



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA MECATRÓNICA

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA VIRTUAL DE DETECCIÓN DE LA POSE 3D DE UN OBJETO MEDIANTE TÉCNICAS DE APRENDIZAJE PROFUNDO PARA APLICACIONES ROBÓTICAS”

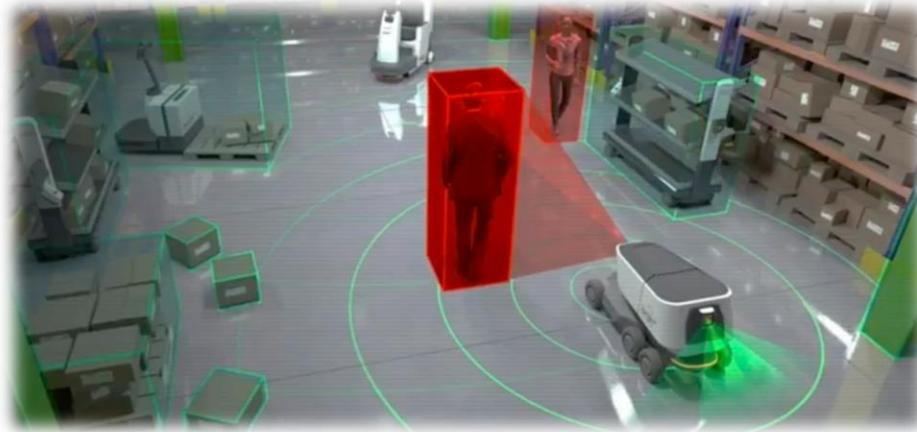
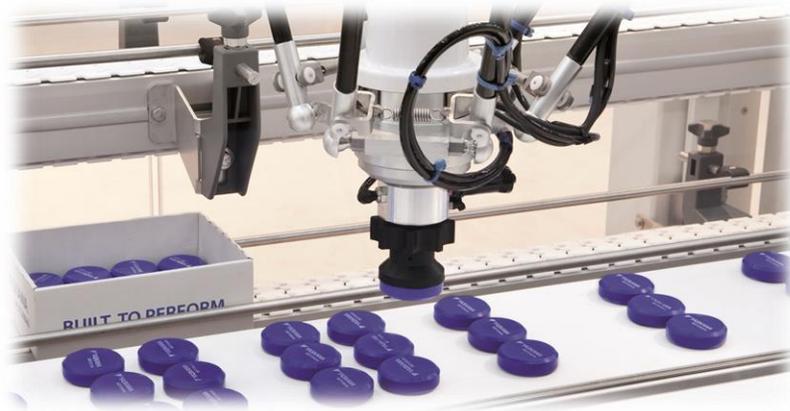
Autor: Chilingua Chuquitarco, Richard Alexander

Directora: Ing. Constante Prócel, Patricia Nataly



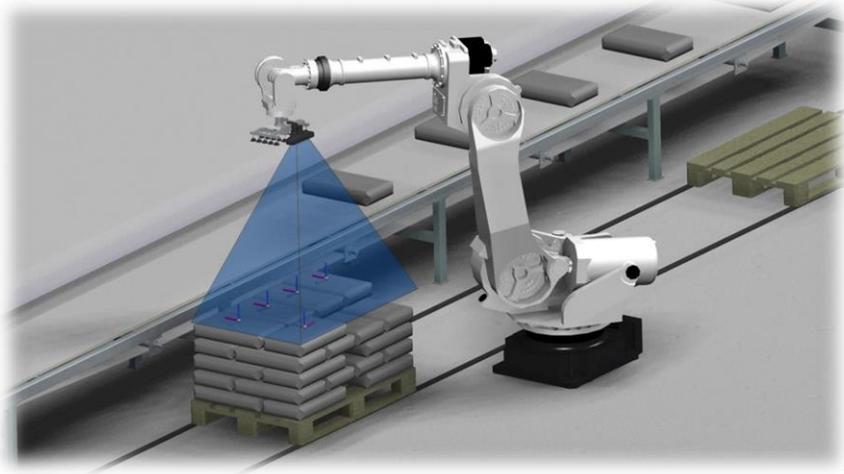
INTRODUCCIÓN

MOTIVACIÓN



INTRODUCCIÓN

PLANTEAMIENTO DEL PROBLEMA



Robot Mitsubishi RV-2SDB



INTRODUCCIÓN

Objetivo General

Diseñar e implementar un sistema virtual de detección de la pose 3D de un objeto mediante técnicas de aprendizaje profundo para aplicaciones robóticas.

