricultura organizada.

Cultivares m. Variedades cultivadas de una especie vegetal, seleccionadas por sus características específicas.

Desviación estándar f. Medida de dispersión que indica cuánto se alejan los valores de una variable con respecto a su media.

Dominancia Apical f. Fenómeno botánico en el que el crecimiento de las ramas laterales se ve inhibido por el brote principal.

Exportación FOB (Free on Board) m. Término de comercio internacional que define que el vendedor entrega la mercancía a bordo del barco designado por el comprador.

Gamificación f. Uso de elementos y dinámicas de juego en contextos no lúdicos para motivar y enganchar a los usuarios.

Generación Procedural f. Técnica de creación automática de contenido mediante algoritmos y reglas predefinidas.

- Generación Procedural de Entornos** f. Aplicación de generación procedural para crear mundos virtuales adaptativos y dinámicos.
- Generación Procedural de Árboles** f. Uso de algoritmos (por ejemplo L-Systems) para modelar la estructura y crecimiento de árboles de manera automatizada.
- Glosario** m. Listado ordenado de términos y sus definiciones, utilizado para aclarar el significado de vocabulario especializado.
- HUD** m. (“Head-Up Display”) Elemento de interfaz que muestra información relevante superpuesta al campo de visión del usuario en VR.
- HTC Vive** m. Sistema de realidad virtual que incluye casco y controles para interacción inmersiva.
- Injerto** m. Técnica de propagación vegetal que une partes de dos plantas para combinarlas en una sola.
- L-Systems** m. (“Sistemas de Lindenmayer”) Gramáticas formales utilizadas para modelar el crecimiento de plantas y estructuras fractales.
- Oculus SDK** m. Kit de desarrollo de software de Oculus para integrar dispositivos VR en Unity y otras plataformas.
- Poda** f. Práctica agrícola que consiste en cortar y eliminar ramas o brotes de las plantas para mejorar su forma y productividad.
- Poda de Cabeza** f. Corte realizado a cierta altura sobre el suelo para fomentar el desarrollo de un tronco robusto.
- Poda de Formación** f. Poda aplicada en las primeras etapas de desarrollo de un árbol para guiar su estructura.
- Propagación Vegetativa** f. Método de reproducción de plantas mediante esquejes u otras partes vegetativas.
- Realidad Virtual (RV)** f. Tecnología que permite la interacción con entornos generados por computadora a través de dispositivos como cascos y guantes.

Retroalimentación Háptica f. Simulación de sensaciones táctiles mediante vibraciones o fuerzas en dispositivos electrónicos.

Retroalimentación Sonora f. Uso de señales de audio para informar al usuario sobre el estado o resultado de una acción.

ScriptableObject m. Clase de Unity que permite guardar datos y configuraciones de forma modular y reutilizable.

Sistema de Crecimiento de Árboles m. Conjunto de técnicas y estructuras (por ejemplo KGB) para organizar la forma de los árboles.

SUS (System Usability Scale) f. Cuestionario de 10 ítems para medir la usabilidad percibida de un sistema.

Tutorial Interactivo m. Guía paso a paso dentro del sistema que enseña al usuario los controles y mecánicas básicas.

Usabilidad f. Grado en que un producto puede ser usado por usuarios específicos para lograr objetivos con efectividad, eficiencia y satisfacción.

Variables Objetivas f. Métricas cuantitativas registradas automáticamente (por ejemplo tiempo, errores, precisión) durante el uso de un sistema.

Visualización de Datos f. Técnica de presentar información cuantitativa en forma de gráficos o tablas para facilitar su interpretación.

Bibliografía

- [1] Rifat Bhat, KM Bhat, Sharbat Hussain, M Maqbool Mir, Umar Iqbal, and Mehvish Bashir. Canopy management in stone fruits. *Production Technology of Stone Fruits*, pages 103–129, 2021.
- [2] JM Donoso, G Lemus, D Arribillaga, and B Sagredo. Evolution of the sweet cherry industry in chile. In *VIII International Cherry Symposium 1235*, pages 141–148, 2017.
- [3] TL Robinson, G Bujdoso, and G Reginato. Influence of pruning severity on fruit size of ‘sweetheart’, ‘lapins’ and ‘hedelfingen’ sweet cherry grown on gisela rootstocks. In *VI International Cherry Symposium 1020*, pages 441–451, 2009.
- [4] L. Jegó, N. Barriga, and F. Besoain. A virtual reality system for fruit processing line personnel training: A usability study. In *2022 IEEE International Conference on Automation/XXV Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, pages 1–5, 2022.
- [5] Ruben Smelik, Tim Tutenel, Klaas Jan De Kraker, and Rafael Bidarra. Integrating procedural generation and manual editing of virtual worlds. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, pages 1–8, 2010.
- [6] Yu Zang, Zhongxiang Zhu, Zhenghe Song, and Enrong Mao. Virtual reality and the application in virtual experiment for agricultural equipment. In *Computer and Computing Technologies in Agriculture IV: 4th IFIP TC 12 Conference, CCTA 2010, Nanchang, China, October 22-25, 2010, Selected Papers, Part III 4*, pages 257–268. Springer, 2011.

- [7] Misha Sra, Sergio Garrido-Jurado, Chris Schmandt, and Pattie Maes. Procedurally generated virtual reality from 3d reconstructed physical space. In *Proceedings of the 22nd ACM conference on virtual reality software and technology*, pages 191–200, 2016.
- [8] Manuel Cervera, Natalia Grandon, Marco Rivera, and Felipe Besoain. Improving the selection of iqf raspberries in processing lines: A virtual reality approach for training and selecting personnel. In *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, pages 1–7. IEEE, 2018.
- [9] Jules Janick. The origins of fruits, fruit growing, and fruit breeding. *Plant breeding reviews*, 25(1):255–320, 2005.
- [10] P. V. Quiñones. Boletín de fruta - abril 2024, 2024. ODEPA, Oficina de Estudios y Políticas Agrarias.
- [11] Gamalier Lemus et al. El cultivo del cerezo. Technical report, Instituto de Investigaciones Agropecuarias (INIA), Centro de Investigación INIA La Platina, Santiago, Chile, 2005.
- [12] L Pefaur, Oficina de Estudios, Políticas Agrarias Odepa, et al. Evolución de la fruticultura chilena en los últimos 20 años. 2020.
- [13] Miguel Ojer, Gabino Reginato Meza, Felipe Vallejos, and Alain Boulet. Poda de formación y producción en cerezos. *Revista de Fruticultura*, 2011.
- [14] Gregory A Lang, Stefano Musacchi, Matthew David Whiting, et al. Cherry training systems. *Journal of Horticultural Sciences*, 42(3):123–134, 2015.
- [15] Javier Andreu Lahoz. El cerezo. *Aragón Agrario. Revista del sector agrícola aragonés*, 3, 2018.
- [16] Hansjörg Haas. *La poda de frutales*. Editorial HISPANO EUROPEA, 2017.
- [17] M Villasante, S Godoy, J Zoffoli, and M Ayala. Efecto de la poda en el crecimiento vegetativo y la calidad de fruta en cerezo dulce (*prunus avium*). *Ciencia e Investigación Agraria*, 39(1):117–126, 2012.

- [18] V Usenik, A Solar, D Meolič, and F Štampar. Effects of summer pruning on vegetative growth, fruit quality and carbohydrates of ‘regina’ and ‘kordia’ sweet cherry trees on ‘gisela 5’. *European Journal of Horticultural Science*, 73(2):62–68, 2008.
- [19] A D Webster and N E Looney. *Cherries: Crop Physiology, Production and Uses*. CAB International, 1996.
- [20] C H Stone, G A Lang, and S Musacchi. Training systems for sweet cherry: Light relations, fruit yield and quality. *Agronomy*, 12(3):643, 2022.
- [21] Hans Stefan, Michael Mortimer, and Ben Horan. Evaluating the effectiveness of virtual reality for safety-relevant training: a systematic review. *Virtual Reality*, 27(4):2839–2869, 2023.
- [22] D. Scorgie, Z. Feng, D. Paes, F. Parisi, T.W. Yiu, and R. Lovreglio. Virtual reality for safety training: A systematic literature review and meta-analysis. *Safety science*, 171:106372, 2024.
- [23] Jaziar Radianti, Tim A. Majchrzak, Jennifer Fromm, and Isabell Wohlgenannt. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. *Computers Education*, 147:103778, 2020.
- [24] Armando de la Re, Francisco Abad, Emilio Camahort, and M Carmen Juan. Tools for procedural generation of plants in virtual scenes. In *Computational Science–ICCS 2009: 9th International Conference Baton Rouge, LA, USA, May 25–27, 2009 Proceedings, Part II 9*, pages 801–810. Springer, 2009.
- [25] O. Št’ava, B. Beneš, R. Měch, D. G. Aliaga, and P. Křištof. Inverse procedural modeling by automatic generation of l-systems. *Computer graphics forum*, 29(2):665–674, 2010.
- [26] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. Springer Science & Business Media, 2012.
- [27] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

- [28] Felipe Besoain, Liza Jago, and Ismael Gallardo. Developing a virtual museum: Experience from the design and creation process. *Information*, 12(6):244, 2021.
- [29] Briony J Oates, Marie Griffiths, and Rachel McLean. *Researching information systems and computing*. Sage, 2022.
- [30] Kapil Chalil Madathil and Joel S Greenstein. An investigation of the efficacy of collaborative virtual reality systems for moderated remote usability testing. *Applied ergonomics*, 65:501–514, 2017.
- [31] Jonathan Linowes. *Unity virtual reality projects: Learn virtual reality by developing more than 10 engaging projects with unity 2018*. Packt Publishing Ltd, 2018.
- [32] Jonathan Linowes. *Unity 2020 virtual reality projects: Learn VR development by building immersive applications and games with Unity 2019.4 and later versions*. Packt Publishing Ltd, 2020.
- [33] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.
- [34] Robert N. Gwynne. Globalisation, commodity chains and fruit exporting regions in chile. *Tijdschrift voor economische en sociale geografie*, 90(2):211–225, 1999.
- [35] CONICYT (Comisión Nacional de Investigación Científica y Tecnológica). El sector frutícola en chile: Capacidades de investigación y áreas de desarrollo científico-tecnológico. Technical report, Comisión Nacional de Investigación Científica y Tecnológica, 2004.
- [36] Comisión Económica para América Latina y el Caribe (CEPAL). Digitalización y cambio tecnológico en mipymes agrícolas y agroindustriales en américa latina y el caribe, junio 2021.

