



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Colaboración humano-robot en paletizado: Diseño de  
escenario virtual y físico centrado en el humano

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Ojeda Gandia, Raül

Tutor/a: Fons Cors, Joan Josep

Director/a Experimental: Mestre Gascón, Antoni

CURSO ACADÉMICO: 2023/2024



# Agradecimientos

---

Me gustaría agradecer a todas las personas e instituciones que, de manera directa o indirecta, han contribuido en la elaboración de este TFG.

En primer lugar, a mis tutores Antoni Mestre y Joan Fons por su gran ayuda, guía e ideas que han construido el presente trabajo.

A mis tutores de prácticas, Vicente Pelechano y Manoli Albert, por su visión y apoyo, que han dirigido las metas y objetivos del proyecto.

A mis compañeros Carlos Ivars y Carles Carbonell por sus ideas y colaboración.

Al instituto de investigación VRAIN por la confianza depositada y el equipo que ha formado.

A la ETSINF y UPV por permitirme aprender y formarme a lo largo de estos años.

Y por último a mi familia, por su apoyo, amor y confianza incondicional.

Sin todos ellos este proyecto no hubiera sido posible, muchas gracias.



# Resumen

---

El Trabajo Fin de Grado consiste en el diseño y desarrollo de un escenario de colaboración humano-robot con el fin de resolver un problema de paletización. El escenario funciona en un entorno virtual que trata de simular fielmente el escenario real, integrando sistemas de visión artificial para permitir al brazo robot colaborativo entender su entorno. El robot distribuye la carga de trabajo del operador asignándole cajas basándose en el modelo del *flow*, facilitando un marco en el que la dificultad de la tarea y la habilidad del trabajador se autorregulen y garanticen una experiencia de trabajo satisfactoria y no fatigante. El humano es capaz de interactuar con el sistema, comunicarse gestualmente con el robot y de apilar cajas en el palé que le hayan sido asignadas por el robot. El robot sigue un algoritmo heurístico que trata de combinar los puntos fuertes de ambos agentes con el objetivo de construir una estructura rígida y estable que corrobore una colaboración satisfactoria. Se ha conseguido resolver el problema planteado en la simulación, pero habrá que ajustar el sistema para que siga funcionando en un entorno real.

**Palabras clave:** Interacción humano-robot; diseño escenarios colaborativos; fábrica del futuro; visión artificial; *cobot*; inteligencia híbrida; espacios inteligentes.

# Resum

---

El Treball Final de Grau consisteix en el disseny i desenvolupament d'un escenari de col·laboració humà-robot amb la finalitat de resoldre un problema de paletització. L'escenari funciona en un entorn virtual que tracta de simular fidelment l'escenari real, integrant sistemes de visió artificial per a permetre al braç robot col·laboratiu entendre el seu entorn. El robot distribuïx la càrrega de treball de l'operador assignant-li caixes basant-se en el model del flow, facilitant un marc en el qual la dificultat de la tasca i l'habilitat del treballador s'autoregulen i garantisquen una experiència de treball satisfactòria i no fatigant. L'humà és capaç d'interactuar amb el sistema, comunicar-se gestualment amb el robot i d'apilar caixes en el palet que li hagen sigut assignades pel robot. El robot segueix un algorisme heurístic que tracta de combinar els punts forts de tots dos agents amb l'objectiu de construir una estructura rígida i estable que corrobore una col·laboració satisfactòria. S'ha aconseguit resoldre el problema plantejat en la simulació, però caldrà ajustar el sistema perquè continue funcionant en un entorn real.

**Paraules clau:** Interacció humà-robot; disseny escenaris col·laboratius; fàbrica del futur; visió artificial; *cobot*; intel·ligència híbrida; espais intel·ligents.

# Abstract

---

The Bachelor's Thesis involves the design and development of a human-robot collaboration scenario to solve a palletizing problem. The scenario operates in a virtual environment that aims to faithfully simulate the real setting, integrating artificial vision systems to enable the collaborative robot arm to understand its surroundings. The robot distributes the operator's workload by assigning boxes based on the flow model, providing a framework where task difficulty and worker skill self-regulate to ensure a satisfying and non-fatiguing work experience. The human can interact with the system, communicate gesturally with the robot, and stack boxes on the pallet as assigned by the robot. The robot follows a heuristic algorithm that seeks to combine the strengths of both agents to construct a rigid and stable structure, confirming a satisfactory collaboration. The problem has been solved in the simulation, but the system will need adjustments to continue functioning in a real environment.

**Keywords:** Human-robot interaction; collaborative scenario design; factory of the future; computer vision; *cobot*; hybrid intelligence; intelligent spaces.



# Índice de Contenidos

---

<b>1. Introducción</b>	<b>11</b>
1.1 Motivación	12
1.2 Objetivos	12
1.3 Impacto Esperado	13
<b>2. Contexto Tecnológico</b>	<b>15</b>
2.1 Fábrica Cognitiva	15
2.2 Paletización	16
2.3 Visión Artificial	17
<b>3. Análisis del Problema</b>	<b>19</b>
3.1 Identificación y Análisis de Problemas y Soluciones Posibles	19
3.2 Soluciones Propuestas	20
<b>4. Tecnología Utilizada</b>	<b>23</b>
4.1 Unity	23
4.4 Robot Operating System	23
4.5 Brazo Robótico UR3e	24
4.6 Google Colab	24
4.7 Keras	24
4.8 Docker	25
<b>5. Diseño de la Solución</b>	<b>27</b>
5.1 Arquitectura del Sistema	27
5.2 Diseño Detallado	28
5.2.1 Modelo de detección de cajas	29
5.2.2 Modelo de cálculo de dimensiones de cajas	31
5.2.3 Almacenamiento temporal de cajas	32
5.2.4 Algoritmo de asignación de cajas	32
5.2.5 Modelo de detección de pose del operario	33
5.2.6 Modelo de estimación de posición de cajas	34
5.2.7 Heurística del algoritmo de asignación de cajas	36
5.2.8 Colaboración humana	37
5.2.9 Algoritmo de paletización	38
5.2.10 Planificador de trayectorias del brazo robot	39
5.2.11 Paletización completada	40
<b>6. Desarrollo de la Solución Propuesta</b>	<b>41</b>
<b>7. Pruebas</b>	<b>45</b>
<b>8. Conclusiones</b>	<b>49</b>
8.1 Relación del trabajo desarrollado con los estudios cursados	50
<b>9. Trabajos Futuros</b>	<b>51</b>
<b>10. Bibliografía</b>	<b>53</b>
<b>Anexo A</b>	<b>55</b>







# Índice de Figuras

---

1. Paletización	11
2. Célula cognitiva	15
3. <i>Pallet loading problem</i> , variante del distribuidor	17
4. Robots UR3e del GIIROB - ETSINF - UPV	21
5. Arquitectura del sistema	27
6. Cámaras del escenario	28
7. Zonas del escenario	28
8. Ejemplo de datos de entrenamiento del modelo de detección de cajas	29
9. Arquitectura del modelo de detección de cajas	30
10. Capa de salida del modelo de cálculo de dimensiones de cajas	31
11. Ejemplo de datos de entrenamiento del modelo de cálculo de dimensiones de cajas	32
12. Escenario virtual con dos cajas en los espacios de almacenamiento	32
13. Ejemplo de datos de entrenamiento del modelo de detección de pose del operario	33
14. Operación XOR con las imágenes de antes y después	35
15. Operación AND con las imágenes XOR	35
16. Arquitectura del modelo de estimación de posición de cajas	36
17. Modelo tradicional de <i>flow</i>	37
18. Ejemplo de colaboración humana en paletización	38
19. Ejemplo de inicio de algoritmo de apilamiento, con cajas en cada esquina	38
20. Ejemplo de ventajas del humano sobre el robot: rotación y entrelazado	39
21. Boceto inicial del primer escenario descartado	41
22. Boceto inicial del escenario de paletización complejo descartado	42
23. Conjunto de imágenes de entrenamiento de escenario genérico descartado	43
24. Ejemplo de palé montado íntegramente por el robot	45
25. Ejemplo de palé montado colaborativamente	46



# Introducción

---

Esta memoria pretende ser una síntesis del trabajo realizado, en ella se tratará de explicar todos los elementos que han hecho posible la elaboración de este proyecto.

Este trabajo forma parte de un proyecto de investigación más amplio en el que se aborda la colaboración entre humanos y máquinas en varios contextos; proyecto que se está llevando a cabo en el instituto VRAIN.

En este trabajo en específico se pretende diseñar e implementar un escenario realista en el que poner en práctica la colaboración entre humano y máquina. Se tratará de utilizar tecnologías que nos permitan acercarnos al estado del arte en distintas áreas, como la inteligencia artificial y la robótica, con el objetivo de que el sistema permita a estos dos agentes, con sus distintas capacidades e ineptitudes, resolver un problema de manera colaborativa, mostrando ventajas a si hubieran resuelto el problema individualmente.

De esta manera, se pretende mostrar una visión de la fábrica del futuro, en la que robots y humanos trabajan en armonía de manera colaborativa. Fábrica en la que los robots son capaces de adaptarse a las necesidades de los humanos y los humanos son capaces de ayudar a los robots en las tareas en las que estos tengan dificultades, y que de esta sinergia surja una mayor productividad a la vez que se mejora el bienestar del humano y se ahorran recursos.

Más concretamente, el problema propuesto a resolver colaborativamente es el problema de la paletización, en el que se pretenderá apilar cajas de distintos tamaños en un mismo palé con el objetivo de construir una estructura rígida y estable ahorrando espacio.

En los siguientes apartados de la introducción se profundizará más en la motivación detrás de este proyecto de investigación, así como del escenario elegido. Se explicará en detalle los objetivos planteados, que nos permitirán completar satisfactoriamente el proyecto y se describirá el impacto esperado del proyecto.

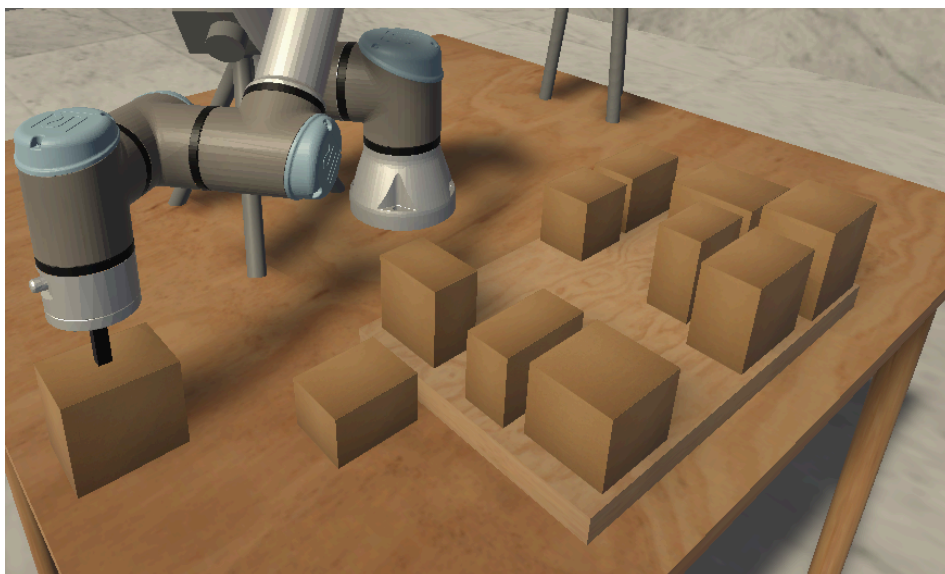


Figura 1. Paletización

## 1.1 Motivación

---

Puesto que el campo recientemente está avanzando aceleradamente en la investigación y desarrollo de la robótica y la superinteligencia, pensamos que el desarrollo de sistemas que pongan al humano en el centro del foco son clave y determinantes para que seamos capaces de integrar estas tecnologías adecuadamente en nuestro día a día.

No obstante, antes de que estas tecnologías puedan llegar a nuestras casas, actualmente muchas empresas ya están utilizando robots colaborativos (*cobots*) para resolver problemas dentro de la industria, por lo que parece más necesario e imperativo actualmente trabajar y refinar estas tecnologías en este contexto, aunque en un futuro estas mismas puedan ser utilizadas en otros.

Estos robots colaborativos llevan bastante tiempo siendo utilizados en la industria y tienen ventajas sobre otro tipo de robots en muchos aspectos, como su fácil programación y adaptación a resolver distintos tipos de problemas y sus mecanismos de seguridad que los permiten operar cerca de humanos, sin que esto suponga un gran riesgo.

No obstante, estos robots por defecto no presentan sistemas que les permitan entender e interactuar con su entorno de una manera efectiva, ya que no disponen de las entradas (como cámaras) o de la inteligencia necesaria para ello.

Por lo que, para resolver nuestro sistema de paletización colaborativa, proponemos una serie de sistemas integrados que permitan extender las capacidades de estos robots. Consideramos que puede tener un gran impacto en la industria si se sigue esta línea de desarrollo y se apuesta por el desarrollo de sistemas inteligentes colaborativos para resolver problemas comunes que ya se resuelven en la industria, pero de otra manera.

Consideramos que el de la paletización es un escenario adecuado a estudio, ya que puede verse como un problema bastante genérico y común a resolver, que tiene un cierto grado de complejidad para humano y máquina, y que puede surgir en muchos puntos del proceso logístico, como en el almacenamiento o transporte de bienes en palés, y que una correcta solución obtiene una mejor productividad, como en el caso del ahorro del tiempo de apilamiento o del espacio de almacenamiento. Además, es un escenario que permite fácilmente mostrar los distintos puntos fuertes y puntos débiles de humanos y robots, y permite encontrar puntos intermedios en los que se potencian los puntos fuertes de ambos cuando se asignan las tareas adecuadamente de acuerdo con las habilidades de estos.

Otro de los motivos importantes de este proyecto es el de poner al humano en el centro de la ecuación. El avance tecnológico que lleva suponiendo durante estos últimos años la inteligencia artificial puede llegar a provocar otra especie de revolución industrial, en la que muchos puestos de trabajo se pierdan a causa de esto. Por lo que la investigación en el ámbito de la colaboración humano-máquina puede ayudar a que se usen a los robots como extensiones que mejoran a los humanos y no como sustitutos directos. Por lo que además en este trabajo también se pretende usar sistemas que tengan en cuenta el bienestar del operario y que traten de ofrecer una experiencia de trabajo cómoda, entretenida y no estresante.

## 1.2 Objetivos

---

El objetivo principal es diseñar e implementar un escenario en el que humano y máquina trabajen colaborativamente para resolver un problema de paletización.

Para conseguir este objetivo se pretende usar un entorno de simulación virtual, tratando de crearlo de la manera más fiel posible a la realidad. Por ejemplo, en el entorno virtual es muy fácil obtener la posición de los objetos en él, pero esto sería imposible si el escenario estuviera montado en un entorno real, por lo que para obtener por ejemplo las posiciones de los objetos, hay que utilizar sistemas que podrían funcionar también una vez se pase íntegramente al escenario físico.

Hay que diseñar el escenario virtual de manera que pueda ser transferido al entorno físico con gran facilidad. Se tratará de ejecutar el escenario en un entorno físico real, una vez funcione en el entorno virtual.

La idea es que el escenario virtual pueda ofrecer exactamente las mismas ventajas que el escenario físico, que tenga en cuenta todas las restricciones que habría en el escenario real, pero que además tenga ventajas sobre este, como la capacidad que tiene el entorno virtual para ser compartido y usado por muchas personas sin necesidad de tener un robot colaborativo físico o la facilidad que ofrece poder simular muchos entornos diferentes con distintas variables en el problema para verificar que el sistema funciona correctamente.

Otro objetivo principal es el de permitir que un humano real sea capaz de interactuar directamente con el entorno, y sea capaz de apilar cajas como lo haría en el entorno real, sin simular esta parte con algoritmos que simulan el comportamiento de apilamiento humano. Es importante tener en cuenta que queremos distinguir claramente entre las ventajas que tiene un humano sobre un robot en distintos aspectos y viceversa, por lo que simulando la inteligencia humana esto no se consigue tan claramente.