Objetivos Específicos

- Recopilar información sobre el aprendizaje profundo aplicados a la detección de la pose 3D de un objeto y el tipo de aplicación robótica en la que se usa.
- Implementar un algoritmo de detección de pose 3D de objetos con redes neuronales convolucionales a base de datos sintéticos.



INTRODUCCIÓN

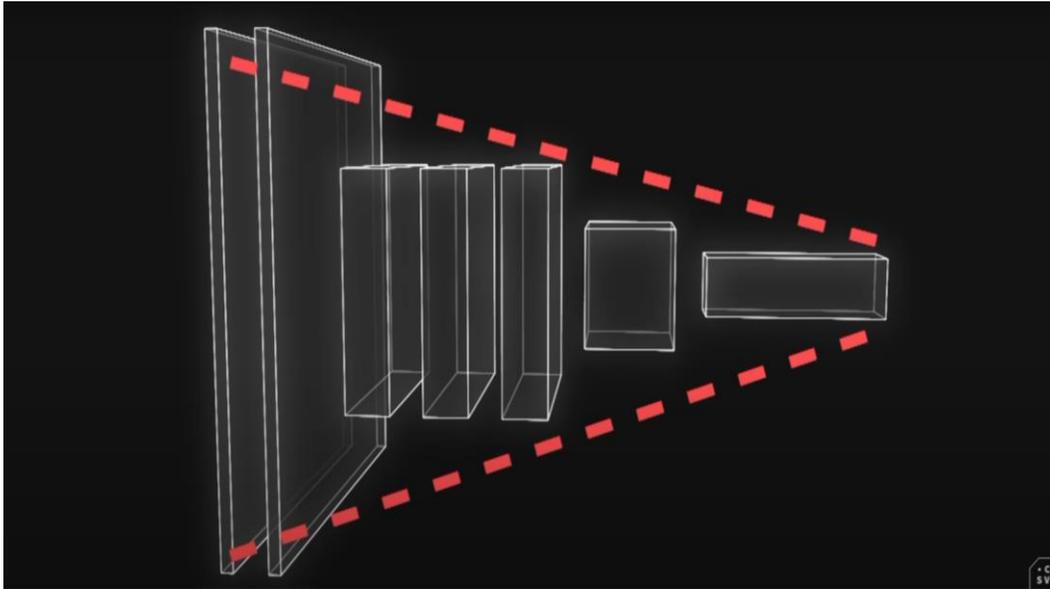
Objetivos Específicos

- Realizar la comunicación del simulador de entorno virtual con el sistema de detección de pose 3D para la aplicación con el robot Mitsubishi conocida su configuración de cinemática inversa.
- Realizar pruebas de funcionamiento del sistema virtual para evaluar los resultados en el campo de la robótica.