A. Casos de Uso

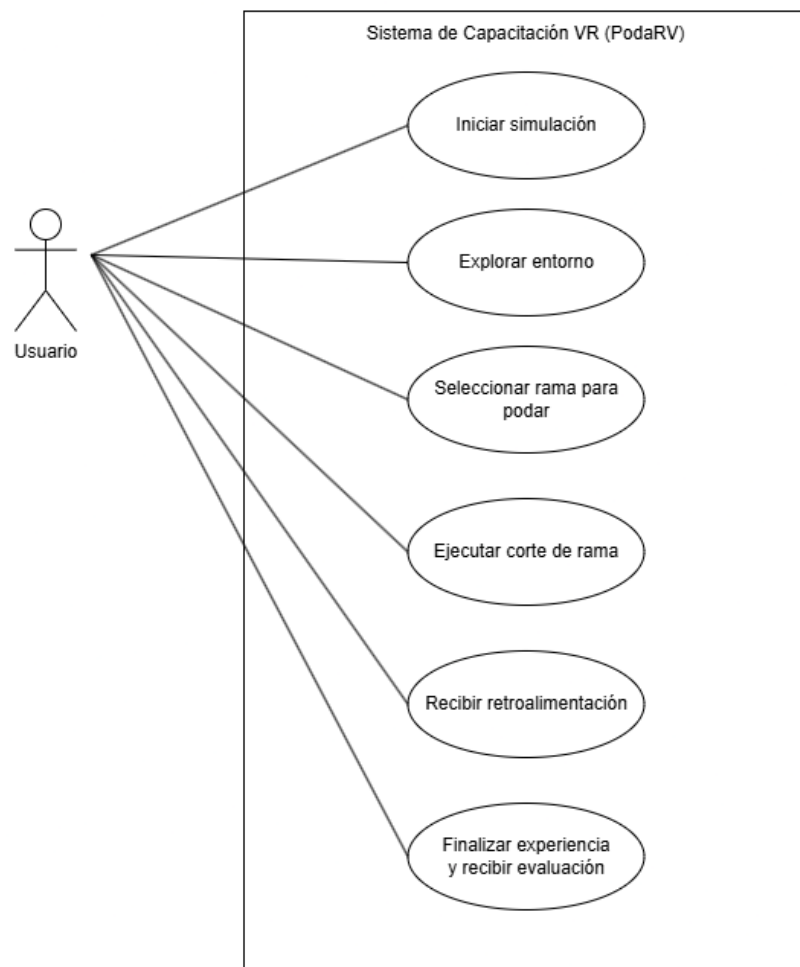


Figura A.1: Diagrama de Casos de Uso del sistema de capacitación en poda VR.

ID	Nombre del Caso de Uso	Actor	Descripción	Resultado Esperado
CU1	Iniciar simulación	Usuario	El usuario accede al sistema y comienza la experiencia en el entorno virtual del huerto.	La simulación comienza correctamente y el usuario es transportado al entorno virtual.
CU2	Explorar entorno	Usuario	El usuario se desplaza libremente por el entorno utilizando controles de movimiento.	El usuario puede navegar sin obstáculos y observar los elementos del entorno.
CU3	Seleccionar rama para poda	Usuario	El usuario selecciona una rama a podar, la cual es resaltada por el sistema.	La rama seleccionada queda marcada y lista para ser cortada.
CU4	Ejecutar corte de rama	Usuario	El usuario ejecuta el corte de la rama mediante una acción específica y el sistema registra la acción.	El corte es ejecutado y validado por el sistema según las reglas predefinidas.
CU5	Recibir retroalimentación	Usuario	El sistema entrega retroalimentación visual y sonora en base a la calidad del corte realizado.	El usuario recibe una respuesta inmediata sobre la calidad del corte.
CU6	Finalizar experiencia y recibir evaluación	Usuario	Al finalizar la sesión, el sistema entrega una evaluación general del desempeño del usuario.	El sistema muestra un resumen con el desempeño total del usuario.

Tabla A.1: Casos de Uso del Sistema

B. Carta Gantt

Año		2024																			
Meses		Agosto				Septiembre				Octubre				Noviembre				Diciembre			
Nº	Actividades / Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Definición de Tema.																				
2	Definición de Objetivo general y específicos.																				
3	Definición Problema o Oportunidad																				
4	Definir Pregunta de Investigación.																				
5	Investigación de la Poda de Cerezos.																				
6	Investigación de Recurso Humano en el Cerezo.																				
7	Investigación de Capacitación en Realidad Virtual.																				
8	Incluir citas en Resumen.																				
9	Entrega de Propuesta de Memoria.																				
10	Revisión y Ajustes en la propuesta.																				
11	Investigación para el Marco Teórico / Estado del Arte.																				
12	Entrega de Avance de Proyecto de Memoria.																				
13	Investigación del Estado del Arte.																				
14	Investigación de Metodología																				
15	Ajustes y detalles del Documento.																				
16	Entrega Final del Proyecto de Memoria																				
17	Creación de la Presentación Proyecto de Memoria.																				
18	Defensa de Proyecto de Memoria.																				

Figura B.1: Carta Gantt - Inicio de Memoria (Proyecto de Memoria de Título)

Año		2025																			
Meses		Marzo				Abril				Mayo				Junio				Julio			
Nº	Actividades / Semanas	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Crear Repositorio en Git hub																				
2	Crear Proyecto en Unity																				
3	Generar distintos Casos de uso para la experiencia VR																				
4	Elaborar esquema para guiar la experiencia del Usuario.																				
5	Agregar Sistema de XR al Proyecto																				
6	Crear Escena Principal																				
7	Implementar Camara y Manos para VR																				
8	Crear Objetos Interactuables																				
9	Crear Mecanica de Interaccion Mano-Objeto																				
10	Crear Sistema de Movimiento																				
11	Buscar Assets Iniciales																				
12	Implementacion de Algoritmo de Generacion Procedural para Arboles																				
13	Diseñar Escenario y Ambiente																				
14	Creacion de Interfaz para el Usuario																				
15	Prueba Inicial de Usabilidad y Análisis																				
16	Pruebas de Interacción																				
17	Optimización de Rendimiento																				
18	Ajustes de Diseño de UI																				
19	Implementación de Sonido y Música																				
20	Pruebas de Usabilidad Intermedia																				
21	Refinamiento de la Generación Procedural de Árboles																				
22	Revisión de Diseño y Flujo de la Experiencia																				
23	Preparación de Contenido Final																				
24	Revisión de Diseño y Flujo de la Experiencia																				
25	Pruebas Finales de Usabilidad y Análisis																				
26	Revisión de Feedback y Ajustes Finales																				

Figura B.2: Carta Gantt - Implementación de Prototipo (Memoria de Título)

Año		2025							
Meses		Junio				Julio			
Nº	Actividades / Semanas	1	2	3	4	1	2	3	4
1	Definir las pruebas con el Prototipo.								
2	Realizar todas las pruebas.								
3	Analizar Pruebas.								
4	Escribir Resultados de las pruebas.								
5	Escribir Conclusiones.								
6	Creacion de Presentacion para Defensa de Memoria.								
7	Defensa de Memoria.								

Figura B.3: Carta Gantt - Prototipo Final y Defensa de Memoria

C. Fruticultura e Impacto en Chile

C.1. Inicio de la Fruticultura

La fruticultura tiene raíces que se remontan a hace miles de años, entre 4000 y 3000 a.C., en un lugar conocido como el Creciente Fértil. Este área, que recorre partes de Mesopotamia y el valle del Nilo en Egipto, fue testigo de un cambio relevante en cómo las sociedades se relacionaban con la naturaleza. En lugar de depender exclusivamente de la recolección de frutas silvestres, las comunidades empezaron a cultivar sus propios árboles frutales de manera más organizada y planificada. Este giro coincidió con el surgimiento de asentamientos urbanos y ciudades-estado, creando una necesidad de tener una agricultura más estable y sostenible. Cultivar árboles como las palmeras datileras y los olivos representaba un compromiso con la tierra, lo que a su vez ayudó a formar sociedades más complejas y organizadas [9]. A medida que la fruticultura se desarrollaba, se comenzaron a aplicar técnicas específicas como la propagación vegetativa, el injerto y la selección de especies con características favorables para el cultivo. Estas prácticas permitieron la mejora y expansión de los cultivos frutales, adaptándolos a las necesidades humanas y las condiciones climáticas locales. El uso de métodos como el injerto, documentado desde épocas tan tempranas como el 3000 a.C., posibilitó la producción de árboles más resistentes y productivos, favoreciendo la diversificación de la agricultura en las primeras civilizaciones. Este enfoque cuidadoso hacia el cultivo de frutas estableció una base tecnológica que ha continuado evolucionando a lo largo de los siglos [9].

C.2. Fruticultura en Chile

En Chile, el sector frutícola ha sido fundamental para el desarrollo agrícola y económico. Desde los años 80, políticas de exportación, incentivos fiscales y reformas agrarias han impulsado su crecimiento, posicionando al país como líder en la producción y exportación de frutas. Esta capacidad de adaptación frente a desafíos económicos ha sido clave para sostener el dinamismo del sector [34]. Durante las décadas posteriores, las exportaciones agrícolas y agroindustriales se diversificaron y crecieron potencialmente. Las frutas, en particular, se destacaron como el motor principal de este impulso económico. A medida que Chile consolidaba su posición en el comercio mundial, las reformas agrarias de las décadas anteriores contribuyeron a que más pequeños y medianos productores se integraran en esta actividad, facilitando el acceso a mercados internacionales [34].

A pesar de las dificultades económicas de los años 90, como la apreciación del peso chileno, el sector frutícola mantuvo su dinamismo. La capacidad de adaptación y la constante innovación en técnicas agrícolas ayudaron a sostener el crecimiento, posicionando a Chile como un líder en la producción y exportación de frutas, algo que sigue siendo crucial para la economía del país en la actualidad [34].

C.3. Impacto Socioeconómico de la Fruticultura

La fruticultura tiene un impacto significativo en la economía y el desarrollo rural de Chile, contribuyendo al PIB, generando empleo y fortaleciendo las exportaciones.

- **Contribución Económica:** El sector frutícola aporta de manera notable al PIB nacional y al sector agropecuario, siendo fuente de empleo directo e indirecto. Por ejemplo, en 2004 representaba el 1,46 % del PIB total y el 31,9 % del PIB del sector agropecuario y forestal, con un aporte global del 2,6 % del PIB [35].
- **Exportaciones:** La diversificación de la producción ha consolidado a Chile como líder en exportaciones frutícolas. Entre enero y marzo de 2024 se exportaron 1.125.220 toneladas de fruta, alcanzando un valor de USD 3.721 millones FOB. La cereza destaca como producto principal, con un crecimiento del 13 % en volumen y un valor de USD 2.005 millones FOB(Free on Board) [10].

- **Desarrollo Rural:** La actividad frutícola es crucial para el desarrollo rural, generando empleo en la producción, procesamiento y actividades complementarias. Su mayor actividad se concentra en regiones como Atacama y Los Lagos, donde reside el 92 % de la población rural, reafirmando su importancia en la economía local [36].