Como objetivo, el sistema ha de ser capaz de visualizar los elementos del entorno y poder actuar correctamente basándose en su conocimiento de este.

En cuanto al algoritmo de apilamiento del robot, es necesario que este tome decisiones adecuadas, que tengan en cuenta las cajas ya depositadas en el palé y trate de seguir construyendo una estructura rígida y estable.

También se pretende que el trabajador tenga una experiencia de trabajo estimulante y no fatigante, apilar cajas intentando ahorrar espacio y mantener la rigidez de la estructura con cajas de distintos tamaños no es un problema fácil y pensamos que este grado de complejidad puede ser positivo para el trabajador si la dificultad del problema se adapta a sus capacidades, por lo que el robot debe adaptarse a las habilidades del trabajador asignándole más o menos cajas dependiendo de la facilidad con la que este las apile.

### 1.3 Impacto Esperado

---

Esperamos que este proyecto suponga un paso adelante en la investigación de la colaboración humano-robot, sirva como punto de partida y de ejemplo para otros proyectos dentro de esta línea para el instituto VRAIN o para otros equipos investigadores y que ayude a fomentar el desarrollo de soluciones robóticas centradas en el humano.

Además, esperamos que el problema de la paletización en específico tome más importancia y que distintas ideas que proponemos para resolverlo puedan inspirar a la industria a avanzar en la dirección colaborativa.



# Contexto Tecnológico

Hasta donde conocemos, no existen trabajos similares que hayan tratado de resolver un problema de paletización con una solución colaborativa humano-robot. No obstante, dividiendo este proyecto en partes, se pueden distinguir distintas áreas de la informática en las que sí se ha investigado profundamente, por lo que hay mucha literatura por analizar. Además, aunque este problema en específico no haya sido abordado de esta manera, sí se ha tratado de resolver otro tipo de problemas con soluciones colaborativas, por lo que se puede también analizar el contexto tecnológico de manera más genérica.

En este apartado se tratará de dividir el proyecto en partes, analizando el contexto tecnológico en cada una de ellas y tratando de analizar también nuestras propuestas y avances en este marco.

## 2.1 Fábrica Cognitiva

En el artículo (Cotta et al., 2023 [1]), se definen conceptos clave para el desarrollo de nuestro trabajo: La Industria 5.0 (I5.0) representa un cambio hacia una industria centrada en el humano que enfatiza en la integración de habilidades de humanos y máquinas. Un concepto muy compatible para habilitar la implementación de I5.0 es el espacio inteligente (IS, por sus siglas en inglés “*Intelligent Spaces*”). Un IS, puede ser definido como un espacio físico equipado con una red de sensores que obtienen información sobre el espacio que observan, una red de actuadores que permite la interacción con usuarios, y cambios en el entorno a través de servicios computacionales. Dado que una fábrica puede estar formada por unidades más pequeñas conteniendo sensores, dispositivos, sistemas, y subsistemas, integrados o no, en ese artículo se denominan células cognitivas, y se definen como un componente fundamental que opera como una unidad inteligente equipada con tecnologías cognitivas para mejorar el proceso de fabricación.

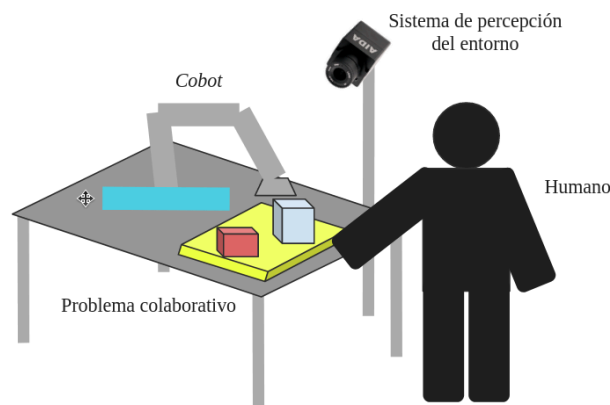


Figura 2. Célula cognitiva

En este marco, en nuestro proyecto se pretende desarrollar una célula cognitiva específica. En el artículo mencionado anteriormente, implementan una célula cognitiva en la que el humano ofrece al robot cubos de distintos colores que este debe recoger de la mano del operador y depositarlos en distintos espacios según su color. Por sus motivaciones y objetivos, este trabajo es el más similar al nuestro hasta donde sabemos.

En el artículo (Dellerman et al., 2019 [2]) definen la inteligencia híbrida como sistemas que tienen la habilidad de completar tareas complejas combinando la inteligencia humana y artificial para conseguir colectivamente resultados superiores que los que habrían conseguido por separado y mejoran continuamente aprendiendo el uno del otro. Nuestro caso de uso pretende ser un acercamiento a la inteligencia híbrida, aunque aún no consigue serlo según su definición, pues la inteligencia de nuestro robot aún no aprende continuamente del humano.

En el artículo (Yaqian et al., 2024 [3]) desarrollaron un método con la idea de mejorar la habilidad cognitiva de los robots en la colaboración humano-robot tratando de predecir las acciones que va a realizar el humano y tratando de adaptar el robot a este. Nuestros modelos, aunque pretenden recibir información y comunicarse con el humano, no llegan a tratar de predecir las acciones que este realizará.

En el artículo (S. Li et al., 2021 [4]) definen la colaboración humano-robot proactiva como “una colaboración bidireccional y autoorganizada entre operadores humanos y robots en actividades de manufactura, donde pueden trabajar proactivamente para un objetivo común en cada ciclo de ejecución a lo largo del tiempo.”.

En el artículo (Claudia et al., 2023 [5]) muestran una revisión que se centra en la carga mental reportada por los operadores que interactúan con *cobots*, sus principales fuentes y las posibles soluciones para optimizarla durante la colaboración humano-robot. (Costanza et al., 2022 [6]) muestran una estrategia de asignación dinámica de tareas para mitigar la fatiga del humano y garantizar una colaboración efectiva en tareas colaborativas con robots. Aunque nuestro trabajo no profundiza en estos aspectos, sí los tiene en cuenta y pretende, mediante una simple heurística, intentar no agobiar al trabajador en la asignación dinámica de cajas.

## 2.2 Paletización

---

El problema de la paletización de cajas de distintos tamaños puede verse como una variante del problema popularmente conocido como *3D bin packing problem* (3D-BPP), con la variación de solo utilizar un contenedor para almacenar, *3D single bin packing problem*.

Además, esta variante es conocida como el *pallet loading problem* (PLP) que es esencialmente un problema 3D-BPP, pero con los requisitos adicionales de asegurar que los palés son estables y que la mercancía puede ser manipulada y transportada con seguridad (Silva et al., 2016 [7]). Además, esta variante al utilizar cajas de distintos tamaños se denomina variante del distribuidor; la variante del fabricante utiliza cajas homogéneas.



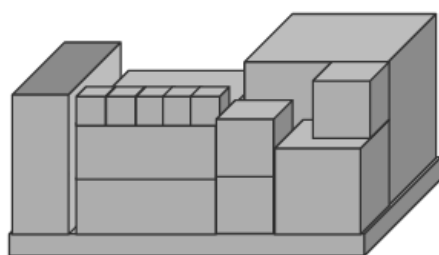


Figura 3. *Pallet loading problem*, variante del distribuidor

También es necesario hacer otra distinción, en este artículo (Sara et al., 2022 [9]), se comparan distintas soluciones a problemas 3D-BPP dividiéndolos en off-line y on-line. El problema off-line se resuelve teniendo el conocimiento de todas las cajas que van a ser depositadas en el palé de antemano, sin embargo, el problema on-line solo es conocedor de las cajas una a una y una vez apiladas no es posible quitarlas o modificar su posición. Nuestro caso de uso, por tanto, forma parte de los problemas on-line, ya que solo conoceremos hasta 3 cajas antes de apilarlas o compartirlas con el trabajador. En el mismo artículo mencionado anteriormente, se analizan distintas soluciones en el estado del arte del on-line 3D-BPP. Este muestra que hay dos tipos de soluciones a este problema, la solución heurística y la solución que usa inteligencia artificial. Las soluciones heurísticas suelen usar el algoritmo simple *deepest bottom left* (DBL), en el que, como su nombre indica, se trata de almacenar las cajas lo más profundo, lo más abajo y lo más a la izquierda posible. Las soluciones que usan inteligencia artificial tratan de resolver el problema utilizando *reinforcement learning*, opción que nos gustaría probar en un futuro.

Para este proyecto hemos tratado de diseñar una heurística nueva con la simpleza del DBL, pero que tenga en cuenta el equilibrio de la estructura y que, por tanto, no concentre todo el peso en una única esquina, y más adiciones que explicaremos en más detalle más adelante.

## 2.3 Visión Artificial

---

Para que nuestro robot sea capaz de entender el entorno y actuar al respecto, los sistemas de visión artificial integrados son muy importantes, en especial las técnicas de *deep learning* en reconocimiento de imágenes son clave para que el proyecto pueda funcionar. Hay muchos tipos de problemas que se pueden abordar con el reconocimiento de imágenes, en nuestro caso, necesitamos ser capaces de reconocer el tamaño de cajas, reconocer la posición en la que se colocan e identificar la pose actual del humano, pues queremos que este sea capaz de comunicarse con el robot gestualmente.

El artículo (S. Hoque et al., 2021 [10]) muestra una revisión amplia de distintos métodos y tecnologías que se usan para resolver este tipo de problemas. La red de aprendizaje profundo más común para el procesamiento de imágenes es la red neuronal convolucional, ya que es capaz de extraer características importantes de estas por su capacidad de agrupar información relacionada. Esta red se puede aplicar de base para resolver todos los problemas de visión artificial que tendremos en el proyecto, aunque también habrá que procesar o generar las imágenes de cierta manera para que puedan ser utilizadas por esta red adecuadamente y pueda generalizar a datos con los que no ha sido entrenada, como explicaremos más adelante.

No obstante, no es un trabajo sencillo extraer información en 3D de imágenes en 2D. En nuestro proyecto, para poder obtener la posición, tamaño y rotación de la nueva caja depositada

en el palé necesitamos usar 2 cámaras simultáneamente, ya que con una única imagen es muy difícil obtener esta información con precisión. Podemos ver trabajos como el siguiente (D. Liang et al., 2014 [11]) que también usan 2 cámaras.

También es muy importante ser capaces de generar nuestros propios datos de entrenamiento sintéticos en simulación, usando *domain randomization* (J.Tobin et al., 2017 [12]), como mostraremos en los capítulos siguientes, para que la red sea capaz de generalizar y funcionar también en imágenes reales, no solo en el simulador.

# Análisis del Problema

---

En este caso de uso, se pretende identificar el problema en términos muy genéricos, que más adelante se dividirán en subproblemas que tendrán sus soluciones específicas.

El problema a resolver en general es el de diseñar e implementar un escenario virtual que pueda ser llevado a un entorno físico en el que robot y máquina colaboren para resolver un problema de paletización.

Una vez identificado el problema principal, se puede tratar de encontrar soluciones dividiéndolo en subproblemas y tratando de resolver cada uno de ellos con una solución específica. En las siguientes secciones se tratará de identificar y analizar las posibles soluciones a esta serie de problemas y mostrar la solución propuesta.

## 3.1 Identificación y Análisis de Problemas y Soluciones Posibles

---

Para poder ejecutar el escenario virtual, es necesario el uso de un simulador en 3D que nos permita integrar con facilidad el uso de físicas reales, el uso de robots colaboradores, que nos posibilite generar datos sintéticos para entrenar los modelos de visión de inteligencia artificial y que nos facilite interactuar con el escenario para que un humano real pueda paletizar cajas con el robot.

En cuanto al robot colaborador a utilizar, existen muchos diferentes en el mercado diseñados específicamente para la tarea de paletizar, no obstante, estos brazos robots tienen la desventaja de tener un gran tamaño y que no disponemos de ellos en la facultad, por lo que para probar estos nuevos sistemas, consideramos que es preferible usar robots más manejables y a los que tengamos acceso. Por lo tanto, por su facilidad de uso y el acceso a robots de la universidad que disponemos, se ha decidido reducir la escala del escenario para adecuarse mejor a nuestras posibilidades. Por lo tanto, la reducción del tamaño del escenario será proporcional a la diferencia de tamaño entre un robot estándar paletizador y nuestro robot de menor tamaño. Por lo que el tamaño de palés y cajas se reducirá para mantener esta proporcionalidad, y que el brazo robot reducido pueda manejar las cajas de la misma manera que lo haría el brazo robot de mayor tamaño.

Es necesario concretar, de qué manera van a colaborar humano y robot. Para este problema, consideramos que es interesante que ambos puedan realizar acciones similares para destacar mejor las diferencias entre ambos que puedan surgir al realizar la misma acción, pero de diferente manera. Por lo que para este problema de paletización se pretende que el robot reciba todas las cajas que entren al sistema, y que este decida si apilarla él mismo o dársela al humano para que sea él el que la apile. De esta manera, ambos son capaces de apilar cajas en el palé y se podrá observar qué puntos fuertes y débiles tienen cada uno de ellos en esta tarea y cómo potenciar sus puntos fuertes.

Se tiene que tener en cuenta, de qué manera va a ser el robot capaz de entender el entorno. Para este problema, creemos que es necesario el uso de varias cámaras en distintas posiciones del escenario con distintas finalidades específicas. Existen otro tipo de cámaras avanzadas

conocidas como LiDAR (acrónimo de Light Detection and Ranging), una técnica de teledetección activa que utiliza un sensor embarcado en diferentes tipos de plataformas que emite pulsos de luz ininterrumpidamente y capta sus retornos y es capaz de reconstruir entornos en 3D con facilidad (Tachella et al., 2019 [13]). No obstante, por la dificultad de integrar esta cámara en entornos virtuales y la capacidad de obtener buenos resultados también con cámaras convencionales, se ha decidido no usar esta tecnología. Como por ejemplo Tesla, que en sus modelos de visión solo usan cámaras convencionales (S. Hoque et al., 2021 [10]).

Es necesario el uso de modelos de inteligencia artificial capaces de reconstruir el mundo en 3 dimensiones a partir de imágenes en 2 dimensiones.

Es necesario un algoritmo que sea capaz de integrar los sensores de las cámaras con el comportamiento del robot, es decir, que tome las imágenes en el momento adecuado, que las envíe a modelos de inteligencia artificial para que sean procesadas y que el robot actúe conforme a esta nueva información y la que ya contaba en su memoria del estado del escenario.

Este algoritmo a su vez debe contar con un algoritmo de toma de decisiones que le permita decidir qué acción realizar con cada caja nueva que recibe, de tres acciones posibles: Guardar la caja para más tarde, dar la caja al operador humano para que la apile él o apilar la caja él mismo en el palé. Teniendo en cuenta también el estado del operario, tratando de no agobiarlo con muchas cajas si este no tiene la destreza para apilarlas o puede encontrarse cansado. Es necesario también poder medir de alguna manera este estado del operario.

Es necesario también un algoritmo que sea capaz de entender el estado del palé, que pueda ser actualizado con nuevas cajas que apile el humano, y que tome la mejor decisión posible para poder apilar nuevas cajas y que se mantenga la estabilidad de la estructura y se ahorre espacio.

Es necesario también poder crear un plan de trayectorias a partir de las decisiones que se tomen que permita transformar las posiciones de *pick and place* dentro del entorno virtual en rotaciones en las articulaciones del robot que le permitan mover cajas de posición.

Todos estos problemas tienen muchas posibles soluciones, por lo que trataremos de enfocarnos en nuestra solución específica. Además, tienen subproblemas más pequeños que permiten integrar todo el sistema y que se tratarán de detallar más adelante.