INTRODUCCIÓN

APRENDIZAJE PROFUNDO

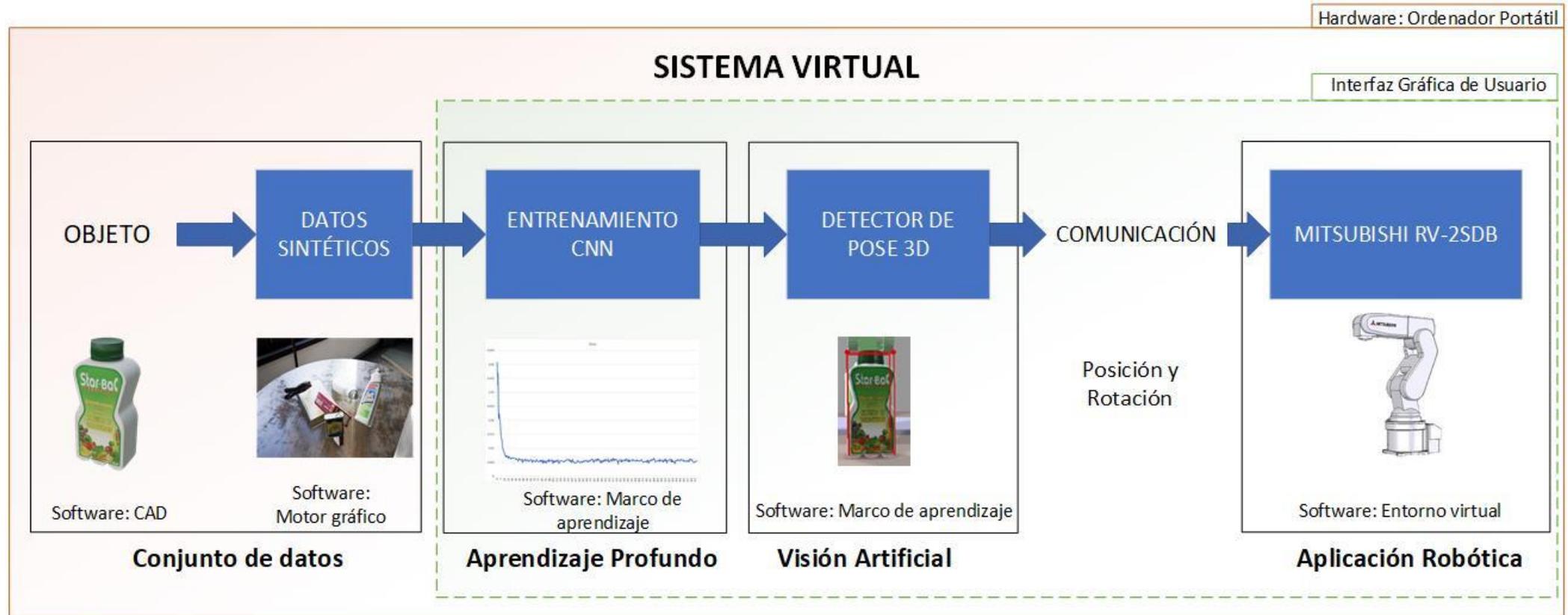


Del término traducido al español de Deep Learning es el subcampo de la Inteligencia Artificial que nombra a las arquitecturas con múltiples capas ocultas con el fin de aprender diversas características con múltiples niveles de abstracción.

La arquitectura que presenta una red convolutiva combina tres etapas: la Convolución, detector RELU y Pooling



DISEÑO E IMPLEMENTACIÓN ARQUITECTURA



DISEÑO E IMPLEMENTACIÓN

MODELADO 3D



Dimensiones: 55x34.4x124 mm,
Caras: 6703 polígonos
Peso: 0.2 Kg de peso
Formato exportación: .fbx y .obj