En conjunto, la fruticultura impulsa el crecimiento económico y el bienestar de las comunidades rurales, constituyendo un pilar fundamental del desarrollo nacional.

C.4. La Cereza: Trayectoria Histórica y su Desarrollo a Nivel Nacional

El cerezo (*Prunus avium* L.) es un árbol caducifolio originario de la región comprendida entre las costas del Mar Negro y del Mar Caspio. Perteneció a la clase Dicotyledonea, orden Rosales y familia Rosaceae, y se caracteriza por su notable adaptabilidad a diversas zonas, tanto nórdicas como continentales. Su estructura presenta una dominancia apical que limita el desarrollo de brotes laterales, lo que da lugar a una ramificación menos densa en comparación con otras especies. Además, sus hojas elípticas y caducas con bordes aserrados, junto a su madera de color pardo rojizo—altamente valorada en el comercio—son rasgos distintivos que han facilitado su cultivo en Chile [11].

El cultivo del cerezo en Chile ha experimentado una evolución significativa, marcada por la selección y adaptación de cultivares para satisfacer las demandas del mercado y las condiciones climáticas locales. Inicialmente se cultivaban variedades tradicionales como *Camelia*, *Ramón Oliva*, *Black Tartarian* y *Black Republican*; sin embargo, a partir de los años 70 la introducción de cultivares como *Bing* y *Van* mejoró la calidad de la fruta y potenció las exportaciones hacia mercados internacionales, incluyendo Brasil y Estados Unidos. Actualmente, con más de 60 variedades plantadas, se ha logrado extender el período de cosecha y optimizar la calidad del producto. Variedades como *Brooks*, *Newstar*, *Lapins*, *Stella*, *Van*, *Celeste*, *Sweet Heart*, *Ruby* y *Garnet* se destacan por su firmeza, resistencia al transporte y características de maduración, factores críticos especialmente en regiones con riesgo de lluvias y temperaturas extremas. Esta continua diversificación es esencial para mantener y mejorar la competitividad de la industria del cerezo en Chile [11].

C.5. Importancia del Recurso Humano en el Cultivo de Cereza

El desarrollo de los cultivos de cerezos destaca la **relevancia del recurso humano en la gestión y evolución de esta industria a lo largo del tiempo**, tema central en esta memoria. Chile se ha consolidado como el productor y exportador de frutas más relevante en el hemisferio sur, gracias a su amplia variedad de frutas frescas y frutos secos. Además, cuenta con una trayectoria significativa que lo posiciona como un proveedor de categoría mundial, reconocido por su fiabilidad y el cumplimiento de altos estándares de inocuidad alimentaria [12].

Según datos del catastro frutícola Odepa-Ciren, en el año 2013 se registraron 568.990 trabajadores en el sector frutícola, de los cuales un 16 % eran trabajadores permanentes y un 84 % temporales. Para el año 2019, la cifra total de trabajadores aumentó a 571.894, con una distribución de 15 % de permanentes y 85 % de temporales [12].

Región	2013			2019			Variación 2013/2019		
	Permanente	Temporal	Total	Permanente	Temporal	Total	Permanente	Temporal	Total
Arica y Parinacota	s/i	s/i	s/i	838	1.439	2.277	--	--	--
Tarapacá	s/i	s/i	s/i	336	354	690	--	--	--
Atacama	3.339	19.596	22.935	2.317	16.708	19.025	-31%	-15%	-17%
Coquimbo	8.192	41.466	49.658	7.973	39.720	47.693	-3%	-4%	-4%
Valparaíso	15.086	52.641	67.727	13.477	51.878	65.355	-11%	-1%	-4%
Metropolitana	12.156	58.714	70.870	12.247	46.741	58.988	1%	-20%	-17%
O'Higgins	27.647	133.852	161.499	19.732	112.848	132.580	-29%	-16%	-18%
Maule	14.307	101.506	115.813	16.376	114.839	131.215	14%	13%	13%
Ñuble	s/i	s/i	s/i	4.415	41.409	45.824	--	--	--
Biobío	4.760	44.329	49.089	1.290	12.614	13.904	-73%	-72%	-72%
La Araucanía	2.162	9.551	11.713	2.769	25.813	28.582	28%	170%	144%
Los Ríos	815	6.703	7.518	980	13.752	14.732	20%	105%	96%
Los Lagos	1.131	11.037	12.168	771	9.548	10.319	-32%	-13%	-15%
Aysén	s/i	s/i	s/i	81	629	710	--	--	--
Total estimado	89.595	479.395	568.990	83.602	488.292	571.894	-6,7%	1,9%	0,5%

Tabla C.1: Distribución de la mano de obra permanente y temporal en predios e industrias frutícolas por región. Número de trabajadores 2013 y 2019 Fuente: [12].

Como se muestra en la tabla C.1, aunque el número total de trabajadores creció solo un 0,5 % durante el período analizado, hubo un notable incremento en los trabajadores temporales y una disminución en los permanentes. Este cambio podría

atribuirse al uso creciente de maquinaria que reemplaza ciertas tareas realizadas anteriormente por trabajadores permanentes, como el riego. El aumento en la mano de obra temporal estaría relacionado con la expansión de la superficie frutícola, que requiere un mayor número de trabajadores para labores intensivas como la cosecha y la **poda** [12].

Un claro ejemplo de esta tendencia se observa en la Región del Maule, que se destaca no solo por su alta densidad de plantas por hectárea, sino también por ser una de las principales regiones productoras y exportadoras de frutas del país. Esta región ha experimentado un crecimiento representativo y valioso en la superficie frutícola durante las últimas dos décadas, lo que ha generado una mayor demanda de mano de obra temporal para satisfacer las necesidades de tareas como la cosecha y la poda. Este aumento en la actividad frutícola en la Región del Maule refuerza la importancia del recurso humano en la sostenibilidad y desarrollo continuo de la industria [12].

D. Clasificación de Tipos de Poda para la Cereza

La poda del cerezo cumple roles fundamentales en su cultivo, ya que permite tanto el control de su crecimiento como la optimización para la producción de fruta [13].

D.1. Clasificación por Objetivo

D.1.1. Poda de Plantación

Este tipo de poda se aplica al momento de plantar, buscando equilibrar tanto la parte aérea como el sistema radicular de la planta. La intención es comenzar a dar forma al sistema de conducción de la planta, eliminando principalmente brotes tempranos para que el crecimiento sea más fuerte y directamente desde el tronco [13].

D.1.2. Poda de Formación

Esta fase es esencial para definir la estructura básica del árbol y mantener su tamaño en un volumen manejable para la recolección y el mantenimiento. Durante esta etapa, el árbol se guía para desarrollar una forma específica. En regiones donde el cerezo se cultiva principalmente para fruta, se suele optar por un estilo de “vaso bajo”, caracterizado por un tronco corto de unos 40 cm y de 4 a 8 ramas principales dispuestas de manera uniforme. Este tipo de poda incluye intervenciones durante el período de crecimiento para guiar la estructura deseada mediante despuntes y ajustes. Luego, en invierno, una poda más intensa elimina las ramas innecesarias para consolidar la estructura básica del árbol [15].

D.1.3. Poda de Fructificación O Producción

Una vez establecida la estructura, se lleva a cabo anualmente para mejorar la producción de fruta y asegurar la salud del árbol. Esta poda se enfoca en la eliminación de ramas viejas y cansadas después de la cosecha, buscando mejorar la penetración de aire y luz en toda la copa y limitar la altura del árbol a un nivel adecuado. Durante la poda de producción, también se manejan ramas específicas para la fructificación: ramas estructurales, ramas mixtas (con yemas de flor y de madera), y las “chifonas” y “ramilletes de mayo”, que concentran la mayoría de las yemas de flor para la producción. Estos ramilletes aparecen en ramas de al menos dos años, siendo los principales responsables de la producción de fruta [15].

D.1.4. Poda de Rejuvenecimiento

La poda de rejuvenecimiento en los cerezos se realiza en el verano, extendiéndose hasta finales de esta estación, cuando el árbol puede recuperarse más eficientemente. Este proceso consiste en redirigir todas las puntas de las ramas estructurales colgantes y envejecidas hacia una rama joven situada más cerca del tronco, manteniendo la estructura y promoviendo el crecimiento de madera joven. Después, se aligeran las nuevas puntas de las ramas para equilibrar el crecimiento [16] .

Es importante evitar heridas de gran tamaño en las ramas, ya que algunos cortes grandes no se curan adecuadamente y pueden comprometer la salud del árbol. Si se requiere eliminar una rama gruesa, se debe dirigir el corte hacia una rama lateral orientada hacia afuera y ubicada cerca de la estructura principal, evitando daños mayores. En caso de que no exista una rama joven para hacer esta transición, se deja un tocón de aproximadamente 10 a 20 cm de longitud [16] .

Un tocón es el remanente de una rama que se deja intencionalmente en la poda cuando no se dispone de una rama joven para el redireccionamiento . Este segmento de rama contribuye a la cicatrización gradual del corte, protegiendo la estructura principal del árbol de daños.

D.2. Clasificación por Época

- **Poda de Invierno:** Se realiza en período de inactividad del árbol (final del invierno o principios de la primavera) o llamado receso vegetativo [13], per-

mitiendo que el árbol concentre energía en el crecimiento nuevo durante la primavera.

- **Poda de Verano o en Verde:** Es una poda que se realiza cuando la planta está en pleno follaje, durante las estaciones de primavera y verano. Se lleva a cabo desde que los brotes empiezan a alargarse, a mediados de primavera, hasta poco antes de que las hojas caigan, mientras aún están activas [13]. La poda durante el verano ayuda a disminuir el crecimiento excesivo del árbol y reduce el riesgo de que los cortes sufran infecciones [14].

D.3. Clasificación según sistema de poda

- **Poda Larga:** Este sistema de poda se caracteriza por dejar las ramas sin recortar ni despuntar. En lugar de cortar, se permite que el peso de los frutos, como los duraznos, doble naturalmente las ramas, estimulando el crecimiento de nuevos brotes en la parte superior para el ciclo siguiente. Este tipo de poda es común en algunos cultivos, especialmente en la producción de duraznos para consumo en fresco [13].
- **Poda Corta:** En la poda corta, se recortan o despuntan las ramas más delgadas para asegurar la renovación del material reproductivo de la planta [13].
- **Poda Mixta:** Este sistema combina los dos métodos anteriores dentro de la misma planta. Los productores que utilizan la poda mixta aplican poda larga en las ramas altas para permitir un crecimiento más natural, mientras que emplean poda corta en las secciones media y baja para mejorar la renovación del material reproductivo [13].

