



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

**Desarrollo en Unity de un Huerto Virtual
Fotorrealista y un Robot con Visión Artificial
para el Entrenamiento y Validación de Algoritmos
en Robótica Agrícola**

DANIEL MATÍAS IGNACIO ARIAS SILVA

Profesor Guía: BENJAMIN INGRAM

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

Talca – Chile
Diciembre, 2025



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

**Desarrollo en Unity de un Huerto Virtual
Fotorrealista y un Robot con Visión Artificial
para el Entrenamiento y Validación de Algoritmos
en Robótica Agrícola**

DANIEL MATÍAS IGNACIO ARIAS SILVA

Profesor Guía: BENJAMIN INGRAM

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

El presente documento fue calificado con nota: _____

Talca – Chile

Diciembre, 2025

AGRADECIMIENTOS

Agradezco a mi profesor guía, Ben, por su apoyo durante el desarrollo de esta memoria, por la confianza depositada en mi trabajo, por la oportunidad y por sus observaciones, que fueron clave para llevar a cabo este proceso.

También agradezco a mis amigos por acompañarme en este camino. En especial, a mi mejor amigo y quienes estuvieron conmigo en los momentos más difíciles, a los que me empujaron a seguir y me levantaron el ánimo, ya fuera conversando, carreteando o jugando.

Finalmente, agradezco a mi familia por su respaldo, apoyo y confianza incondicional en mí. Y, por sobre todo, a mi mamá, que si no fuera por ella no existiría esta tesis ni mi formación. Gracias por aguantarme, por apoyarme, por sostenerme, por creer en mí incluso cuando ni yo lo hice, por cada sacrificio que hiciste, haces y seguirás haciendo por mí, por todo el amor incondicional que me das para que hoy pueda llegar hasta aquí. Este logro es más tuyo que mío.

TABLA DE CONTENIDOS

	página
Agradecimientos	I
Tabla de Contenidos	II
Índice de Figuras	VI
Índice de Tablas	VIII
Resumen	IX
1. Introducción	1
1.1. Problema	2
1.2. Objetivos	3
1.2.1. Objetivos Específicos	3
1.3. Pregunta de investigación	4
1.4. Hipótesis	4
2. Revisión Bibliográfica	5
2.1. Visión por Computador aplicada a la Agricultura	5
2.2. Aprendizaje Automático y Detección de Objetos	6
2.3. Desafíos visuales en entornos agrícolas	7
2.3.1. Iluminación cambiante	7
2.3.2. Oclusión	7
2.3.3. Variabilidad morfológica de las frutas	8
2.4. Generación de datos sintéticos y su uso	8
2.5. Simulación fotorrealista en entornos agrícolas	9
2.6. Visión Estereoscópica	9
2.7. Evaluación y Métricas de Detección Agrícola	10
3. Metodología	11
3.1. Flujo de trabajo	11
3.2. Diseño del Huerto Virtual	11

3.2.1.	Elementos del Huerto Virtual	12
3.2.2.	Parametrización y diversidad controlada	13
3.3.	Diseño del Robot Virtual	13
3.3.1.	Patrón de recorrido y captura	14
3.3.2.	Configuración de cámaras	14
3.4.	Generación de Datos Sintéticos	15
3.4.1.	Etiquetado automático con Unity Perception	15
3.4.2.	Estructuración del dataset y conversión de etiquetas	16
3.5.	Entrenamiento y Validación Preliminar de Modelos	16
3.5.1.	Validación preliminar y evaluación de generalización	16
3.5.2.	Parámetros de entrenamiento y configuración experimental	17
3.5.3.	Diseño experimental aplicado	18
3.5.4.	Diagnóstico del dominio sintético y variantes E1a–E1d	18
3.5.5.	Evaluación por tamaños y Rango de Validez Operacional (RVO)	20
3.6.	Hardware y software utilizados	21
3.6.1.	Rendimiento computacional	21
3.7.	Capacidades de experimentación de la plataforma	22
3.7.1.	Parámetros ambientales	22
3.7.2.	Parámetros del cultivo	23
3.7.3.	Parámetros del sensor y captura	23
3.7.4.	Implicancias experimentales	23
4.	Resultados y análisis	24
4.1.	Resultados globales	24
4.2.	Análisis del caso E1 y variantes E1a–E1d	25
4.3.	Resultados por tamaños	26
4.4.	RVO: métricas operacionales vs globales	28
4.5.	Análisis cualitativo	29
4.5.1.	E1	30
4.5.2.	E2	30
4.5.3.	E3	31
4.5.4.	E4	32
4.5.5.	Conexión con RVO: Lectura operacional	33

5. Conclusión	34
5.1. Conclusiones principales	34
5.2. La plataforma como contribución reutilizable	35
5.3. Limitaciones de trabajo	37
5.4. Trabajos futuros	38
5.5. Conclusión final	39
Bibliografía	41
A: Diagramas	46
A.1. Diagrama de flujo de trabajo	46
A.2. Diagrama de arquitectura del Sistema	47
B: Diseño del Sistema	48
B.1. Elementos del Huerto Virtual	48
B.1.1. Terreno	48
B.1.2. Árboles y Cerezas	50
B.1.3. Iluminación Dinámica	53
B.1.4. Condiciones Climáticas	54
B.1.5. Control de Condiciones	61
B.1.6. Diseño Modular y extensible	65
B.2. Diseño del Robot Virtual	68
B.2.1. Robot	68
B.2.2. Configuraciones del Robot	71
B.2.3. Unity Perception	73
C: Rendimiento computacional	76
C.1. Simulación	76
C.2. Entrenamiento	78
D: Recursos externos	79
D.1. Disponibilidad de código, notebooks y resultados	79
D.1.1. Repositorio del proyecto Unity	79
D.1.2. Notebooks de experimentación (Google Colab)	79
D.1.3. Resultados y datos generados	79

D.2. Paquete de materiales y entorno (Unity Asset Store)	79
D.3. Modelo 3D del robot (TurboSquid)	80
D.4. Atribuciones de íconos	80

ÍNDICE DE FIGURAS

	página
4.1. Comparación cualitativa del experimento E1 (Sintético $\rightarrow$ Sintético) en el conjunto de prueba sintético.	30
4.2. Comparación cualitativa del experimento E2 (Sintético $\rightarrow$ Real) en CherryCO test.	31
4.3. Comparación cualitativa del experimento E3 (Sintético + Real, fine-tuning) sobre CherryCO test.	32
4.4. Comparación cualitativa del experimento E4 (Nocturno con linterna $\rightarrow$ Real diurno) sobre CherryCO test.	33
 A.1. Diagrama de flujo de trabajo.	 46
A.2. Diagrama de arquitectura del sistema.	47
 B.1. Distribución del terreno.	 48
B.2. Vista en sitio del terreno.	49
B.3. Árbol en conducción espaldera simple.	50
B.4. Árbol en conducción espaldera complejo.	51
B.5. Modelo de las Cerezas.	52
B.6. Horario am.	53
B.7. Horario pm.	53
B.8. Iluminacion artificial (linternas del robot).	54
B.9. Clima Despejado.	54
B.10.Clima Nuboso.	55
B.11.Clima cielo nublado.	55
B.12.Clima tormentoso.	56
B.13.Nubes personalizadas.	56
B.14.Configuración de intensidad de las Estrellas.	57
B.15.Condición no nominal del entorno bajo una configuración extrema de iluminación: se modificaron deliberadamente la temperatura de color e intensidades del Sol.	57
B.16.Condición no nominal del entorno bajo una configuración extrema de iluminación: se modificaron deliberadamente la temperatura de color e intensidades de la Luna.	58

B.17.Niebla moderada.	59
B.18.Niebla suave.	59
B.19.Niebla fuerte.	60
B.20.Controles de Tiempo.	61
B.21.Configuración de Nubes.	62
B.22.Configuración de Astros.	63
B.23.Configuración de la Niebla.	64
B.24.Configuración de distribución y generación de árboles.	65
B.25.Configuración del Spawn de cerezas.	66
B.26.Continuación del Spawner de cerezas.	67
B.27.Modelo del Robot.	68
B.28.Controles del Robot.	69
B.29.Previsualización de cámaras del Robot.	70
B.30.Configuraciones del Robot.	71
B.31.Continuación de configuraciones del Robot.	72
B.32.Configuración de la cámara.	73
B.33.Configuración de la etiqueta Cherry.	74
B.34.Asignación de etiqueta al modelo de la Cereza.	75

ÍNDICE DE TABLAS

	página
3.1. Configuración base de entrenamiento utilizada en la mayoría de los experimentos.	17
3.2. Resumen de hiperparámetros por experimento.	18
3.3. Variantes del experimento E1 y sus hiperparámetros.	20
4.1. Resultados globales de detección para los experimentos E0–E4.	25
4.2. Caso de estudio del dominio sintético: resultados globales para E1 y variantes controladas (E1a–E1d).	26
4.3. Distribución de tamaños de frutos en el conjunto de test sintético filtrado, usando umbrales basados en el área de la bounding box.	27
4.4. Recall por tamaño para E1 y sus variantes, evaluado en el test sintético filtrado.	27
4.5. Recall desagregado por tamaño para E0, E3 y E4.	27
4.6. Recall global versus Recall operacional en el conjunto real (CherryCO test). El RVO se define como objetos con bounding boxes de área $\geq 32 \times 32$ px (MEDIUM+LARGE).	28
4.7. Recall global versus Recall operacional en el conjunto sintético. El RVO32 se define como objetos con bounding boxes de área $\geq 32 \times 32$ px (MEDIUM+LARGE).	29
C.1. Medición observacional de rendimiento usando <i>Stats</i> de Unity en escenarios representativos.	76
C.2. Tiempos de entrenamiento por experimento (según logs de Ultralytics).	78

RESUMEN

La escasez y el alto costo de la mano de obra en el sector agrícola han impulsado el desarrollo de tecnologías que faciliten la automatización de labores como la cosecha de frutas. Uno de los mayores desafíos para lograrlo es detectar y localizar con precisión las frutas en condiciones reales, donde factores como la iluminación cambiante, la oclusión de los frutos por el follaje y la variabilidad natural entre cada fruta complican el desempeño de los algoritmos de visión computacional.

En Chile, algunos estudios han mostrado que las redes neuronales profundas pueden ser útiles para esta tarea, pero también han dejado en evidencia ciertas limitaciones técnicas, especialmente cuando se trabaja solo con datos capturados en campo. Esto refuerza la necesidad de contar con entornos controlados que permitan entrenar, probar y mejorar estos modelos en condiciones variadas pero reproducibles.

Dentro de este marco, este proyecto propuso crear una plataforma modular compuesta por un huerto virtual fotorrealista y un robot simulado con visión artificial, ambos desarrollados en Unity. El huerto recrea visualmente un entorno con árboles de cerezo conducidos en espaldera, incluyendo condiciones como cambios de luz, hora, clima, oclusión por ramas y frutos con distintos tamaños. El robot, por su parte, recorre el entorno capturando imágenes RGB que son usadas para generar conjuntos de datos sintéticos etiquetados. Estos datos se utilizaron para entrenar algoritmos de aprendizaje automático para la detección y localización de frutas.

Esta plataforma funciona como un espacio de prueba virtual donde se puede evaluar el desempeño de los modelos en distintas condiciones visuales. Si bien esta memoria se centrará en validar el sistema dentro del entorno simulado utilizando árboles de cerezo como caso base, los desafíos que aborda, como la detección en condiciones complejas de iluminación, oclusión y variabilidad morfológica son comunes a una amplia variedad de cultivos frutales. Por ello, el enfoque planteado tiene potencial de ser adaptado y escalado a otros contextos agrícolas.

Este trabajo busca aportar una herramienta práctica, accesible y reutilizable para el desarrollo de soluciones en robótica agrícola. Gracias a la simulación, se pueden evitar muchas de las limitaciones de las pruebas en campo, acelerar el desarrollo de modelos más robustos y facilitar su aplicación en escenarios reales.

1. Introducción

Durante los últimos años el aumento del costo de las labores agrícolas, la escasez de mano de obra y trabajadores, la necesidad de producir cada día mas y la oportunidad de automatizar el área agrícola ha convertido el desarrollo de esta área en un tema de relevancia mundial [1]. Es por esto que la robótica aplicada en lo agrícola nace como una herramienta futurista pero prometedora para tareas que van desde la siembra, el monitoreo y cuidado de cultivos y especialmente la cosecha automatizada de productos.

Uno de los grandes obstáculos técnicos para la adecuada implementación de sistemas de cosecha automatizada es el cómo percibir adecuadamente el entorno, y en particular, la localización y detección precisa de las frutas en los árboles sigue siendo una tarea compleja, principalmente debido a la variabilidad de condiciones lumínicas, la oclusión que recae sobre la fruta por las hojas, las ramas y la diversidad morfológica natural de los frutos; todas estas complejidades visibles complican la eficacia de los algoritmos de visión por computador que permiten a un sistema robótico “ver” y actuar [2, 3].

Diferentes trabajos han tratado la aplicación del aprendizaje automático y de las redes neuronales profundas para la mejora de la percepción en la robótica agrícola obteniendo unos resultados importantes [4]. Sin embargo, muchas de estas soluciones requieren un volumen elevado de datos reales, cuya obtención es costosa, limitada y muy dependiente de factores externos como la climatología o la disponibilidad estacional de cultivos [5]. Esto ha alentado la búsqueda de nuevas metodologías que permitan entrenar y validar algoritmos de forma eficiente y en condiciones controladas.

Dentro de este contexto, la utilización de las simulaciones virtuales se perfila como una opción potente. Creando réplicas digitales fotorrealistas de los entornos agrícola-

las, podemos generar grandes cantidades de datos sintéticos etiquetados, reproducir condiciones visuales complejas y realizar pruebas con seguridad y bajo coste [4]. Asimismo, el uso de entornos virtuales, también facilita la experimentación, la iteración, la mejora de los modelos antes de llevarlo a la práctica en el campo. Y la posibilidad de integrar robots simulados [6] dentro de estos entornos, incluso permite llegar a recrear los escenarios completos de una operación autónoma, lo que representa una nueva vía de investigación y desarrollo para obtener soluciones inteligentes.

Este proyecto se encuentra precisamente en esa intersección de la robótica, la simulación y la inteligencia artificial. El objetivo de este será dotar de una plataforma modular que combine un huerto virtual fotorrealista con un robot simulado con visión estereoscópica, implementados ambos con Unity. El sistema se desarrollará inicialmente a partir de un entorno de árboles de cerezo, sin embargo, los problemas tratados, como los de detección de bajo condiciones complejas de luminosidad, oclusión y variabilidad morfológica, son comunitarios en muchos ambientes agrícolas. La plataforma será útil para el entrenamiento y la evaluación de algoritmos de detección de frutas mejorando su posterior aplicación en sistemas reales de recolección, sin importar la especie de cultivo. Mediante este proyecto se espera poder contribuir a la automatización agrícola ayudando en el desarrollo de soluciones más accesibles y robustas así como más adaptables a las condiciones del mundo real.

1.1. Problema

A pesar de los logros y avances de la visión por computador y el aprendizaje automático en el área de la robótica agrícola es necesario resolver importantes retos para alcanzar una correcta detección y posterior localización de las frutas en escenario real. Aspectos como una iluminación cambiante, la oclusión de hojas y ramas o la variabilidad morfológica de los frutos afectan negativamente al rendimiento de los algoritmos y reducen la fiabilidad de estos en escenarios cambiantes o complejos [2, 3].

Por otro lado la generación de modelos robustos requiere muchos datos etiquetados en el mundo real, los cuales son difíciles de obtener, estacionales y difíciles de controlar [5]. Esta limitación afecta la posibilidad de experimentar, limita la validación en las distintas condiciones visuales y retrasa la llegada de soluciones efectivas para la cosecha automática.

Hoy en día no existe una herramienta accesible, modular y fotorrealista que per-

mita la simulación de situaciones de agricultura realistas y que al mismo tiempo integre un robot virtual que cuente con visión estereoscópica que capture imágenes y evalúe directamente algoritmos de detección en condiciones controladas y reproducibles. Es por ello que se hace necesaria una plataforma que permita acelerar y facilitar el desarrollo de soluciones de robótica agrícola, teniendo en cuenta las restricciones del trabajo en el campo y contribuir a la implementación de sistemas más precisos, generalizables y listos para su futura implementación en entornos reales.

1.2. Objetivos

Para sentar las bases para la futura implementación en sistemas robóticos reales, es necesario desarrollar un entorno de simulación fotorrealista en Unity en el que se implemente de manera modular un huerto virtual y un sistema de robot virtual, que cuente con visión estereoscópica, de tal manera que se puedan generar datos sintéticos y pueda evaluarse el rendimiento de los algoritmos de detección de frutas haciendo uso del aprendizaje automático.

1.2.1. Objetivos Específicos

Objetivo específico 1: Diseñar y construir un huerto virtual de cerezos con condiciones visuales realistas en Unity.

Objetivo específico 2: Generar un conjunto de datos sintéticos que incluya imágenes con etiquetas de referencia para el entrenamiento de modelos de detección.

Objetivo específico 3: Desarrollar un robot virtual equipado con visión artificial que sea capaz de tomar imágenes del entorno simulado y permita la posibilidad de validar modelos de aprendizaje automático para la detección y localización de cerezas.

Objetivo específico 4: Validar el desempeño de los modelos de detección en el entorno simulado de acuerdo a las métricas de precisión y robustez e incluir hacer una validación preliminar en un pequeño conjunto de imágenes reales para observar su capacidad de generalización y dejar la arquitectura habilitada para sus múltiples usos futuros.

1.3. Pregunta de investigación

¿Puede un entorno de simulación que combine un huerto virtual y un robot con visión artificial en Unity, servir como una herramienta efectiva para entrenar, evaluar y validar preliminarmente algoritmos de detección y localización de frutas, para una aplicación posterior en contextos reales?