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DISEÑO E IMPLEMENTACIÓN AMBIENTES FOTORREALISTAS



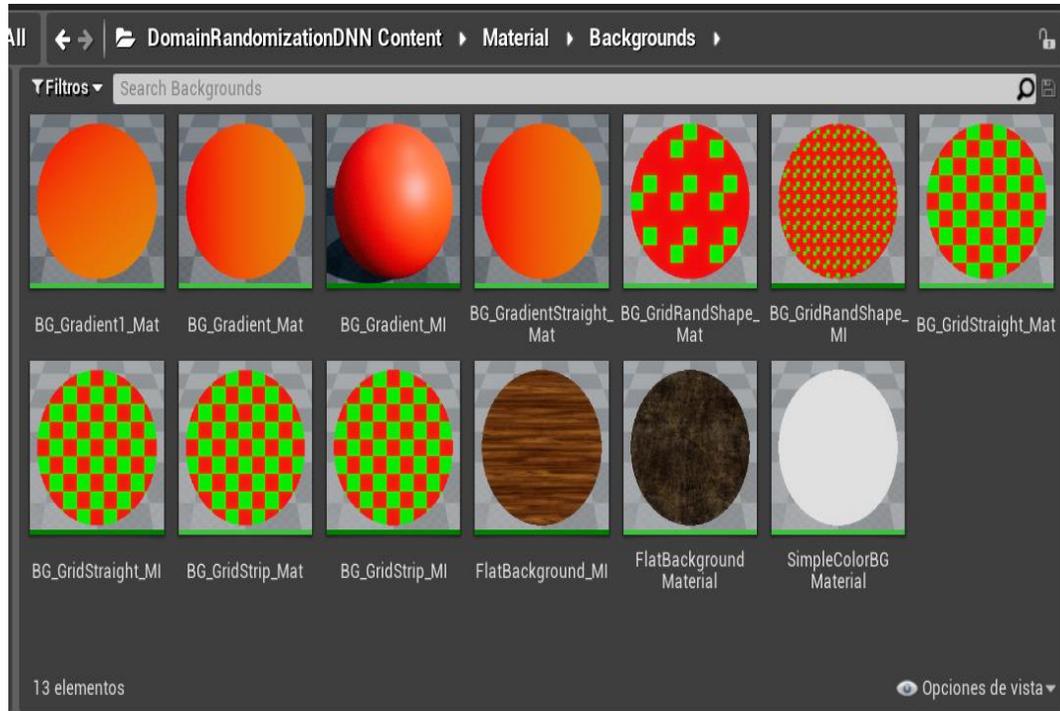