D.4. Tipos de Entrenamiento para la Poda

A continuación, se describen distintos sistemas de poda utilizados en la cereza:

D.4.1. Kym Green Bush (KGB)

El Kym Green Bush (KGB) es un árbol autoportante que consiste en múltiples unidades de fructificación vertical temporales para moderar el vigor del árbol, per-

mitiendo que el sistema se cultive con patrones de raíces de tamaño completo o semi-enanizantes. Las unidades de fructificación vertical se renuevan regularmente para mantener los espolones jóvenes y productivos. El KGB es el único sistema cubierto en este manual que crea un huerto completamente peatonal, uno que se puede cosechar sin escaleras ni plataformas. La formación del árbol es fácil y requiere un trabajo mínimo; la poda madura es simple y sigue un plan repetible [14].

D.4.2. Spanish Bush (SB)

El Spanish Bush (SB) es un árbol autoportante que consiste en múltiples líderes permanentes para moderar el vigor del árbol, permitiendo que el sistema se cultive con patrones de raíces de tamaño completo o semi-enanizantes. Las unidades de fructificación laterales se desarrollan y renuevan en los líderes. El SB produce un huerto semi-peatonal que se puede cosechar desde escaleras pequeñas. La formación del árbol es fácil y requiere un trabajo mínimo; la poda madura consiste en renovar la madera de fructificación horizontal en los andamiajes verticales permanentes [14].

D.4.3. Steep Leader (SL)

El Steep Leader (SL) se puede describir mejor como un árbol de múltiples líderes en tri- o quad-ejes autoportantes con ramas de andamiaje horizontales que proyectan desde la base de cada líder. Esta arquitectura de dosel crea una forma piramidal que facilita una buena distribución de luz en todo el árbol. Debido al número limitado de líderes, los árboles SL en patrones de raíces de tamaño completo pueden crecer de 18 a 20 pies o más. La madera de fructificación se desarrolla en los andamiajes verticales y horizontales permanentes, y cada andamiaje se trata como un árbol de huso de un solo lado. La poda madura consiste en la renovación regular de una porción de toda la madera de fructificación [14].

D.4.4. Super Slender Axe (SSA)

El Super Slender Axe (SSA) es un sistema de muy alta densidad, semi-peatonal, de hasta 2,000 árboles de un solo líder semi-autoportantes por acre, que generalmente requieren patrones de raíces enanizantes y un enrejado de alambre superior. El hábito de producción de los árboles SSA es significativamente diferente de otros sistemas, con la mayoría de la fruta creciendo en yemas florales axilares producidas en la base

de ramas de un año. Este aspecto, combinado con el hecho de que la fructificación se ubica en proximidad al eje, tiene un efecto positivo en el tamaño de la fruta. La poda madura consiste en la renovación anual de todas las ramas laterales en el eje, lo que requiere un trabajo simplificado pero extenso anualmente [14].

D.4.5. Tall Spindle Axe (TSA)

El Tall Spindle Axe (TSA) es una evolución de los sistemas Vogel Central Leader y Zahn Spindle, incorporando nuevos conceptos a estos sistemas anteriores de un solo líder autoportante. El dosel del árbol TSA se caracteriza por un verticilo continuo de ramas laterales moderadamente vigorosas desarrolladas por la activación de yemas seleccionadas (a través de scoring de yemas, eliminación de yemas o aplicación de reguladores de crecimiento), generalmente en patrones de raíces semi-enanizantes para un huerto semi-peatonal. El TSA se diferencia aún más por dos conceptos clave: 1) la cabezada anual de brotes laterales para equilibrar las proporciones de hojas a fruta y futuras cargas de cultivo, y 2) la renovación anual de las ramas de fructificación más antiguas, de modo que la única estructura permanente sea el líder central [14].

D.4.6. Upright Fruiting Offshoots (UFO) y “Y” Trellis (UFO-Y)

El Upright Fruiting Offshoots (UFO) es un sistema completamente enrejado que optimiza la eficiencia laboral y la calidad de la fruta al crear una pared de fructificación estrecha que es precoz y fácil de cosechar y podar. Al igual que el sistema KGB, produce fruta en unidades de fructificación vertical renovables, que surgen de un tronco o cordón horizontal bajo. La poda madura se reduce a un simple proceso de dos pasos. Las ventajas de este sistema semi-peatonal son altos rendimientos tanto en etapas tempranas como en madurez, y un buen movimiento de aire y distribución de luz para reducir la incidencia de enfermedades y promover frutas uniformes y de alta calidad en todo el dosel. Aunque el sistema es fácil de podar y mantener en madurez, el establecimiento de los verticales del dosel enrejados es más intensivo, laborioso y costoso que otros sistemas [14].

El UFO puede ser entrenado en una arquitectura de dosel en “Y” de doble plano (UFO-Y) que se desarrolla y gestiona de manera similar al enrejado vertical de un solo plano UFO; sin embargo, las unidades de fructificación renovables se entrenan

alternativamente para crecer aproximadamente a 30 grados de la vertical, formando ambos lados de un enrejado en “Y”. El UFO-Y ofrece la ventaja de mayores rendimientos debido a una mayor interceptación de luz, así como el potencial para la cosecha mecánica o asistida mecánicamente [14].

D.4.7. Vogel Central Leader (VCL)

El Vogel Central Leader (VCL) es un árbol autoportante de un solo líder con andamiajes de fructificación renovables que promueve rendimientos tempranos en huertos de alta densidad moderada. El VCL fomenta la precocidad mediante una poda mínima temprana y el establecimiento de ángulos de ramas planos, lo que requiere relativamente altos insumos laborales en los años de establecimiento, pero bajos niveles de mantenimiento en la madurez. La forma piramidal del árbol promueve una buena distribución de luz en todo el dosel. Debido a la naturaleza de un solo líder de este sistema, es necesario un patrón de raíces enanizante para ayudar a mantener la altura del árbol entre 10 y 12 pies en madurez, creando un huerto semi-peatonal [14].

E. Diseño E Implementación

E.1. Módulo de Poda Interactiva

Objetivo: Permitir al usuario ejecutar cortes de ramas de forma interactiva dentro del entorno de realidad virtual, simulando una experiencia de poda realista.

Funciones principales:

- Detectar colisión o selección de una rama por parte del usuario.
- Ejecutar animación y acción de corte al activar el input correspondiente.
- Enviar datos al módulo de evaluación para validar el corte.

Entradas:

- Posición del controlador VR.
- Señal de activación (botón de corte).
- Árbol generado con ramas seleccionables.

Salidas:

- Confirmación visual del corte ejecutado.
- Activación de feedback (sonido).

Interacción con otros módulos: Este módulo se comunica directamente con el módulo de feedback visual y el módulo de evaluación de resultados.

E.2. Módulo de Feedback Visual y Sonoro

Objetivo: Proporcionar retroalimentación inmediata al usuario sobre la calidad de los cortes realizados, utilizando estímulos visuales y sonoros para reforzar el aprendizaje.

Funciones principales:

- Activar sonidos específicos según el tipo de corte (correcto o incorrecto).
- Cambiar el color de la rama cortada según su evaluación.
- Mostrar indicadores visuales como mensajes breves o íconos contextuales.

Entradas:

- Resultado del análisis del corte (correcto, incorrecto).
- Señal de finalización de acción desde el módulo de poda.

Salidas:

- Efectos sonoros específicos (positivo o negativo).
- Cambio de color o indicadores visuales sobre el árbol o interfaz.

Interacción con otros módulos: Este módulo recibe información desde el módulo de evaluación y está directamente vinculado al módulo de poda interactiva para generar feedback inmediato.

E.3. Módulo de Evaluación y Resultados

Objetivo: Analizar la calidad de las acciones del usuario durante la simulación y generar una evaluación final basada en criterios técnicos de poda.

Funciones principales:

- Comparar cada corte con una base de reglas definidas.
- Calcular una puntuación de desempeño por árbol y total.
- Generar una retroalimentación final sobre la sesión de práctica.

Entradas:

- Datos del corte recibido desde el módulo de poda (posición, tipo de rama, grosor).
- Tiempo total de la sesión.

Salidas:

- Resultado por corte (correcto/incorrecto).
- Informe resumido del desempeño.
- Puntuación global.

Interacción con otros módulos: Este módulo recibe entradas del módulo de poda y envía resultados al módulo de feedback y a la interfaz final de usuario.

E.4. Módulo de Generación Procedural de Árboles

Objetivo: Generar árboles virtuales con estructuras variadas y realistas utilizando gramáticas L-System especializadas en el patrón de poda KGB.

Funciones principales:

- Aplicar gramáticas L-System personalizadas para construir árboles tridimensionales.
- Generar variedad estructural en cada instancia de simulación.
- Definir características visuales como grosor, altura, ramificación y textura.

Entradas:

- Parámetros definidos por el sistema (grosor, longitud).
- Gramáticas de generación configuradas por el desarrollador.

Salidas:

- Árboles 3D completos y listos para interacción.
- Datos estructurales para detección de ramas.

Interacción con otros módulos: Este módulo entrega los modelos generados al módulo de poda interactiva, permitiendo su uso directo en la simulación.

Con el objetivo de simular con precisión los distintos niveles de complejidad en la poda de cerezos bajo el sistema KGB (Kym Green Bush), se diseñaron tres gramáticas paramétricas independientes, correspondientes a los niveles 1, 2 y 3 del sistema de capacitación. Cada una de estas gramáticas fue implementada mediante un componente `ScriptableObject` en Unity, configurado para controlar la estructura del árbol mediante reglas de producción que gobiernan el crecimiento del tronco, las ramas primarias y las ramas secundarias.

Las diferencias entre niveles se manifiestan principalmente en:

- El grosor y longitud inicial del tronco.
- El comportamiento de ramificación (cantidad y tipo de bifurcaciones).
- La complejidad geométrica de las ramas secundarias.