1.4. Hipótesis

El desarrollo de un entorno de simulación fotorrealista modular que integre un huerto virtual junto a un sistema robótico virtual, nos ayudará a entrenar y validar algoritmos de detección y localización de frutas mediante aprendizaje automático. De esta manera, se podrán hacer grandes recortes a las extensas pruebas físicas. Esto supone un ahorro relevante en tiempos y recursos para el desarrollo de los sistemas robóticos agrícolas. Una primera validación en imágenes reales nos permitirá constatar su capacidad de generalización y su ayuda en la futura implementación de sistemas reales en plataformas robóticas.

2. Revisión Bibliográfica

2.1. Visión por Computador aplicada a la Agricultura

La visión por computador ha emergido como una tecnología esencial para la transformación digital del ámbito agrícola dado que para los sistemas automáticos permite lograr la interpretación, comprensión y tratamiento de la información visual del medio, sin la intervención de una persona. Esta herramienta se ha utilizado ampliamente en tareas como la clasificación de cultivos, la detección de enfermedades, el seguimiento fenológico y sobre todo en la identificación y localización de frutos [7]. Su capacidad para automatizar tareas agrícolas ha permitido contribuir significativamente hacia una agricultura más eficiente, más sostenible y menos dependiente de la mano de obra humana intensiva.

El uso de modelos de deep-learning en los sistemas de visión ha permitido abandonar las limitaciones impuestas por los métodos convencionales fundamentados en reglas manuales, umbrales y segmentación de imágenes por color, logrando así una mayor precisión y una adaptación efectiva a situaciones que antes eran problemáticas, por ejemplo, cambios de iluminación, oclusiones, variación morfológica de los cultivos, etc [8]. Gracias a ello, se han visto muy favorecidas y superadas las técnicas en agricultura de precisión, ya que permite la detección específica y oportuna de los objetos de interés relacionados con la agricultura, es decir, los agricultores pueden tomar decisiones unipersonalmente adecuadas y localizadas, de este modo se crea una interconexión más directa entre el sistema máquina y el agricultor.

Por otro lado, la progresión de sensores avanzados, cámaras RGB y cámaras multiespectrales ha abierto la puerta a nuevas posibilidades para los sistemas de visión por computador, que permite combinar información visual con otra variable

medioambiental para llegar al análisis de los datos con una mayor resolución. Esto ha llevado a una transformación hacia los sistemas que permiten una visión autónoma e inteligente en el ámbito agrícola, dando lugar a una nueva generación de soluciones robóticas que hacen un mayor uso de la visión artificial [9].

Finalmente, esta realidad tecnológica crea un planteamiento para el desarrollo de ambientes simulados capaces de generar datos visuales de ámbito agrícola, ya que permite experimentar y entrenar modelos sin necesidad de grandes volúmenes de datos reales, línea que es realmente la que se pretende abordar en el presente proyecto.

2.2. Aprendizaje Automático y Detección de Objetos

La detección y el reconocimiento de frutas a través del Deep Learning (DL) [10], dentro de la agricultura de precisión y de la automatización de procesos es en la actualidad, una línea de investigación relevante. Esta línea de investigación trata de poder construir o desarrollar sistemas que sean capaces de detectar e identificar frutas en imágenes de situación real, debiendo afrontar dificultades como los cambios de la luz, las oclusiones, las diferencias morfológicas que presenta cada fruta y la gran afinidad cromática existente entre las distintas frutas y entre estas y su entorno.

El DL (aprendizaje profundo), en particular a través de las CNN (redes neuronales convolucionales) [11], ha supuesto un avance considerable en la aplicación de la visión por computador a la agricultura. Debido a su capacidad para detectar objetos en tiempo real y de forma precisa; se han explotado técnicas como YOLO (You Only Look Once) [12], Faster R-CNN [13] o SSD (Single Shot MultiBox Detector) [14], que son potentes técnicas, incluso en complejos escenarios de detección positiva en la que los métodos tradicionales basados en reglas o la segmentación por color no funcionan.

Los sistemas de detección basados en Deep Learning, además de alcanzar una mejor tasa de aciertos en el reconocimiento de frutas, han potenciado el desarrollo de aplicaciones también importantes como la recolección automatizada, el seguimiento del desarrollo y la estimación del rendimiento. De todas maneras, uno de los principales desafíos es la necesidad de conjuntos de datos etiquetados de gran volumen y de buena calidad para el entrenamiento de los modelos. La obtención de estos conjuntos de datos es muy costosa y laboriosa y su diversidad es importante para poder

garantizar la habilidad del modelo para la generalización en los distintos entornos agrícolas [2].

2.3. Desafíos visuales en entornos agrícolas

A pesar de los avances en visión por computador en el ámbito agrícola, los entornos naturales presentan unas condiciones visuales que dificultan la precisión y la fiabilidad de sus sistemas automatizados, lo que afecta directamente al rendimiento de los algoritmos de detección y reconocimiento. A continuación, se describen algunas de las principales dificultades visuales que deben ser resueltas para lograr desarrollar soluciones eficientes:

2.3.1. Iluminación cambiante

El cultivo en el exterior está sometido a una iluminación natural y de calidad variable por el paso de las nubes, las estaciones del año, la hora del día, el clima y la inclinación del terreno; por lo que esta variabilidad puede generar sombras, reflejos o zonas sobreexpuestas que afectan la visibilidad y el contraste entre el fondo y los frutos. Hoy en día, los modelos de aprendizaje profundo han mejorado la robustez frente a este tipo de condiciones, aunque todavía pueden experimentar degradaciones en términos de precisión si las condiciones lumínicas se alejan de las que se exhibieron durante su entrenamiento [15].

2.3.2. Oclusión

Uno de los problemas más relevante en la detección de frutos es la oclusión, especialmente en cultivos de densa frondosidad. Los frutos pueden hallarse parcial o totalmente ocultos por follaje, ramas u otros frutos, lo que da lugar a problemas serios de identificación y localización. A modo de solución de esta problemática han surgido modelos de detección conscientes de la oclusión, como es el caso del modelo AHG-YOLO [16], que agrupa las frutas en función del nivel de oclusión, por lo que, viene a contribuir en incrementar la precisión en situaciones complejas. También la combinación de técnicas de visión estereoscópica con el uso de técnicas para estimar la profundidad han sido exitosas en inferir la presencia y la localización de frutas que se hallan parcialmente ocultas [17].

2.3.3. Variabilidad morfológica de las frutas

Pese a que las frutas puedan proceder de la misma variedad o especie, estos frutos presentan una gran variación de formas, colores, texturas y tamaños. Esta variación morfológica, la cual es capaz de ser influenciada por los cambios que produce la genética, el medio o el manejo agronómico, conlleva una difícil generalización de los modelos de detección de las propias frutas. Es por esto que los modelos entrenados a partir de conjuntos de datos limitados podrían no ser capaces de identificar correctamente frutas con rasgos inusuales a los comunes [2]. Para poder fortalecer la robustez y fiabilidad de los modelos además de tener conjuntos de datos variados y representativos, es necesario poder proporcionar la variabilidad mencionada anteriormente que existe en las frutas que se estén trabajando.

2.4. Generación de datos sintéticos y su uso

La creación de datos sintéticos se ha convertido en una práctica habitual y estratégica en el aprendizaje automático [18], sobre todo en aquellas situaciones en las que la recolección y la anotación de datos reales se convierte en una actividad costosa, tediosa y limitada. En el caso de la agricultura, donde las condiciones climáticas, morfológicas y de campo están en constante cambio y es difícil llevar un estándar común, los datos sintéticos vienen a ser una alternativa potente y flexible [19]. Éstos, que son generados de forma artificial mediante algoritmos o motores de simulación, intentan reproducir las características y las distribuciones de los datos reales. Estos datos pueden incluir imágenes, secuencias de vídeo, anotaciones de objetos, mapas de profundidad, etc. En lo que se refiere a la visión natural por ordenador en la agricultura se han utilizado entornos de simulación impulsados por motores de simulación como Unity o Blender, que simplifican la modelización controlada del crecimiento de cultivos, de frutas, así como condiciones del medio ambiente y la modelización de la iluminación. [20, 21].

Las ventajas que los datos sintéticos presentan en comparación con los reales son numerosas, ya que su forma de generar requiere de una menor inversión y es más escalable al poder generar muchas muestras anotadas automáticamente y sin intervención humana. También se pueden controlar y manipular factores relevantes como la luz, el ángulo de vista, la maduración del fruto, las formas o el nivel de

oclusión y se puede generar una diversidad de condiciones complicadas de recoger en campo, además los datos sintéticos permiten la combinación con técnicas de dominio mixto lo que permite alcanzar un equilibrio de realismo visual y precisión geométrica, lo que a su vez mejora la capacidad de generalización de los modelos [22].

2.5. Simulación fotorrealista en entornos agrícolas

La simulación fotorrealista en el ámbito agrícola ha adquirido creciente importancia como método alternativo para llevar a cabo el desarrollo, la validación y la optimización de los sistemas inteligentes sin recurrir únicamente a los entornos reales. Con esta práctica se pueden generar representaciones digitales fotorrealistas y a escala real de los campos agrícolas, de las plantas y de los robots mediante entornos tridimensionales, los cuales son la base de la simulación de pruebas de forma segura, reproducible y económica.

Mediante el uso de motores gráficos de simulación como Unity, se pueden modelar entornos en tres dimensiones suficientemente flexibles y detallados como para conseguir la generación de datos sintéticos a partir de la iluminación, perspectiva y morfologías de los cultivos que varían. Esta técnica ha mostrado efectividad en mejorar los modelos para la detección de frutas ya que produce un conjunto de datos variado con anotaciones precisas [20].

2.6. Visión Estereoscópica

La visión estereoscópica, que se basa en la percepción binocular de los humanos, se ha convertido en una herramienta fundamental en la agricultura de precisión. Por medio de dos cámaras a una distancia conocida, la visión estereoscópica permite estimar profundidad y reconstruir tridimensionalmente las escenas del medio agrícola para aplicaciones como la detección de frutos, la estimación de los rendimientos y la conducción autónoma de robots agrícolas en los campos [23]. Esta visión es una ventaja en entornos agrícolas en donde las condiciones de iluminación y la oclusión del follaje pueden variar mucho ya que permite tener información de profundidad más robusta que los sistemas de visión monocular. Por ejemplo, en la detección de uvas de mesa; combinar visión estereoscópica y técnicas de aprendizaje profundo ha demostrado ser un método mejorado en la estimación de la profundidad y la

localización de los frutos frente a la variabilidad a nivel de condiciones de iluminación y la oclusión [24].

Este tipo de visión además de ser un método pasivamente estimador de la profundidad sin sensores activos, presenta un método relativamente accesible y que se adapta fácilmente a las distintas plataformas robóticas. De esta forma, las capacidades de la visión estereoscópica permiten realizar una reconstrucción tridimensional precisa del entorno y útil para poder superar múltiples tareas de la agricultura como la segmentación de cultivos, la navegación autónoma entre hileras o la estimación del volumen de la cosecha [25, 26]. De todos modos, esta tecnología enfrenta retos tecnológicos en su implementación. En determinadas situaciones, como por ejemplo, en una intensidad de luz elevada, en superficies con bajo relieve, en un entorno con una densa vegetación o con plantas enredadas, los errores de correspondencia entre cámaras son mucho más probables. Sumado a esto, en la literatura se ha documentado que la oclusión que se produce entre hojas, ramas y frutos dificulta la estimación de la profundidad y que los algoritmos de coincidencia estereoscópica tienen también altos tiempos de procesamiento y producen errores en escenas complejas [27]. A pesar de estas limitaciones, la visión estereoscópica se está estableciendo como un método de percepción espacial en los sistemas de agricultura de precisión, versátil y eficaz.

2.7. Evaluación y Métricas de Detección Agrícola

Analizar de forma minuciosa y detallada los modelos de detección es de vital importancia para validar el uso de un modelo real. Las métricas que más se suelen utilizar son la precisión, la exhaustividad (recall) y la puntuación F1, que ayudan a calcular la capacidad o certeza de un modelo para detectar bien los objetos de interés como pueden ser en este caso los frutos [28]. Estas métricas resultan ser muy útiles cuando hay desequilibrio entre clases, como en el caso de imágenes donde los frutos ocupan solo una pequeña parte del área total. En el caso de las tareas de localización, se usa la métrica Intersection over Union (IoU) para comparar las predicciones con las anotaciones reales, añadiendo umbrales para indicar cuando una detección es válida [29]. La elección de estas métricas resulta muy importante en agricultura donde las condiciones visuales pueden cambiar ampliamente. Una evaluación exhaustiva permite medir el rendimiento del modelo pero también desafía la habilidad del modelo para indicar sus límites e incitar futuras mejoras.

3. Metodología

La base de este proyecto es una plataforma de simulación modular desarrollada en Unity. La cual cuenta con dos elementos principales, un huerto virtual fotorrealista y un robot simulado con visión estereoscópica, los cuales tienen como finalidad la generación de datos sintéticos que serán utilizados para entrenar y testear los algoritmos de detección y localización de la fruta. En los siguientes puntos se definen la fases que integra el diseño y desarrollo del sistema A.2.

3.1. Flujo de trabajo

En términos generales, el proceso comienza con la configuración controlada del escenario simulado, seguida por la ejecución del recorrido del robot para capturar imágenes y generar anotaciones automáticas mediante Unity Perception. Posteriormente, las anotaciones se convierten a formato YOLO y se organiza el dataset en particiones *train/val/test*, habilitando el entrenamiento de modelos de detección y su evaluación mediante métricas estándar (*mAP*, *precisión* y *recall*). Finalmente, los resultados se interpretan bajo el concepto de Rango de Validez Operacional (RVO) para delimitar las condiciones en que el modelo mantiene un desempeño aceptable A.1.

3.2. Diseño del Huerto Virtual

El primer paso consiste en la construcción del medio en el cual se trabajará, este huerto virtual de carácter fotorrealista se llevó a cabo en el motor gráfico de Unity. El diseño del mundo incluye la incorporación de elementos visuales que pretenden

ser realistas, como lo pueden ser los árboles de cerezos, el terreno y sus materiales, la iluminación dinámica causada por el ciclo del día y la noche, sumándole las variantes de las estaciones del año que influyen respecto a las nubes y demás condiciones climatológicas que pueden ocurrir [30, 31]. Para reproducir todas aquellas variaciones que pudieran suceder durante una época de cosecha de frutas, estas se pueden modificar con el fin de contar con distintos escenarios sobre los cuales experimentar para generar o crear un espacio virtualmente representativo de las distintas situaciones que se darían en el mundo real.

3.2.1. Elementos del Huerto Virtual

- **Terreno:** Se implementó una superficie o terreno con relieves para simular la topografía común en los huertos, utilizando texturas y materiales que representen realismo visual B.1.1.
- **Árboles:** Los Cerezos se organizarán en filas paralelas y en conducción espaldera, a modo de huertos comerciales. Por temas de rendimiento y escalabilidad del entorno sólo se trabajará predominantemente con un modelo de árbol único, optimizado para instanciarse un gran número de veces en la escena. La variedad de ese entorno se determinará mediante el uso de frutos procedimentales, donde su *cantidad*, su *colocación* y su *tamaño* en cada árbol variarán de forma que se puedan obtener diferentes configuraciones de cerezas entre las capturas B.1.2.
- **Iluminación Dinámica:** Se incluirá un ciclo de día y noche mediante el cual se gestionen distintos momentos del día por medio del uso horario para la captura de imágenes con diferentes posiciones del sol, sombras generadas y colores del ambiente [32, 33, 34] B.1.3.
- **Condiciones climáticas:** Se incorporaron parámetros ajustables de neblina y nubosidad, que tienen la opción de modificarse manualmente a fin de presentar diferentes condiciones atmosféricas en el momento de la captura de datos. Estas modificaciones permiten variar la visibilidad y la iluminación global de la escena, lo que aporta diversidad en las condiciones de captura de imágenes. [31]. B.1.4

- **Control de condiciones:** Se puede simular distintos escenarios a partir del uso de sliders, botones o parámetros, los que permiten ajustar las horas, nubes y visibilidad, generando así los distintos datasets [30] B.1.5.
- **Diseño modular y extensible:** La estructura del entorno estará diseñada para que sea sencillo añadir nuevos elementos o cambiar los que existan sin tener que rehacer el proyecto completo, de esta forma el huerto podría tener una evolución continua [31] B.1.6.

3.2.2. Parametrización y diversidad controlada

Para aumentar la capacidad de generalización del modelo y por ende corregir la discrepancia entre simulación y realidad, las variaciones del entorno se implementaron a partir de un enfoque de diversidad controlada. Cada ejecución se llevó a cabo combinando parámetros fijos, para asegurar una cierta consistencia durante una sesión de captura, con factores variables para provocar diversidad. Se controló la variación de las siguientes variables: *hora del día*, *nubosidad*, *neblina*, variaciones ligeras en la *iluminación global*, *distribución* de frutos y sus *tamaños*.

3.3. Diseño del Robot Virtual

Lo que se busca con este paso es crear un sistema robótico agrícola de forma funcional [35] que sea capaz de realizar el desplazamiento entre las hileras de los cerezos que conforman el huerto virtual mientras que se van capturando imágenes útiles para el entrenamiento de modelos de detección de frutos. Contará por lo tanto con un sistema de cámaras virtuales configuradas para la captura de imágenes en dos vistas cercanas, emulando una visión estereoscópica [36] que permita generar imágenes RGB.

El robot podrá trasladarse de modo automático sobre el sistema de navegación por mallas (NavMesh) que proporciona Unity [37], el cual encuadra tanto áreas transitables como rutas optimizadas, permitiendo al robot moverse de manera automática entre árboles imitando una exploración del huerto.

Finalmente, el robot tendrá la funcionalidad de tomar y almacenar las imágenes que haya capturado con sus cámaras a lo largo del recorrido, imágenes que servirán

como insumo para la posterior fase de entrenamiento y validación de modelos de detección de frutas [38] B.2.1.