## 3.2 Soluciones Propuestas

---

Para cumplir todos los requisitos anteriormente mencionados, se usará el entorno de simulación Unity, un entorno creado principalmente para el desarrollo de videojuegos, pero que es lo suficientemente genérico como para resolver otro tipo de problemas.

En especial, existen distintos paquetes que se pueden añadir a Unity y que facilitarán mucho la integración de distintos sistemas, como el paquete de percepción de Unity, que nos permitirá crear datos sintéticos para entrenar nuestros modelos de visión artificial, y el paquete de robótica de Unity que permite incluir diversos robots en él, y comunicarse con ellos y manejarlos de la misma manera que se haría con el robot real, a modo de *digital twin* (Lianhui Li et al., 2022 [14]).

El brazo robot que se utilizará será el UR3e, de Universal Robots, dado que en un laboratorio de la ETSINF (Escuela Técnica Superior de Ingeniería Informática), perteneciente al GIIROB (Grado de Informática Industrial y Robótica) se cuenta con varios de ellos.



Figura 4. Robots UR3e del GIROB - ETSINF - UPV

La idea es que el robot reciba las cajas en un espacio para depositarlas preestablecido. Habrá una cámara monitorizando continuamente esta zona, que tomará imágenes de la zona cuando espere recibir más cajas. Estas imágenes serán enviadas al modelo de inferencia de inteligencia artificial diseñado para detectar si hay una caja depositada en este espacio. Una vez se haya detectado si hay una caja, el sistema tomará otra imagen que será enviada al modelo encargado de calcular las dimensiones de la caja. Para detectar cambios en el estado del palé, habrá 2 cámaras, una en cada esquina del palé. Una de ellas tendrá también visión del operario de frente, la idea es que esta cámara tome fotos continuamente del operario cuando esté colocando la caja y que estas sean enviadas a un modelo de inteligencia artificial que trate de detectar un gesto del pulgar arriba del operario. Esta es la manera que el operario tiene de comunicarse con el robot. Cuando el operario haya colocado la caja en el palé, este le hará un gesto de aprobación con la mano a la cámara, lo que hará que las 2 cámaras enfocando al palé tomen otra foto del estado del palé y envíen al modelo de inteligencia artificial cuatro imágenes, 2 correspondientes al palé antes de apilar la nueva caja, y 2 correspondientes al palé después de apilar la nueva caja. De esta manera, el modelo de inteligencia artificial debe ser capaz de comunicar al sistema del robot, en qué posición y en qué rotación específicas se ha colocado esta nueva caja, y el sistema reajustará su conocimiento del sistema acorde a esta adición.

El algoritmo de toma de decisiones es bastante sencillo. Este algoritmo siempre tratará de tener el espacio de almacenamiento de cajas del robot lleno, para así siempre tener posibilidades de recibir una mejor caja que apilar a las ya encontradas en el almacenamiento. Una vez el almacenamiento queda lleno, se analiza si es momento de compartir la caja con el trabajador. La manera de la que esto funciona, es con un simple contador que cuenta cuántas cajas hay que apilar antes de ofrecer la caja al trabajador para que él la apile. Cuando este contador llega a 0, quiere decir que no hay más cajas que apilar en el palé, lo que significa que habrá que compartir la caja con el trabajador. Una vez el trabajador apile la caja, habrá que tener en cuenta el tiempo que ha tardado en hacerlo para calcular cuántas cajas tiene que apilar el robot antes de que se compartan con el operario, de manera que cuanto más tiempo ha tardado en apilar, más cajas habrá que esperar, con la heurística proveniente de pensar que si el operario ha tardado un tiempo excesivo en apilar la caja es porque está excesivamente cansado o porque no tiene la destreza necesaria o porque el estado del palé no le permite operar con facilidad. Por el contrario, si el operario es capaz de apilar la caja con velocidad, esto puede decir que el operario está trabajando a un buen ritmo, tiene la habilidad necesaria para la tarea que está realizando o el estado del palé facilita esta rápida maniobrabilidad, por lo que no queremos cortar el ritmo de trabajo o de estado de *flow* (Csikszentmihalyi, 1988 [15]) y tardaremos menos cajas en volver a asignarle una.

También puede pasar, que el algoritmo de apilamiento del robot no encuentre una posición de apilamiento lo suficientemente adecuada para las cajas a las que tiene acceso. Este algoritmo funciona asignando una puntuación a cada caja en cada una de sus 2 posibles rotaciones (explicación detallada más adelante) en cada uno de los puntos que se consideran adecuados para apilar. Por lo que, una puntuación demasiado baja puede llevar al sistema a decidir dar la caja al trabajador para que sea él mismo el encargado de encontrar una mejor posición que las encontradas por el sistema.

El algoritmo propuesto para el apilamiento de cajas en el palé es un nuevo algoritmo que hasta nuestro conocimiento no había sido utilizado anteriormente para resolver el problema de paletización, que trata de mantener la distribución de peso en el palé y de crear oportunidades de entrelazado de cajas para mejorar la rigidez de la estructura. Será explicado con más detalle más adelante.

Para poder crear el plan de trayectorias para los movimientos del robot, se ha utilizado ROS (*Robot Operating System*) utilizando el *framework* Moveit, como detallaremos más adelante.

# Tecnología Utilizada

---

En esta sección se tratará de describir la tecnología más importante que se ha utilizado en el desarrollo del proyecto.

## 4.1 Unity

---

Unity es el motor de videojuego más popular, con la posibilidad de crear juegos, aplicaciones y experiencias en 3D para multitud de industrias. Unity ofrece distintos motores de física que permiten simular correctamente la aceleración de objetos, colisiones, gravedad y otras fuerzas diversas. Unity ofrece la posibilidad de desarrollar en distintos lenguajes de programación, pero el más utilizado y extendido en la comunidad es C#.

Es la tecnología más importante del proyecto, ya que va a ser utilizada como cliente principal desde el que simular el entorno y es donde se encuentran los algoritmos principales del sistema que permiten controlar el robot y comunicarse con los otros sistemas del proyecto.

Unity también se utiliza para generar datos sintéticos para los modelos de inteligencia artificial con el paquete Perception.

El paquete Perception de Unity ofrece una serie de herramientas para generar a gran escala conjuntos de datos sintéticos para el entrenamiento y validación de modelos de visión artificial. Está centrada en unos específicos casos de uso basados en cámaras con la intención de expandirse a otras formas de sensores y tareas de aprendizaje automático.

En este proyecto se cuenta con distintos modelos de inteligencia artificial cuyos datos se han generado simulando distintas variaciones de escenarios utilizando *domain randomization* para que los modelos sean capaces de generalizar y transferir lo aprendido a cámaras reales.

El paquete URDF Importer de Unity permite importar robots a Unity definidos con el formato URDF (Universal Robotic Description Format). URDF define la geometría, las mallas visuales, los atributos cinemáticos y dinámicos de un robot en formato XML.

El paquete ROS TCP Connector permite la comunicación entre Unity y los servicios ROS, que será esencial para controlar el movimiento del *cobot*.

## 4.4 Robot Operating System

---

Robot Operating System (ROS) es un conjunto de librerías de *software* libre que define los componentes, interfaces y herramientas para elaborar robots avanzados. Los robots generalmente cuentan con sensores, actuadores y sistemas de control, ROS ayuda a los desarrolladores a desarrollar estos componentes y después comunicarlos fácilmente utilizando herramientas ROS llamadas temas y mensajes. Estos mensajes pueden ser fácilmente enviados a una variedad de herramientas de visualización y teleoperación como Unity.

En nuestro proyecto, ROS es el encargado de generar una lista de trayectorias para el movimiento del robot, a partir de un punto de *pick* y otro punto de *place*, para así mover las cajas de sitio adecuadamente, utilizando el *framework* Moveit.

Moveit es la plataforma por excelencia de código abierto para construir aplicaciones de manipulación comercial. Moveit combina cinemática avanzada, planificación de movimiento y capacidad de detección de colisiones para planificar y ejecutar movimientos óptimos para cualquier manipulador robótico, puesto que es agnóstico al robot.

## 4.5 Brazo Robótico UR3e

---

UR3e se trata de un robot desarrollado por Universal Robots compacto de sobremesa, con el formato compacto ideal para espacios de trabajo reducidos. Por sus dimensiones, se puede utilizar en mesas de trabajo o instalarlo directamente en maquinaria, por lo que resulta ideal para aplicaciones ligeras de montaje y atornillado.

Este tiene un rango de alcance de 50 centímetros y puede levantar una carga útil de 3 kg. Este *cobot* cuenta con sistemas de seguridad aprobados y certificados por TÜV (asociación alemana de inspección técnica). Estos sistemas, por ejemplo, paran el movimiento del robot cuando se detecta una colisión no esperada, para así reducir posibles daños en el caso de colisionar con un humano.

Para este proyecto, hay que importar el URDF de este robot a Unity para controlarlo y visualizarlo, y al sistema Moveit de ROS, para que este tenga información sobre las respectivas articulaciones y pueda planear el movimiento adecuadamente.

La herramienta instalada en la punta del brazo del robot es una ventosa de vacío que permite desplazar objetos no muy pesados. Como restricción para nuestro problema, este solo puede coger y dejar cajas desde el punto central arriba de la caja, ya que otras rotaciones de la caja pueden ser complejas y no garantizan que el centro de gravedad de la caja esté alineado con el brazo robot y, por tanto, puede causar que la caja caiga.

## 4.6 Google Colab

---

Colab es un servicio alojado de Jupyter Notebook que no requiere de configuración y que ofrece acceso gratuito a recursos de computación, como GPUs y TPUs, ofreciendo una solución especialmente adecuada para el aprendizaje automático y la ciencia de datos.

Este sistema ha sido esencial para nuestro proyecto, pues contratando distintos servicios, hemos sido capaces de entrenar y realizar pruebas para nuestros distintos modelos de inteligencia artificial. En específico para este proyecto, se han utilizado aproximadamente 300 unidades informáticas en el testeo y entrenamiento de los distintos modelos, ejecutándose en una TPU con un gasto aproximado de 1.76 de unidades informáticas por hora.

## 4.7 Keras

---

Keras es una API de alto nivel para el uso de Tensorflow que proporciona una interfaz accesible y altamente productiva para resolver problemas de aprendizaje automático, con un



enfoque en el aprendizaje profundo moderno. Keras cubre cada paso del flujo de trabajo del aprendizaje automático, desde el procesamiento de datos hasta el ajuste de hiperparámetros y la implementación.

En nuestro caso, Keras ha sido utilizado con la arquitectura de redes convolucionales VGG16 preentrenada con los pesos de problemas de clasificación de imagenet. Aunque estos pesos están entrenados con un objetivo muy distinto al de estimación de poses de objetos en 3D, estos modelos basados en redes convolucionales son capaces de aprender características intrínsecas de las imágenes que también aplican a nuestro problema, por lo que empezar con una red ya entrenada para el reconocimiento de imágenes o cualquier otro objetivo, acelera el tiempo de aprendizaje significativamente y muestra la potencia del *fine tuning* y el *transfer learning* (S. J. Pan. et al., 2009 [16]).

## 4.8 Docker

---

Docker es una plataforma que ofrece la habilidad de empaquetar y lanzar aplicaciones en un entorno aislado de manera laxa en un entorno llamado contenedor. Este aislamiento del sistema anfitrión permite elaborar contenedores fiables que pueden ser diseñados para funcionar de la misma manera en diferentes sistemas Linux, asegurando la portabilidad del sistema y garantizando que funcionará de la misma manera independientemente del ordenador anfitrión.

En nuestro proyecto, con esta idea en mente de portabilidad, hemos incluido los distintos servidores que proveen servicios al cliente Unity en distintos contenedores utilizando *docker compose* para orquestarlos, para asegurar que en cada contenedor existe el *software* necesario para funcionar, separar esta dependencia del ordenador anfitrión y facilitar las labores de despliegue del proyecto. En específico, hay dos contenedores: uno con ROS, con el que se comunica el cliente de Unity para las labores del control del movimiento del robot, y otro con un servicio web de API REST con el *framework* de *routing* Flask al que Unity envía las imágenes para ser procesadas por los modelos de inferencia de inteligencia artificial.





# Diseño de la Solución

En este capítulo, se unirán los términos, tecnologías e ideas que se han ido introduciendo a lo largo del trabajo de manera concreta y detallada para mostrar de manera precisa cómo se ha resuelto el problema planteado.

Finalmente, el escenario solo ha sido desplegado en el entorno virtual, en el siguiente apartado se explicará qué pasos habría que seguir para llevar el escenario al mundo real.

## 5.1 Arquitectura del Sistema

El cliente encargado de ejecutar el código principal de la aplicación, con los algoritmos principales del control del robot, es Unity. Unity inicia la ejecución de la aplicación y se comunica con los dos servidores de ROS y Flask, para obtener el plan de movimiento del robot y analizar las imágenes tomadas por Unity, respectivamente.

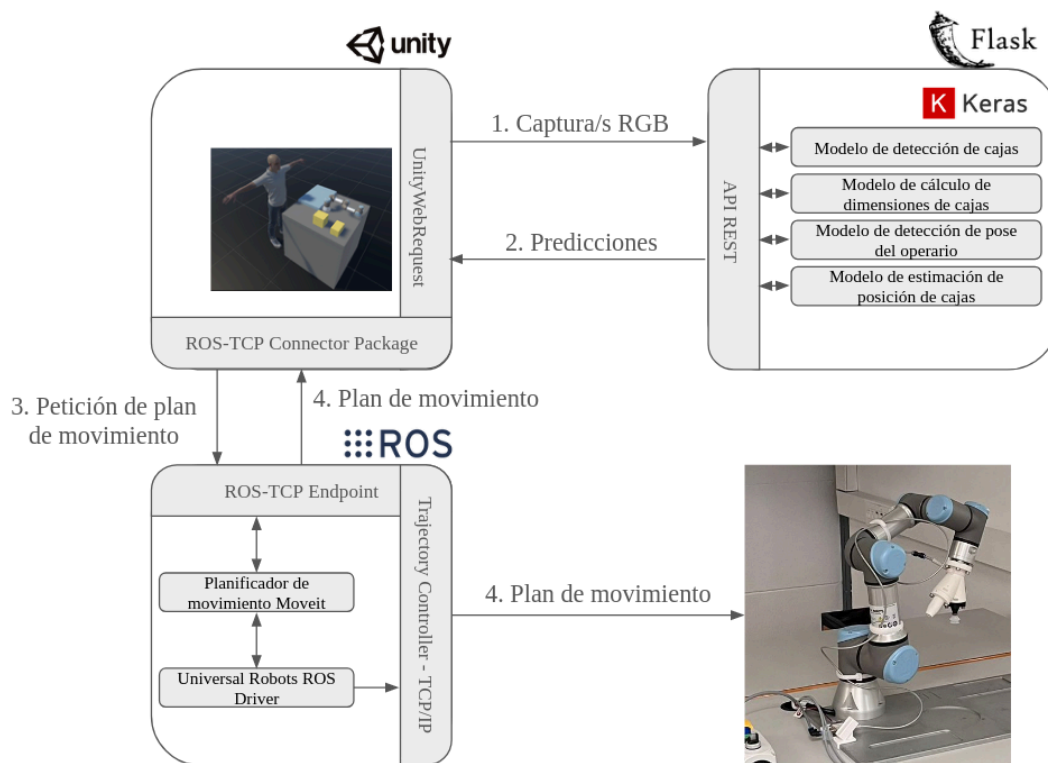


Figura 5. Arquitectura del sistema

Como este caso de uso ocurre íntegramente en el mundo virtual de Unity, las imágenes del escenario se toman directamente de capturas de pantalla de las distintas cámaras montadas en este escenario. No obstante, el código está preparado modularmente para incluir una nueva clase



que se encargue de hacer peticiones a otro tercer posible servidor desde el que se obtendrían las imágenes de un entorno real.