INTERIOR

EXTERIOR

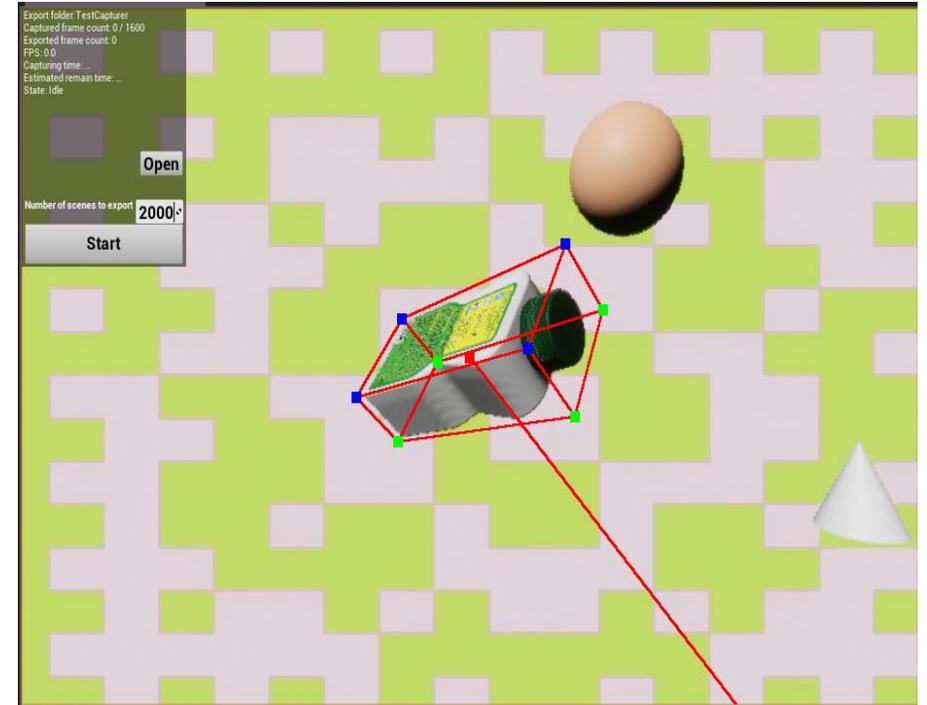
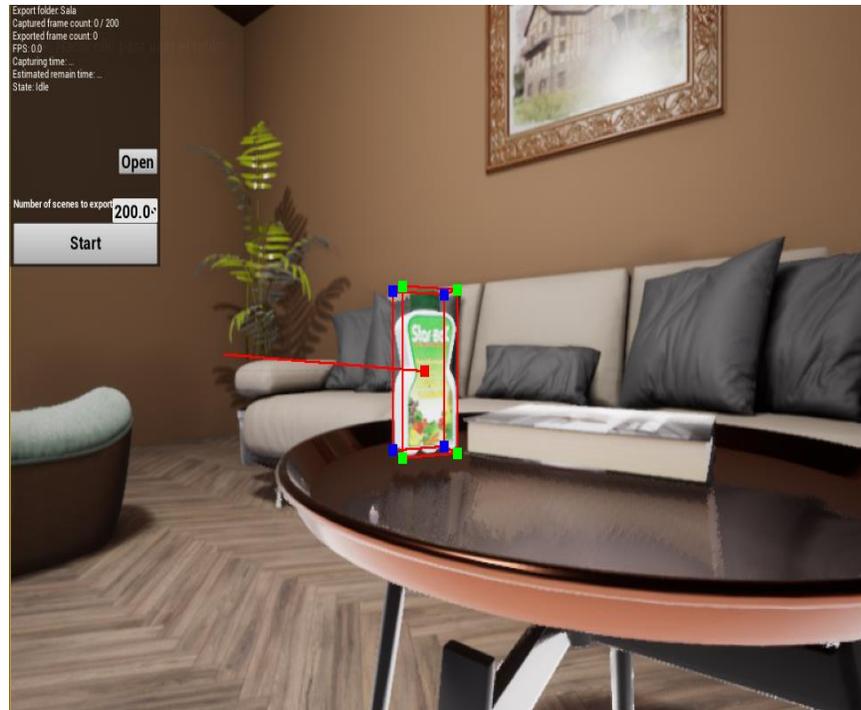
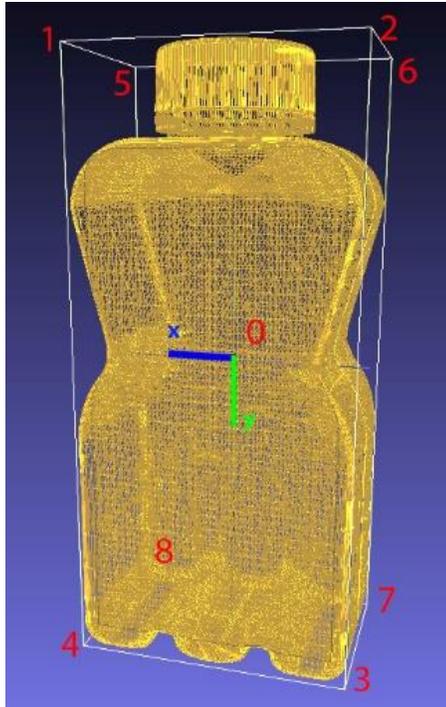