3.3.1. Patrón de recorrido y captura

Con la intención de garantizar distintas perspectivas y no concentrar las capturas únicamente en las mismas áreas de las plantaciones, el recorrido del robot también fue diseñado con el propósito de realizar movimientos coordinados entre hileras de manera que las capturas de los frutos fueran realizadas desde distintos puntos de vista y distancias. Y de la misma manera para limitar las capturas desenfocadas, relacionadas con el movimiento e imágenes borrosas, las imágenes se tomaron mediante capturas constantes y estables, asegurando una homogeneidad visual del dataset.

3.3.2. Configuración de cámaras

Las cámaras del robot se configuraron para mantener parámetros en conjunto entre sesiones. En el caso de visión estereoscópica, se mantuvo una separación fija entre cámaras y la opción de sincronización de capturas, de modo que cada escena pudiera generar pares comparables o individuales. Para alcanzar una diversidad visual alta sin necesidad de perder el control de las condiciones iniciales de la escena a fotografiar, las cámaras del robot fueron ajustadas para que, al ir capturando fotografías, fuesen capaces de cambiar los parámetros de configuración en un rango reducido. En el caso del rango de aleatoriedad establecido, se ajustaron las siguientes propiedades:

- **Altura de la cámara:** Variaciones verticales en la altura de la cámara para capturar los frutos desde distintas alturas del árbol manteniendo la misma distancia relativa a la hilera.
- **Orientación:** Variaciones en un rango angular (rotación) para simular cambios en el encuadre y dirección de mirada del sensor.
- **Campo de visión (FOV):** Variaciones en el campo de observación dentro de un intervalo variado de distancias de captura para intentar capturar distintos tamaños que aparezcan en el fruto; es decir, el tamaño relativo del objeto en la imagen varía.

Estos valores configurados permitieron generar capturas con variación en tamaño aparente del fruto, oclusión por hojas/frutos y cambios de contraste asociados al encuadre, complementando además la variación provista por las condiciones del entorno B.2.2.

3.4. Generación de Datos Sintéticos

En esta fase, el reto principal fue conseguir un set de imágenes lo más variadas y controladas posibles, que mostrara condiciones de observación realistas en huertos, conservando el grado de similitud que existe entre el tipo de anotaciones que ofrece el conjunto de pruebas real. El sistema de cámaras del robot, se configuró haciendo hincapié en el hecho de que el conjunto de imágenes obtenidas debería mostrar la mayor similitud con el dataset real de referencia por lo que se generaron imágenes RGB junto con las anotaciones 2D pertinentes [39, 38]. Aunque la plataforma creada permite activar una configuración para tomar pares estereoscópicos, en este trabajo se utilizó fundamentalmente para obtener dos puntos de vista e imágenes complementarias, no para estimación de la profundidad. Las dos cámaras pueden moverse y aplicar variaciones coordinadas (altura, orientación y FOV) dentro de unos rangos predefinidos. Y la adquisición se llevó a cabo alternando los espacios laterales de la plantación; una cámara captura un lateral de la hilera y la otra registró el lateral opuesto. Con esto se aumentó la cobertura y diversidad del set de imágenes obtenidas por los diferentes puntos de vista capturados sin dejar de seguir un protocolo homogéneo para la validación inicial Sim2Real.

3.4.1. Etiquetado automático con Unity Perception

El etiquetado del conjunto de datos sintéticos fue el resultado del uso de Unity Perception, con la que se configuró la captura automática de imágenes y la anotación automática de los frutos visibles en cada imagen [38]. En coherencia con el objetivo experimental y las características del conjunto de datos real usado como referencia, en este trabajo únicamente se considera el esquema de 2D Bounding Boxes (cajas delimitadas en el plano de imagen). Por lo tanto, no se consideraron anotaciones en profundidad, mapas de distancia, segmentaciones ni cualquier otra etiqueta alternativa. Esta elección se justifica ya que el conjunto de datos real seleccionado (Cherry CO) trae anotaciones comparables en 2D, circunstancia que facilitará una compa-

ración directa entre los dominios sintético versus real y la posibilidad de evaluar el rendimiento bajo un protocolo homogéneo. En términos de práctica, a partir de cada captura RGB se disponía de una anotación de bounding box relacionada con cada fruto que el sistema de etiquetado había conseguido detectar, quedando ambas asociadas a un mismo identificador o archivo, pasando a ser los elementos objeto de posterior conversión y uso en entrenamiento B.2.3.

3.4.2. Estructuración del dataset y conversión de etiquetas

Al finalizar el proceso de recolección de datos, las imágenes y anotaciones fueron organizadas en base a una estructura por defecto para facilitar el entrenamiento de forma supervisada. Las anotaciones 2D generadas por Unity Perception fueron transformadas a la forma que requería YOLO generando un archivo por imagen con las coordenadas normalizadas de cada caja delimitadora y la clase correspondiente. Este proceso de convertirlas al formato de etiquetas necesario por el sistema YOLO, permitió construir un dataset sintético completamente utilizables en entrenamiento. Consiguiendo así que el tipo de etiqueta sintética contuviera la misma estructura que el tipo de etiquetas de dataset real que se usaría en la fase de validación. En consecuencia, las comparaciones de las inferencias de los datos de prueba con sus imágenes originales, se ejecutarían con el mismo criterio de etiquetas, facilitando la interpretabilidad en el estudio Sim2Real.

3.5. Entrenamiento y Validación Preliminar de Modelos

El objetivo principal que debe lograrse en esta etapa es que los modelos aprendan como reconocer e identificar correctamente las frutas, en escenarios que simulen los retos visuales del entorno agrícola real. Para comprobar el rendimiento generado durante el entrenamiento, se utilizarán métricas como la precisión, el recall y el mAP [40], de forma que se pueda monitorizar el comportamiento y ajustar los parámetros para mejorar el modelo.

3.5.1. Validación preliminar y evaluación de generalización

Una vez finalizada la etapa de entrenamiento, se llevó a cabo una validación preliminar destinada a evaluar la capacidad de generalización del sistema siguiendo

un enfoque Sim2Real teniendo como referencia el dataset proveniente del documento *Cherry CO Dataset: A Dataset for Cherry Detection, Segmentation and Maturity Recognition* [41]. La idea de esta fase de validación era poder medir la brecha entre el dominio sintético versus el dominio real y, al mismo tiempo, determinar patrones de error o situaciones en que el detector es operacionalmente válido. Si bien Cherry CO incluye anotaciones ricas para detección, segmentación y reconocimiento de madurez, en esta memoria se ha priorizado detectar la fruta a partir de bounding boxes 2D. La razón de esta elección ha sido la de mantener un protocolo experimental más sencillo y controlado en la etapa de validación inicial del flujo de trabajo completo y la de asegurar comparabilidad directa respecto con el tipo de etiquetado puesto en práctica en Unity Perception. Como consecuencia, la evaluación se ha limitado a las métricas estándar de detección (precisión, recall y mAP) [42], dejando las extensiones a segmentación y madurez como futuras líneas de trabajo.

3.5.2. Parámetros de entrenamiento y configuración experimental

Para asegurar trazabilidad y permitir replicación, se documentan los hiperparámetros utilizados en los entrenamientos realizados con YOLOv8. Cuando un parámetro no fue fijado manualmente, se utilizó la configuración por defecto del framework (Ultralytics).

Parámetro	Valor
Modelo inicial (weights)	yolov8n.pt
Tamaño de imagen (<i>imgsz</i>)	640
Épocas (<i>epochs</i>)	30
Batch (<i>batch</i>)	16
Learning rate inicial (<i>lr0</i>)	1×10^{-3}
Optimizador (<i>optimizer</i>)	SGD
Scheduler coseno (<i>cos_lr</i>)	True

Tabla 3.1: Configuración base de entrenamiento utilizada en la mayoría de los experimentos.

3.5.3. Diseño experimental aplicado

La validación se ha estructurado a partir de una secuencia de experimentos con la finalidad de responder ciertas preguntas sobre generalización:

- **E0 (Punto de referencia inicial Set Real):** Entrenamiento y evaluación con imágenes reales (Cherry CO diurno), como referencia de rendimiento en dominio real.
- **E1 (Sintético diurno):** Entrenamiento y evaluación en el dominio sintético, para medir desempeño “in-domain” y establecer el punto de partida sintético.
- **E2 (Sintético + Real):** Evaluación del modelo entrenado con sintético directamente sobre el test real, para medir la brecha Sim2Real.
- **E3 (Sintético + Real fine-tuning):** Fine-tuning con datos reales a partir del modelo sintético y evaluación en test real, para observar cuánto se reduce la brecha al incorporar muestras reales.
- **E4 (Sintético nocturno con linterna + Real día):** Entrenamiento con sintético nocturno iluminado por linterna y evaluación en test real diurno, como caso extremo para analizar el efecto de cambios fuertes de iluminación.

Exp.	Modelo	imgsz	Épocas	Batch	lr0
E0	yolov8n.pt	640	30	16	1×10^{-3}
E1	yolov8n.pt	640	30	16	1×10^{-3}
E3	best.pt (E1)	640	20	16	5×10^{-4}
E4	yolov8n.pt	640	30	16	1×10^{-3}

Nota: En todos los casos se mantuvo `optimizer=SGD` y `cos_lr=True`. En E2 y E3 se reutiliza el modelo entrenado en E1 y para E3 se aplica *fine-tuning* con datos reales.

Tabla 3.2: Resumen de hiperparámetros por experimento.

3.5.4. Diagnóstico del dominio sintético y variantes E1a–E1d

Al analizar el E1 se identificó un patrón significativo, el modelo puede mantener una precisión moderada/alta, pero su recall es bajo, lo que implica que deja una gran cantidad de frutas sin detectar. Antes de volver a atrevernos a complicar el

diseño experimental (añadiendo nuevas condiciones), se decidió hacer un diagnóstico para ver de dónde podía venir el problema revisando curvas, métricas por confianza, ejemplos de falsos positivos/falsos negativos y distribución de tamaños de bounding boxes. Este análisis nos hizo lograr ver que el dataset sintético tiene una proporción alta de frutos pequeños y/o ocluidos, comportándose como un “stress test” más exigente que el set de datos real.

Para aislar si la limitación era del modelo o del dominio, se construyeron variantes controladas sobre E1:

- **E1a:** Se aumentó la cantidad de épocas manteniendo arquitectura y resolución. El objetivo fue verificar si el recall bajo era producto de subentrenamiento o convergencia insuficiente.
- **E1b:** Se incrementó la resolución de entrenamiento, manteniendo el modelo base. El objetivo fue aumentar el número de píxeles disponibles en frutas pequeñas, reduciendo el colapso de información espacial que ocurre al reescalar imágenes a resoluciones menores.
- **E1c:** Se utilizó una variante de YOLO de mayor capacidad (YOLOv8s) manteniendo resolución. El objetivo fue evaluar si una red con mayor número de parámetros y capacidad representacional mejora el recall bajo escenarios con oclusión y señales visuales débiles.
- **E1d:** Combinación de E1b y E1c. El objetivo fue observar si los beneficios de mayor capacidad y mayor resolución se suman y si eso reduce el cuello de botella asociado a objetos pequeños del dominio sintético.

Estas variantes permitieron verificar que el desempeño mejora, especialmente al aumentar la resolución, consistente con la hipótesis de que el tamaño aparente del fruto es un factor crítico.

Exp.	Modelo	imgsz	Épocas	Batch
E1a	yolov8n.pt	640	60	16
E1b	yolov8n.pt	960	30	16
E1c	yolov8s.pt	640	30	16
E1d	yolov8s.pt	960	30	16

Nota: En todas las variantes se mantuvieron los valores predeterminados del `framework(lr0,optimizer,cos lr)`

Tabla 3.3: Variantes del experimento E1 y sus hiperparámetros.

3.5.5. Evaluación por tamaños y Rango de Validez Operacional (RVO)

Dado que la principal dificultad se concentró en objetos pequeños, la evaluación se refinó incorporando métricas por rangos de tamaño basados en el área del bounding box:

- **Small:** $\text{área} < 32 \times 32 \text{ px}$
- **Medium:** $32 \times 32 \leq \text{área} < 48 \times 48 \text{ px}$
- **Large:** $\text{área} \geq 48 \times 48 \text{ px}$

Bajo la geometría de captura estándar del robot (distancias y FOV en rangos definidos) los casos de $32 \times 32 \text{ px}$ constituyen los casos extremos (muy lejos, oclusión importante y en bordes o límites del frame). Por lo tanto, se definió un Rango de Validez Operacional (RVO) asociado a objetos de al menos MEDIUM, el cual se entiende que es el rango que más razonablemente se puede exigir para un primer uso robótico. En términos de reporte, se presentan dos niveles de resultados:

- **Métricas globales:** Métricas sin filtrar, por transparencia.
- **Métricas dentro del RVO:** Métricas enfatizadas como lectura operacional del sistema.

Finalmente y para cerrar la comparación de forma justa, este análisis por tamaños se extiende también al conjunto real y a los modelos evaluados en test real (E0, E2 y E3). Esto permite comparar el desempeño entre dominios dentro de rangos equivalentes, evitando conclusiones sesgadas por diferencias en la distribución de tamaños entre datasets.

3.6. Hardware y software utilizados

Para asegurar reproducibilidad, se documenta el entorno de ejecución utilizado en el desarrollo, simulación y entrenamiento del proyecto.

- CPU: AMD Ryzen 7 7800X3D
- GPU: NVIDIA GeForce RTX 4070Ti
- RAM: 32GB
- Sistema operativo: Windows 11
- Versión de Unity: Unity 6 6000.0.33f1
- Render Pipeline: HDRP
- Versión de Unity Perception: 1.0.0-preview.1

3.6.1. Rendimiento computacional

En esta sección se caracteriza el rendimiento computacional del sistema tanto en simulación (Unity) como en entrenamiento (Ultralytics YOLO), con el fin de documentar costos prácticos de ejecución bajo el hardware descrito previamente. La caracterización presentada es de tipo descriptiva y busca entregar una referencia útil para reproducibilidad y futuras extensiones, más que establecer un *benchmark* exhaustivo.

Simulación (Unity)

El rendimiento del simulador se evaluó de manera observacional utilizando el panel *Stats* de la vista *Game* en Unity durante ejecuciones representativas. Se compararon tres condiciones:

- **Navegación alejada de los árboles:** movimiento del robot/usuario sin activar instanciación de frutos.
- **Navegación cercana a los árboles:** navegación a distancia de activación, donde opera la lógica de *spawn/despawn* de frutos por proximidad.

- **Instanciación de Árboles:** Generación de más hileras con árboles.

La segunda condición corresponde al caso de uso relevante, ya que la instanciación dinámica introduce carga adicional (renderizado y actualización de objetos). La estrategia de *spawn/despawn* se incorporó como medida de optimización, dado que una instanciación masiva de frutos por hilera (o múltiples hileras simultáneamente) hacía el sistema impracticable en tiempo real. Los rangos de FPS observados y evidencia complementaria se incluyen en el Anexo C.1.

Entrenamiento (Ultralytics YOLO)

El rendimiento computacional del entrenamiento se caracterizó registrando los tiempos reportados por Ultralytics durante el proceso de entrenamiento de los modelos. Para cada experimento se recopiló el tiempo total de entrenamiento y, cuando fue posible, el tiempo promedio por época, permitiendo estimar el costo de replicación y comparar configuraciones (p.ej., cambios en resolución, capacidad del modelo o número de épocas). El detalle de tiempos por experimento se reporta en el Anexo C.2.

3.7. Capacidades de experimentación de la plataforma

El diseño modular de la plataforma le permite ejecutar experimentos controlados al ir modificando los parámetros del entorno, cultivos y sistema de captura. Es decir, el propio proyecto no sólo sería funcional para la detección de las cerezas, sino que se puede considerar como una herramienta re-utilizable para estudios de visión artificial en robótica agrícola.

3.7.1. Parámetros ambientales

La plataforma permite modificar condiciones visuales del entorno, relevantes para robustez de detectores y creación de múltiples escenarios hipotéticos:

- Iluminación: hora del día (posición), intensidad (incluyendo las estrellas) y temperatura del sol y la luna.
- Condiciones atmosféricas: nubosidad y niebla (visibilidad).

3.7.2. Parámetros del cultivo

El huerto virtual se diseñó para controlar variaciones del cultivo que afectan directamente la detección:

- Densidad de frutos por árbol y distribución espacial (concentración por zonas).
- Distribución de tamaños de frutos y proporciones controladas.
- Variación morfológica entre árboles (instancias, escala y disposición en las hileras).

3.7.3. Parámetros del sensor y captura

La plataforma permite ajustar el sistema de captura del robot, lo cual impacta la geometría de observación:

- Altura de captura y distancia a las hileras.
- Ángulos de cámara y campo de visión (FOV).
- Frecuencia y cantidad de capturas.

3.7.4. Implicancias experimentales

Estas capacidades habilitan estudios reproducibles como:

- Sensibilidad a iluminación (día/noche, contraluz, sombras).
- Impacto de oclusión y densidad de frutos en el desempeño del detector.
- Comparación entre arquitecturas de detección (YOLOv8n vs YOLOv8s, etc.).
- Optimización del protocolo de captura (distancia, FOV, resolución) para mejorar métricas y estabilidad del sistema.

4. Resultados y análisis

En este capítulo se presentan los resultados de los experimentos E0–E4 y sus variantes, junto con un análisis cuantitativo y cualitativo con el fin de intentar dar respuesta al comportamiento del detector en un marco Sim2Real. Primero se presentan las métricas globales para transparencia y comparación amplia. Posteriormente se presenta el caso de estudio del dominio sintético (E1) y sus variantes controladas, posteriormente se lleva a cabo un análisis por tamaños y la definición del Rango de Validez Operacional (RVO) junto con ejemplos representativos de error.

4.1. Resultados globales

En esta sección se reportan las métricas globales de los experimentos principales E0–E4 utilizando un protocolo homogéneo de evaluación. El objetivo es comparar el rendimiento de referencia en real (E0), el desempeño en sintético (E1), la transferencia directa sintético a real (E2), la reducción de la brecha mediante fine-tuning con real (E3) y el impacto de un cambio extremo de dominio de iluminación (E4).