De manera que para transferir el funcionamiento del sistema virtual a un sistema físico real, solo habría que añadir este servidor dedicado que se encargaría de recibir peticiones del cliente y devolver la imagen deseada en ese instante de tiempo y configurar la IP que usará el sistema ROS para comunicarse con el robot. Esto habilitaría que todos los movimientos que efectuara el robot en el entorno virtual los efectuara de la misma manera el robot del entorno real y que, por tanto, si los sistemas de visión artificial funcionan correctamente, que el sistema funcione completamente en el escenario real también.

## 5.2 Diseño Detallado

---

En primer lugar se mostrará la estructura del escenario final. El tamaño de los palés y las cajas se ha reducido para poder adaptarse al tamaño reducido del robot UR3e.

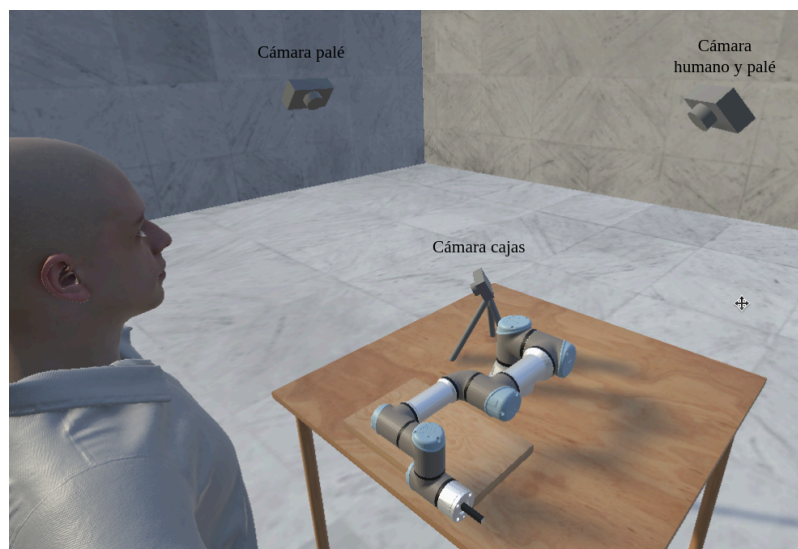


Figura 6. Cámaras del escenario

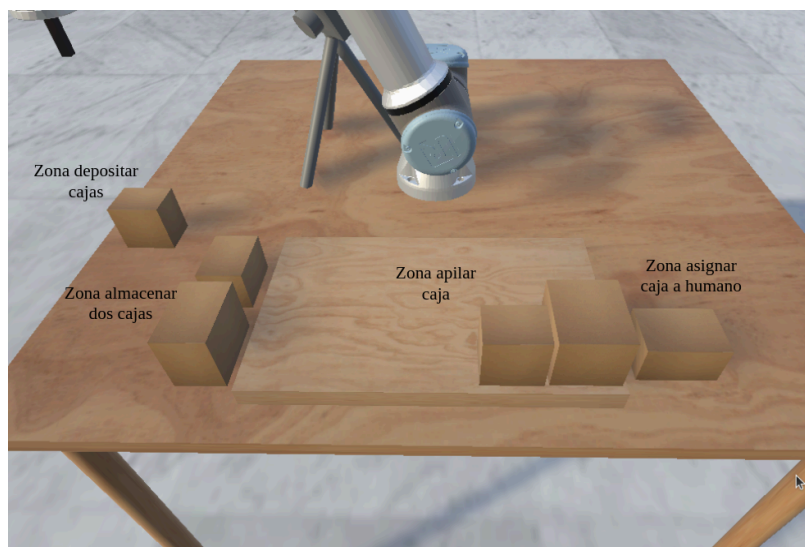


Figura 7. Zonas del escenario

La manera más sencilla de explicar todo el diseño del sistema detalladamente es siguiendo el hilo de ejecución desde que arranca la aplicación cliente hasta que acaba, ya que esta sigue una ejecución determinista y de esta manera analizaremos todos los elementos que componen la solución final de manera ordenada.

### 5.2.1 Modelo de detección de cajas

---

El primer paso del algoritmo del robot es esperar a que se coloque una caja en la zona diseñada especialmente para este propósito. Esta espera funciona utilizando la cámara, que toma fotos de la región en la que se colocará la nueva caja. Estas imágenes son enviadas con una petición POST a la API REST del servicio Flask que funciona en Docker y que es el encargado de hacer la inferencia de los modelos de inteligencia artificial. El único propósito de este primer modelo de inteligencia artificial es detectar si hay una caja o no en la región explicada, por lo que la respuesta esperada es un valor real entre 0 y 1, indicando la probabilidad de que esta imagen contenga una caja en la zona destinada para ello o no.

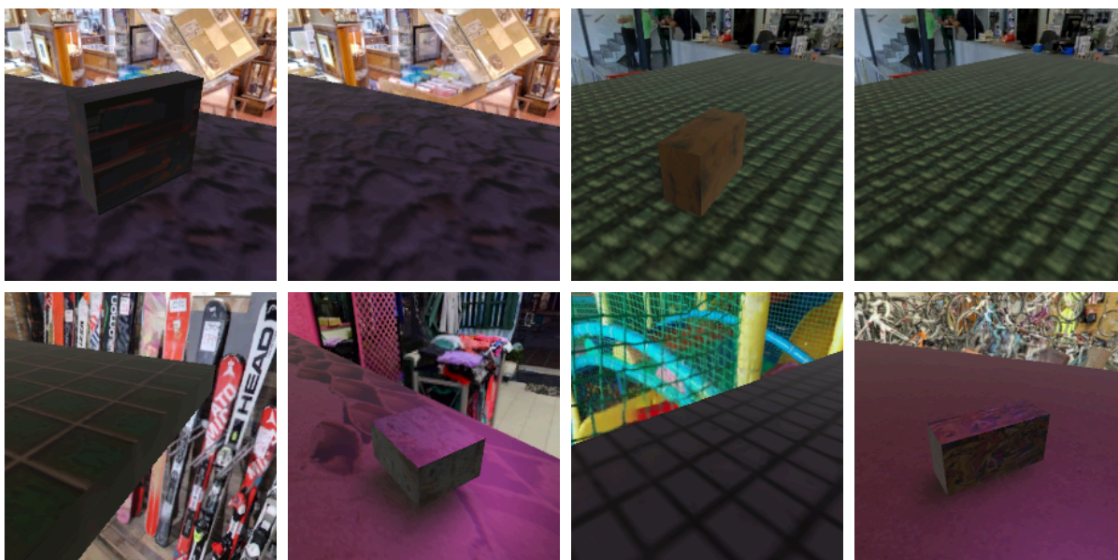


Figura 8. Ejemplo de datos de entrenamiento del modelo de detección de cajas

El conjunto de datos para el entrenamiento de este modelo ha sido generado utilizando un escenario diferente de Unity que trata de imitar las condiciones del escenario de simulación del entorno, pero con distintos cambios claves. Este escenario utiliza el paquete de percepción de Unity para generar imágenes para el modelo de entrenamiento. Este paquete se encarga de facilitar la aleatoriedad de las características de los elementos del entorno para garantizar que el modelo pueda llegar a funcionar en una gran variedad distinta de escenarios posibles pero con las mismas características base. A continuación se enumeran los elementos del escenario que han sido aleatorizados:

- El color y la dirección de la luz.
- La imagen de fondo correspondiente a una imagen panorámica de 360° de una habitación real, unas 250 imágenes distintas obtenidas de habitaciones interiores de Google Street View.
- La rotación, tamaño, posición y textura aleatoria entre 120 texturas de la mesa sobre la que se colocan la cámara y la caja.
- Posición fija de la caja, pero con variación del tamaño y ligera rotación de la caja, y variación de la textura entre las mismas 120 texturas que la mesa.

- La posición y ángulo de la cámara con respecto a la posición en la que se deposita la caja son fijas, para garantizar la precisión del modelo.

Las distintas iluminaciones y texturas generan un conjunto extenso de imágenes variadas que son capaces de generalizar a cualquier escenario y caja, hasta con las que no ha sido directamente entrenado, usando el principio de *domain randomization* y, por tanto, se cree que este modelo podrá generalizar adecuadamente en el momento de usar imágenes reales, siempre y cuando se use la misma posición para caja y cámara.

Este modelo de inteligencia artificial fue entrenado desde Google Colab utilizando Keras, Tensorflow y redes convolucionales utilizando la arquitectura VGG16 preentrenadas con las imágenes de la base de datos imagenet. Los pesos de las últimas 2 capas de la red son entrenados con las nuevas imágenes y se añaden más capas que conectan completamente las neuronas de esta arquitectura con la salida, consistente de una única neurona con el valor esperado real entre 0 y 1.

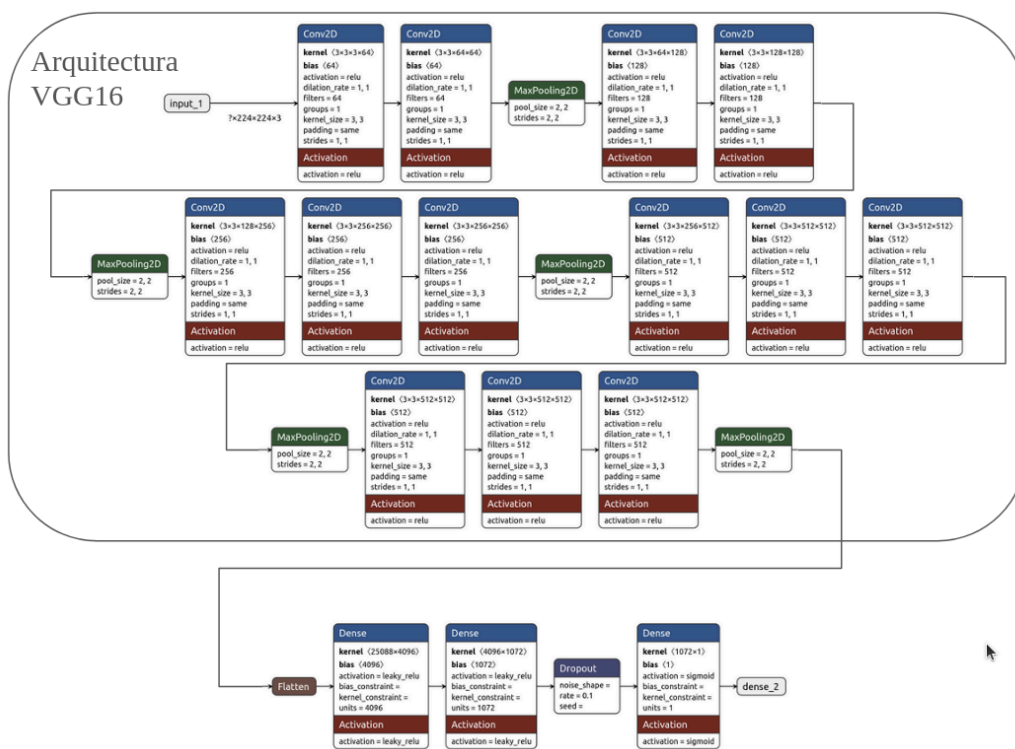


Figura 9. Arquitectura del modelo de detección de cajas

Se ha entrenado con 20.000 imágenes. La mitad de los datos de entrenamiento no contienen cajas, la otra mitad sí. Las primeras corresponden a una salida del valor 0 y las que contienen la caja esperan un valor de salida de 1.

Con los datos de validación se alcanza una precisión del 99.8%. Para los modelos entrenados para resolver este problema, no se ha incluido un corte de datos para testear la solución, puesto que se cree que, por ejemplo, el valor de precisión obtenido se corresponde con la precisión verdadera, puesto que los datos de validación y de entrenamiento están altamente vinculados a la manera con la que han sido generados, y por tanto es normal esperar un alto porcentaje de acierto. No obstante, no podemos verificar que la precisión es la verdadera, hasta que se cree un conjunto de datos con imágenes de cajas reales y podamos verificar que el modelo funciona y es capaz de generalizar al mundo real como sería esperado. Esta verificación queda aplazada a trabajos futuros en los que se consiga hacer funcionar al modelo en el escenario real.

También surgen dudas sobre el funcionamiento de este sistema, si tenemos en cuenta que la mitad de las imágenes tienen cajas y las otras no, pero ¿qué pasaría si la caja está en la imagen pero no está en la posición adecuada o está todavía de camino a ser depositada en la posición correcta? Para el modelo simulado es suficiente con cómo está planteado el modelo, pues las cajas simplemente van apareciendo en la posición deseada, pero una importante calibración a futuro entrenaría el modelo con imágenes en estas malas condiciones e indicaría al sistema que todavía la caja no está lista para ser recogida.

## 5.2.2 Modelo de cálculo de dimensiones de cajas

---

Una vez detectada una nueva caja, se vuelve a realizar otra foto, que esta vez es enviada de la misma manera al servidor de Flask, pero a una ruta diferente, esta vez para calcular las dimensiones de esta caja, en sus 3 dimensiones. El paquete de percepción de Unity permite generar imágenes anotadas en las que cada imagen puede venir anotada con distintos datos relativos a ella. Estos datos vienen dados en metros y vienen anotados en ficheros JSON separados complejos, por lo que para una fácil manipulación de estos datos desde Google Colab, es necesario procesarlos con un lenguaje de scripting, en nuestro caso se eligió el uso de PHP por su sencillez y manejabilidad y por cada conjunto de datos que generamos para los distintos modelos se tienen que procesar los distintos JSON y convertirlos en ficheros CSV para manipularlos con más facilidad desde Python en Google Colab.

Este modelo es prácticamente idéntico al anterior con el cambio significativo de la última capa, que pasa de tener una activación *sigmoid* a no tener activación, y de una neurona en la salida a 4, correspondientes a las 3 dimensiones de la caja en centímetros y a la ligera rotación que pueda tener la caja en el eje vertical “y”, aunque este último dato no se usa de momento y se asume que todas las cajas vienen dadas con la misma rotación por simplicidad.

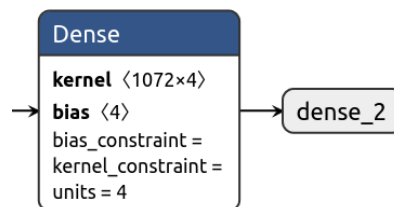


Figura 10. Capa de salida del modelo de cálculo de dimensiones de cajas

Las cajas pueden tener dimensiones de entre 4 y 9 centímetros en cada eje y han sido entrenadas con 100.000 imágenes de cajas de entre 3 y 15 centímetros inicialmente, y finalmente, cuando se decidió acotar el tamaño de las cajas, se volvió a entrenar con 10.000 imágenes con cajas de entre 4 y 9 centímetros para una mayor precisión en estos casos.

Las pruebas con el conjunto de validación dan un error cuadrático medio de 1.32 centímetros.