DISEÑO E IMPLEMENTACIÓN

ALEATORIZACIÓN DE DOMINIO

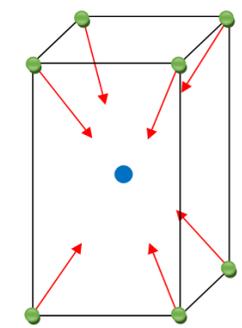
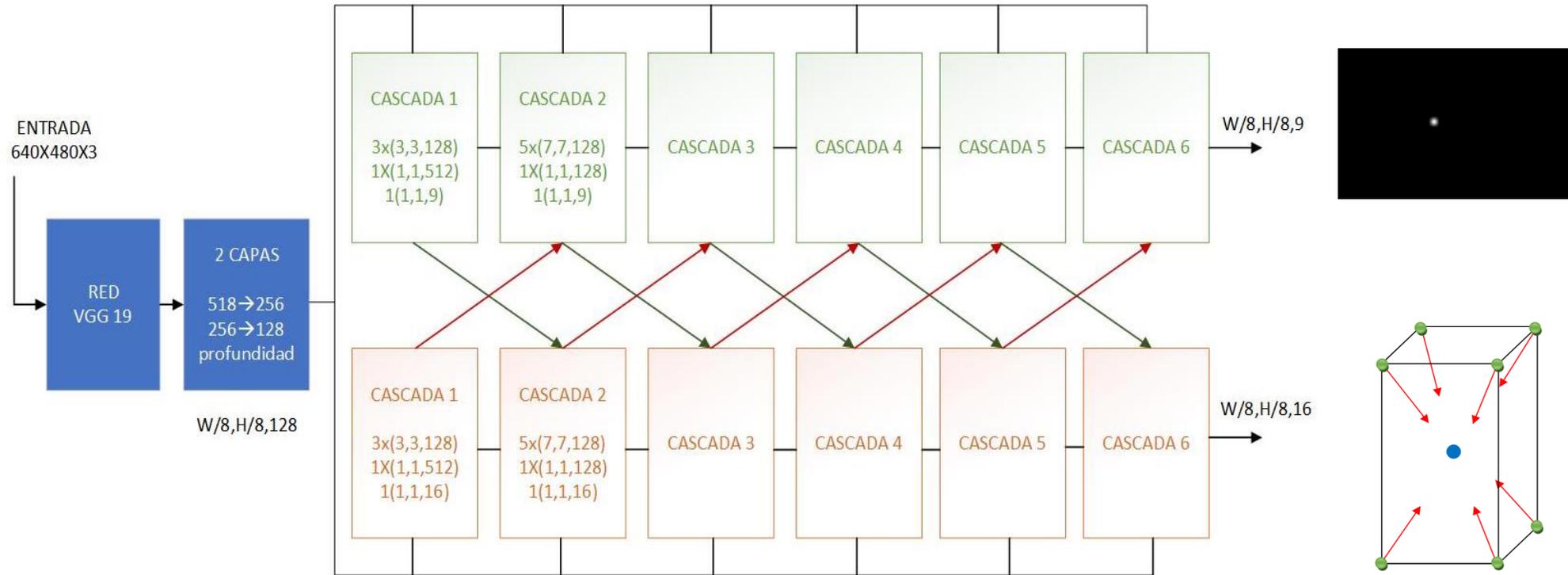