Experimento	P (BoxP)	R (Recall)	mAP@0.5	mAP@0.5:0.95
E0	0.827	0.744	0.838	0.550
E1	0.584	0.473	0.456	0.266
E2	0.482	0.149	0.193	0.096
E3	0.732	0.612	0.698	0.429
E4	0.735	0.103	0.269	0.132

Tabla 4.1: Resultados globales de detección para los experimentos E0–E4.

A nivel global, E0 establece el desempeño de referencia en el dominio real (Cherry CO). E1 refleja el rendimiento “in-domain” al entrenar y evaluar en el dominio sintético, pero sus métricas globales resultan más bajas de lo esperado, lo que sugiere que el conjunto sintético es particularmente exigente. Esto se explica más adelante mediante el análisis por tamaño, el dataset sintético concentra una alta proporción de frutos small y casos ocluidos, actuando como un stress test que penaliza especialmente el recall.

Al evaluar el modelo sintético directamente sobre el test real (E2) se observa una caída adicional atribuible al domain gap. En cambio, al aplicar fine-tuning con datos reales (E3), el rendimiento en real mejora de forma clara, indicando que el sintético es útil como preentrenamiento cuando se complementa con adaptación al dominio real. Finalmente, E4 muestra que entrenar bajo condiciones nocturnas con linterna y evaluar en día afecta fuertemente la generalización, evidenciando sensibilidad a cambios extremos de iluminación. Estos resultados globales permiten comprender tendencias generales, pero no explican completamente las causas de degradación. Por ello, en las siguientes secciones se analizan en detalle el comportamiento del dominio sintético y el impacto del tamaño aparente del fruto.

4.2. Análisis del caso E1 y variantes E1a–E1d

Esta sección profundiza en el experimento E1 y en cuatro variantes controladas (E1a–E1d). El objetivo es identificar qué factores influyen más en el rendimiento

dentro del dominio sintético y determinar si la principal limitación proviene de entrenamiento insuficiente, resolución de entrada limitada o capacidad del modelo.

Variante	Modelo	imgsz	P (BoxP)	R (Recall)	mAP@0.5	mAP@0.5:0.95
E1	YOLOv8n	640	0.584	0.473	0.456	0.266
E1a	YOLOv8n	640	0.683	0.493	0.534	0.334
E1b	YOLOv8n	960	0.662	0.525	0.556	0.361
E1c	YOLOv8s	640	0.630	0.487	0.504	0.316
E1d	YOLOv8s	960	0.695	0.526	0.557	0.372

Tabla 4.2: Caso de estudio del dominio sintético: resultados globales para E1 y variantes controladas (E1a–E1d).

Las variantes corresponden a que incrementar la resolución (imgsz) está asociado a ser el motivo que más incide sobre el rendimiento de detección, especialmente en recall, en comparación con los cambios debidos a aumentar épocas o poder del modelo, los cuales aportan pequeñas mejoras. Esto tiene sentido en un escenario de objetos pequeños, cuando el fruto ocupa pocos píxeles, la resolución viene a ser una limitación principal, dado que las mejoras del modelo no son suficiente para apoyar la información visual que falta. Este diagnóstico también da paso al análisis por tamaños, ya que permite estudiar si directamente la disminución del rendimiento se encuentra precisamente el rango de frutos “**Small**”.

4.3. Resultados por tamaños

El rendimiento según el tamaño aparente del fruto es el que se descompone en esta sección, utilizando los umbrales que se habían definido con anterioridad (Small, Medium y Large) con la intención de localizar cual es el rango de tamaños en el que predominan los errores y cómo cambian las métricas, entre experimentos o variantes.

Categoría	Criterio (área bbox)	Cantidad	Proporción
SMALL	$< 32 \times 32$ px	2791	$\sim 72\%$
MEDIUM	$32 \times 32 \leq \text{área} < 48 \times 48$ px	899	$\sim 23\%$
LARGE	$\geq 48 \times 48$ px	107	$\sim 3\text{--}4\%$

Tabla 4.3: Distribución de tamaños de frutos en el conjunto de test sintético filtrado, usando umbrales basados en el área de la bounding box.

Variante	R (SMALL)	R (MEDIUM)	R (LARGE)
E1	0.511	0.976	0.991
E1a	0.527	0.974	0.981
E1b	0.604	0.992	0.972
E1c	0.518	0.970	1.000
E1d	0.603	0.989	0.981

Tabla 4.4: Recall por tamaño para E1 y sus variantes, evaluado en el test sintético filtrado.

Experimento	R (SMALL)	R (MEDIUM)	R (LARGE)
E0	0.880	0.968	0.994
E3	0.814	0.934	0.985
E4	0.396	0.494	0.691

Tabla 4.5: Recall desagregado por tamaño para E0, E3 y E4.

Los resultados obtenidos en función de los tamaños mostraron un desempeño

elevado y relativamente estable tanto en “**Medium**” como en “**Large**”, mientras que la degradación del rendimiento se concentraba en “**Small**”. Esto explica por qué las métricas globales pueden verse penalizadas cuando el dataset de evaluación contiene una proporción elevada de objetos pequeños. Además, la distribución de tamaños evidencia que el dataset sintético funciona como un stress test, ya que contiene una proporción mayor de frutos small que el dataset real, lo que hace que comparaciones globales directas entre dominios puedan ser injustas si no se controla este factor.

Dado que los casos “**Small**” representan escenarios extremos (muy lejos, oclusión fuerte o borde de FOV), se formaliza a continuación un criterio operacional para interpretar el rendimiento del sistema.

4.4. RVO: métricas operacionales vs globales

En esta sección se supone y se define lo que se entiende por Rango de Validez Operacional, con el objetivo de poder finalmente codificar el desempeño haciendo uso de un criterio práctico de validez de funcionamiento del detector. En este trabajo, el RVO se define principalmente en función del tamaño mínimo del fruto, se considera operacional el rango de objetos con tamaño, al menos MEDIUM, ya que en esta geometría el detector presenta un buen rendimiento con la estabilidad necesaria para empezar un uso robótico inicial.

Experimento	R global (REAL)	R en RVO (REAL)
E0	0.744	0.778
E2	0.149	0.168
E3	0.612	0.639
E4	0.103	0.122

Tabla 4.6: Recall global versus Recall operacional en el conjunto real (CherryCO test). El RVO se define como objetos con bounding boxes de área $\geq 32 \times 32$ px (MEDIUM+LARGE).

Experimento	R global (SYNTH)	R en RVO32 (SYNTH)
E1	0.473	0.603
E1a	0.493	0.625
E1b	0.525	0.645
E1c	0.487	0.599
E1d	0.526	0.669

Tabla 4.7: Recall global versus Recall operacional en el conjunto sintético. El RVO32 se define como objetos con bounding boxes de área $\geq 32 \times 32$ px (MEDIUM+LARGE).

Se reportan métricas globales por transparencia y honestidad científica, ya que incluyen todos los objetos del conjunto de prueba, incluso los casos extremos. Sin embargo, para determinar la viabilidad del sistema en un entorno robótico, se prevén métricas del RVO, dado que representan el desempeño en condiciones donde el objetivo tiene suficiente tamaño como para ser visto con cierta consistencia. Con este diseño se evita que una distribución sesgada en objetos realmente pequeños o muy ocluidos altere la interpretación del sistema en condiciones operativas. A través del RVO se concluye que la comparación entre E0, E2 y E3 es una estimación más justa de la brecha Sim2Real dado que la comparación entre MEDIUM versus MEDIUM y LARGE versus LARGE introducirá un sesgo por diferencias de distribución y eso implica que se puedan atribuir las mejoras de E3 a la etapa de adaptación real (fine-tuning) y no a un simple cambio implícito de la distribución típica de tipo de objetos.

4.5. Análisis cualitativo

Esta sección complementa las métricas globales y del RVO mediante inspección visual de predicciones representativas, con el fin de identificar patrones de falsos negativos (FN) y falsos positivos (FP) que expliquen la brecha Sim2Real y el efecto del fine-tuning.

4.5.1. E1

En el conjunto sintético se observan escenas con oclusión foliar, agrupaciones densas y frutas con escala aparente reducida o parcialmente visible, lo cual tiende a concentrar errores en casos de alta dificultad.

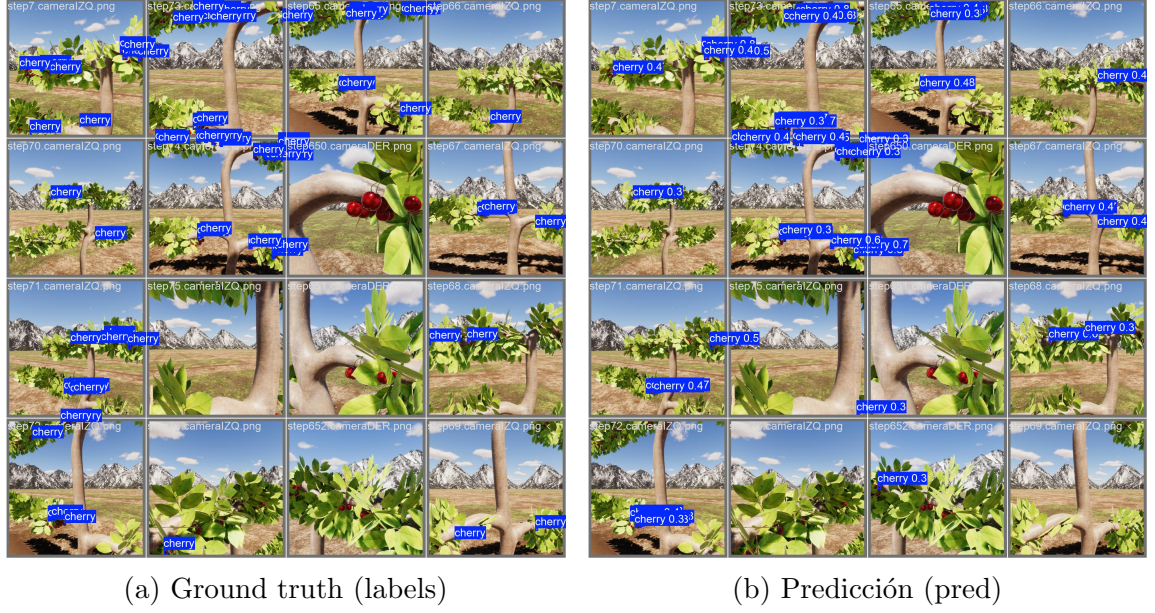


Figura 4.1: Comparación cualitativa del experimento **E1** (Sintético → Sintético) en el conjunto de prueba sintético.

Conclusión parcial: el desempeño global de E1 se ve penalizado principalmente por condiciones exigentes de visibilidad y escala, más que por fallas sistemáticas.

4.5.2. E2

En el test real CherryCO se aprecia una baja densidad de detecciones en múltiples escenas, consistente con una caída marcada del recall. La variabilidad de texturas naturales, sombras, contraluz y complejidad del follaje favorece omisiones.

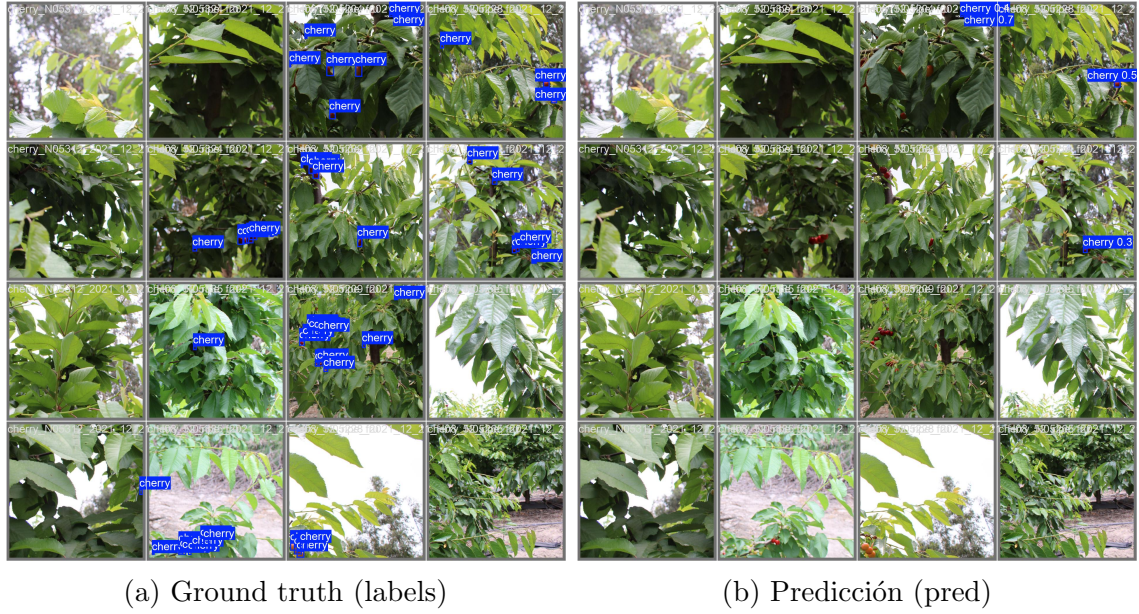


Figura 4.2: Comparación cualitativa del experimento E2 (Sintético $\rightarrow$ Real) en CherryCO test.

Conclusión parcial: la brecha Sim2Real se manifiesta mayoritariamente como falsos negativos (omisión de frutos) al cambiar del dominio simulado al real.

4.5.3. E3

Tras el ajuste con datos reales, se observa recuperación de detecciones respecto a E2, en concordancia con el aumento de recall. No obstante, aparecen FP típicos en regiones ambiguas (texturas/reflejos) y en algunos casos, múltiples cajas en zonas con agrupaciones.

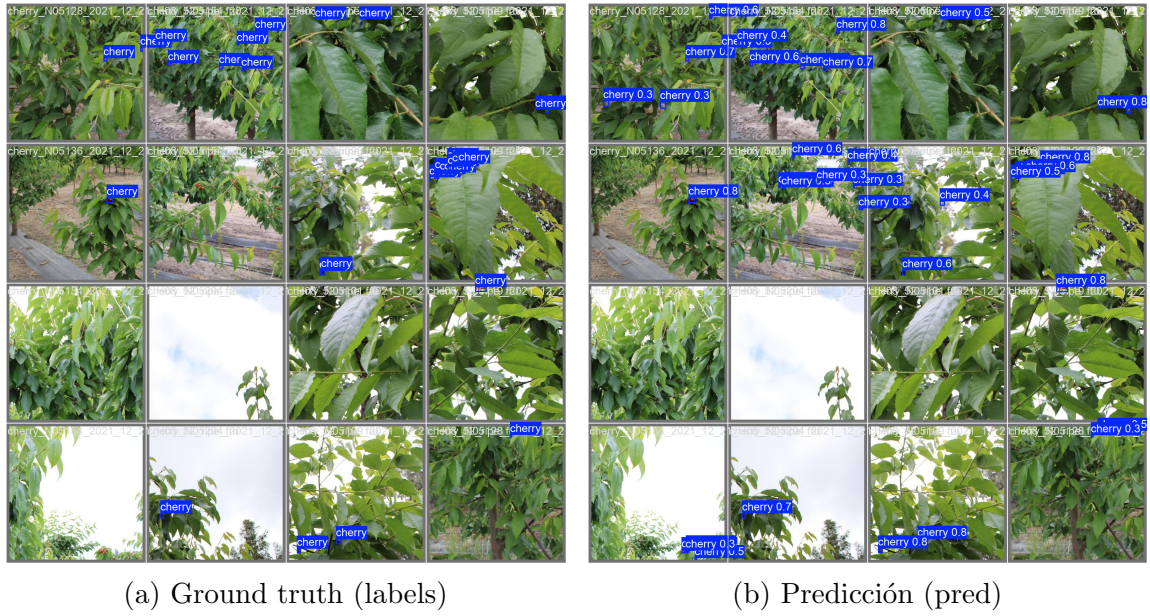


Figura 4.3: Comparación cualitativa del experimento E3 (Sintético + Real, fine-tuning) sobre CherryCO test.

Conclusión parcial: el fine-tuning reduce FN y mejora la generalización al dominio real, a costa de un incremento relativo de FP en escenas complejas.

4.5.4. E4

El modelo entrenado bajo iluminación nocturna con linterna casi no propone detecciones en el dominio real diurno, reflejando un cambio fotométrico extremo entre dominios.

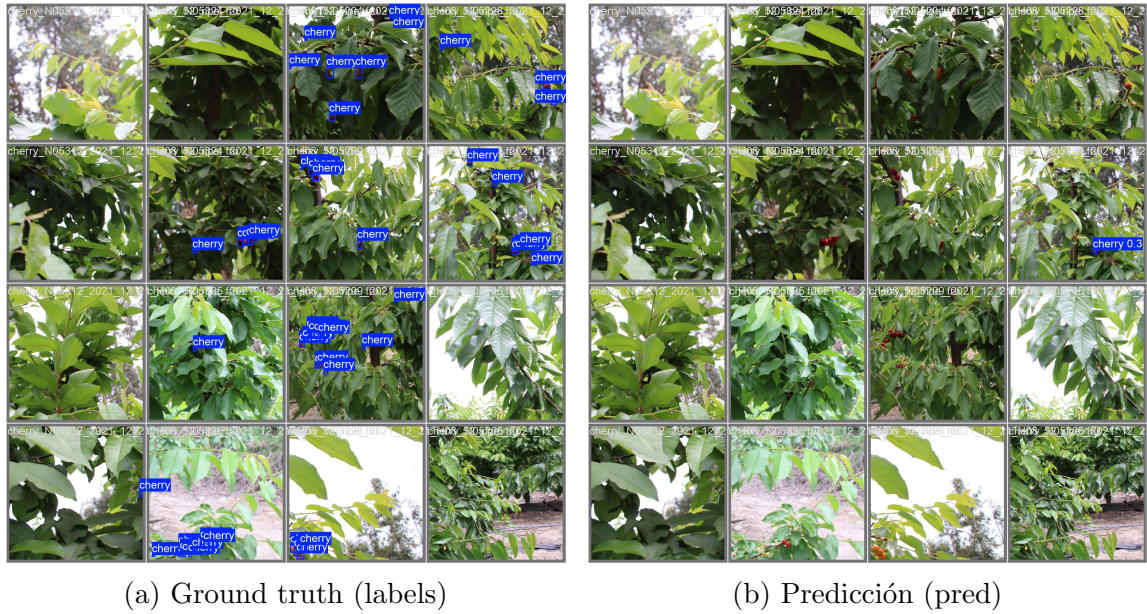


Figura 4.4: Comparación cualitativa del experimento E4 (Nocturno con linterna → Real diurno) sobre CherryCO test.