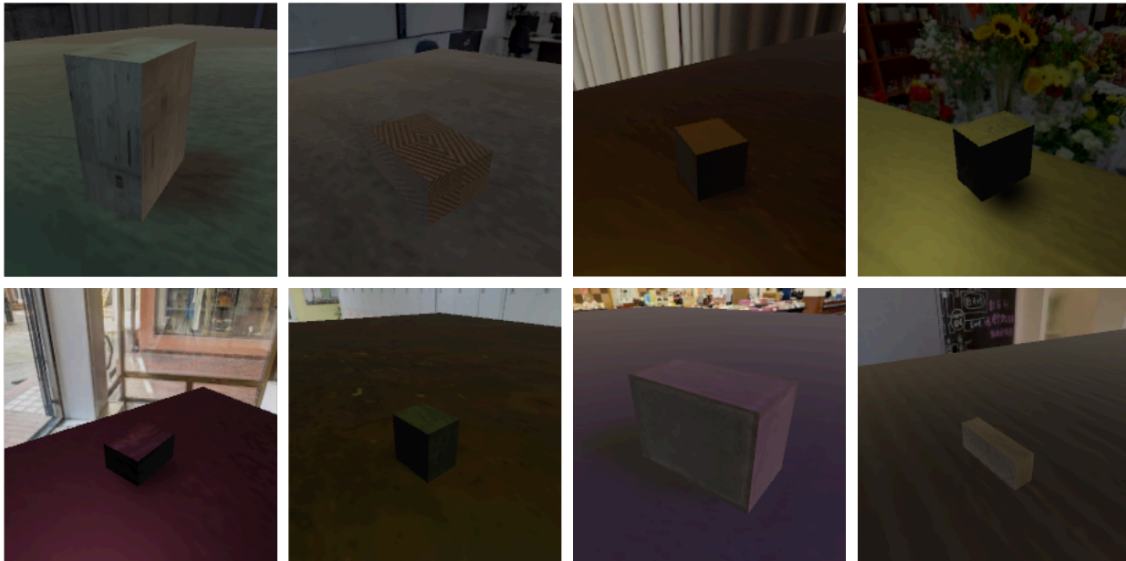


Figura 11. Ejemplo de datos de entrenamiento del modelo de cálculo de dimensiones de cajas

### 5.2.3 Almacenamiento temporal de cajas

---

Una vez se obtienen las dimensiones de la caja, ya se sabe la posición sobre la que coger la caja, puesto que esta posición viene dada por las posiciones “x” y “z” en las que se espera que se deposite y la altura a la que hay que recogerla dada por las dimensiones calculadas.

El siguiente paso es decidir qué acción realizar. El algoritmo para esta decisión es bastante sencillo. Lo primero y prioritario es siempre llenar los espacios de almacenamiento del robot primero que son 2. Es más probable encontrar una buena caja a depositar en el palé entre 3 posibilidades (los dos espacios de almacenamiento más la nueva caja) que entre una caja. Esta cantidad de dos espacios son un punto intermedio entre el compromiso de tratar de llenar el palé con velocidad y no tener que esperar a muchos espacios de almacenamiento para empezar a llenar el palé, y el optimismo de que con más cajas es más probable encontrar una adecuada para el palé.

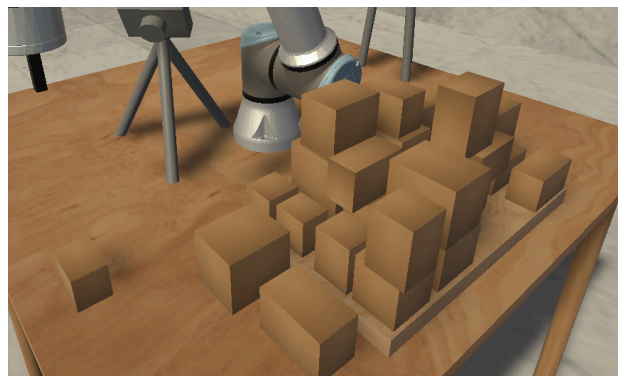


Figura 12. Escenario virtual con dos cajas en los espacios de almacenamiento

### 5.2.4 Algoritmo de asignación de cajas

---



Una vez todos los espacios de almacenamiento quedan llenos, la siguiente decisión es entre poner la caja en el palé o dar la caja al humano. Para realizar esta acción se usa un contador de cajas que empieza con el valor de 5, y que se va decrementando con cada caja que se incluye en el palé. De esta manera, las 5 primeras cajas del palé siempre son depositadas por el robot, ya que el algoritmo de apilamiento de esta funciona con una heurística que trata de balancear la distribución de cajas en el palé y, por tanto, para obtener buenos resultados es importante que sea inicializado por el robot y que el humano se adapte a ello consecuentemente.

### 5.2.5 Modelo de detección de pose del operario

Una vez 5 cajas han sido depositadas en el palé de entre las que llegan nuevas y las que se van incluyendo en el espacio de almacenamiento, la siguiente caja será ofrecida al trabajador. Antes de dar la caja al trabajador, se toman dos fotos del estado actual del palé desde dos cámaras elevadas en lados opuestos del palé que serán utilizadas más adelante, como explicaremos a continuación.

Mientras el operario maneja la caja ofrecida por el robot y la apila en el palé, el robot queda inmóvil esperando a que el humano coloque su caja por motivos de seguridad. Al mismo tiempo que esto ocurre, hay un contador de tiempo que analiza el tiempo que tarda el trabajador en cargar la caja.

También se toman fotos continuamente desde una de las cámaras elevadas que enfocan al palé y también enfocan al humano. Estas imágenes son enviadas continuamente al servidor de Flask, donde hay un servicio preparado para analizar la imagen en un paso parecido al que sigue el primer modelo de detectar si una caja se ha colocado en posición.

La diferencia es que este modelo ha sido entrenado con distintas imágenes, de distintos escenarios, con distintos humanos realizando distintas animaciones de movimiento. La idea es que el modelo aprenda a detectar al humano haciendo un gesto a la cámara con el pulgar arriba, expresando que ha apilado la caja en el palé satisfactoriamente y que el algoritmo de apilamiento puede continuar.



Figura 13. Ejemplo de datos de entrenamiento del modelo de detección de pose del operario

Este modelo ha sido entrenado de una manera similar a los anteriores, pero con los siguientes parámetros aleatorios modificados para asegurar variabilidad en los datos de entrenamiento:

- El humano tiene una posición aleatoria, una rotación aleatoria, es uno de entre 20 modelos de humanoides descargados gratuitamente desde la plataforma Mixamo y se mueve con una animación aleatoria de entre 40, también descargadas desde la plataforma Mixamo, donde 20 de ellas corresponden a variaciones con las dos manos de levantar el pulgar a la cámara y las otras 20 son animaciones distintas normales variadas.
- Como en el modelo anterior, la imagen de fondo, la luz, la mesa y las cajas tienen posiciones y texturas aleatorias.
- La posición y tamaño del palé son fijas, solo cambia su textura. En el palé aparecen distintas configuraciones de cajas apiladas aleatoriamente para simular que el algoritmo de apilamiento ha ido añadiendo cajas y añadir ruido para ayudar en el *domain randomization*.

Las estructuras fijas son la posición y ángulo de la cámara, la posición y rotación del palé y la posición y rotación del brazo robot, que está parado esperando a que se posicione la caja en el palé. Nos hubiera gustado que estos parámetros estuvieran más aleatorizados para poder asegurar que el escenario funcionara con múltiples variaciones a este, no obstante, distintas pruebas en esta dirección mostraron que era necesario mantener algunas posiciones fijas para mantener la precisión. En este modelo probablemente no sea tan imperativo mantener estas estructuras fijas, pues lo más importante probablemente para la inteligencia artificial a la hora de clasificarlos será si aparece un pulgar arriba o no, pero es clave para el siguiente y último modelo de inteligencia artificial que se comentará.

La arquitectura y funcionamiento de este modelo es similar al primero que detecta si se coloca una caja, pero con distintos datos de entrenamiento.

El entrenamiento se realizó con 30.000 imágenes, con una tasa de acierto en los datos de validación del 90%. Aunque el modelo funciona adecuadamente dentro de la simulación, queda verificar que este funcione en un escenario real, ya que los datos de validación incluyen los mismos modelos de humanos que los datos de entrenamiento. No obstante, el escenario final en el que se ejecutan las pruebas finales sí usa un modelo de personaje nuevo que no aparece en los datos de entrenamiento y el modelo funciona perfectamente.

### 5.2.6 Modelo de estimación de posición de cajas

---

Una vez el operario coloca la caja y comunica al robot que así ha sido, se toman otras dos imágenes del palé desde las dos cámaras elevadas que van a ser enviadas, junto a las fotos tomadas antes de apilar la última caja a un servicio de inteligencia artificial en el servidor Flask.

Múltiples versiones de este último modelo fueron desarrolladas hasta hallar la combinación que mejores resultados daba. La idea principal de este modelo es, utilizando la imagen del antes y después, detectar en qué posición, qué rotación y qué tamaño tiene la nueva caja depositada. La salida del modelo es de 6 valores, 3 correspondientes a la posición de la caja y 3 correspondientes a sus dimensiones, independientemente de la rotación de la caja. Unity da la posición de la caja en relación a la cámara que toma la imagen, por lo que es necesario realizar una transformación para que los valores vengan dados basándose en la posición del robot, puesto que hay dos cámaras tomando la foto. El modelo final que mejores resultados ha dado para este cometido procesa las imágenes de la manera explicada a continuación.

Primero realiza una operación de XOR en todos los bits de las 2 imágenes de cada cámara, de manera que se cree una imagen que solo destaca los píxeles con diferencias entre estas dos. Al acabar, tendremos dos imágenes correspondientes al antes y después de cada imagen, resaltando la posición de la caja nueva.



Figura 14. Operación XOR con las imágenes de antes y después

Después, para quitar el ruido producido por el movimiento del humano, que tendrá una posición diferente a la que tenía cuando se efectuó la foto por primera vez, se hace una operación AND con dos imágenes creadas anteriormente, diseñadas para solo dejar a 1 los bits de la imagen que se corresponden con el espacio que tienen las cajas en el palé, de manera que se ignore el movimiento que ha podido efectuar el humano o el entorno alrededor del palé.

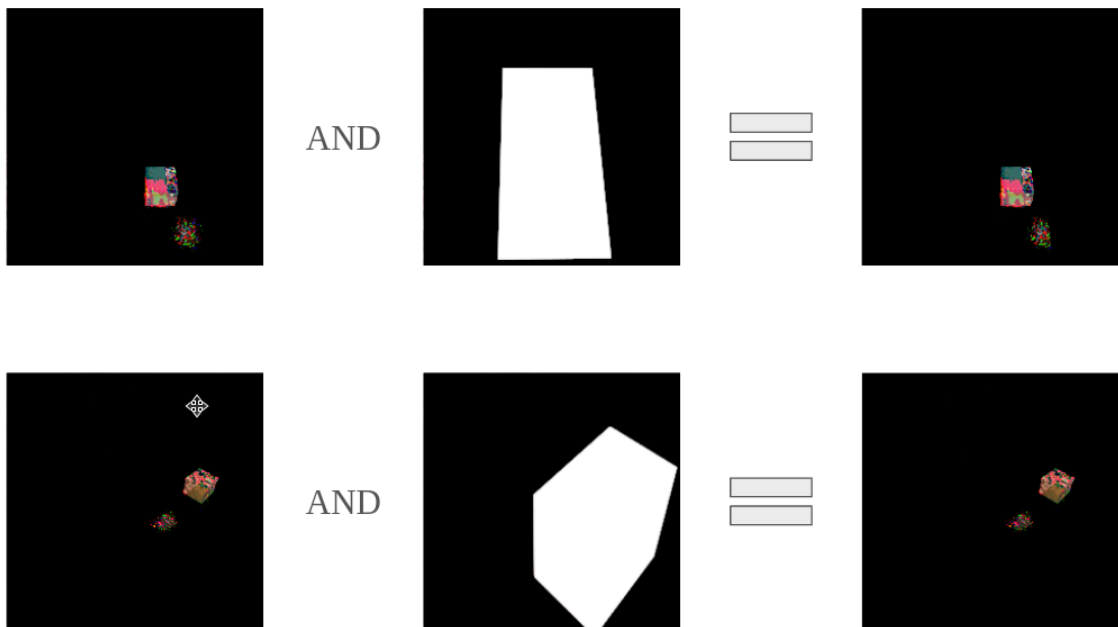


Figura 15. Operación AND con las imágenes XOR

En la figura 15 no hay un cambio significativo puesto que la posición del humano no ha cambiado, pero en algunos datos de entrenamiento esta operación sí tiene mucho más efecto.

Esta manera específica de tratar la imagen funciona con suficiente precisión, como para que el conocimiento del estado del palé sea actualizado por el algoritmo de apilamiento del robot y siga apilando de acorde a este nuevo cambio.

La arquitectura de este modelo es similar a los modelos anteriores, no obstante, al tener que procesar dos imágenes, se usa la arquitectura VGG16 dos veces.

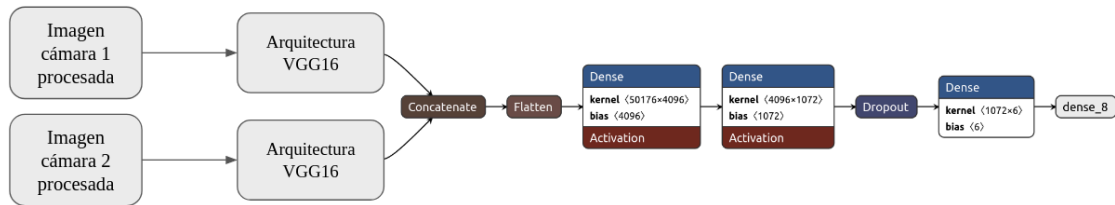


Figura 16. Arquitectura del modelo de estimación de posición de cajas

Las pruebas con el conjunto de validación dan un error cuadrático medio de 1.8 centímetros.

No obstante, analizando el modelo con más detenimiento, este puede que sea el que más dificultad tenga para ser transferido a cámaras reales, ya que ligeras vibraciones en la cámara o cambios en la luz entre la primera y la segunda imagen pueden hacer que la operación de XOR funcione de manera muy diferente a las imágenes virtuales, y, por tanto, haya que balancearlas mejor añadiendo ruido a los datos de entrenamiento, por ejemplo, para que el sistema siga funcionando. Esta hipótesis queda pendiente para ser analizada en trabajos futuros.

### 5.2.7 Heurística del algoritmo de asignación de cajas

Una vez todo este proceso de apilar del humano ha tomado lugar, se para el contador de tiempo que ha estado analizando el tiempo que ha tardado el humano en apilar la caja y este tiempo se utilizará para calcular cuántas cajas debe esperar el humano para apilar una caja de nuevo. Esta función es muy simple, cada 10 segundos que tarde el operario en apilar la caja, este deberá esperar una caja más para apilar de nuevo, más una caja por defecto. Por ejemplo, si tarda 26 segundos en apilar la caja, se divide entre 10 (2.6), se trunca al entero inferior (2) y se añade 1 (3), por lo que tendrá que esperar 3 cajas hasta añadir otra caja más.

Esta heurística viene dada por distintos razonamientos que ahora se explicarán.

Los parámetros específicos del cálculo pueden ser retocados para adaptarse mejor a cada trabajador o manera de resolver el problema. Lo que sí es intrínsecamente importante es la base de la heurística, en la que cuanto más tiempo tarde en apilar la caja, más tiempo se tardará en ofrecer otra caja para apilar, y cuanto menos tiempo tarde, menos tiempo se tardará en ofrecer otra caja. Esto viene primero de la idea de tratar de adaptar la dificultad del desafío para el humano a la habilidad que este tenga y segundo de la idea de que cuanto más cansado esté, más tardará en apilar, y por tanto para no penalizar aún más este cansancio, es mejor no ofrecerle más trabajo inmediatamente.

La primera idea viene dada por el modelo de *flow* estudiado intensivamente por Csikszentmihalyi, (Csikszentmihalyi, 1988 [15]). La idea principal de este modelo psicológico es encontrar el estado óptimo con el que realizar una actividad, y argumenta que para alcanzar este estado óptimo, la dificultad de la tarea y la habilidad del humano para realizarla deben encontrarse en un punto intermedio, pues si la tarea es muy difícil para el humano, este puede

llegar a agobiarse, y si es muy fácil, este puede llegar a aburrirse, por lo que facilitando un entorno en el que la dificultad (cantidad de cajas ofrecidas al trabajador) y la habilidad (tiempo que tarda en apilarlas) se autorregulen, podemos llegar a alcanzar este estado óptimo o de *flow*.