E.4.1. Codigos Necesarios para el funcionamiento del Sistema

```

1  using Antlr.Runtime.Tree;
using System;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using Utils = ParametrizedUtilities;

[System.Serializable]
public class Generator
{
11  [System.Serializable]
    public class PrefPairs
    {
        public string name;
        public List<GameObject> prefs;
    }

    public float alphaAngle = 45f;
    public float betaAngle = 60f;
    public float segmentSize = 1f;
    public List<PrefPairs> prefPairs = new();
21  /// <summary>
/// Cada acci n retorna el nuevo SubStruct creado (o el
    mismo 'last' si no se crea nada nuevo).
/// </summary>
    [System.Serializable]
    public class GenerationAction
    {
        public char variable;
        public Func<int, Transform, Stack<tortoiseData>,
            TreeStruct, SubStruct, List<float>, SubStruct>
            action;
    }

    public List<GenerationAction> generationAction = new();
31  /// <summary>

```

```

41  /// Instancia y configura un segmento (prefab) para el
    s mbolo dado, y devuelve el SubStruct creado.
    /// </summary>
    public SubStruct SimpleGeneration(
        int index,
        Transform tortoise,
        Stack<tortoiseData> pushdown,
        TreeStruct tree,
        SubStruct last,
        List<float> _params,
        string value)
    {
        /// Valores predeterminados en caso de que no se
        /// provean par metros
        float width = 1f;           // Valor por defecto
        /// para el grosor
        float length = segmentSize; // Valor por defecto
        /// para la longitud

        /// Si se proporcionan par metros y son al menos 2,
        /// asume:
        /// - _params[0]: ancho (w)
        /// - _params[1]: longitud (L)
        if (_params != null && _params.Count >= 2)
        {
            width = _params[0];
            length = _params[1];
        }

        /// Buscar el prefab asociado al s mbolo (por ejemplo,
        /// "F", "A", etc.)
        var tPart = prefPairs.FirstOrDefault(t => t.name ==
            value);
        if (tPart == null || tPart.prefs.Count == 0)
        {

```

```
        Debug.LogWarning($"No Prefab found for value: {  
            value}");  
        return last;  
    }  
  
    // Instanciar el prefab en la posici n actual de la  
    // tortuga.  
    var part = GameObject.Instantiate(tPart.prefs[0],  
        tortoise.position, Quaternion.identity);  
    part.transform.SetParent(last.transform,  
        worldPositionStays: true);  
  
    // Asegurarse de que el objeto padre tenga escala  
    // (1,1,1)  
    part.transform.localScale = Vector3.one;  
    // Acceder al primer hijo del prefab y asignarle la  
    // escala deseada:  
    // Se asigna 'width' a X y Y, y 'length' a Z.  
    Transform child = part.transform.GetChild(0);  
    child.localScale = new Vector3(width, width, length);  
  
    // Alinear el prefab de modo que su eje Z (forward) se  
    // oriente seg n la direcci n de la tortuga.  
    part.transform.up = tortoise.up;  
  
    // Agregar el componente SubStruct y guardar los  
    // par metros si se requiere.  
    var sub = part.AddComponent<SubStruct>();  
    sub.size = length;  
    sub.parent = tree;  
    tree.subStructs.Add(sub);  
  
    // Calcular el desplazamiento real basado en la  
    // longitud del objeto hijo (eje Z):  
    Renderer rend = child.GetComponent<Renderer>();  
    if (rend != null)
```

```

    {
        float actualLength = rend.bounds.size.y;
        // Mover la tortuga en la direcci n en la que se
        // ha orientado el prefab.
        tortoise.position += tortoise.up * actualLength;
    }
    else
    {
        // Sino, usar la longitud definida.
        tortoise.position += tortoise.up * length;
    }

    return sub;
}

/// <summary>
/// Inicializa las acciones para cada s mbolo.
/// </summary>
public List<GenerationAction> InitGenerateAction()
{
    var toR = new List<GenerationAction>();

    // Rotaciones: actualizan la tortuga y devuelven 'last'
    // sin crear nuevos segmentos.
    toR.AddRange(new[]
    {
        new GenerationAction() {
            variable = '+',
            action = (i, tortoise, pushdown, tree, last,
                _params) => {
                float angle = (_params.Count > 0) ?
                    _params[0] : UnityEngine.Random.Range
                    (15f, 45f);
                tortoise.Rotate(tortoise.forward, angle);
                return last;
            }
        }
    });
}

```

```
121      },
      new GenerationAction() {
        variable = '-',
        action = (i, tortoise, pushdown, tree, last,
          _params) => {
          float angle = (_params.Count > 0) ?
            _params[0] : UnityEngine.Random.Range
              (15f, 45f);
          tortoise.Rotate(tortoise.forward, -angle);
          return last;
        }
      },
      new GenerationAction() {
        variable = '&',
        action = (i, tortoise, pushdown, tree, last,
          _params) => {
          float angle = (_params.Count > 0) ?
            _params[0] : UnityEngine.Random.Range
              (15f, 45f);
          tortoise.Rotate(tortoise.right, angle);
          return last;
        }
      },
      new GenerationAction() {
        variable = '^',
        action = (i, tortoise, pushdown, tree, last,
          _params) => {
          float angle = (_params.Count > 0) ?
            _params[0] : UnityEngine.Random.Range
              (15f, 45f);
          tortoise.Rotate(tortoise.right, -angle);
          return last;
        }
      },
      new GenerationAction() {
        variable = '\\',
```

```

        action = (i, tortoise, pushdown, tree, last,
            _params) => {
            float angle = (_params.Count > 0) ?
                _params[0] : UnityEngine.Random.Range
                    (60f, 60f);
            tortoise.Rotate(tortoise.forward, angle);
            return last;
        }
    },
    new GenerationAction() {
        variable = '/',
        action = (i, tortoise, pushdown, tree, last,
            _params) => {
            float angle = (_params.Count > 0) ?
                _params[0] : UnityEngine.Random.Range
                    (60f, 60f);
            tortoise.Rotate(tortoise.forward, -angle);
            return last;
        }
    }
});

// Símbolos que generan un segmento: usamos
SimpleGeneration.
var symbols = new[] { 't', 'g', 'f', 'r', 'h', 'p' };
foreach (var symbol in symbols)
{
    toR.Add(new GenerationAction()
    {
        variable = symbol,
        action = (i, tortoise, pushdown, tree, last,
            exps) => {
            return SimpleGeneration(i, tortoise,
                pushdown, tree, last, exps, symbol.
                    ToString());
        }
    })
}

```

```
171         });  
        }  
  
        // Acciones explícitas para F, A, B y C (con C usando  
        // la lógica de B).  
        toR.Add(new GenerationAction()  
        {  
            variable = 'F',  
            action = (i, t, p, tr, l, exps) => {  
                return SimpleGeneration(i, t, p, tr, l, exps,  
                    "F");  
            }  
        });  
181        toR.Add(new GenerationAction()  
        {  
            variable = 'A',  
            action = (i, t, p, tr, l, exps) => {  
                return SimpleGeneration(i, t, p, tr, l, exps,  
                    "A");  
            }  
        });  
        toR.Add(new GenerationAction()  
        {  
191            variable = 'B',  
            action = (i, t, p, tr, l, exps) => {  
                return SimpleGeneration(i, t, p, tr, l, exps,  
                    "B");  
            }  
        });  
        toR.Add(new GenerationAction()  
        {  
            variable = 'C',  
            action = (i, t, p, tr, l, exps) => {  
                return SimpleGeneration(i, t, p, tr, l, exps,  
                    "B");  
            }  
201        });  
    }
```

```

    });

    // Corchetes: push y pop de la tortuga
    toR.AddRange(new[]
    {
        new GenerationAction() {
            variable = '[',
            action = (i, tortoise, pushdown, tree, last,
                exps) => {
                pushdown.Push(new tortoiseData() {
                    pos = tortoise.position,
                    dir = tortoise.rotation.eulerAngles,
                    last = last.transform
                });
                return last;
            }
        },
        new GenerationAction() {
            variable = ']',
            action = (i, tortoise, pushdown, tree, last,
                exps) => {
                if (pushdown.Count > 0)
                {
                    var t = pushdown.Pop();
                    tortoise.position = t.pos;
                    tortoise.rotation = Quaternion.Euler(t
                        .dir);
                    var sub = t.last.GetComponent<
                        SubStruct>();
                    return sub;
                }
                return last;
            }
        }
    });

```

```

        return toR;
    }
    /// <summary>
    /// Genera el rbol 3D a partir de la cadena L-System.
    /// Procesa la cadena caracter a caracter. Si detecta un
        s mbolo parametrizado,
    /// extrae sus par metros y, si hay un sufijo que
        comience con '#' despu s de los par ntesis,
    /// lo interpreta como el uniqueId y lo asigna al
        SubStruct correspondiente.
    /// </summary>
241 public TreeStruct GenerateTree(string value, Transform
    tortoise)
    {
        var pushdown = new Stack<tortoiseData>();

        // Crear el GameObject ra z para el rbol
        var root = new GameObject("Tree");
        var tree = root.AddComponent<TreeStruct>();
        tree.LSys = value;
        Debug.Log("LSys: " + value);

251 // 'last' inicial es un SubStruct vac o en el root.
        var last = root.AddComponent<SubStruct>();
        last.parent = tree;
        last.uniqueId = Deriver.GenerateId(); // Asigna un ID
            al root
        int i = 0;
        while (i < value.Length)
        {
            char v = value[i];
            var actions = generationAction.Where(a => a.
                variable == v).ToList();

261 if (Utils.IsParameterized(value, i))
            {

```

271

281

291

```

// Extrae el contenido entre parntesis
string subStr = value.Substring(i + 1);
var (param, endIndex) = Utils.
    ExtractFromParentheses(subStr);
var exps = param.Split(';').Select(float.Parse
    ).ToList();

// Procesar el smbolo parametrizado
if (actions.Count > 0 && actions[0].action !=
    null)
{
    var newLast = actions[0].action(i,
        tortoise, pushdown, tree, last, exps);
    if (newLast != null)
        last = newLast;
}

// Avanzar el ndice hasta justo despu s del
// cierre de parntesis
i += (endIndex + 2); // 1 para la letra, (
    endIndex + 1) para "(...)"

// Procesar sufijos de ID consecutivos (por
// ejemplo, "#0", "#1", etc.)
while (i < value.Length && value[i] == '#')
{
    i++; // Salta el '#'
    int idStart = i;
    while (i < value.Length && char.IsDigit(
        value[i]))
    {
        i++;
    }
    int idLength = i - idStart;
    if (idLength > 0)
    {

```

301

```

        int uniqueId = int.Parse(value.
            Substring(idStart, idLength));
        last.uniqueId = uniqueId;
    }
}
else
{
    // S mbolo no parametrizado
    if (actions.Count > 0 && actions[0].action !=
        null)
    {
        var newLast = actions[0].action(i,
            tortoise, pushdown, tree, last, new
            List<float>());
        if (newLast != null)
            last = newLast;
    }
    i++;
}

return tree;
}

```

311

```

public TreeStruct GenerateTreeOnExisting(TreeStruct
    existingTree, string value, Transform tortoise)
{
    var pushdown = new Stack<tortoiseData>();

    // 1) Limpiar el TreeStruct existente (borrar hijos)
    existingTree.subStructs.Clear();
    for (int k = existingTree.transform.childCount - 1; k
        >= 0; k--)
    {