Conclusión parcial: cambios severos de iluminación pueden provocar especialización del detector y colapso de generalización, dominado por FN.

4.5.5. Conexión con RVO: Lectura operacional

Los patrones visuales observados son consistentes con el enfoque del RVO: gran parte de los errores se concentra en escenarios extremos (frutos pequeños/ocluídos o iluminación adversa). Reportar métricas globales y métricas en RVO permite mantener comparabilidad y a su vez representar mejor el rendimiento esperable en condiciones operacionales.

5. Conclusión

Esta memoria abordó el desarrollo de un entorno de simulación modular en Unity que integra un huerto y un robot (ambos virtuales) para captura de imágenes, con el objetivo de generar datos sintéticos anotados y evaluar detectores de cerezas en un escenario controlado. A partir de los experimentos realizados y el análisis de estos, se identificaron factores críticos que explican la brecha Sim2Real y se formalizó un Rango de Validez Operacional (RVO) para realizar comparaciones más justas y útiles para una futura aplicación robótica.

Los principales hallazgos de esta memoria son:

- Se validó un workflow completo de simulación → generación de datos → entrenamiento → evaluación Sim2Real.
- La principal degradación del detector se concentra en frutos pequeños/ocluídos; en tamaños operacionales el rendimiento se estabiliza.
- El RVO permite reportar métricas operacionales y comparar dominios de forma más justa.
- El fine-tuning con datos reales reduce significativamente la brecha Sim2Real respecto al entrenamiento sintético puro.

5.1. Conclusiones principales

A partir del desarrollo del entorno y de los experimentos realizados, esta memoria permite concluir que un flujo de trabajo basado en simulación puede usarse de manera efectiva para generar datos sintéticos anotados, entrenar detectores y medir la brecha

Sim2Real en un escenario controlado. El aporte central no es únicamente el rendimiento numérico de un modelo específico, sino la construcción y validación de una arquitectura modular reproducible: huerto virtual + robot capturador + generación automática de etiquetas + exportación a un formato compatible con entrenamiento y evaluación.

Los resultados indican que la forma de responder del detector es altamente dependiente de la geometría de captura (distancia y FOV) y de la magnitud de su fruto a analizar, encontrando en particular que el análisis por tamaños muestra que el bajo rendimiento global del dominio sintético se explica, en gran medida, por la alta proporción de frutos en rangos small (objetos muy pequeños o extremadamente ocluidos), convirtiendo así al dataset sintético en un entorno más complicado que el conjunto real.

En este contexto, la formalización del Rango de Validez Operacional (RVO) permite reportar métricas con mayor sentido para un uso robótico al restringir la evaluación a frutos con tamaño aparente suficiente, el desempeño se vuelve más estable y comparable entre dominios. Esto respalda la idea de que la comparación “global” puede ser injusta si los dominios no comparten distribuciones de tamaño similares, y que el RVO actúa como un criterio objetivo para evaluar la viabilidad práctica del sistema.

Acercas de la generalización, los experimentos de transferencia muestran que entrenar únicamente con sintético y verificar la capacidad del sistema con real lleva a una caída importante en recall, mostrando que la brecha Sim2Real es importante. No obstante, el experimento de fine-tuning con datos reales muestra que el preentrenado sintético puede ser un buen punto de partida, ya que complementando ese preentrenado con muestras reales el sistema recupera buena capacidad de detección del dominio objetivo. Finalmente el experimento con iluminación nocturna muestra que los cambios extremos de la condición visual tienden a especializar el detector y perjudicar su desempeño en condiciones diurnas, lo que refuerza la importancia de cubrir cierta variabilidad de la iluminación si la intención es conseguir robustez.

5.2. La plataforma como contribución reutilizable

Además de los resultados experimentales concretos que se muestran en relación con la detección de cerezas, la presente memoria forma una plataforma modular que

contribuye a la investigación y desarrollo en visión artificial agrícola. La principal contribución de este trabajo es la construcción de un entorno de simulación y de generación de datos sintéticos que permite controlar ciertas variables importantes del escenario, automatizar la anotación y ejecutar evaluaciones repetibles bajo el protocolo Sim2Real.

Componentes reutilizables

La plataforma implementa los siguientes componentes reutilizables:

- **Huerto virtual parametrizable:** permite modificar visuales del entorno y del cultivo de manera que sea posible generar diferentes escenarios controlados.
- **Robot capturador configurable:** implementa navegación y un sistema de captura basado en cámaras ajustable en resolución y en geometría de observación.
- **Pipeline de anotación automática:** genera etiquetas a partir de Unity Perception, de forma que el costo de etiquetado manual se reduce significativamente.
- **Protocolo de evaluación Sim2Real con RVO:** método para evaluar el desempeño en función de las condiciones del escenario del experimento, definiendo un Rango de Validez Operacional que permite definir las condiciones bajo las que un modelo de la acción controla el desempeño de forma aceptable.

Experimentos habilitados por la plataforma

El modularidad del sistema permite que futuros trabajos puedan ejecutar de manera directa experimentación controlada como:

- Estudios de sensibilidad a iluminación y condiciones de captura.
- Análisis del impacto de oclusión foliar, densidad de frutos y solapamiento en la detección.
- Comparación de arquitecturas/modelos de detección y configuraciones de entrenamiento.

- Optimización de estrategias de captura (distancias, FOV, resolución, trayectorias) para maximizar métricas de detección.

Extensiones en el corto plazo

Por último, la plataforma se puede ampliar sin que sea necesario volver a diseñar el pipeline completo:

- **Cambio de cultivo:** Alternar a otros frutos o bienes agrícolas a partir de la estructura huerto→captura→anotación→entrenamiento.
- **Nuevas modalidades de sensores:** como puede ser la profundidad o cualquier otro aspecto que nos permite Unity Perception, si es lo que pretendemos cuando queramos añadir tareas de percepción más complejas.
- **Variabilidad fenológica:** imágenes de estados como la maduración del fruto (cambio en tamaño, color) así como del tiempo más variado, para que la cobertura del dominio simulado sea más amplia.

5.3. Limitaciones de trabajo

Este estudio presenta limitaciones asociadas tanto al dominio real de evaluación como a las restricciones técnicas de la simulación y del entrenamiento.

En primer lugar, el dataset real utilizado como referencia (CherryCO) representa un escenario específico (huerto, época y condiciones particulares), por lo que las conclusiones sobre generalización deben interpretarse como validación preliminar y no como garantía de desempeño universal. En otras palabras, el análisis Sim2Real se realiza con un conjunto real acotado y no con múltiples huertos o configuraciones de captura.

En segundo lugar, el entorno de simulación se construyó bajo restricciones de rendimiento propias del motor. En particular, se contó con un modelo de árbol más complejo que no fue incorporado como activo principal debido al impacto en la tasa de cuadros y en la estabilidad de ejecución. De forma similar, aunque la intención inicial era aproximar un escenario de mayor escala (una hectárea completa), en la práctica el huerto simulado se implementó y evaluó principalmente sobre una configuración reducida (una hilera) para mantener tiempos de simulación razonables y asegurar

capturas consistentes. Esto implica que el comportamiento del sistema en escenas masivas (mayor densidad de árboles, más oclusión global y mayor carga de render) queda fuera del alcance de esta memoria.

En tercer lugar, el entrenamiento se centró en una configuración base con YOLOv8n, privilegiando viabilidad computacional y rapidez experimental. Esto limita el alcance de las conclusiones respecto al “mejor modelo posible”, ya que no se exploró en profundidad el espacio de arquitecturas más grandes, ni un ajuste exhaustivo de hiperparámetros o estrategias de entrenamiento. Además, aunque el robot cuenta con una configuración de cámaras preparada para capturas coordinadas, en esta memoria se trabajó principalmente con detección 2D mediante bounding boxes, alineado con el objetivo de comparación y validación preliminar del flujo de trabajo.

5.4. Trabajos futuros

A partir de los resultados obtenidos, se proponen las siguientes líneas de trabajo futuro para fortalecer la validez externa del sistema y mejorar el rendimiento Sim2Real:

- **Escalar el huerto virtual y mejorar fidelidad visual:** Incorporar modelos de árboles más complejos y mayor densidad de plantación (múltiples hileras y escenarios extensos), junto con estrategias de optimización que permitan mantener rendimiento sin sacrificar realismo. Esto permitiría estudiar cómo cambia la detección con oclusión más realista y mayor complejidad geométrica.
- **Explorar detectores de mayor capacidad y configuraciones más robustas:** Evaluar variantes más grandes o arquitecturas alternativas y estudiar de manera sistemática el efecto de resolución, aumentos de datos, umbrales de confianza y estrategias de entrenamiento orientadas a objetos pequeños. Dado que el tamaño aparente del fruto fue un factor crítico, este punto es especialmente relevante.
- **Extender la evaluación real y robustecer Sim2Real:** Validar el modelo y el pipeline en conjuntos reales adicionales (idealmente de distintos huertos, cámaras o épocas), para cuantificar mejor la generalización y evitar depender de un solo dominio real. También sería útil incluir pruebas en condiciones más variadas.

- **Mejorar el diseño del dataset sintético según el RVO:** Usar el RVO como criterio de diseño para controlar la distribución de tamaños del fruto en el sintético (por ejemplo, ajustar distancias y FOV para no sobrerrepresentar casos extremos), manteniendo al mismo tiempo una fracción controlada de ejemplos “stress test” para robustez.
- **Aprovechar la arquitectura para tareas adicionales:** Como la configuración del robot está preparada para capturas coordinadas, un paso natural es habilitar experimentos complementarios (por ejemplo, anotaciones adicionales o tareas de localización más avanzadas) sin modificar el núcleo del workflow. Esto ampliaría el uso del simulador como banco de pruebas.
- **Fenología y maduración procedural del huerto:** Una extensión lógica del entorno simulado sería integrar un modelo de fenología del cultivo que permita representar el crecimiento y desarrollo de la fruta en el tiempo, haciendo que las cerezas (o la fruta que se trabaje) puedan aparecer, crecer y madurar poco a poco, variando de forma controlada atributos como tamaño, color, brillo y textura, de forma que se obtenga una diversidad que sea consistente en las distintas etapas de madurez. Este camino permitiría también la generación de datasets orientados no sólo a detección, sino también a tareas de clasificación por estado de maduración, manteniendo el workflow ya desarrollado.
- **Condiciones meteorológicas y estacionales más realistas:** También se considera el conjunto de condiciones atmosféricas que tienen que ver con fenómenos como lluvia, nieve, variaciones de humedad y fenómenos que surgen como los charcos, el barro, las gotas que caen sobre la cámara, sobre la lente, cambios en la reflectancia de las superficies; pero además, un sistema estacional más detallado podría modificar la densidad y el color de las hojas, la presencia de flores o la caída de hojas; podría incluir además variaciones en la iluminación general, en la nubosidad, nieblas, etc., ampliando aún más la gama de escenarios que cubre el dataset sintético.

5.5. Conclusión final

Esta memoria demuestra que es factible construir en Unity un entorno modular que integra simulación, captura y anotación automática para soportar entrenamiento

y evaluación de detectores de cerezas bajo un enfoque Sim2Real. El análisis experimental permitió establecer la existencia de que la brecha de dominio no depende solo del “realismo” global de la escena, sino que, de hecho, está altamente mediatizada por las características del entorno operativo, tales como el tamaño aparente del fruto, la oclusión, iluminación e incluso el tipo y potencia del modelo a utilizar. En este sentido, la delimitación del RVO se convierte en un criterio práctico y reproducible para reportar resultados y comparar dominios de una forma equilibrada y justa. A pesar de la existencia de limitantes, tanto de escala como de validación, el workflow puesto en práctica y los hallazgos son una buena base para extender el sistema en dirección a situaciones más complejas y parecidas a una aplicación robótica real.

Bibliografía

- [1] J Edward Taylor, Diane Charlton, and Antonio Yúnez-Naude. The end of farm labor abundance. *Applied Economic Perspectives and Policy*, 34(4):587–598, 2012.
- [2] Feng Xiao, Haibin Wang, Yueqin Xu, and Ruiqing Zhang. Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review. *Agronomy*, 13(6):1625, 2023.
- [3] C Beldek, J Cunningham, M Aydin, E Sariyildiz, SL Phung, and G Alici. Sensing-based robustness challenges in agricultural robotic harvesting. In *2025 IEEE International Conference on Mechatronics (ICM)*, pages 1–6. IEEE, 2025.
- [4] Omeed Mirbod, Daeun Choi, and John K Schueller. From simulation to field validation: A digital twin-driven sim2real transfer approach for strawberry fruit detection and sizing. *AgriEngineering*, 7(3):81, 2025.
- [5] Juan Fernando Villacrés and Fernando Auat Cheein. Detection and characterization of cherries: A deep learning usability case study in chile. *Agronomy*, 10(6):835, 2020.
- [6] Chenglin Wang, Suchwen Liu, Lijun Zhao, and Tianhong Luo. Virtual simulation of fruit picking robot based on unity3d. In *Journal of Physics: Conference Series*, volume 1631, page 012033. IOP Publishing, 2020.
- [7] Hongkun Tian, Tianhai Wang, Yadong Liu, Xi Qiao, and Yanzhou Li. Computer vision technology in agricultural automation—a review. *Information processing in agriculture*, 7(1):1–19, 2020.
- [8] Sumaira Ghazal, Arslan Munir, and Waqar S Qureshi. Computer vision in smart agriculture and precision farming: Techniques and applications. *Artificial Intelligence in Agriculture*, 2024.

- [9] P Muthulakshmi and T Tamilarasi. Computer vision in agritech: A review. 2022.
- [10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [11] Mohammad El Sakka, Mihai Ivanovici, Lotfi Chaari, and Josiane Mothe. A review of cnn applications in smart agriculture using multimodal data. *Sensors*, 25(2):472, 2025.
- [12] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 5(4):1680–1716, 2023.
- [13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.
- [15] Maryam Rahnemoonfar and Clay Sheppard. Deep count: fruit counting based on deep simulated learning. *Sensors*, 17(4):905, 2017.
- [16] Na Ma, Yile Sun, Chenfei Li, Zonglin Liu, and Haiyan Song. Ahg-yolo: multi-category detection for occluded pear fruits in complex orchard scenes. *Frontiers in Plant Science*, 16:1580325, 2025.
- [17] Omeed Mirbod, Daeun Choi, Paul H Heinemann, Richard P Marini, and Long He. On-tree apple fruit size estimation using stereo vision with deep learning-based occlusion handling. *Biosystems Engineering*, 226:27–42, 2023.
- [18] Sergey I Nikolenko et al. *Synthetic data for deep learning*, volume 174. Springer, 2021.
- [19] Manya Afonso and Valerio Giufrida. Synthetic data for computer vision in agriculture. *Frontiers in Plant Science*, 14:1277073, 2023.

- [20] Leonardo Saraceni, Ionut M Matoi, Daniele Nardi, and Thomas A Ciarfuglia. Self-supervised data generation for precision agriculture: Blending simulated environments with real imagery. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 71–77. IEEE, 2024.
- [21] Zhenghao Fei, Alexander G Olenskyj, Brian N Bailey, and Mason Earles. Enlisting 3d crop models and gans for more data efficient and generalizable fruit detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1269–1277, 2021.
- [22] Mikolaj Cieslak, Umabharathi Govindarajan, Alejandro Garcia, Anuradha Chandrashekar, Torsten Hadrich, Aleksander Mendoza-Drosik, Dominik L Michels, Soren Pirk, Chia-Chun Fu, and Wojciech Palubicki. Generating diverse agricultural data for vision-based farming applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5422–5431, 2024.
- [23] Qingyu Wang, Dihua Wu, Wei Liu, Mingzhao Lou, Huanyu Jiang, Yibin Ying, and Mingchuan Zhou. Plantstereo: A high quality stereo matching dataset for plant reconstruction. *Agriculture*, 13(2):330, 2023.
- [24] Gabriel Coll-Ribes, Iván J Torres-Rodríguez, Antoni Grau, Edmundo Guerra, and Alberto Sanfeliu. Accurate detection and depth estimation of table grapes and peduncles for robot harvesting, combining monocular depth estimation and cnn methods. *Computers and Electronics in Agriculture*, 215:108362, 2023.
- [25] Wan-Soo Kim, Dae-Hyun Lee, Yong-Joo Kim, Taehyeong Kim, Won-Suk Lee, and Chang-Hyun Choi. Stereo-vision-based crop height estimation for agricultural robots. *Computers and Electronics in Agriculture*, 181:105937, 2021.
- [26] Md Rejaul Karim, Shahriar Ahmed, Md Nasim Reza, Kyu-Ho Lee, H Jin, Mohammod Ali, Sun-Ok Chung, and Joonjea Sung. A review on stereo vision for feature characterization of upland crops and orchard fruit trees. *Precis. Agric*, 6:104–122, 2024.
- [27] Li Zhang, Qun Hao, Yefei Mao, Jianbin Su, and Jie Cao. Beyond trade-off: An optimized binocular stereo vision based depth estimation algorithm for designing harvesting robot in orchards. *Agriculture*, 13(6):1117, 2023.