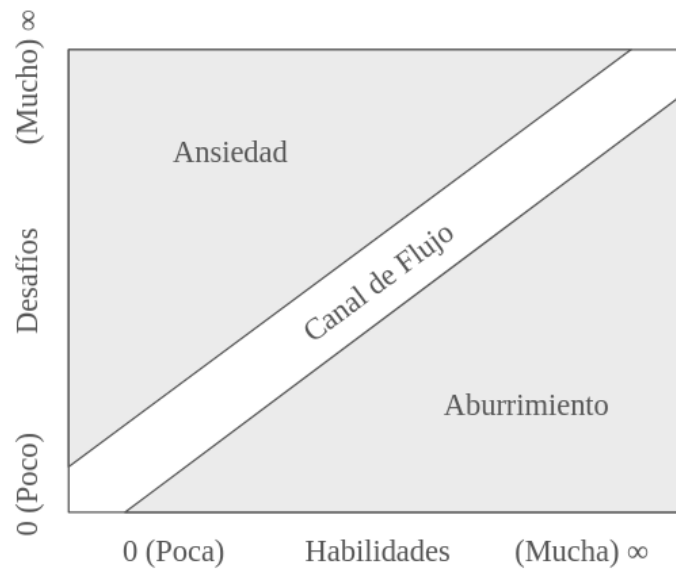


Figura 17. Modelo tradicional de *flow*

La segunda idea también se alcanza al derivarse de la primera, pues si no está cansado probablemente pueda apilar cajas a más velocidad y si lo está probablemente necesite tiempo para recuperarse.

Al fin y al cabo, estas ideas vienen potenciadas por mantener la calidad de trabajo del humano al máximo nivel posible y tenerlo siempre en el centro de la toma de decisiones del robot, de manera que este se adapte a las preferencias del humano sin que este lo tenga que solicitar directamente, y que, por tanto, sean lo menos intrusivas posible.

### 5.2.8 Colaboración humana

Para que el sistema utilice la inteligencia humana de la manera más fiel posible, sin tratar de simularla, un humano desde el editor de Unity debe desplazar las cajas al palé de la manera que desee y señalar al sistema que ha colocado la caja cambiando la animación del humano de una posición normal de espera a una en la que muestre el dedo del pulgar arriba a la cámara. El humano tiene libertad para mover y rotar la caja de cualquier manera, usando las mismas herramientas que ofrece Unity para desplazar objetos dentro de la escena, puesto que al estar integrado en el editor esta era la manera más sencilla de hacerlo, aunque una mejora a futuro puede ser convertir el entorno completamente en una especie de videojuego que integre estos sistemas de nuevo completamente.

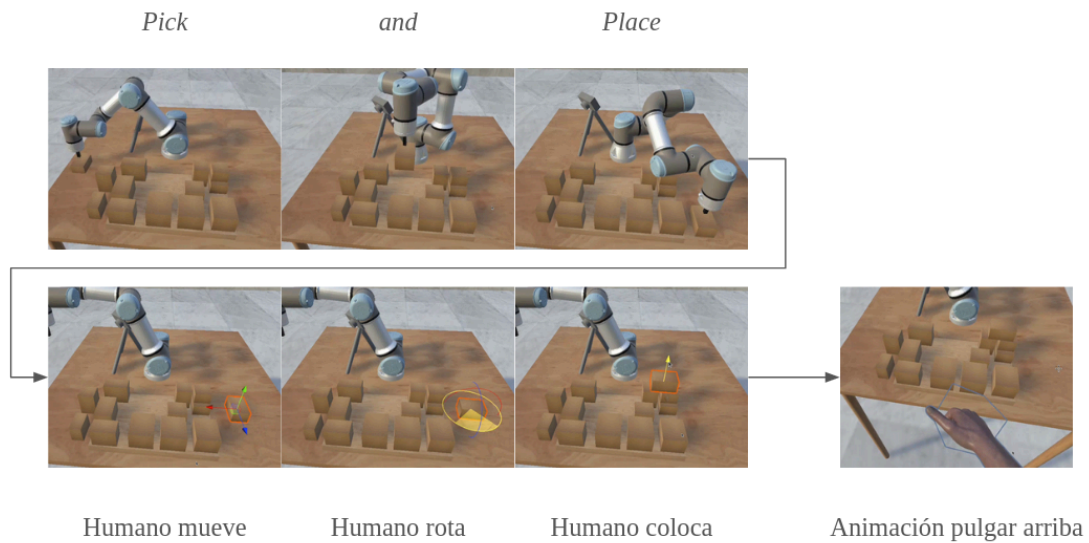


Figura 18. Ejemplo de colaboración humana en paletización

### 5.2.9 Algoritmo de paletización

El robot utiliza un algoritmo heurístico para cargar el palé. El algoritmo es bastante sencillo, aunque un tanto más complejo que el algoritmo heurístico estándar utilizado para resolver este tipo de problemas, el nombrado DBL (Deepest Bottom Left), que trata de almacenar cajas siempre lo más al fondo a la izquierda posible.

Nuestro algoritmo, en cambio, trata de mantener la distribución de peso equitativa en todas las esquinas, por lo que trata de apilar cajas siguiendo 4 esquinas, lo más DBL, DBR, DTL y DTR (Deepest Bottom Left, Deepest Bottom Right, Deepest Top Left y Deepest Top Right). Además, para no tener en cuenta todas las posibles posiciones en las que se puede colocar cada caja, inicialmente solo se consideran 4 posiciones, correspondientes a cada esquina del palé.

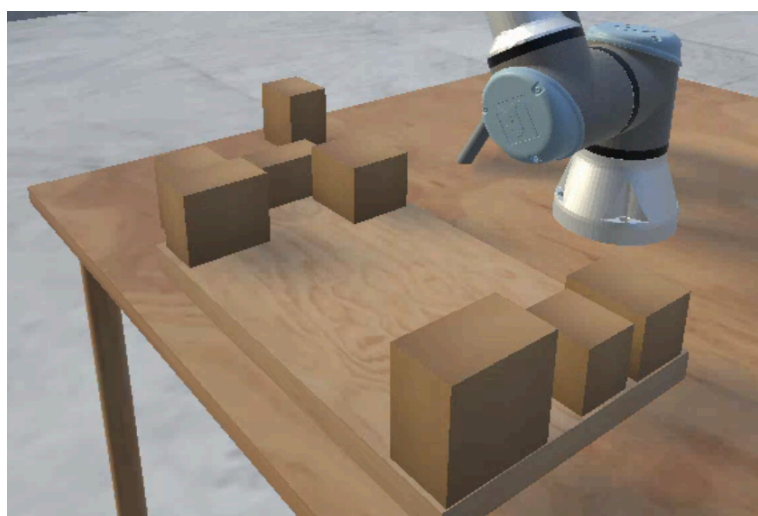


Figura 19. Ejemplo de inicio de algoritmo de apilamiento, con cajas en cada esquina

Una vez se coloca una caja en una de las esquinas, esa posibilidad se elimina y surgen 3 posibilidades más en las que colocar una caja, 2 a cada uno de los lados de la caja más cerca del

centro de la estructura y una arriba de esta. Este principio funciona para todas las cajas que se colocan, por lo que el crecimiento del palé en la primera de cajas crece desde las esquinas hacia el centro del palé.

El robot siempre tiene en cuenta las dimensiones y posiciones de todas las cajas del palé para asegurarse de que la nueva caja a colocar no colisiona con ninguna de las ya posicionadas.

Entre las 3 cajas posibles a colocar en cada momento, se decide entre una de las 3 teniendo en cuenta diversos factores y heurísticas, estos son los más importantes: dimensiones de la caja, dimensiones de la caja sobre la que se coloca y si la altura de la caja coincide con la altura de otra caja ya colocada. Esta última es muy importante, puesto que si colocamos juntas cajas con la misma altura, esto creará oportunidades para el operario humano de entrelazado.

Dado que la visión del robot es limitada, no podemos asegurar que estas oportunidades sean verdaderamente útiles, por lo que el robot no es capaz de utilizar esta oportunidad de entrelazado para poner una caja entre las dos que tienen la misma altura. El entrelazado asegura la rigidez de la estructura si funciona a distintos niveles, y muestra una de las ventajas que tiene el humano para apilar sobre el robot y por qué esta simbiosis puede llegar a ser útil para resolver el problema.

Además, el robot trata de encajar cada caja con las dos posibles rotaciones perpendiculares que esta puede tener al ser rotada desde el eje vertical “y”. Esto muestra otra ventaja que tiene el humano sobre el robot, y es que el robot sólo puede rotar la caja en 2 posiciones diferentes, en cambio, el humano puede rotarla en los 3 ejes principales, ofreciendo 6 posibles orientaciones diferentes para cada caja.

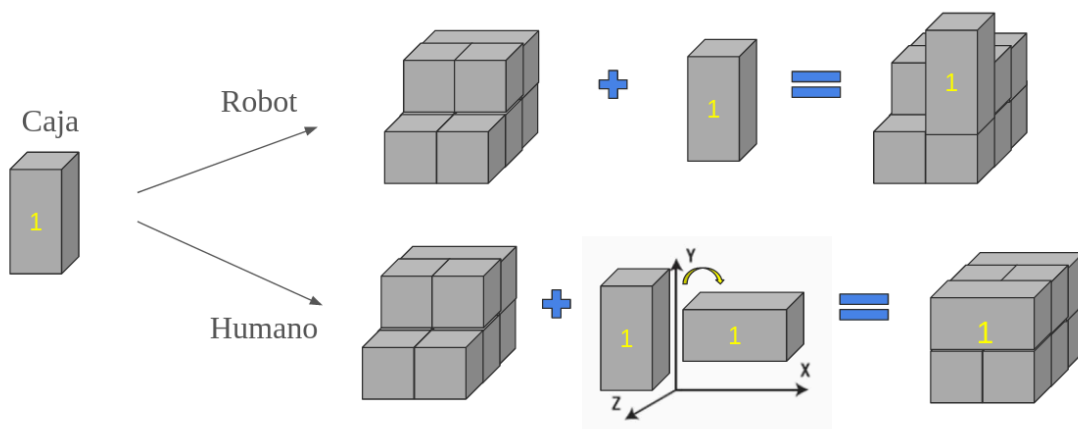


Figura 20. Ejemplo de ventajas del humano sobre el robot: rotación y entrelazado

Estos ejemplos muestran la capacidad que tiene el humano para mejorar las carencias del robot y que ambos se vean beneficiados.

### 5.2.10 Planificador de trayectorias del brazo robot

Una vez decididas las posiciones de *pick* y *place* de la caja, es necesario transformar estos 2 puntos del escenario de Unity en 2 puntos dentro del escenario del planificador de trayectorias del brazo robot. Este planificador se encuentra dentro de un servicio lanzado desde el contenedor con el servidor de ROS, la petición se hace con mensajes TCP de ROS, Unity transforma estas posiciones al marco de referencia del planificador en ROS. Este planificador,

planifica cada trayectoria paso a paso, punto por punto, teniendo en cuenta restricciones y colisiones como la que tendría con la mesa y en total para cada secuencia usa 11 puntos intermedios. El brazo robot deberá pasar por cada uno de ellos para completar la trayectoria. Esto garantiza que la planificación se ejecute correctamente y las trayectorias sean simples.

Es posible que una trayectoria sea imposible de realizar por el brazo robot, porque no sea alcanzable o el sistema tarde demasiado en encontrar una solución posible. En este caso, Unity descarta la orden dada y trata de encontrar otra de entre las posibilidades de posicionamiento que quedan.

El robot trata de no colisionar con las cajas situadas en el palé, alzando el brazo sobre estas, no obstante, esto puede llegar a fallar, ya que solo Unity tiene idea de las cajas apiladas en el palé. Una posible mejora a futuro es informar también al planificador de las nuevas cajas en el palé y que este las tenga en cuenta a la hora de calcular las nuevas trayectorias. No obstante, distintas pruebas muestran que cuantas más restricciones se añaden al planificador, más lento es su algoritmo de encontrar el camino adecuado, hasta llegar al punto en el que le cueste demasiado obtener una solución factible incluso cuando esta sea posible. Por lo que hay que tener cuidado de no añadir muchas restricciones y habrá que estudiar mejor este tema para posibles mejoras del sistema.

### 5.2.11 Paletización completada

---

Finalmente, llegará un punto en el que el robot no tenga más opciones para apilar, puesto que la estructura sea demasiado alta y, por tanto, el planificador de trayectorias no encuentre ninguna posición adecuada, o que las cajas sean demasiado grandes para apilarlas en estas partes superiores del palé con seguridad. Cada posible posición, caja y rotación realiza un cálculo heurístico al que se le establece una puntuación teniendo en cuenta qué tan adecuada puede llegar a ser, por lo que también existe un sistema pensado para no apilar cajas cuando no se encuentra una posición lo suficientemente adecuada. En este caso, la solución es enviar la caja al trabajador, y que este sea el encargado de situar la caja en la mejor posición que él considere, aunque no se haya alcanzado el número de cajas restantes estipuladas para dar la caja al trabajador.

En este punto, el sistema espera a que el trabajador apile la caja y puede pasar que el operador considere que el palé ya está lo suficientemente lleno y que, por tanto, no es posible añadir más cajas. Se puede considerar que el sistema se acaba cuando el trabajador no coloca la caja que se le ha ofrecido o la coloca, pero no le señala al robot que continúe con un gesto de aprobación alzando el pulgar arriba. Una posible mejora a futuro del sistema puede contemplar que el trabajador sea el encargado de desplazar el palé lleno a otro sitio, este coloque un palé vacío en la posición del anterior y este señalice a la cámara con otro tipo de gesto que se desea resetear el sistema y que, por tanto, se empiece a apilar otro palé de cero.



## Desarrollo de la Solución Propuesta

El desarrollo de esta solución ha supuesto un largo trabajo desde la concepción de la idea a implementación de la misma, que muchas veces ha derivado en caminos sin salida y ha hecho que muchas veces se tuviera que cambiar de idea o empezar de cero. En esta sección se explicará el proceso de desarrollo que tuvo la solución, con especial hincapié en las ideas iniciales y cómo estas han ido variando en el tiempo.

El primer paso era idear un escenario en el que humano y robot trabajaran colaborativamente. La primera idea de escenario consistía en un problema simple de empaquetamiento, en el que una serie de productos individuales venían en una cinta transportadora y era la labor del humano y del robot de empaquetarlos en distintas cajas de 3 tamaños posibles, de 1 unidad, 3, o 6. Estos distintos paquetes serían asignados a humano y robot dependiendo de distintos factores, por ejemplo, se pretendía que hubiera sensores, como un analizador de frecuencias cerebrales en la cabeza del trabajador, y de esta manera tratar de asignarle cajas más fáciles o más difíciles dependiendo de su estado, si fuéramos capaces de medir con este casco, por ejemplo, su nivel de cansancio.

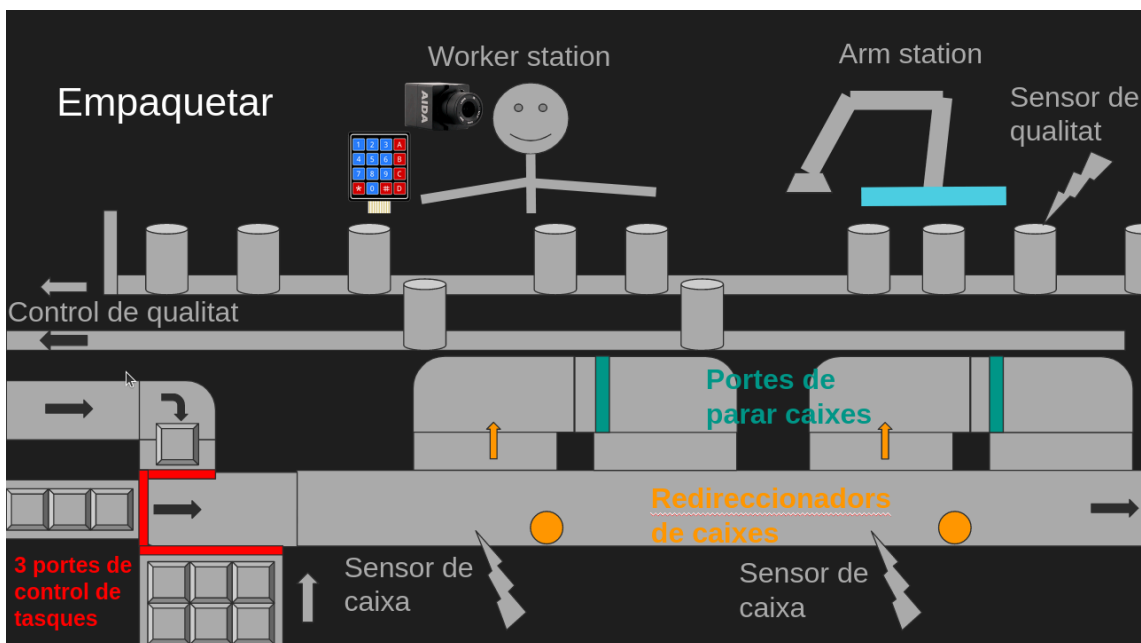


Figura 21. Boceto inicial del primer escenario descartado

Esta idea, aunque prometedora, no cumple con muchos de los requisitos que se buscaban con el escenario. Por ejemplo, la tarea no es lo suficientemente compleja como para verdaderamente mostrar diferencias cognitivas o de resolución de problemas entre humano y máquina como el escenario final sí muestra. La colaboración directa entre humano y máquina también es muy escasa. Además, el casco medidor puede ser un tanto intrusivo para el trabajador y no permite su fácil implementación en las fábricas reales, pues probablemente existan regulaciones que obstaculicen el uso de estas tecnologías en el trabajo, pues pueden ser negativas para el trabajador si se usan con objetivos malintencionados, como tratar de comparar el rendimiento de estos basándose en sus señales vitales.