DISEÑO E IMPLEMENTACIÓN GENERADOR DE DATOS SINTÉTICOS



DISEÑO E IMPLEMENTACIÓN

ARQUITECTURA RED CNN



MAPAS DE AFINIDAD □
 MAPAS DE CREENCIA □
 RED VGG19 □

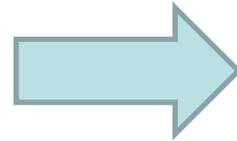


DISEÑO E IMPLEMENTACIÓN

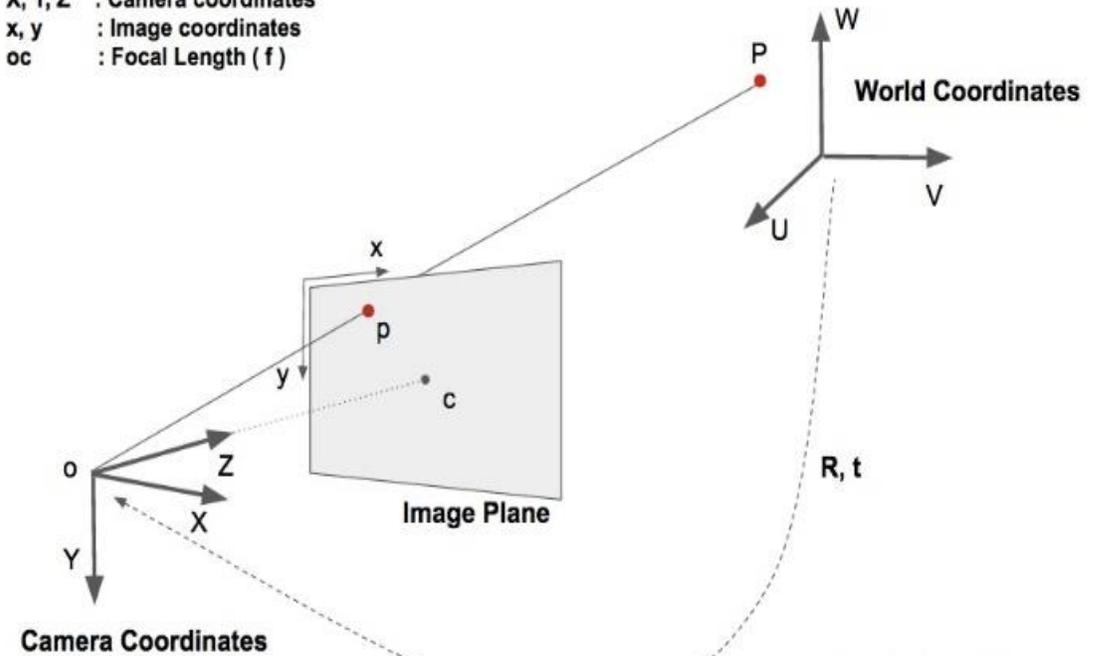
ALGORITMO PNP

$$\text{Matriz de cámara} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Distorsión} = (k_1 \ k_2 \ p_1 \ p_2 \ k_3)$$