- [28] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.
- [29] José Naranjo-Torres, Marco Mora, Ruber Hernández-García, Ricardo J Barrientos, Claudio Fredes, and Andres Valenzuela. A review of convolutional neural network applied to fruit image processing. *Applied Sciences*, 10(10):3443, 2020.
- [30] Hamza Noueihed, Heba Harb, and Joe Tekli. Simulating weather events on a real-world map using unity 3d. In *SMARTGREENS*, pages 86–93, 2022.
- [31] Hamza Noueihed, Heba Harb, and Joe Tekli. Knowledge-based virtual outdoor weather event simulator using unity 3d. *The Journal of Supercomputing*, 78(8):10620–10655, 2022.
- [32] Unity Technologies. Lighting strategy. <https://docs.unity3d.com/2019.3/Documentation/Manual/BestPracticeMakingBelievableVisuals3.html>, 2019. Accedido en junio de 2025.
- [33] Unity Technologies. Lighting window reference. <https://docs.unity3d.com/Manual/lighting-window.html>, 2023. Accedido en junio de 2025.
- [34] Unity Technologies. Lighting overview. <https://docs.unity3d.com/Manual/LightingOverview.html>, 2023. Accedido en junio de 2025.
- [35] Arturo Haces-Garcia and Weihang Zhu. Building an accessible and flexible multi-user robotic simulation framework with unity-matlab bridge. *Computers*, 13(11):282, 2024.
- [36] Anna Konrad. Simulation of mobile robots with unity and ros: A case-study and a comparison with gazebo, 2019.
- [37] Unity Technologies. Navmesh. <https://docs.unity3d.com/ScriptReference/AI.NavMesh.html>, 2023. Accedido en junio de 2025.
- [38] Unity Technologies. Unity Perception package. <https://github.com/Unity-Technologies/com.unity.perception>, 2020.
- [39] Steve Borkman, Adam Crespi, Saurav Dhakad, Sujoy Ganguly, Jonathan Hogins, You-Cyuan Jhang, Mohsen Kamalzadeh, Bowen Li, Steven Leal, Pete Parisi, et al.

Unity perception: generate synthetic data for computer vision. *arXiv preprint arXiv:2107.04259*, 2021.

- [40] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.
- [41] Luis Cossio-Montefinale, Javier Ruiz-del Solar, and Rodrigo Verschae. Cherry co dataset: a dataset for cherry detection, segmentation and maturity recognition. *IEEE Robotics and Automation Letters*, 2024.
- [42] Rafael Padilla, Wesley L Passos, Thadeu LB Dias, Sergio L Netto, and Eduardo AB Da Silva. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, 10(3):279, 2021.

A. Diagramas

A.1. Diagrama de flujo de trabajo

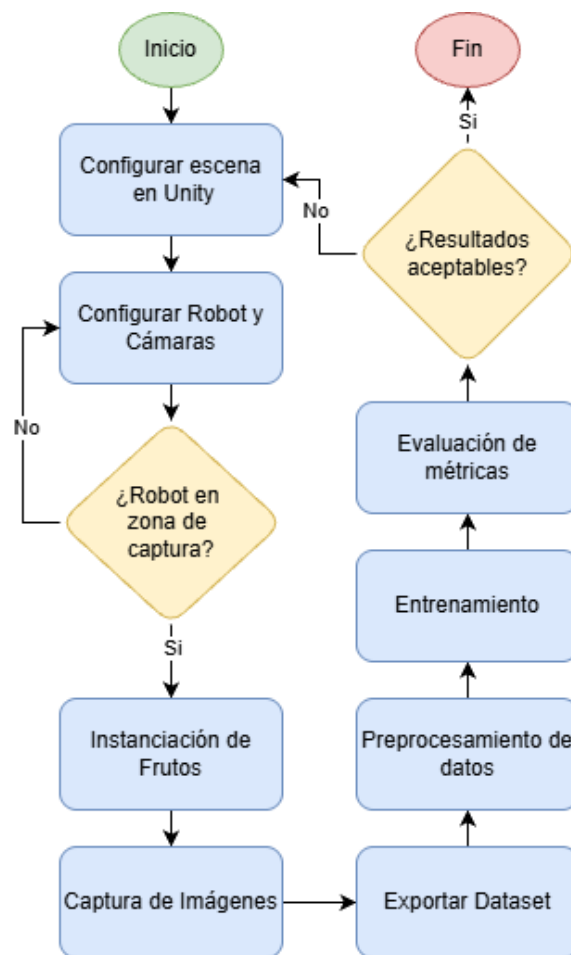


Figura A.1: Diagrama de flujo de trabajo.

A.2. Diagrama de arquitectura del Sistema

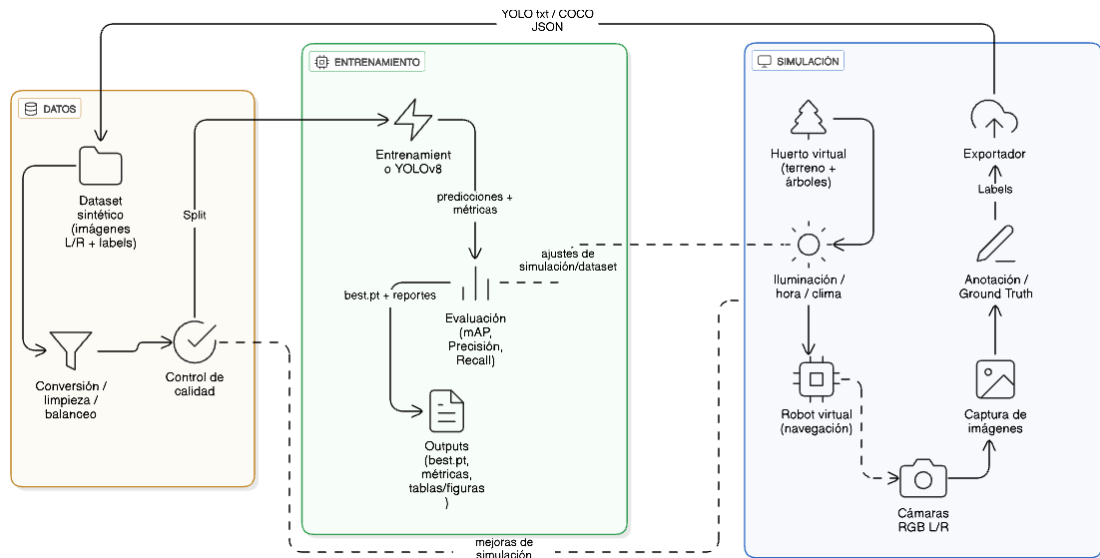


Figura A.2: Diagrama de arquitectura del sistema.

B. Diseño del Sistema

B.1. Elementos del Huerto Virtual

B.1.1. Terreno

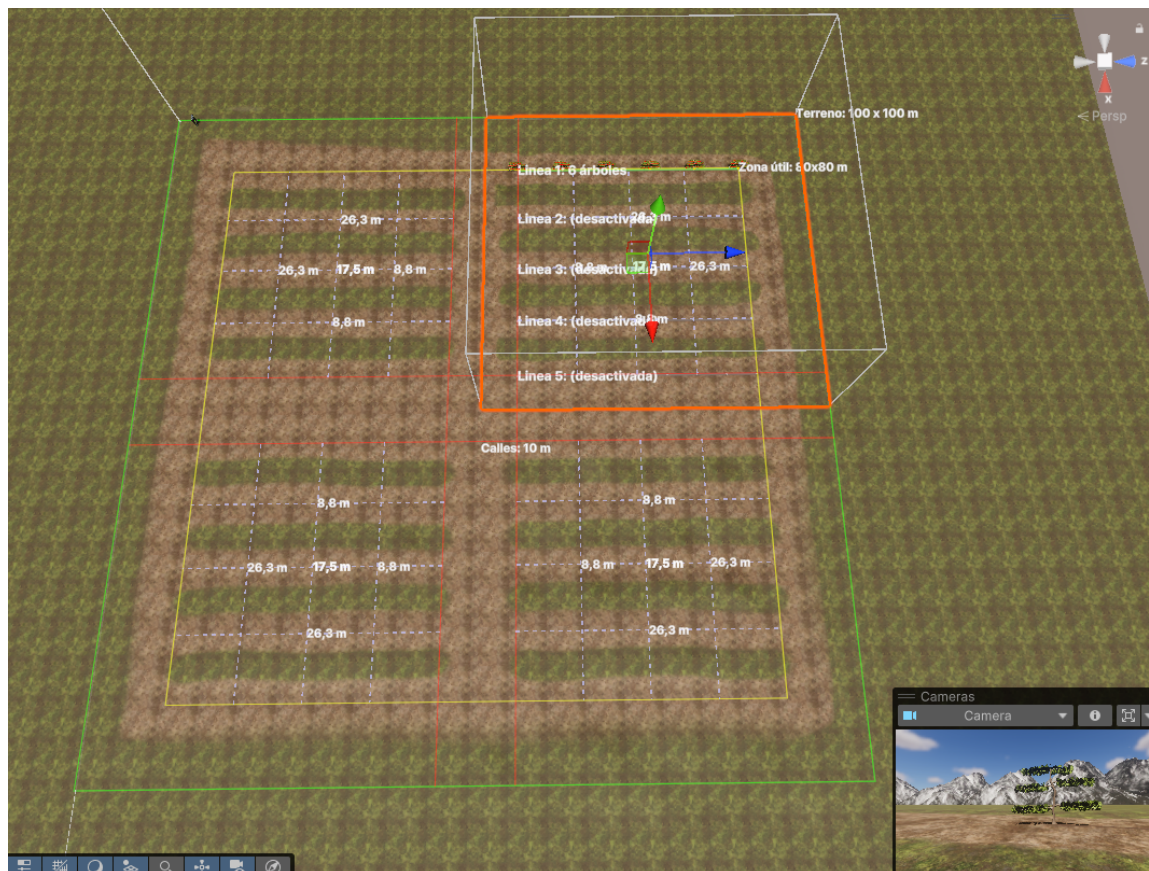


Figura B.1: Distribución del terreno.



Figura B.2: Vista en sitio del terreno.

B.1.2. Árboles y Cerezas

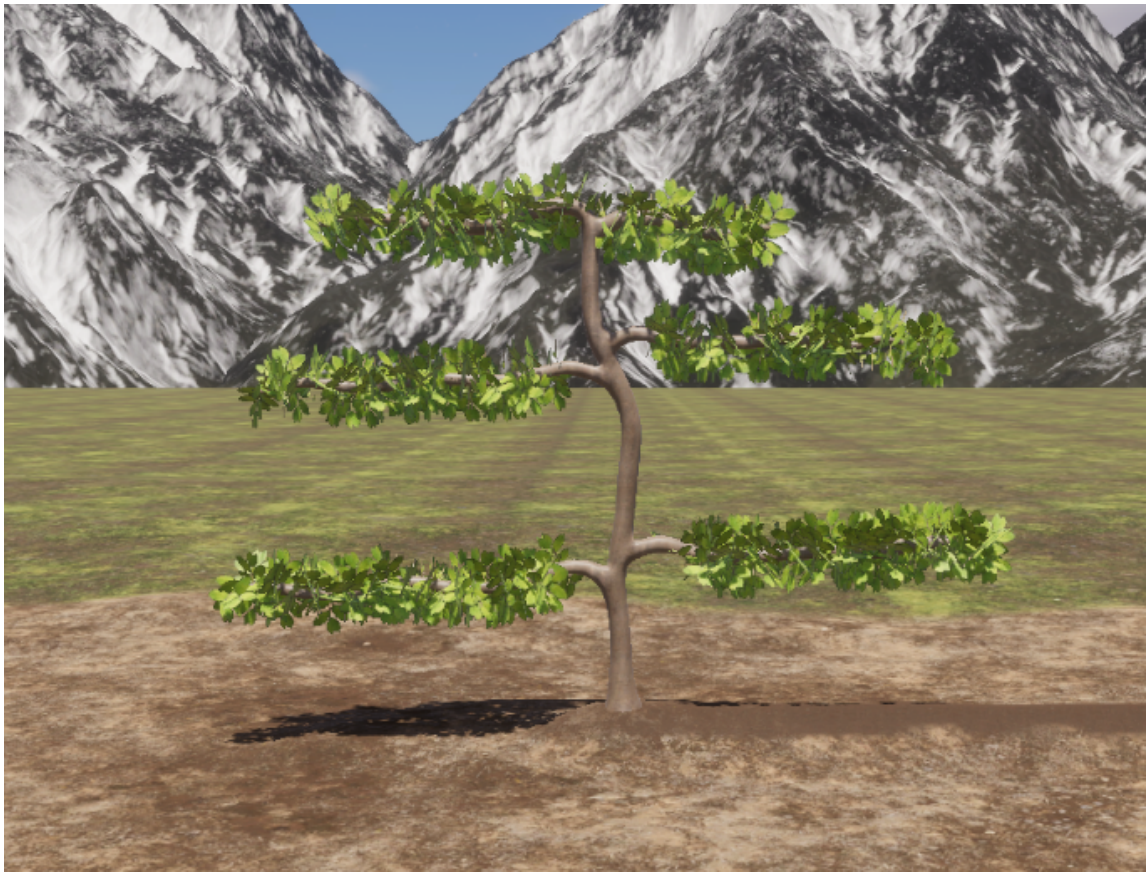


Figura B.3: Árbol en conducción espaldera simple.



Figura B.4: Árbol en conducción espaldera complejo.



Figura B.5: Modelo de las Cerezas.

B.1.3. Iluminación Dinámica



Figura B.6: Horario am.



Figura B.7: Horario pm.



Figura B.8: Iluminacion artificial (linternas del robot).

B.1.4. Condiciones Climáticas

Nubes



Figura B.9: Clima Despejado.



Figura B.10: Clima Nuboso.

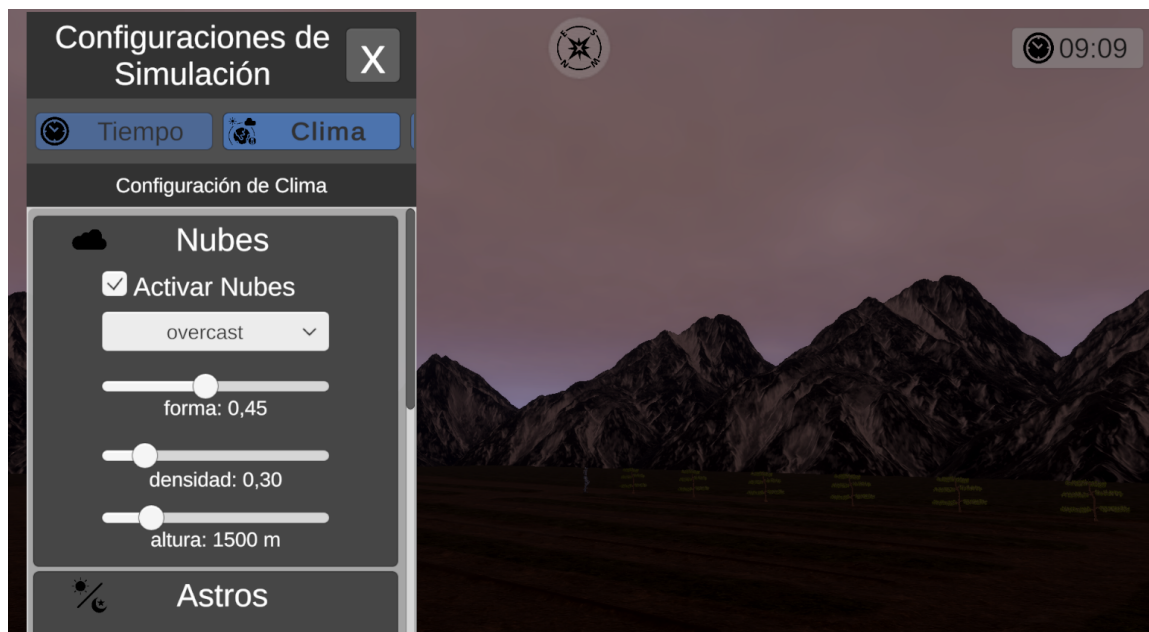


Figura B.11: Clima cielo nublado.



Figura B.12: Clima tormentoso.



Figura B.13: Nubes personalizadas.

Astros

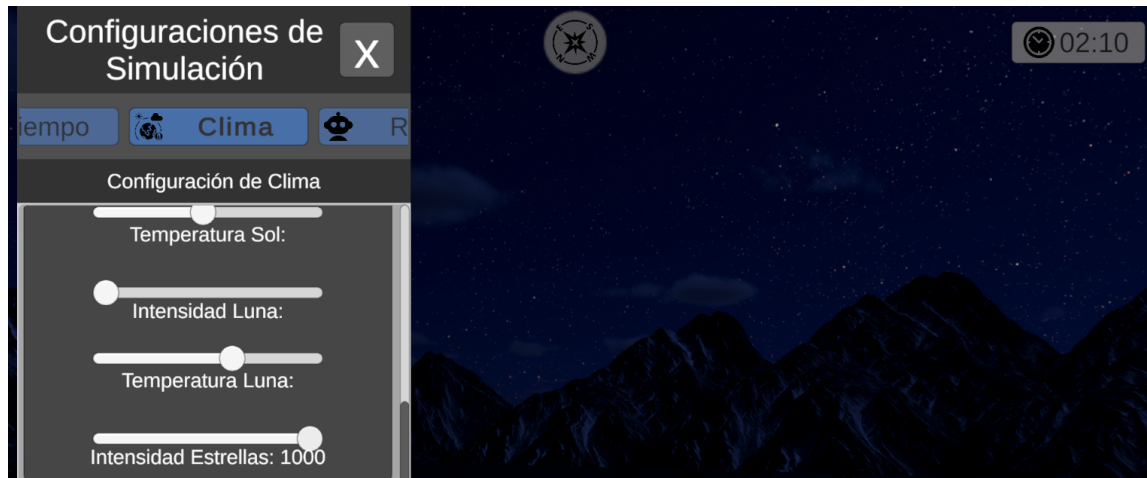


Figura B.14: Configuración de intensidad de las Estrellas.



Figura B.15: Condición no nominal del entorno bajo una configuración extrema de iluminación: se modificaron deliberadamente la temperatura de color e intensidades del Sol.