```

```
321         Transform child = existingTree.transform.GetChild(k);
        UnityEngine.Object.Destroy(child.gameObject);
    }
    // Actualizar la cadena LSys
    existingTree.LSys = value;
    Debug.Log("LSys: " + value);

    // 2) Destruir el SubStruct del root, si existe.
    var oldLast = existingTree.GetComponent<SubStruct>();
    if (oldLast != null)
331         UnityEngine.Object.Destroy(oldLast);

    // 3) Crear un nuevo SubStruct "last" en el mismo
    GameObject.
    var last = existingTree.gameObject.AddComponent<
        SubStruct>();
    last.parent = existingTree;
    // 4) Procesar la cadena LSys
    int pos = 0;
    while (pos < value.Length)
    {
341         char v = value[pos];
        var actions = generationAction.Where(a => a.
            variable == v).ToList();

        if (Utils.IsParameterized(value, pos))
        {
            // Extrae el bloque de par metros
            string subStr = value.Substring(pos + 1);
            var (param, end) = Utils.
                ExtractFromParentheses(subStr);
            var exps = param.Split(';').Select(float.Parse
            ).ToList();
        }
    }
}
```

```
351         if (actions.Count > 0 && actions[0].action !=  
            null)  
        {  
            var newLast = actions[0].action(pos,  
                tortoise, pushdown, existingTree, last,  
                exps);  
            if (newLast != null)  
                last = newLast;  
        }  
  
        // Avanzar pos: 1 (letra) + (end+1) para el  
        // bloque entre paréntesis.  
        pos += (end + 2);  
  
        // Procesar sufijos de ID consecutivos, si  
        // existen.  
361        while (pos < value.Length && value[pos] ==  
            '#')  
        {  
            pos++; // Saltar el '#'  
            int idStart = pos;  
            while (pos < value.Length && char.IsDigit(  
                value[pos]))  
            {  
                pos++;  
            }  
            int idLength = pos - idStart;  
            if (idLength > 0)  
            {  
371                int uniqueId = int.Parse(value.  
                    Substring(idStart, idLength));  
                last.uniqueId = uniqueId;  
            }  
        }  
    }  
    else
```

```

        {
            if (actions.Count > 0 && actions[0].action !=
                null)
            {
                var newLast = actions[0].action(pos,
                    tortoise, pushdown, existingTree, last,
                    new List<float>());
                if (newLast != null)
                    last = newLast;
            }
            pos++;
        }
    }
    return existingTree;
}
}

391 /// <summary>
    /// Clase para almacenar la posici n, rotaci n y el "last"
    /// transform cuando se hace push/pop.
    /// </summary>
    public class tortoiseData
    {
        public Vector3 pos;
        public Vector3 dir;
        public Transform last;
    }
}

```

Listing E.1: Código completo de Generator.cs

```
1 using System;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using System.Text.RegularExpressions;
using Utils = ParametrizedUtilities;

public static class Deriver
{
    private static int nextId = 0;
11 public static int GenerateId()
    {
        return nextId++;
    }
    /// <summary>
    /// Resetea el contador de ID, til para recalcular IDs
    /// tras un corte.
    /// </summary>
    public static void ResetIdCounter()
    {
21         nextId = 0;
    }

    public static string RecalculateUniqueIds(string lSys)
    {
        ResetIdCounter();
        // Este regex busca smbolos parametrizados que
        // empiecen con F, A, B o G,
        // que tengan contenido entre parntesis y,
        // opcionalmente, un sufijo "#nmero".
        Regex regex = new Regex(@"([FABG])\[([^\]]*)\](#\d+)?");
        string result = regex.Replace(lSys, match =>
        {
31             // Toma el simbolo sin el sufijo, si es que existe
            .
```

```

        string baseSymbol = match.Value;
        int hashIndex = baseSymbol.IndexOf("#");
        if (hashIndex >= 0)
        {
            baseSymbol = baseSymbol.Substring(0, hashIndex);
        }
        int newId = GenerateId();
        return baseSymbol + "#" + newId;
    });
    return result;
}

public static List<string> Derive(GrammarTree grammar,
    string axiom, int amount)
{
    //ResetIdCounter();
    List<string> generations = new List<string>();
    string current = axiom;
    generations.Add(current);
    Debug.Log("Generacin 0: " + current);

    for (int i = 0; i < amount; i++)
    {
        string outputGen = "";
        Debug.Log("----- Derivando generacin " + (i + 1)
            + " -----");

        for (int j = 0; j < current.Length; j++)
        {
            char symbol = current[j];
            Debug.Log("Procesando simbolo : ' " + symbol + "
                ' en posicin " + j);

            if (Utils.IsParameterized(current, j))
            {
                string substring = current.Substring(j +

```

71

81

```

1);
var (parameters, endIndex) = Utils.
    ExtractFromParentheses(substring);
Debug.Log(" Símbolo parametrizado detectado
    : '" + symbol + "' con parámetros: " +
    parameters);

var exps = parameters.Split(';');
var rules = grammar.GetRules(symbol);
Debug.Log("Reglas encontradas para '" +
    symbol + "': " + rules.Count);

// Evaluar condición de cada regla
var validRules = rules.Where(rule => {
    bool conditionOk = rule.CheckCondition
        (exps, grammar.generalVariables);
    Debug.Log(" Regla: '" + rule.input +
        "' -> '" + rule.output + "'",
        "Condición: '" + rule.condition + "'
        evaluada como: " + conditionOk);
    return conditionOk;
}).ToList();

if (validRules.Count <= 0)
{
    Debug.Log("No se encontró regla
        válida para el símbolo '" + symbol
        + "' en posición " + j + ". Se
        conserva el símbolo original.");
    outputGen += current.Substring(j, 1 +
        (endIndex + 1));
}
else
{
    // Selección de regla: estocástica o
    determinista.

```

91

101

```

        var selectedRule = grammar.
            isStochastic ? validRules.
                RandomRullete(r => r.weight) :
                validRules[0];
        Debug.Log("Regla seleccionada para '"
            + symbol + "': '" + selectedRule.
                input + "' -> '" + selectedRule.
                output + "'");

        string result = selectedRule.
            CalcOutput(exps, grammar.
                generalVariables);
        Debug.Log("Resultado de la evaluacin:
            " + result);
        outputGen += result;
    }
    // Avanza el dice para saltar la parte
    // de los parametros procesados.
    j += (endIndex + 1);
}
else
{
    var rules = grammar.GetRules(symbol);
    if (rules.Count <= 0)
    {
        Debug.Log("No hay reglas para el
            simbolo no parametrizado '" +
                symbol + "'. Se aade tal cual.");
        outputGen += symbol;
    }
    else
    {
        Debug.Log("Smbolo no parametrizado '"
            + symbol + "' con regla aplicada:
                '" + rules[0].output + "'");
        outputGen += rules[0].output;
    }
}

```

```

    }
    }
}

111    Debug.Log("Generación " + (i + 1) + ": " +
        outputGen);
    generations.Add(outputGen);
    current = outputGen;
}
return generations;
}
}

```

Listing E.2: Código completo de Deriver.cs

E.4.2. Reglas Paramétricas del Nivel 1

Símbolos Utilizados

- **T(w;L)**: Tronco, con grosor w y longitud L .
- **P(w;L;a)**: Rama primaria, con grosor w , longitud L y nivel de bifurcación a .
- **S(w;L;a)**: Rama secundaria, derivada desde P, con los mismos parámetros.
- **F(w;L)**: Segmento visible del árbol, renderizado con el prefab `Palo_MC_1`.
- **[]**: Inicio y cierre de una bifurcación.
- **+, -, ^, &**: Rotaciones en diferentes ejes espaciales.

Símbolo	Regla de Producción
T1	$T(w; L) \rightarrow F(w; 0, 3) T(w + \text{trunkIncrement}; L + \text{trunkLengthIncrement})$ <i>Condición:</i> $w < \text{maxTrunkWidth}$
T2	$T(w; L) \rightarrow F(w; 0, 3) [P(w * \text{branchTaper}, \text{branchStartL}, 0)]$ $[+(\text{branchAngle}) P(\dots)] [-(\text{branchAngle}) P(\dots)]$ $[^(\text{branchAngle}) P(\dots)] [\&(\text{branchAngle}) P(\dots)]$ <i>Condición:</i> $w \geq \text{maxTrunkWidth}$
P1	$P(w; L; a) \rightarrow F(w; \text{branchGrowth}) P(w * \text{branchTaper}; L + \text{branchGrowth}; a)$ <i>Condición:</i> $L < \text{maxBranchLength}$
P2	$P(w; L; a) \rightarrow F(w; \text{branchGrowth})$ $[S(w * \text{branchTaper}; \text{branchStartL} * \text{secScale}; 0)]$ $[+(\text{branchAngle}) S(\dots)] [-(\text{branchAngle}) S(\dots)]$ $[^(\text{branchAngle}) S(\dots)] [\&(\text{branchAngle}) S(\dots)]$ $P(w * \text{branchTaper}; L + \text{branchGrowth}; a + 1)$ <i>Condición:</i> $L \geq \text{maxBranchLength} \wedge a < 2$
P3	$P(w; L; a) \rightarrow F(w; \text{branchGrowth}) P(w; \text{branchStartL}; 0)$ <i>Condición:</i> $a \geq 2$
S1	$S(w; L; a) \rightarrow F(w; \text{branchGrowth}) S(w * \text{branchTaper}; L + \text{branchGrowth}; a)$ <i>Condición:</i> $L < \text{maxBranchLength}$
S2	$S(w; L; a) \rightarrow F(w; \text{branchGrowth})$ $[+(\text{branchAngle}) S(\dots)] [-(\text{branchAngle}) S(\dots)]$ $[^(\text{branchAngle}) S(\dots)] [\&(\text{branchAngle}) S(\dots)]$ $S(w * \text{branchTaper}; L * \text{secScale}; a + 1)$ <i>Condición:</i> $L \geq \text{maxBranchLength} \wedge a < 2$
S3	$S(w; L; a) \rightarrow F(w; \text{branchGrowth})$ <i>Condición:</i> $a \geq 2$

Tabla E.1: Reglas paramétricas utilizadas en la generación procedural del patrón KGB.

E.4.3. Parámetros Globales

Variable	Valor	Descripción
maxTrunkWidth	0.2	Ancho máximo del tronco antes de ramificar.
trunkIncrement	0.04	Incremento de grosor por iteración.
trunkLengthIncrement	0.5	Incremento de longitud por iteración.
branchStartW	0.04	Grosor inicial de ramas primarias.
branchStartL	0.4	Longitud inicial de ramas primarias.
branchTaper	0.7	Reducción porcentual de grosor por nivel.
branchGrowth	0.6	Longitud añadida por cada crecimiento.
maxBranchLength	0.6	Longitud máxima antes de bifurcar.
branchAngle	25	Ángulo de bifurcación en grados.
secScale	1.5	Escala de longitud para ramas secundarias.

Tabla E.2: Parámetros globales utilizados en la generación procedural.