Al siguiente escenario al que se llegó sí era un escenario de paletización, no obstante, había muchos elementos más complejos en el sistema como cintas transportadoras de cajas y de palés para compartir cajas entre humano y máquina y llenar diversos palés al mismo tiempo. Este escenario es bastante más ambicioso que el usado finalmente, pero al fin y al cabo es menos realista si se pretende hacer pruebas fácilmente en el futuro, puesto que no contamos con cintas transportadoras en el laboratorio, y resuelve el problema de una manera menos general, ya que las cintas transportadoras se usaban también para asignar cajas a humano y máquina. De esta manera, el escenario es un poco más sencillo y controlado, ya que solo hay un punto de entrada de cajas en el sistema y el robot es el encargado directamente de decidir qué hacer con ella.



Figura 22. Boceto inicial del escenario de paletización complejo descartado

Una vez ya se acotó más el escenario final, aún quedaban ideas en el aire sobre cómo medir el estado del trabajador para adaptarse a este de la mejor manera. Una de ellas era utilizar un reloj *tracker* como Fitbit para contar por ejemplo el número de pulsaciones por minuto del trabajador y poder analizar mejor si se encuentra cansado, agobiado o aburrido, entre otros. Esta idea también ha sido desechada de momento por las mismas dificultades planteadas en el caso del casco analizador, pero no se descarta para trabajos futuros, ya que puede ser un área interesante de investigación.

En este punto del desarrollo se tenía claro el escenario final, que es el mismo que se ha presentado en este trabajo, y se dividió el trabajo en 4 grandes bloques de problemas a solucionar. El primer bloque es el modelo de asignación de cajas, el segundo es el algoritmo encargado de apilar las cajas en el palé, el tercero es el problema de transformar una posición de *pick* y otra de *place* en trayectorias que pudiera seguir el robot para mover las cajas y el cuarto es el modelo de percepción encargado de entender el entorno del robot.

Estos bloques, aunque pudieran haberse completado en cualquier orden, se completaron en el orden inverso al que se ha comentado anteriormente, puesto que se quería abordar los problemas más difíciles primero y el problema de la percepción era una gran incógnita que hasta que no se empezara a probar no se saldría de dudas de cómo resolver.

El primer paso en el modelo de percepción, fue tratar de analizar el escenario entero con una única cámara. Esta cámara trataría de entender el escenario completamente, y se trataría de que esta fuera capaz de detectar la posición, tamaño y rotación de todos los elementos del entorno, desde la mesa, al palé, al robot, a las cajas, al humano y al robot. Además, se trataba de que la posición y la rotación de la cámara fueran aleatorias también y que, por tanto, toda esta información se calculara en relación a la cámara. Este primer paso fue fruto de la inexperiencia a la hora de abordar estos problemas, ya que rápidamente fue claro que el escenario era demasiado ambicioso y genérico como para funcionar, era necesario acotar mejor la variedad del escenario y siguientes iteraciones del desarrollo trataron de fijar algunos valores como la posición de la cámara y del robot o de otros elementos, etcétera. Además, tratar de obtener la posición de todos los elementos del escenario no era tarea sencilla, había que concretar mejor qué información nueva era necesaria saber y qué información ya habíamos obtenido anteriormente y, por tanto, se podía reducir y mejorar el resultado de los modelos de visión artificial.

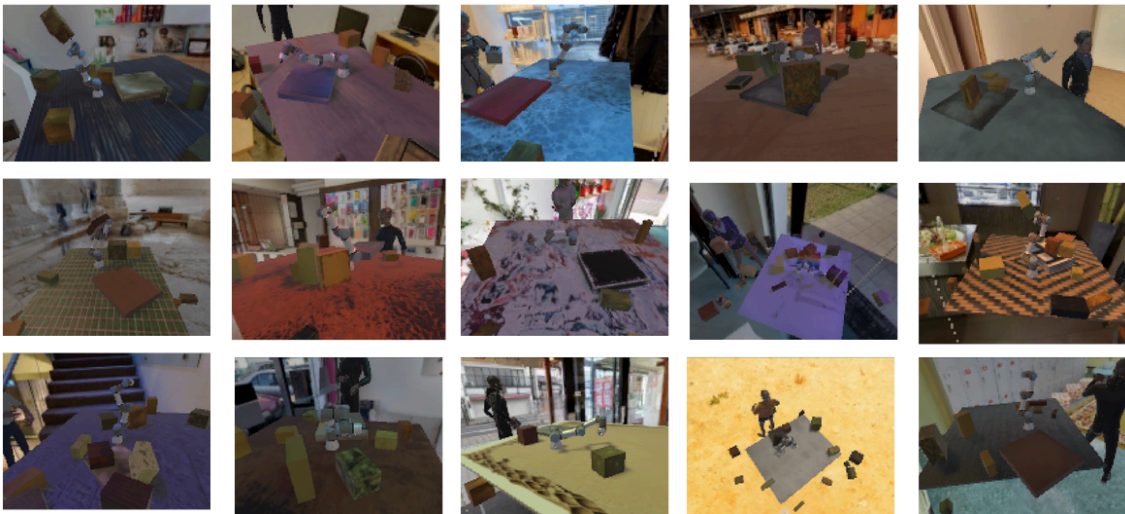


Figura 23. Conjunto de imágenes de entrenamiento de escenario genérico descartado

Para estos modelos de visión artificial, la idea inicial era utilizar redes conocidas como Liquid Time-Constant Neural Networks (Hasani et al., 2021 [17]), ya que habíamos analizado otros trabajos que habían logrado implementarlas junto a redes convolucionales y mostraron resultados prometedores (Chahine et al., 2023 [18]). Estas redes son muy útiles para entradas de distinto tamaño y de información secuencial, ya que son modelos causales capaces de extrapolar la causalidad del problema que están resolviendo. Y dado que se pretendía grabar continuamente la escena, se pretendía probar si estas redes eran lo suficientemente potentes como para entender el mundo en 3D a través de imágenes en 2D. Como los resultados no fueron satisfactorios y era más fácil desarrollar un sistema que tomara fotos en vez de grabar continuamente, se desechó la idea de utilizar este tipo de redes y se volvió a redes neuronales convolucionales convencionales, ya que ofrecían más garantías de funcionar.

Al final, se acotó el modelo de percepción de la manera que ha sido descrito anteriormente, con pequeños cambios y ajustes que han ido surgiendo a lo largo del desarrollo de este y de los otros bloques para tratar de mejorar el funcionamiento de estos sistemas de visión hasta que se alcanzó un estado satisfactorio.

Después se continuó con el planificador de trayectorias del robot, al que hubo que añadir muchos puntos intermedios para que este encontrara soluciones sencillas y rápidas. Si la distancia entre un punto y otro era muy larga, el planificador podía o no llegar a resolver el problema y quedarse atascado, o encontrar la solución difícil dando la vuelta al robot por el lado contrario al que se esperaba que el robot rotara, tomando una rotación más larga. Se trató de

añadir restricciones para que el robot no tratara de dar la vuelta por el lado equivocado pero al añadirlas, el robot era incapaz de resolver el problema puesto que las restricciones hacían que el robot calculara la trayectoria mucho más lento hasta llegar a un punto de *time out* y parar. Por lo que usar muchos puntos intermedios fue la mejor solución encontrada y se deja para el futuro probar otros planificadores más inteligentes que no tengan estos problemas de cálculo y restricciones.

Para el algoritmo apilador de cajas, había poca literatura sobre soluciones heurísticas válidas para palés, la mayoría simplemente trataban de resolver el problema para contenedores en los que la rigidez o estabilidad no fuera importante, por lo que se tuvo que diseñar un algoritmo desde cero adaptado a nuestras necesidades y a la posibilidad de que el humano ayudara a apilar. Otra opción que contemplamos muy al principio era la de utilizar un algoritmo de aprendizaje por refuerzo para resolver este problema, pero al final se decidió optar por la solución más sencilla y dejar esta posible solución, muy prometedora, para el futuro.

Este último algoritmo heurístico y el de asignación de cajas a humano o robot se realizaron a la misma vez, pues se acabaron integrando los dos algoritmos en uno por sencillez y estar altamente conectados. Además en este punto ya se podía conectar todos los demás elementos con el bucle del algoritmo principal y probar que todo funcionara correctamente y hacer las primeras pruebas en las que colaborar humano y máquina para resolver el problema de paletización.

# Pruebas

---

Como se ha descrito anteriormente, este proyecto cuenta con distintos elementos altamente conectados que dependen directamente entre sí, y, por tanto, las distintas pruebas que se hagan al sistema pueden verse afectadas en gran parte por uno de estos elementos y acabar afectando en cascada a todos los demás. Por ejemplo, si el modelo de detección del tamaño de la caja falla, el tamaño de la caja será erróneo y, por tanto, esto afectará al correcto sistema de apilamiento de cajas. Por lo que es importante que todos los sistemas estén bien calibrados y funcionen correctamente todos ellos.

Este es un sistema cerrado completo que trata de resolver el problema de paletización de inicio a fin y, por tanto, es difícil de medir con exactitud la calidad de la solución ofrecida en comparación a otras, pues hasta donde conocemos no existen soluciones parecidas colaborativas a las que se ofrecen en este proyecto.

Por lo tanto, las pruebas que vamos a realizar en esta sección están más centradas en tratar de ver si las heurísticas que se han incluido en los distintos algoritmos de la solución son lo suficientemente buenas, si el palé apilado llega a ser de la suficiente calidad teniendo en cuenta todas las dificultades del sistema, los puntos fuertes y flacos de humano y máquina y la experiencia que tiene el trabajador al operar colaborativamente con el robot.

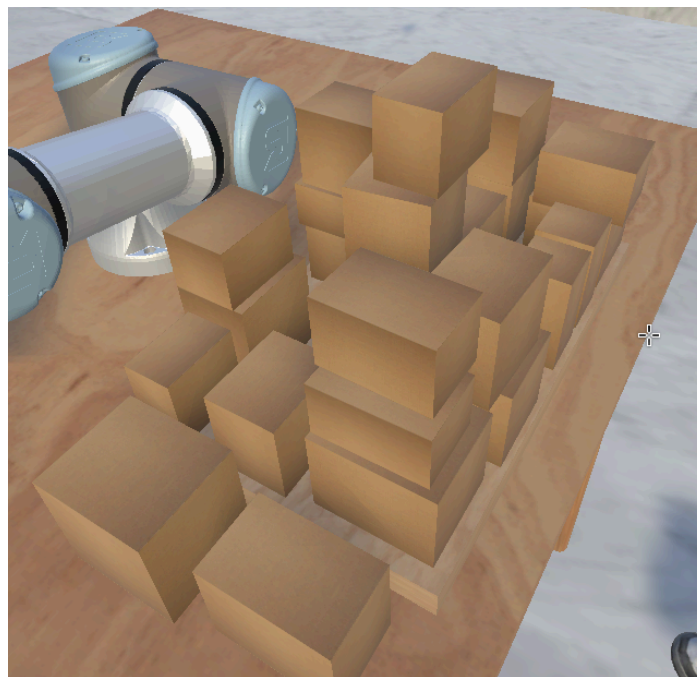


Figura 24. Ejemplo de palé montado íntegramente por el robot

El algoritmo de apilamiento del robot no entrelaza las cajas, por lo que la estructura creada en la figura 24 está formada por una serie de monolitos independientes. Además, ha llegado a un punto en el que las cajas que tiene disponibles son demasiado grandes como para colocarlas encima de las ya presentes en el palé, por lo que no puede continuar.

Tras probar el escenario, la heurística de tratar de poner cajas de la misma altura juntas es bastante útil, aunque dada la variación de tamaño de cajas continua, es difícil conseguir que otra oportunidad de entrelazado vuelva a surgir en niveles más elevados de la estructura, pues simplemente la altura a la que consigue apilar el robot no es lo suficientemente alta como para que sigan apareciendo estas oportunidades. Lo que reduce la calidad de la estructura y nos da indicios para poder mejorar el sistema en un futuro, por ejemplo, desplazando la altura del palé para poder seguir apilando cajas y que surjan más oportunidades de entrelazamiento.

Las estructuras que se forman son bastante adecuadas y en ningún instante se han venido abajo hasta que por accidente el brazo robot o el humano han entrado en contacto con ellas, por lo que, aun pudiendo observarse margen de mejora en algunas estructuras en las que hay demasiado espacio entre cajas, la solución parece bastante adecuada teniendo en consideración todos los posibles problemas que surgen al tratar de resolver este problema.

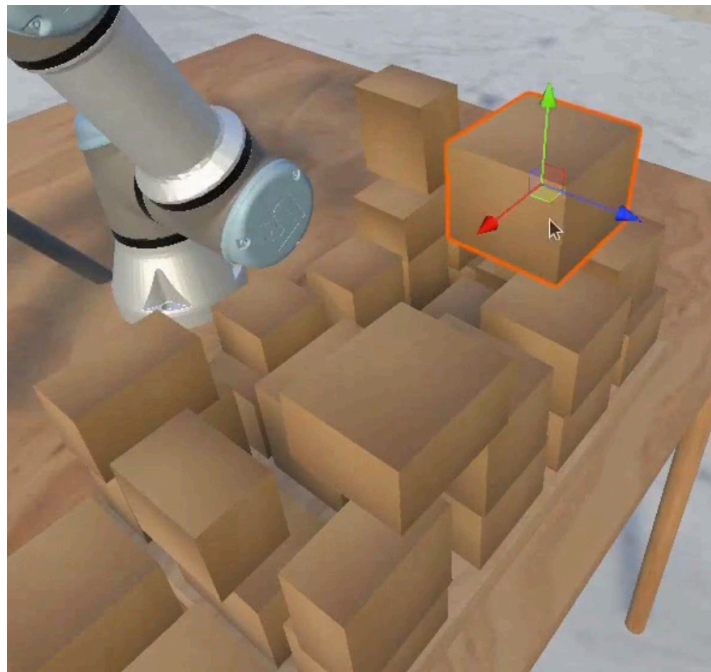


Figura 25. Ejemplo de palé montado colaborativamente

En cuanto a las diferencias entre humano y robot, el robot puede llegar a funcionar a mucha más velocidad que el humano y sus fallos de precisión vienen dados por los sistemas de visión artificial que usa o la precisión de este también sería mucho mejor que la del humano. Sin embargo, por motivos ergonómicos, el robot tiene menos libertad a la hora de rotar la caja y, por tanto, el humano ofrece ventajas a la hora de encajar cajas en lugares en los que el robot no puede al no poder efectuar la rotación necesaria. En este caso, el algoritmo de apilamiento del robot es bastante rígido, y por tanto, la creatividad del humano puede ser una gran ventaja, no obstante el robot puede tratar de buscar rápidamente todas las soluciones posibles y probar si funcionan directamente hasta encontrar rápidamente la posición que él cree más adecuada, mientras el humano puede debatir y tardar en escoger la posición más adecuada en este momento. En este caso de uso, se trata de utilizar al robot para realizar la mayor parte del trabajo, ya que es más rápido que el humano y no se cansa y se utiliza la inteligencia y creatividad del humano para posicionar cajas en posiciones que el robot no había tenido en cuenta, para de esta manera encontrar una mejor solución al problema que la que pudieran haber encontrado individualmente (ahorrando más espacio, utilizando menos tiempo, creando una mejor estructura, etc.).