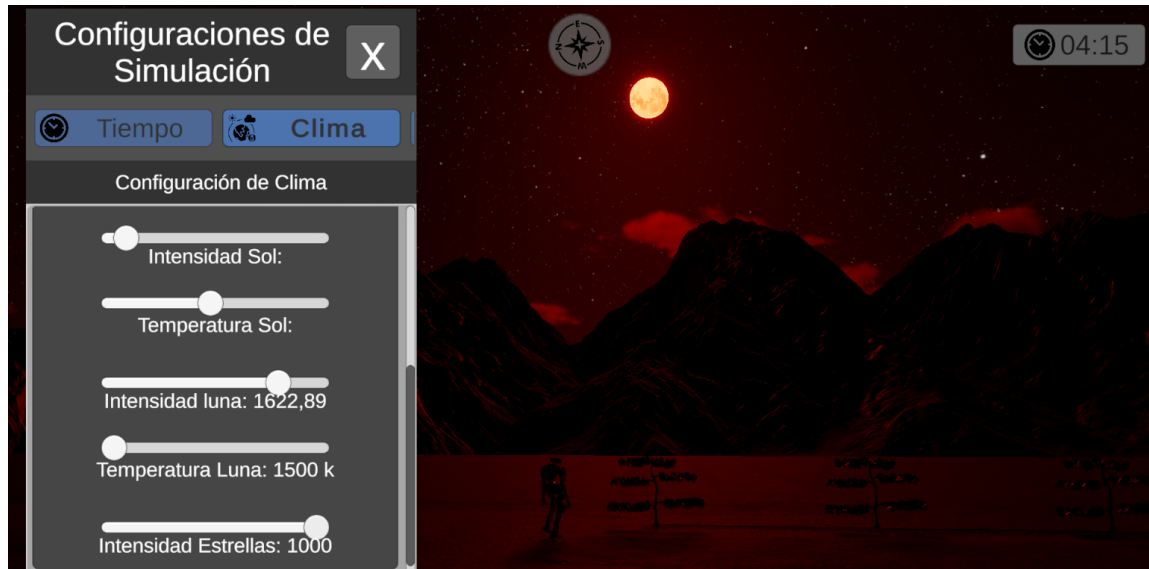


Figura B.16: Condición no nominal del entorno bajo una configuración extrema de iluminación: se modificaron deliberadamente la temperatura de color e intensidades de la Luna.

Estas capturas se incluyen como *prueba de estrés* para evidenciar los límites de parametrización visual del simulador (intensidad de iluminación y temperaturas de los astros), por lo que **no representan** condiciones realistas de operación o captura.

Niebla



Figura B.17: Niebla moderada.



Figura B.18: Niebla suave.



Figura B.19: Niebla fuerte.

B.1.5. Control de Condiciones



Figura B.20: Controles de Tiempo.



Figura B.21: Configuración de Nubes.



Figura B.22: Configuración de Astros.



Figura B.23: Configuración de la Niebla.

B.1.6. Diseño Modular y extensible

Configuración de Distribución del Huerto

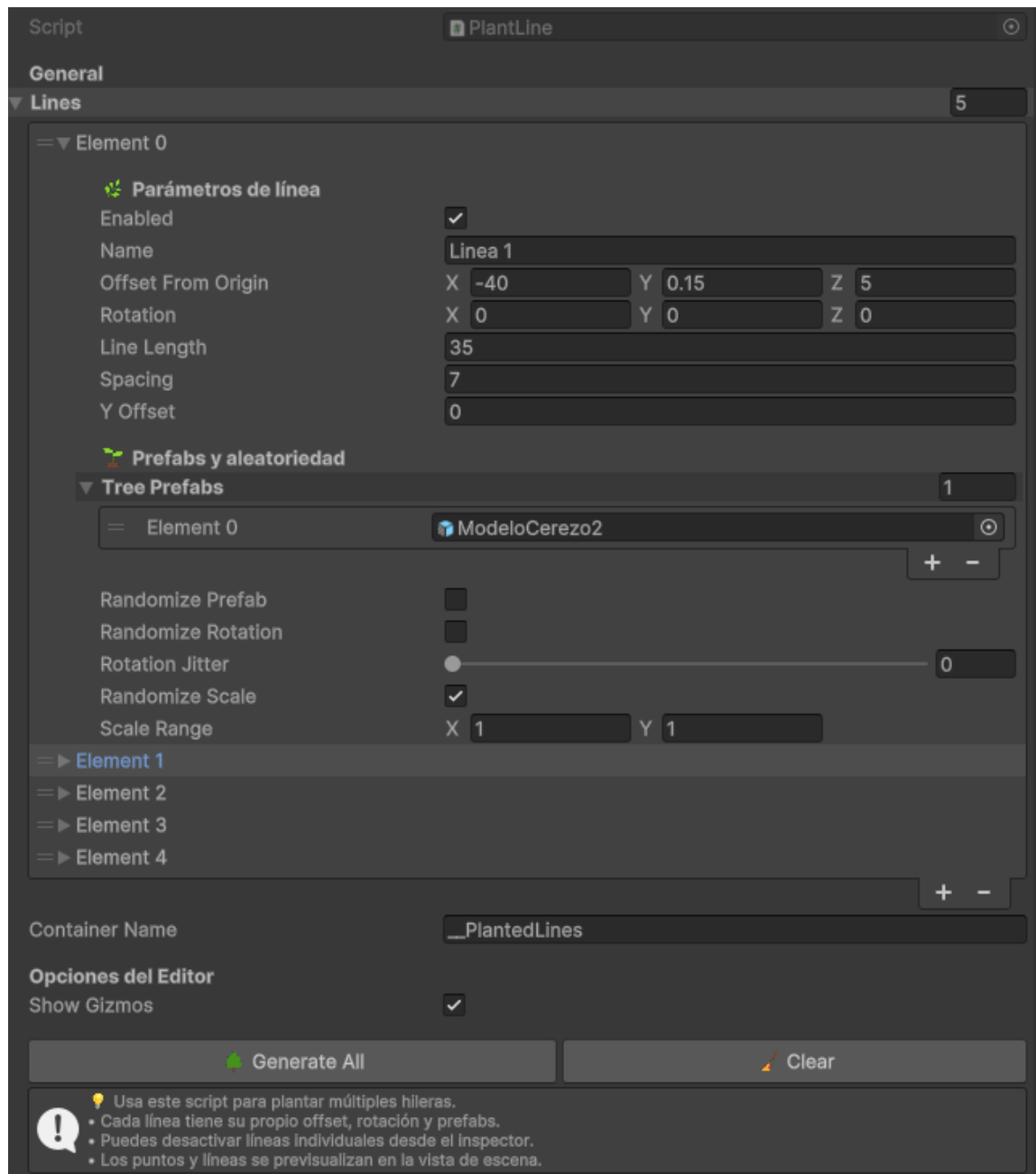


Figura B.24: Configuración de distribución y generación de árboles.

Configuración de Generación de Cerezas

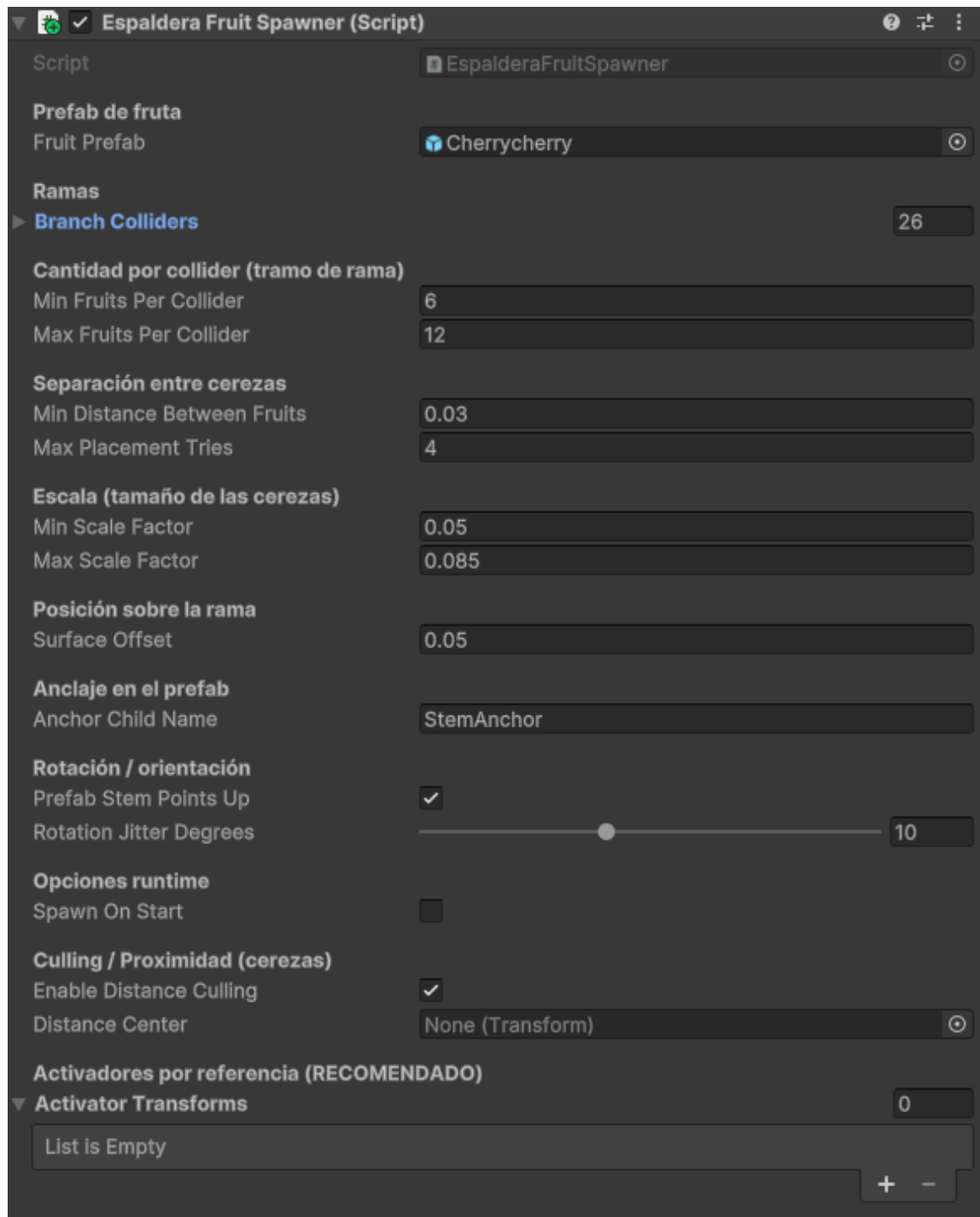


Figura B.25: Configuración del Spawn de cerezas.

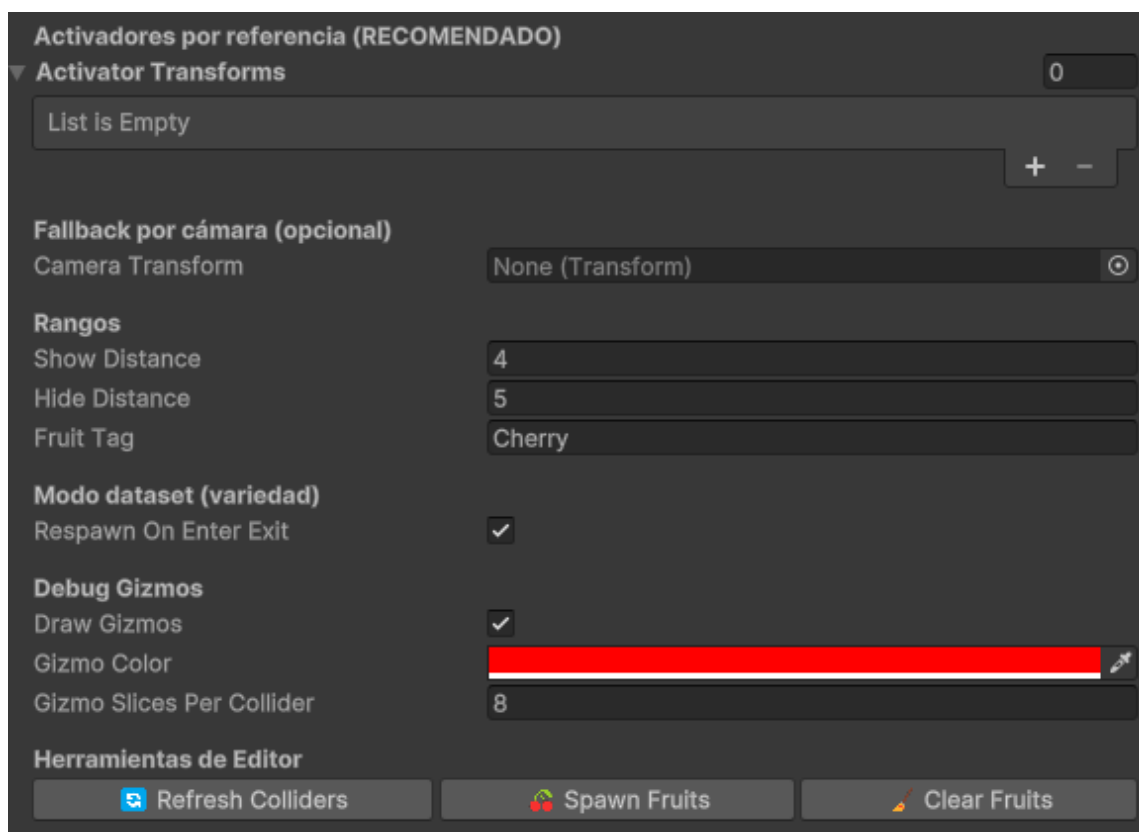


Figura B.26: Continuacion del Spawner de cerezas.

B.2. Diseño del Robot Virtual

B.2.1. Robot



Figura B.27: Modelo del Robot.



Figura B.28: Controles del Robot.



Figura B.29: Previsualización de cámaras del Robot.

B.2.2. Configuraciones del Robot



Script: AutoNavRobotEspaldera

Movimiento base

- Cruise Speed: 1.6
- Acceleration: 6
- Angular Look Speed: 8
- Stop Distance: 0.55
- Wait Time After Tree: 0.5
- Loop: ☒

Espaldera / Ruta

- Lane Offset: 1.2
- Row Tolerance: 0.8
- Turn Forward Extra: 1.5
- Go And Return Same Row: ☒

Distancia al Árbol (base + random por árbol)

- Photo Distance: 1.5
- Randomize Photo Distance: ☒
- Photo Distance Range: X 1 Y 1.8

Fotos por árbol (base + random por árbol)

- Photos Per Tree: 6
- Randomize Photos Per Tree: ☒
- Photos Per Tree Range: X 4 Y 8

Barrido / Sweep (base + random por árbol)

- Sweep Span: 4
- Randomize Sweep Span: ☒
- Sweep Span Range: X 1 Y 4
- Sweep Speed: 0.55
- Continuous Sweep: ☐

Variación realista por foto (rotación)

- Yaw Jitter: 2
- Pitch Jitter: 2
- Roll Jitter: 1.5
- Shot Delay: 0.08

Altura de cámaras (variación por foto)

- Randomize Camera Height Per Shot: ☒
- Camera Height Offset Range: X -60 Y 0

Zoom / Distancia extra por foto (Dolly)

- Randomize Dolly Per Shot: ☒
- Dolly Offset Range: X 0 Y 40

Figura B.30: Configuraciones del Robot.

Altura de cámaras (variación por foto)
Randomize Camera Height Per Shot ☒
Camera Height Offset Range X Y

Zoom / Distancia extra por foto (Dolly)
Randomize Dolly Per Shot ☒
Dolly Offset Range X Y

Zoom óptico (FOV) por foto (opcional)
Randomize Fov Per Shot ☒
Fov Range X Y

CAPTURE FIX
Pause Agent During Capture ☒
Settle Frames Before Capture
Settle Seconds Before Capture
Instant Look At For Shot ☒

Cámaras Perception
Left Perception Camera
Right Perception Camera
Capture Side Only ☒

Perception runtime
Keep Perception Cameras Always On ☐
Perception Warmup Frames

Luces / Linternas (seguir cámaras)
Sync Lights To Cameras ☒
▼ Left Lights To Follow
= Element 0
+ -

▼ Right Lights To Follow
= Element 0
+ -

Sync Lights Using Local If Possible ☒

Figura B.31: Continuacion de configuraciones del Robot.