Este sistema permite generar árboles visual y estructuralmente coherentes con el patrón KGB, con niveles de complejidad adaptables según iteraciones y condiciones lógicas.

E.4.4. Reglas Paramétricas del Nivel 2

Símbolo	Regla de Producción
T1	$T(w; L) \rightarrow F(w; 0,35) T(w + \text{trunkIncrement}; L + \text{trunkLengthIncrement})$ <i>Condición:</i> $w < \text{maxTrunkWidth}$
T2	$T(w; L) \rightarrow F(w; 0,35)$ $[+(\text{branchAngle}) P(w * \text{branchTaper}; \text{branchStartL}; 0)]$ $[-(\text{branchAngle}) P(\dots)]$ $[\wedge(\text{branchAngle}) P(\dots)]$ $[\&(\text{branchAngle}) P(\dots)]$ <i>Condición:</i> $w \geq \text{maxTrunkWidth}$
P1	$P(w; L; a) \rightarrow F(w; \text{branchGrowth}) P(w * \text{branchTaper}; L + \text{branchGrowth}; a)$ <i>Condición:</i> $L < \text{maxBranchLength}$
P2	$P(w; L; a) \rightarrow F(w; \text{branchGrowth})$ $[+(\text{branchAngle}) S(w * \text{branchTaper}; \text{branchStartL} * \text{secScale}; 0) S(w * \text{branchTaper}^2; \text{branchStartL} * \text{secScale}; 0)]$ Repetido con signos $[-]$, $[\wedge]$, $[\&]$ $P(w * \text{branchTaper}; L + \text{branchGrowth} * \text{secScale}; a + 1)$ <i>Condición:</i> $L \geq \text{maxBranchLength} \wedge a < 2$
P3	$P(w; L; a) \rightarrow F(w; \text{branchGrowth}) P(w; \text{branchStartL}; 0)$ <i>Condición:</i> $a \geq 2$
S1	$S(w; L; a) \rightarrow F(w; \text{branchGrowth}) S(w * \text{branchTaper}; L + \text{branchGrowth}; a)$ <i>Condición:</i> $L < \text{maxBranchLength}$
S2	$S(w; L; a) \rightarrow F(w; \text{branchGrowth})$ $[+(\text{branchAngle}) S(\dots)]$ $[+(\text{branchAngle}) S(\dots)]$ $[-(\text{branchAngle}) S(\dots)]$ $[-(\text{branchAngle}) S(\dots)]$ $[\wedge(\text{branchAngle}) S(\dots)]$ $[\&(\text{branchAngle}) S(\dots)]$ $S(w * \text{branchTaper}; L * \text{secScale}; a + 1)$ <i>Condición:</i> $L \geq \text{maxBranchLength} \wedge a < 2$
S3	$S(w; L; a) \rightarrow F(w; \text{branchGrowth})$ <i>Condición:</i> $a \geq 2$

Tabla E.3: Reglas paramétricas utilizadas en el Nivel 2 del sistema KGB.

Parámetros Nivel 2

Variable	Valor	Descripción
maxTrunkWidth	0.35	Grosor máximo antes de ramificar
trunkIncrement	0.04	Incremento del tronco por iteración
trunkLengthIncrement	0.5	Aumento de longitud por segmento
branchStartW	0.05	Grosor inicial de las ramas primarias
branchStartL	0.4	Longitud inicial de ramas
branchTaper	0.7	Reducción por bifurcación
branchGrowth	0.5	Crecimiento por iteración
maxBranchLength	0.7	Longitud máxima antes de bifurcar
branchAngle	20	Ángulo de bifurcación
secScale	1.5	Escala para ramas secundarias

Tabla E.4: Parámetros utilizados en el Nivel 2 del sistema procedural.

E.4.5. Reglas Paramétricas del Nivel 3

Símbolo	Regla de Producción
T1	$T(w; L) \rightarrow F(w; 0,45) T(w + \text{trunkIncrement}; L + \text{trunkLengthIncrement})$ <i>Condición:</i> $w < \text{maxTrunkWidth}$
T2	$T(w; L) \rightarrow F(w; 0,45)$ $[+(\text{branchAngle}) P(w * \text{branchTaper}; \text{branchStartL}; 0)]$ $[-(\text{branchAngle}) P(\dots)]$ $[(\text{branchAngle}) P(\dots)]$ $[\&(\text{branchAngle}) P(\dots)]$ <i>Condición:</i> $w \geq \text{maxTrunkWidth}$
P1	$P(w; L; a) \rightarrow F(w; \text{branchGrowth}) P(w * \text{branchTaper}; L + \text{branchGrowth}; a)$ <i>Condición:</i> $L < \text{maxBranchLength}$
P2	$P(w; L; a) \rightarrow F(w; \text{branchGrowth})$ $[+(\text{branchAngle}) S(w * \text{branchTaper}; \text{branchStartL}; 0)]$ $[-(\text{branchAngle}) S(\dots)] [(\text{branchAngle}) S(\dots)] [\&(\text{branchAngle}) S(\dots)]$ $P(w * \text{branchTaper}; L + \text{branchGrowth}; a + 1)$ <i>Condición:</i> $L \geq \text{maxBranchLength} \wedge a < 2$
P3	$P(w; L; a) \rightarrow F(w; \text{branchGrowth}) P(w; \text{branchStartL}; 0)$ <i>Condición:</i> $a \geq 2$
S1	$S(w; L; a) \rightarrow F(w; \text{branchGrowth}) S(w * \text{branchTaper}; L + \text{branchGrowth}; a)$ <i>Condición:</i> $L < \text{maxBranchLength}$
S2	$S(w; L; a) \rightarrow F(w; \text{branchGrowth})$ $[+(\text{branchAngle}) S(\dots)] [-(\text{branchAngle}) S(\dots)]$ $S(w * \text{branchTaper}; L * \text{secScale}; a + 1)$ <i>Condición:</i> $L \geq \text{maxBranchLength} \wedge a < 2$
S3	$S(w; L; a) \rightarrow F(w; \text{branchGrowth})$ <i>Condición:</i> $a \geq 2$

Tabla E.5: Reglas paramétricas utilizadas en el Nivel 3 del sistema KGB.

Parámetros Nivel 3

Variable	Valor	Descripción
maxTrunkWidth	0.45	Grosor máximo antes de ramificar
trunkIncrement	0.04	Incremento del tronco por iteración
trunkLengthIncrement	0.5	Aumento de longitud por segmento
branchStartW	0.1	Grosor inicial de las ramas primarias
branchStartL	0.4	Longitud inicial de ramas
branchTaper	0.75	Reducción por bifurcación
branchGrowth	0.5	Crecimiento por iteración
maxBranchLength	0.7	Longitud máxima antes de bifurcar
branchAngle	25	Ángulo de bifurcación
secScale	1.5	Escala para ramas secundarias

Tabla E.6: Parámetros utilizados en el Nivel 3 del sistema procedural.

E.5. Módulo de Interfaz de Usuario e Inmersión

Objetivo: Facilitar la interacción fluida del usuario con el entorno de realidad virtual, mediante una interfaz intuitiva, clara y adecuada al contexto de capacitación.

Funciones principales:

- Mostrar información relevante durante la simulación (estado, instrucciones).
- Permitir al usuario iniciar, pausar o finalizar la experiencia.
- Integrar elementos de HUD, menús y ayudas contextuales.

Entradas:

- Comandos del usuario (input VR).
- Estado actual del sistema (inicio, poda, evaluación).

Salidas:

- Visualización de elementos en pantalla (HUD, textos, íconos).

- Activación de funciones como comenzar o terminar la práctica.

Interacción con otros módulos: Este módulo se comunica con todos los demás, sirviendo como puente entre el usuario y las funcionalidades del sistema.

F. Cuestionario SUS y Métricas

F.1. Datos del Cuestionario SUS Ronda 1

Los datos crudos de la primera ronda del SUS pueden consultarse en:

[https://docs.google.com/spreadsheets/d/
1vKT1pjecv0rwid-3kbDUTM4bS1tox9_V/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1vKT1pjecv0rwid-3kbDUTM4bS1tox9_V/edit?usp=sharing)

F.2. Datos del Cuestionario SUS Ronda 2

Los datos crudos de la segunda ronda del SUS pueden consultarse en:

[https://docs.google.com/spreadsheets/d/
1RrCKL77t6eh6GIut4ir6fyUnYh9FJ0XB/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1RrCKL77t6eh6GIut4ir6fyUnYh9FJ0XB/edit?usp=sharing)

F.3. Métricas Internas de Unity

Las métricas de desempeño extraídas de Unity están disponibles en:

[https://drive.google.com/file/d/1qKv_OFbaeUac2qhGfGEnem-H1I1QI3eG/
view?usp=sharing](https://drive.google.com/file/d/1qKv_OFbaeUac2qhGfGEnem-H1I1QI3eG/view?usp=sharing)

G. Consentimiento Informado y Cuestionario SUS con Pregunta Cualitativa

G.1. Consentimiento Informado

El Consentimiento informado hecho a los participantes del estudio están disponibles en:

https://drive.google.com/file/d/1C-vTjenquU0CGmKpkTbQ73ZLJr2H7Nm_/view?usp=sharing

G.2. Cuestionario SUS con Pregunta Cualitativa

El cuestionario SUS hecho a los participantes del estudio están disponibles en:

https://drive.google.com/file/d/1T9DWy-WilyhR4ZaP11J8Jvftof4XoH_F/view?usp=sharing



Figura H.2: Herramienta de poda en acción.

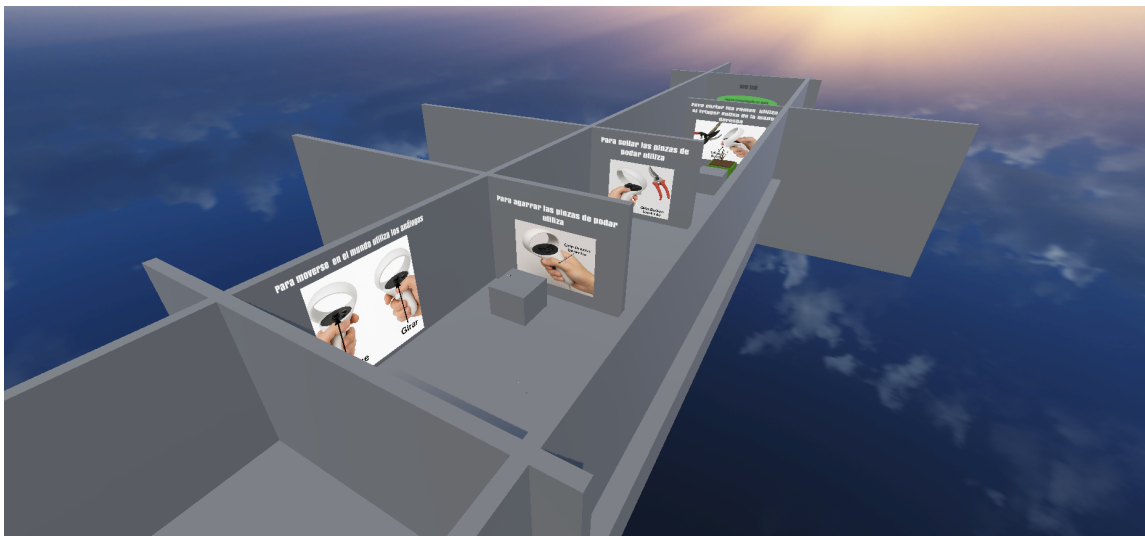


Figura H.3: Menú e interfaz definitiva del entorno en el Tutorial.