En cuanto a la experiencia del trabajador, esta es bastante satisfactoria, mucho mejor que si el trabajador tuviera que montar todo el palé por supuesto, y además el trabajador es capaz de

ajustar cuánta carga de trabajo desea, tardando más o menos en apilar la caja que se le asigna. Además, su esfuerzo obtiene recompensa cuando con su creatividad apila una caja en una posición que no era posible para el robot y crea una mejor estructura en general, lo que puede transformarse en mayor satisfacción con menor esfuerzo.

Además, la labor del trabajador puede llegar a ser muy útil cuando el robot comete algún fallo, lo que sería muy difícil si no hubiera trabajadores humanos revisando el escenario. Por ejemplo, el robot podría fallar y colocar una caja en una posición no adecuada. En este momento, el operario puede decidir intervenir y colocar la caja en la posición en la que el robot deseaba colocarla para que no haya una gran diferencia entre el mapa mental que tiene el robot sobre el palé y las cajas reales en el palé.

En resumen, el sistema es lo suficientemente potente como para resolver el problema, aunque todavía queda margen para mejorar.







# Conclusiones

---

El objetivo principal del trabajo era diseñar e implementar un escenario en el que humano y máquina trabajen colaborativamente para resolver un problema de paletización. Dados los resultados de las pruebas mostradas anteriormente, se puede asegurar que se ha cumplido este propósito principal.

Desafortunadamente, no se ha conseguido ejecutar el sistema en un entorno real por limitaciones de tiempo, pero al fin y al cabo, se cree que este objetivo ha estado muy cerca de cumplirse, puesto que el escenario virtual trata de imitar en lo máximo posible al real y solo quedaría adaptar un poco los sistemas si fuera necesario, aunque no se puede asegurar con certeza hasta que no se implemente.

Se ha tratado de mantener el sistema lo más fiel a la realidad posible, aunque por algunas restricciones, como por ejemplo, las diferencias que habría entre una imagen tomada en un entorno virtual en 3D y una imagen tomada con una cámara real, no se puede asegurar con total seguridad que el sistema pudiera funcionar sin tener que adaptar algunos de sus elementos para un escenario completamente real.

Se ha conseguido el objetivo de permitir al humano apilar cajas con el robot y que no se tenga que simular esta parte.

El sistema de visión artificial, aunque con sus limitaciones para entender el entorno en su totalidad, es lo suficientemente bueno como para permitir al sistema funcionar. Queda comprobar que este sistema funcione en cámaras reales.

El algoritmo de apilamiento es lo suficientemente adecuado como para funcionar correctamente y conseguir ocupar el máximo espacio, teniendo en cuenta también que los sistemas de visión artificial pueden llegar a hacer que la calidad de este sistema no reluzca tanto como podría.

El último objetivo de ofrecer una experiencia estimulante y no fatigante al trabajador es un tanto subjetiva, pero dado que los sistemas se han diseñado con esta idea en mente y funcionan adecuadamente, creemos que es adecuado decir que se ha cumplido.

En general, dado que el proyecto es muy extenso y requiere del correcto funcionamiento de muchos sistemas con diferentes tecnologías, este ha resultado una experiencia bastante desafiante, de la que se ha podido aprender mucho de distintas tecnologías que no se ven en la carrera, o se han podido poner en práctica algunos de los conocimientos que sí se habían visto y que han facilitado la elaboración de este TFG.

Las áreas que más problema han planteado han sido las relacionadas con la robótica, la creación de datos sintéticos de entrenamiento desde Unity para los distintos modelos de inteligencia artificial, y sobre todo, los distintos modelos de visión artificial que permiten que el robot pueda relacionarse con su entorno.

No obstante, al final todos estos problemas han podido superarse en menor o mayor medida y valoramos que este TFG ha resultado ser una experiencia muy enriquecedora.

## 8.1 Relación del trabajo desarrollado con los estudios cursados

---

El algoritmo diseñado para llenar el palé de cajas está muy vinculado a la asignatura de Algorítmica, dado que se estudian variaciones diferentes de este tipo de problema.

Los distintos elementos de redes y de despliegue de aplicaciones en el uso de Docker y distintos sistemas de comunicación entre Unity y los distintos servidores están muy relacionados con la asignatura de Tecnología de Sistemas de Información en la Red.

El algoritmo de comportamiento del robot está muy relacionado con la asignatura de Agentes Inteligentes.

Los distintos modelos de inteligencia artificial están muy vinculados a las asignaturas de Aprendizaje Automático y Percepción.

El manejo del entorno de Unity está muy relacionado con las asignaturas de Sistemas Gráficos Interactivos y Desarrollo de Videojuegos 3D.

En cuanto a competencias transversales clave para el desarrollo del proyecto, destacaría el análisis y resolución de problemas, por la variedad de problemas a resolver a lo largo del proyecto, la innovación, creatividad y emprendimiento, por el aspecto innovador de la solución planteada y la comunicación efectiva, pues ha sido muy importante la comunicación y poder transmitir las diferentes ideas adecuadamente.

## Trabajos Futuros

---

A lo largo de la memoria ya se han ido indicando distintos puntos mejorables o que se dejan para trabajos futuros y en este apartado se tratará de enumerarlos.

- Llevar el sistema virtual al entorno real. Por suerte, este objetivo no se ha quedado muy lejos y puede que se alcance pronto.
- Desarrollar videojuego interactivo a partir del sistema virtual diseñado para hacer la solución más portable y que pueda ser compartida más fácilmente.
- Mejorar el comportamiento del planificador de trayectorias del brazo robot.
- Modificar el sistema para que la ejecución continúe después del primer palé y se pueda seguir apilando palés infinitamente, añadiendo gestos que entienda el sistema para comunicar que el palé se ha llenado.
- Utilizar un reloj *tracker* para obtener diversa información vital del trabajador, como sus pulsaciones por minuto, para tratar de mejorar su bienestar.
- Palé con altura desplazable para poder continuar añadiendo cajas.
- Algoritmo de apilamiento de cajas que utilice aprendizaje por refuerzo.
- Acercarse más a la inteligencia híbrida, permitiendo al robot aprender del humano.
- Mejorar y calibrar modelos de visión artificial para que funcionen con más precisión en el escenario virtual y lleguen a funcionar en el escenario real.



---

# Bibliografía

---

[1] Cotta WAA, Lopes SI, Vassallo RF. Towards the Cognitive Factory in Industry 5.0: From Concept to Implementation. *Smart Cities*. 2023, 6(4), pp. 1901-1921. <https://doi.org/10.3390/smartcities6040088>.

[2] Dellermann, Dominik & Calma, Adrian & Lipusch, Nikolaus & Weber, Thorsten & Weigel, Sascha & Ebel, Philipp. The Future of Human-AI Collaboration: A Taxonomy of Design Knowledge for Hybrid Intelligence Systems. 2019. <https://doi.org/10.24251/HICSS.2019.034>.

[3] Yaqian Zhang, Kai Ding, Jizhuang Hui, Sichao Liu, Wanjin Guo, Lihui Wang, Skeleton-RGB integrated highly similar human action prediction in human-robot collaborative assembly. *Robotics and Computer-Integrated Manufacturing*. 2024, Vol. 86, 102659. ISSN 0736-5845. <https://doi.org/10.1016/j.rcim.2023.102659>.

[4] S. Li, R. Wang, P. Zheng, L. Wang. Towards proactive human-robot collaboration: A foreseeable cognitive manufacturing paradigm. *J. Manuf. Syst.* 2021, Vol. 60, pp. 547-552.

[5] Claudia Carissoli, Luca Negri, Marta Bassi, Fabio Alexander Storm & Antonella Delle Fave. Mental Workload and Human-Robot Interaction in Collaborative Tasks: A Scoping Review. *International Journal of Human-Computer Interaction*. 20 Sep 2023. <https://doi.org/10.1080/10447318.2023.2254639>.

[6] C. Messeri, A. Bicchi, A. M. Zanchettin and P. Rocco. A Dynamic Task Allocation Strategy to Mitigate the Human Physical Fatigue in Collaborative Robotics. *IEEE Robotics and Automation Letters*. April 2022, Vol. 7, No. 2, pp. 2178-2185. <https://doi.org/10.1109/LRA.2022.3143520>.

[7] Yusen Li, Xueyan Tang, and Wentong Cai. 2014. On dynamic bin packing for resource allocation in the cloud. In: *Proceedings of the 26th ACM symposium on Parallelism in algorithms and architectures (SPAA '14)*. New York: Association for Computing Machinery, 2014, pp. 2-11. <https://doi.org/10.1145/2612669.2612675>.

[8] Silva, E., Oliveira, J.F. and Wäscher, G. (2016), The pallet loading problem: a review of solution methods and computational experiments. *International Transactions in Operational Research*. 2016, Vol. 23, pp. 147-172. <https://doi.org/10.1111/itor.12099>.

[9] Sara Ali, António Galvão Ramos, Maria Antónia Carravilla, José Fernando Oliveira. On-line three-dimensional packing problems: A review of off-line and on-line solution approaches. *Computers & Industrial Engineering*. 2022, Vol. 168, 108122. ISSN 0360-8352. <https://doi.org/10.1016/j.cie.2022.108122>.

[10] S. Hoque, M. Y. Arafat, S. Xu, A. Maiti and Y. Wei. A Comprehensive Review on 3D Object Detection and 6D Pose Estimation With Deep Learning. *IEEE Access*. 2021, Vol. 9, pp. 143746-143770. <https://doi.org/10.1109/ACCESS.2021.3114399>.

[11] D. Liang, K. Weng, C. Wang, G. Liang, H. Chen and X. Wu. A 3D object recognition and pose estimation system using deep learning method. In: *2014 4th IEEE International*

Conference on Information Science and Technology. Shenzhen, China: IEEE, 2014, pp. 401-404. <https://doi.org/10.1109/ICIST.2014.6920502>.

[12] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, BC, Canada: IEEE, 2017, pp. 23-30. <https://doi.org/10.1109/IROS.2017.8202133>.

[13] Tachella, J., Altmann, Y., Mellado, N. *et al.* Real-time 3D reconstruction from single-photon lidar data using plug-and-play point cloud denoisers. *Nature Communications*. 2019, Vol. 10, 4984. <https://doi.org/10.1038/s41467-019-12943-7>.

[14] Lianhui Li, Bingbing Lei, Chunlei Mao. Digital twin in smart manufacturing. *Journal of Industrial Information Integration*. 2022, Vol. 26, 100289. ISSN 2452-414X. <https://doi.org/10.1016/j.jii.2021.100289>.

[15] Csikszentmihalyi M. The flow experience and its significance for human psychology. In: CSIKSZENTMIHALYI, M., CSIKSZENTMIHALYI, I. S., eds. *Optimal Experience: Psychological Studies of Flow in Consciousness*. Cambridge: Cambridge University Press, 1988, pp. 15-35.

[16] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*. October 2010, Vol. 22, No. 10, pp. 1345-1359. <https://doi.org/10.1109/TKDE.2009.191>.

[17] Hasani, R., Lechner, M., Amini, A., Rus, D., & Grosu, R. Liquid Time-constant Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021, 35(9), pp. 7657-7666. <https://doi.org/10.1609/aaai.v35i9.16936>.

[18] Makram Chahine *et al.*, Robust flight navigation out of distribution with liquid neural networks. *Science Robotics*. 2023, Vol. 8, eadc8892. <https://doi.org/10.1126/scirobotics.adc8892>.

# Objetivos De Desarrollo Sostenible



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible.

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.		X		
ODS 2. Hambre cero.			X	
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.			X	
ODS 5. Igualdad de género.		X		
ODS 6. Agua limpia y saneamiento.			X	
ODS 7. Energía asequible y no contaminante.			X	
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.			X	
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.			X	
ODS 15. Vida de ecosistemas terrestres.			X	
ODS 16. Paz, justicia e instituciones sólidas.			X	
ODS 17. Alianzas para lograr objetivos.		X		

## Reflexión sobre los Objetivos de Desarrollo Sostenible.

---

Los Objetivos de Desarrollo Sostenible (ODS) constituyen un llamamiento universal a la acción para poner fin a la pobreza, proteger el planeta y mejorar las vidas y las perspectivas de las personas en todo el mundo.

Este proyecto está relacionado en menor o mayor medida con los siguientes objetivos:

- ODS 1: Fin de la pobreza. Es posible relacionar el proyecto con el fin de la pobreza, dado que uno de los objetivos ha sido mejorar la productividad utilizando sistemas de colaboración humano-máquina. La mayor productividad puede conducir a mayores ingresos y a una mayor estabilidad económica para las empresas, lo que, a su vez, puede traducirse en mejores salarios y más empleos. La colaboración con robots puede hacer que ciertas tareas industriales sean accesibles para personas con diferentes niveles de habilidad, incluyendo a aquellas personas que puedan tener dificultad para encontrar empleo.
- ODS 3: Salud y bienestar. El diseño de la solución trata de adaptar el sistema a las necesidades del operario, adaptando la dificultad de la tarea a su habilidad, para así mantener su salud y bienestar.
- ODS 5: Igualdad de género. La colaboración con robots puede ayudar a eliminar sesgos de género, ofreciendo las mismas oportunidades de empleo a hombres y mujeres en roles tecnológicos.
- ODS 8: Trabajo decente y crecimiento económico. La investigación en el área de la colaboración humano-máquina pretende mejorar las condiciones de trabajo en la industria. En nuestro caso específico, es fácil comparar tener que apilar todas las cajas a mano con poder esperar a que el robot se encargue de las más fáciles y repetitivas y que el humano solo tenga que intervenir cuando sea necesario. Los descubrimientos en la investigación de la inteligencia artificial y la robótica pueden llevar a mejoras significativas de la productividad que pueden suponer un gran crecimiento económico.
- ODS 9: Industria, innovación e infraestructuras. Industria e innovación son palabras clave en este trabajo, se ha tratado de ofrecer una solución innovadora a un problema muy común en la industria. La adopción de sistemas avanzados de colaboración humano-robot requiere la modernización de la infraestructura industrial.
- ODS 10: Reducción de las desigualdades. La colaboración con robots puede hacer que ciertas tareas industriales sean accesibles para personas con diferentes niveles de habilidad, reduciendo las desigualdades. Esto puede incluir a personas con discapacidades, quienes pueden encontrar en la tecnología robótica una herramienta que les permita participar plenamente en el mercado laboral.



- ODS 12: Producción y consumo responsables. La colaboración puede optimizar el uso de recursos y reducir el desperdicio, promoviendo prácticas de producción más sostenibles y responsables.
- ODS 13: Acción por el clima. La colaboración humano-robot puede aumentar la eficiencia energética, contribuyendo a la mitigación del cambio climático.
- ODS 17: Alianzas para lograr objetivos. El avance en estas tecnologías en la industria requerirá de la colaboración entre empresas, universidades y gobiernos, fomentando la cooperación global para alcanzar los ODS. La colaboración entre diferentes sectores puede también influir en la creación de políticas y regulaciones que apoyen el desarrollo sostenible y la adopción de tecnologías innovadoras en la industria.