B.2.3. Unity Perception

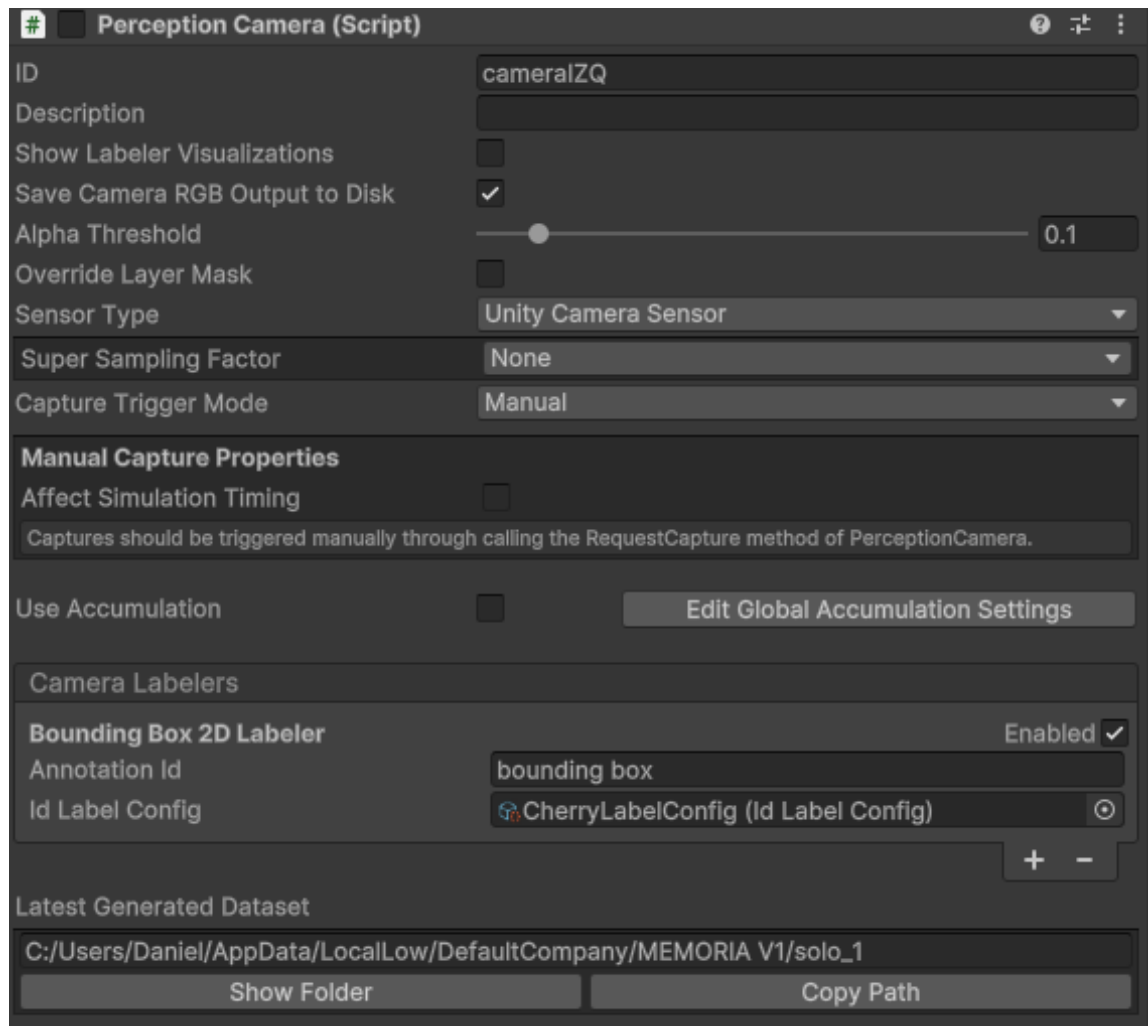


Figura B.32: Configuración de la cámara.

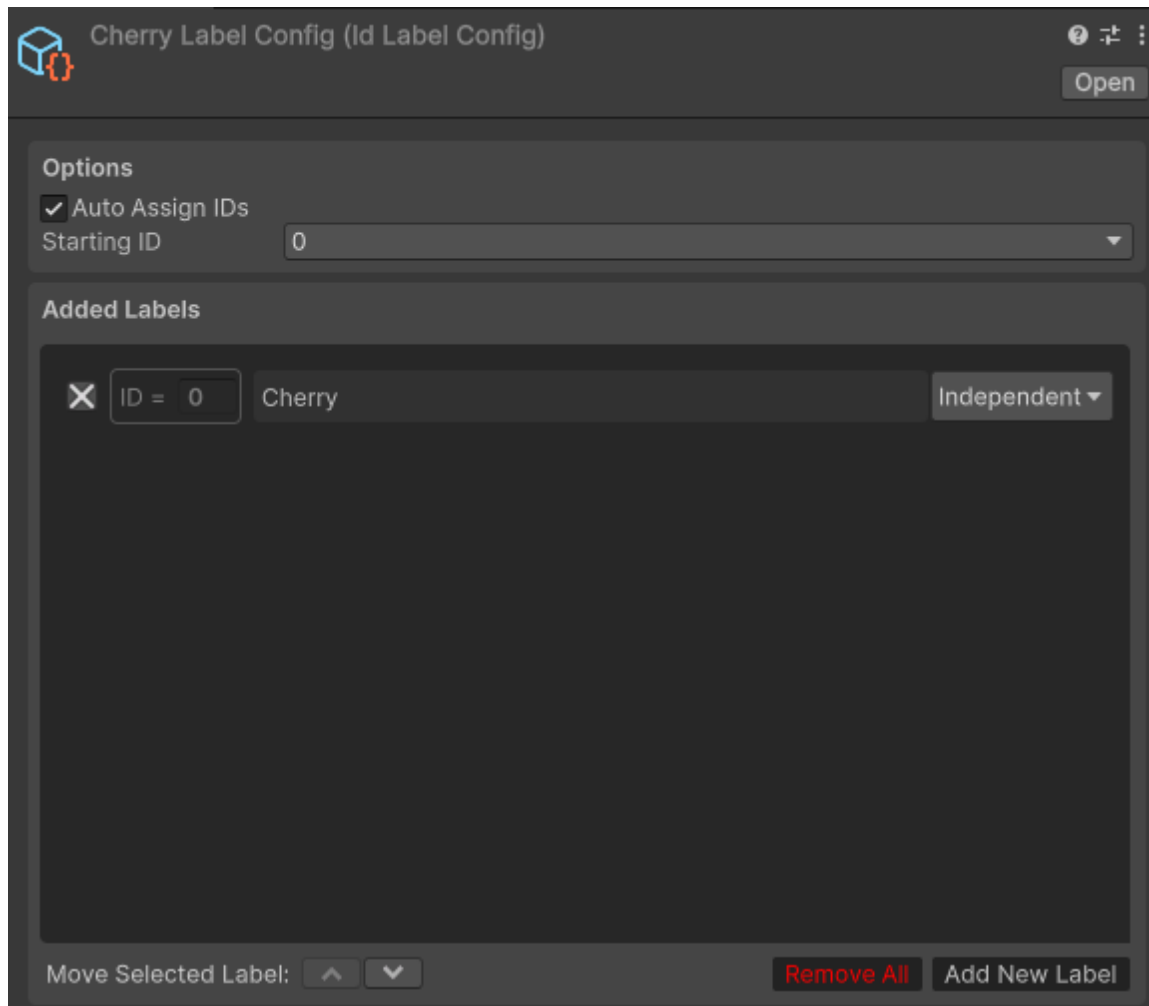


Figura B.33: Configuración de la etiqueta Cherry.

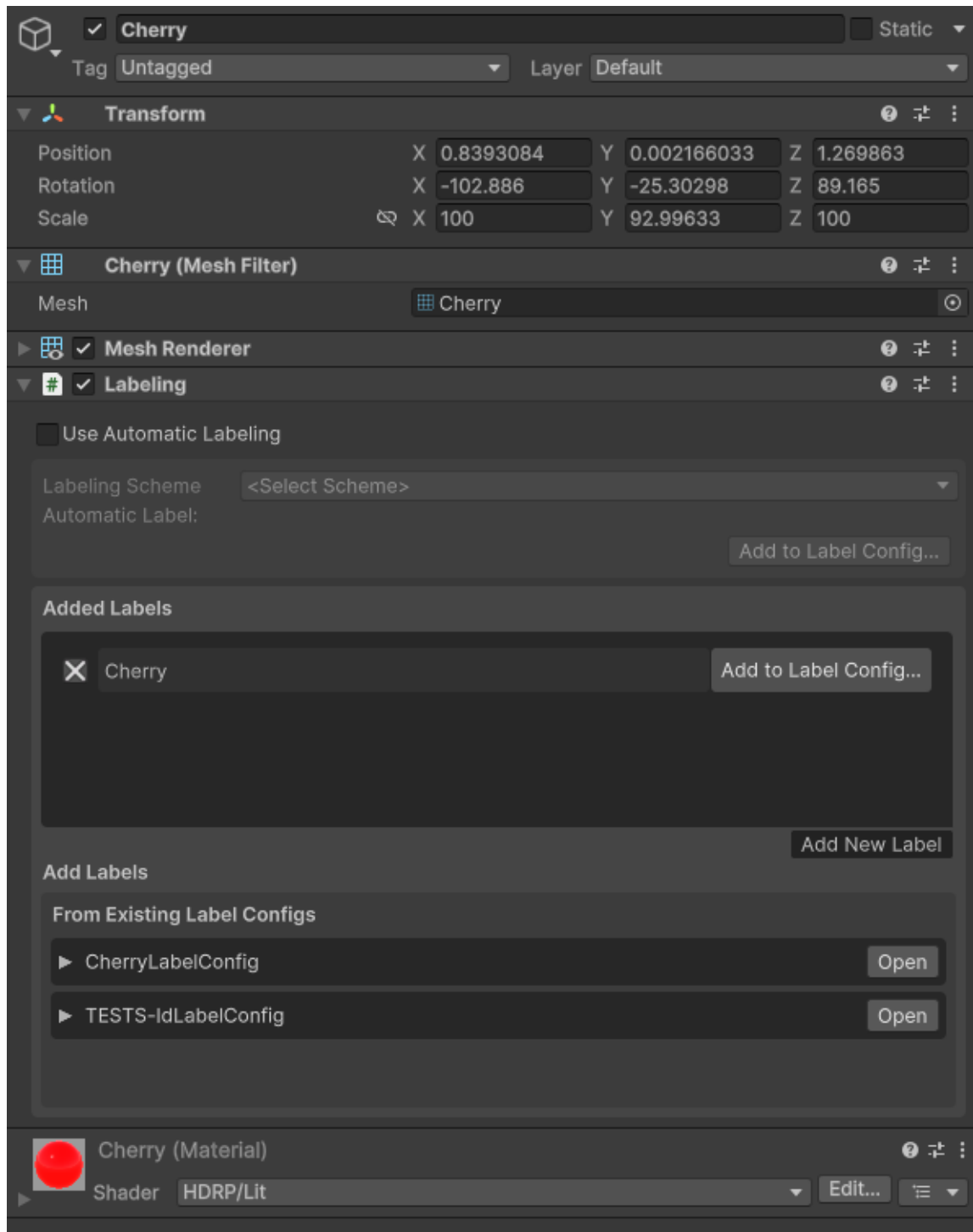


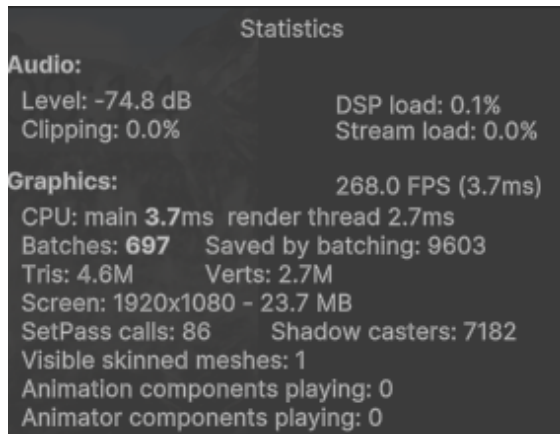
Figura B.34: Asignación de etiqueta al modelo de la Cereza.

C. Rendimiento computacional

C.1. Simulacion

Escenario	FPS	ms/frame
Base (huerto a distancia, sin cerezas activas)	≈ 260	$\approx 3,5$
Cercano (árbol con cerezas activas)	≈ 15	≈ 80
Carga alta (huerto/filas adicionales)	≈ 20	≈ 50

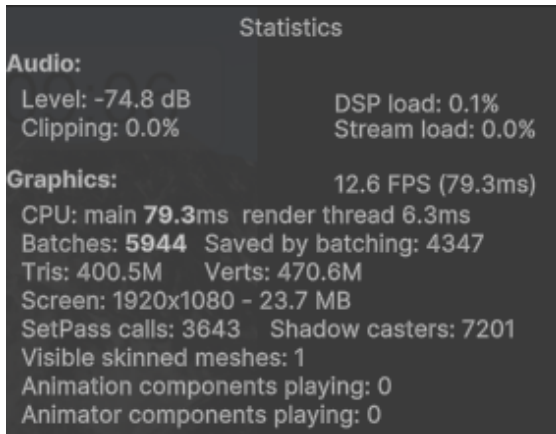
Tabla C.1: Medición observacional de rendimiento usando *Stats* de Unity en escenarios representativos.



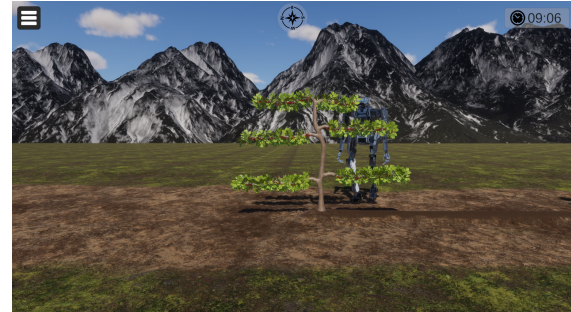
(a) Stats en condición base: ≈ 260 FPS.



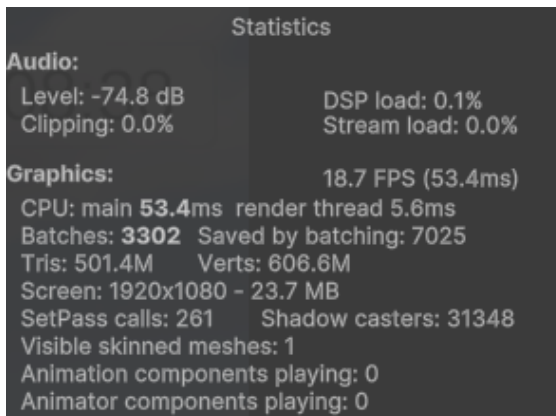
(b) Vista del huerto en condición base.



(a) Stats cercano: ≈ 15 FPS (alta carga geométrica).



(b) Robot cercano a un árbol con cerezas.



(a) Stats con carga alta: ≈ 20 FPS.



(b) Escena con mayor densidad de huer-to/filas.

El rendimiento del simulador se evaluó mediante observación del panel de Stats de Unity, obteniendo unos ≈ 260 FPS como base de la evaluación, bajo la condición de hilera de 6 árboles, sin cerezas activas, y el robot a distancia. Activar el acercamiento y la generación/visualización de las cerezas hizo caer el rendimiento a ≈ 15 FPS, algo muy perjudicial para estabilizar las tomas. La disminución del rendimiento se relaciona con el aumento de la carga de render, cambio de ≈ 4.6 M triángulos en la condición base a cerca de ≈ 400 – 500 M triángulos con alto detalle, también se vio afectada la cantidad de vértices y de batches. Esta disminución de los precio se tuvo que compensar con modelos de simplificación y estrategias de instanciación controlada, buscando la mayor impresión de realismo sin que el rendimiento se viese muy perjudicado.

C.2. Entrenamiento

Experimento	Epochs	Tiempo (h)	Tiempo (min)	Prom. por epoch (s)
E0	30	0.092	5.52	11.04
E1	30	0.178	10.68	21.36
E1a	30	0.259	15.55	15.55
E1b	30	0.217	13.02	26.04
E1c	30	0.175	10.50	21.00
E1d	30	0.367	22.02	44.04
E3	20	0.056	3.36	10.08
E4	30	0.125	7.50	15.00

Tabla C.2: Tiempos de entrenamiento por experimento (según logs de Ultralytics).

En la Tabla C.2 se reportan los tiempos de entrenamiento por experimento (según los logs de Ultralytics) y el promedio de segundos por época. Con el fin de mantener comparabilidad entre configuraciones, se definió un **presupuesto base de 30 épocas** para los entrenamientos desde cero (E0, E1, E4). Este valor se utilizó como un compromiso práctico entre permitir que el entrenamiento alcance una fase estable de aprendizaje y controlar el costo computacional total, manteniendo además consistencia metodológica entre corridas.

E0 y E1 corresponden a entrenamientos base independientes. En cambio, E1a y E3 no se entrenan desde cero: ambos continúan el entrenamiento del modelo E1, iniciando desde su mejor *checkpoint*. Por ello, E1a (30 épocas) y E3 (20 épocas) representan **épocas adicionales** sobre E1, usadas para extender el aprendizaje bajo una configuración específica (E1a) y ajustar el modelo hacia el dominio real con un refinamiento más acotado (E3). En particular, en E3 se utilizaron 20 épocas adicionales para priorizar un ajuste controlado hacia datos reales sin incrementar innecesariamente el tiempo total de entrenamiento, dado que parte del aprendizaje ya está incorporado en el modelo inicial (E1).

D. Recursos externos

Este anexo reúne los recursos externos utilizados en el proyecto.

D.1. Disponibilidad de código, notebooks y resultados

D.1.1. Repositorio del proyecto Unity

- Repositorio: https://github.com/das3110/Das_Memoria.git

D.1.2. Notebooks de experimentación (Google Colab)

- Carpeta/Drive: <https://drive.google.com/drive/folders/1p6B4ew7cCe6eLDmh4uqhYA1usp=sharing>

D.1.3. Resultados y datos generados

- Resultados (Drive): <https://drive.google.com/drive/folders/1p6B4ew7cCe6eLDmh4uqhusp=sharing>

D.2. Paquete de materiales y entorno (Unity Asset Store)

Para la construcción visual del entorno (materiales/superficies y elementos del paisaje), se utilizó el paquete:

- *HDRP Surfaces Starter Pack* (Unity Asset Store):
<https://assetstore.unity.com/packages/2d/textures-materials/hdrp-surfaces-starter-pack/content>

El uso de este contenido se rige por los términos del *Unity Asset Store Terms of Service and EULA*: <https://unity.com/legal/as-terms>

D.3. Modelo 3D del robot (TurboSquid)

El modelo 3D del robot empleado en la simulación fue obtenido desde TurboSquid:

- *Robot Free 3D Model* (TurboSquid):
<https://www.turbosquid.com/3d-models/robot-free-3d-model-2284745>

El uso de este modelo se encuentra sujeto a la licencia de TurboSquid (*TurboSquid 3D Model License*): <https://blog.turbosquid.com/turbosquid-3d-model-license/>

D.4. Atribuciones de íconos

- Grabadora de cámara: “Grabadora de cámara iconos creados por laterunlabs - Flaticon”.
- Cambio climático: “Cambio climático iconos creados por Iconjam - Flaticon”.
- Día y noche: “Día y noche iconos creados por Md Tanvirul Haque - Flaticon”.
- Linterna: “Linterna iconos creados por Freepik - Flaticon”.
- Nubes: “Nubes iconos creados por zky.icon - Flaticon”.
- Temporada: “Temporada iconos creados por Freepik - Flaticon”.
- Robot: “Robot icons created by MetamorphLab - Flaticon”.
- Person walking: “Person walking icons created by Romica - Flaticon”.