Figura H.4: Vista en VR del Menú e interfaz.

Imágenes representativas de las distintas vistas y funcionalidades del prototipo final.

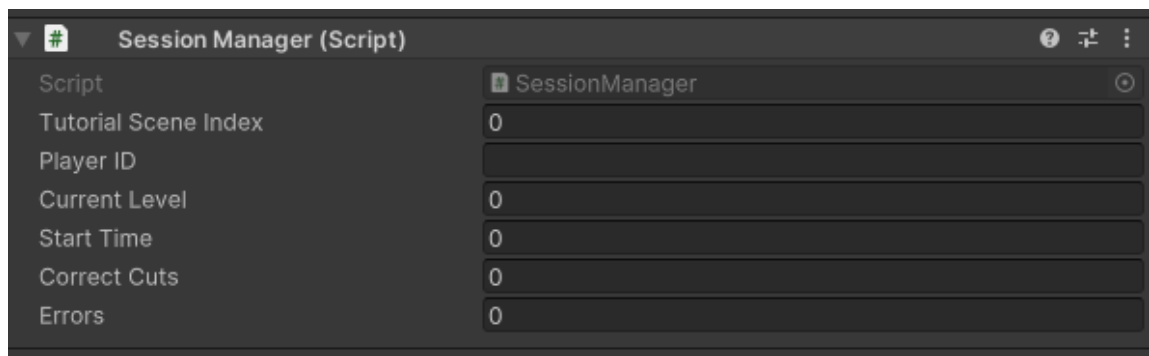


Figura H.5: Interfaz del *Script SessionManager* en Unity, mostrando las variables principales de la sesión: índice de escena del tutorial, ID del jugador, nivel actual, tiempo de inicio, cortes correctos y errores.

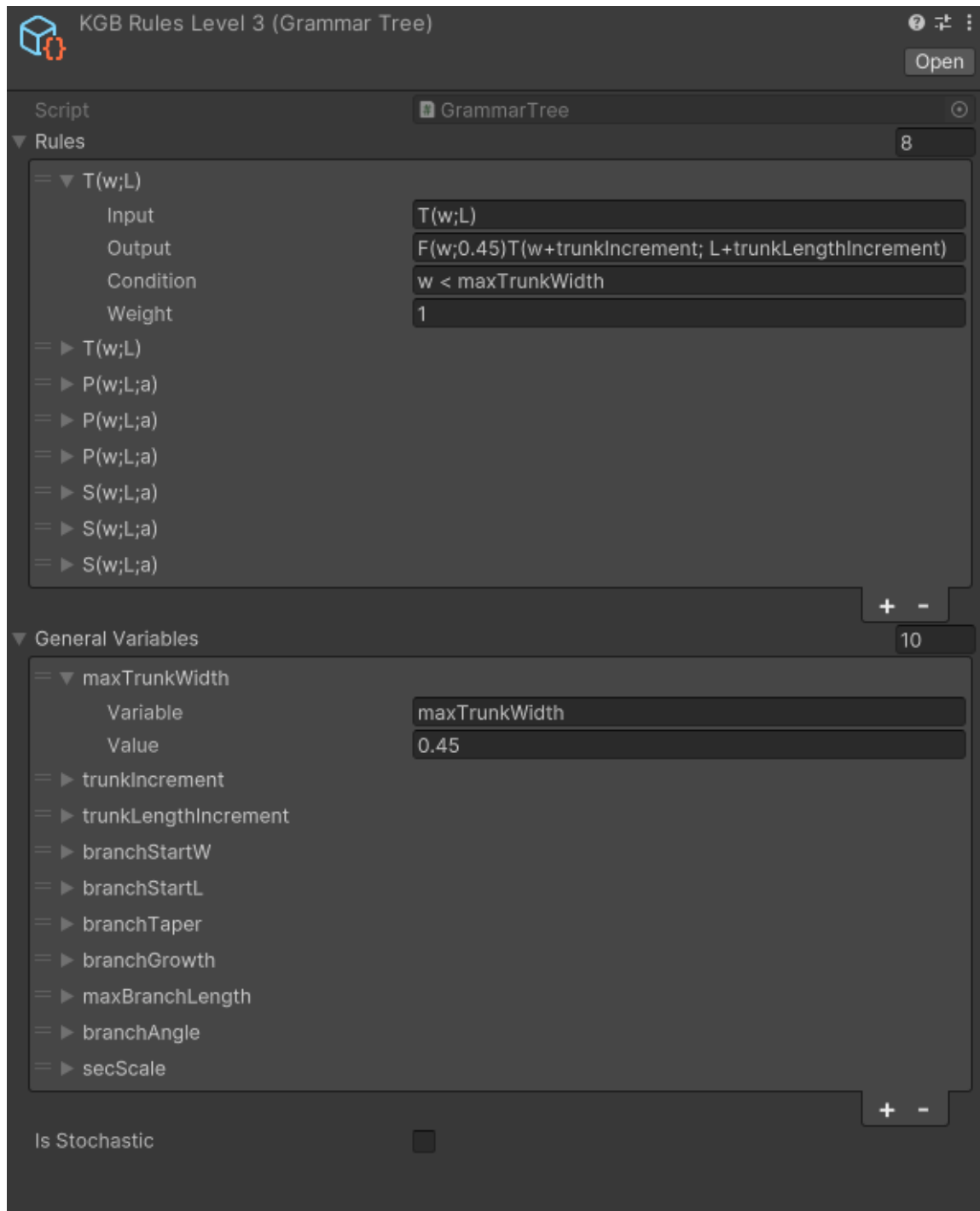


Figura H.6: Interfaz del módulo **GrammarTree** en Unity, mostrando las reglas L-System paramétricas y las variables generales definidas para la generación de árboles KGB.

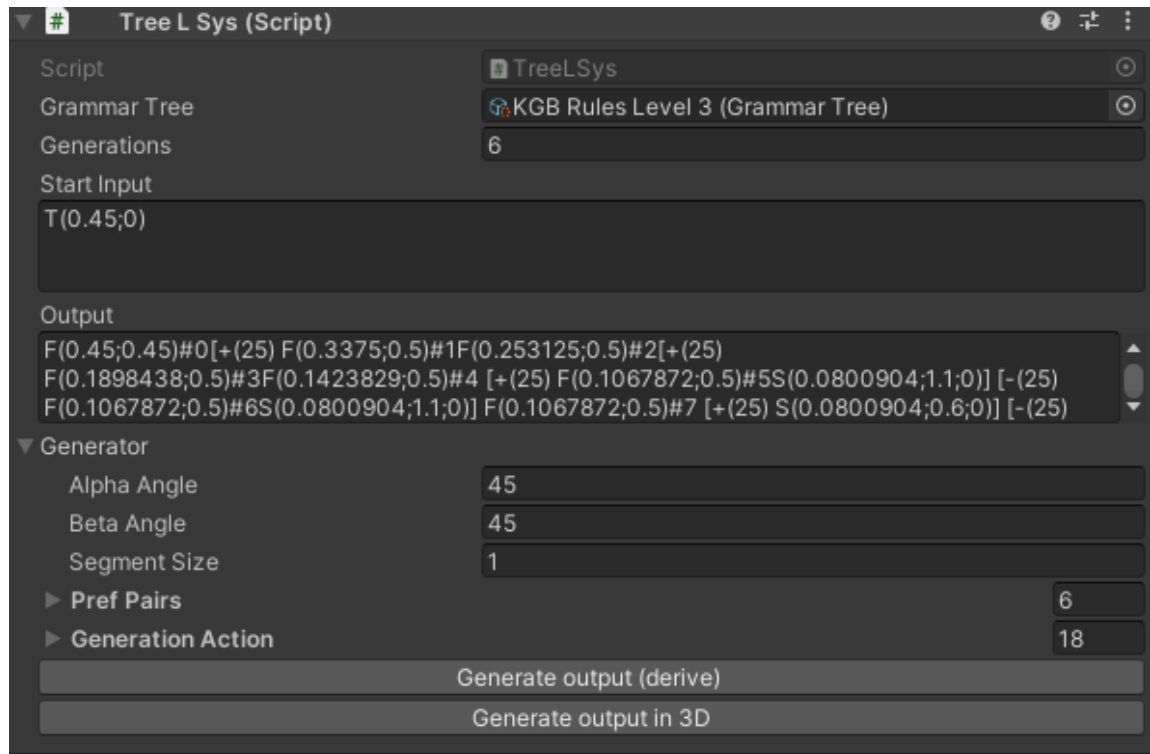


Figura H.7: Interfaz del script `TreeLSys` en Unity, encargado de procesar las reglas del `GrammarTree` y generar la estructura del árbol en 3D.

E.6. Build Final de la Aplicación y Video de la Experiencia

Con el propósito de complementar la descripción técnica y práctica de **PodaRV**, se incluye el enlace de descarga de la **build final en formato APK** para su ejecución en dispositivos Oculus Quest 2. Esta versión corresponde a la implementación completa, integrando los módulos de poda interactiva, retroalimentación visual y sonora, interfaz de usuario, generación procedural de árboles y sistema de métricas de desempeño.

Asimismo, se proporciona un **video de demostración del gameplay** grabado directamente desde el visor Oculus Quest. Este material audiovisual muestra la experiencia inmersiva tal como la percibe el usuario final, evidenciando la interacción con las herramientas de poda, la navegación por el entorno virtual y la retroalimentación en tiempo real.

- **Enlace de descarga de la build (APK):** <https://drive.google.com/file/d/1z1v51v0H0vGL-0w5A6BjTyfiukRxJrcy/view?usp=sharing>
- **Enlace de gameplay de la experiencia:** <https://drive.google.com/file/d/1x9yaAogX0R22JkbbkLE1Dpaf6hoP9eLfq/view?usp=sharing>

Ambos recursos tienen como objetivo respaldar la comprensión práctica del prototipo desarrollado, permitiendo a revisores y futuros investigadores ejecutar y observar el sistema en funcionamiento.