U, V, W : World coordinates
 X, Y, Z : Camera coordinates
 x, y : Image coordinates
 oc : Focal Length (f)



LearnOpenCV.com

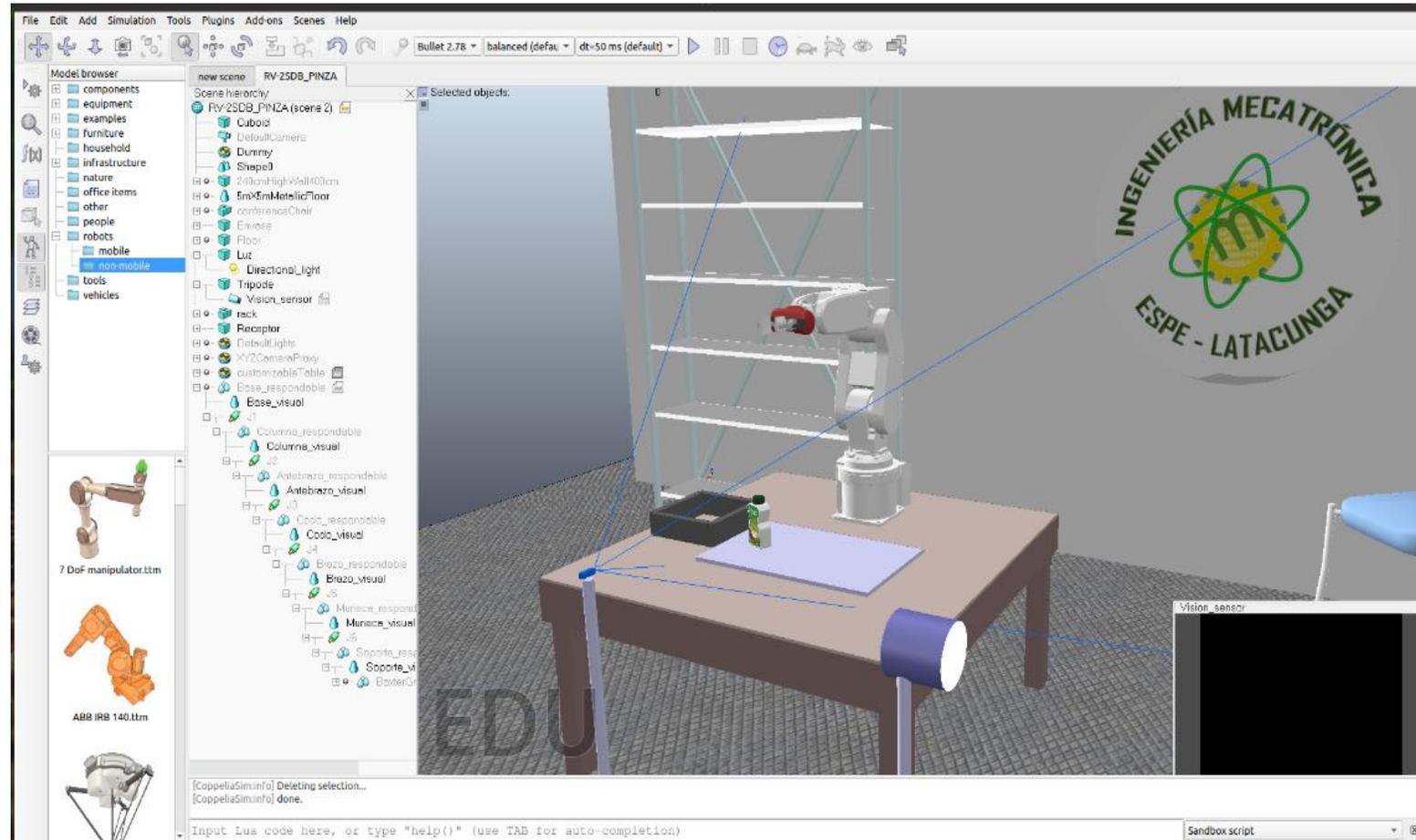


ESPE
 UNIVERSIDAD DE LAS FUERZAS ARMADAS
 INNOVACIÓN PARA LA EXCELENCIA

CONTROL ENTORNO VIRTUAL



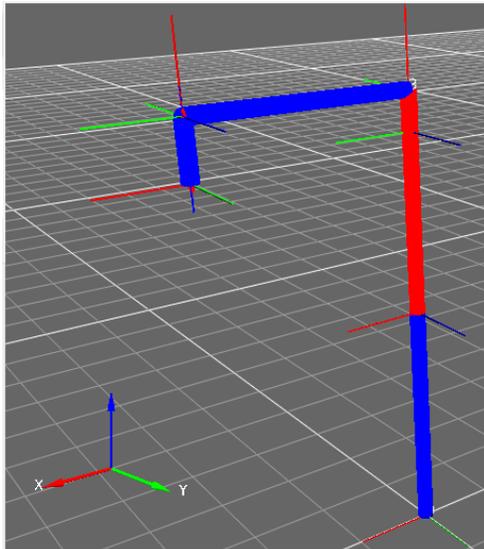
CoppeliaSim
from the creators of V-REP



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

CONTROL

PARÁMETROS DENAVIT-HARTENBERG



θ	d	l	α
q_1	0.295	0	-90°
$q_2 - 90^\circ$	0	0.230	0
q_3	0	0.05	-90°
q_4	0.270	0	90°
$q_5 + 90^\circ$	0	0	-90°
q_6	0.07	0	180°



CONTROL

CINEMÁTICA DIRECTA E INVERSA

$$T_{\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_l = \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_z = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_{\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_i = R_{\theta} * T_z * T_l * R_{\alpha}$$



SymPy

Symbolic Mathematics Python Library

```
try:
    q = sp.nsolve((T - D), (q1, q2, q3, q4, q5, q6), (0, 0, 0, 0, 0, 0), prec=6)
except:
    print('no se encontró la solución')
    q = [0, 0, 0, 0, 0, 0]
print(q)
```



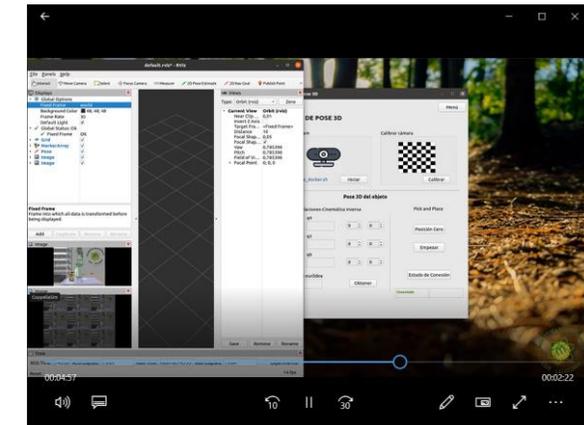
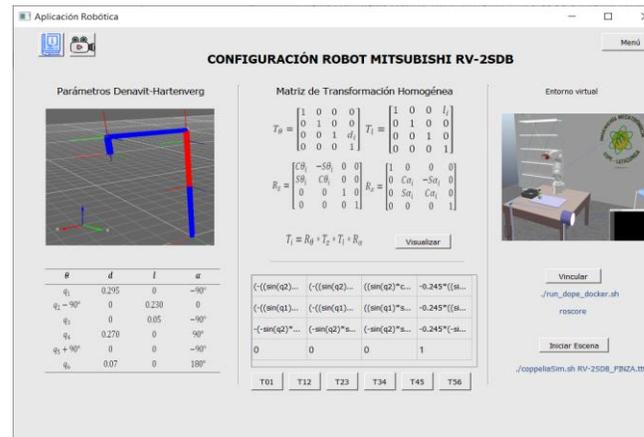
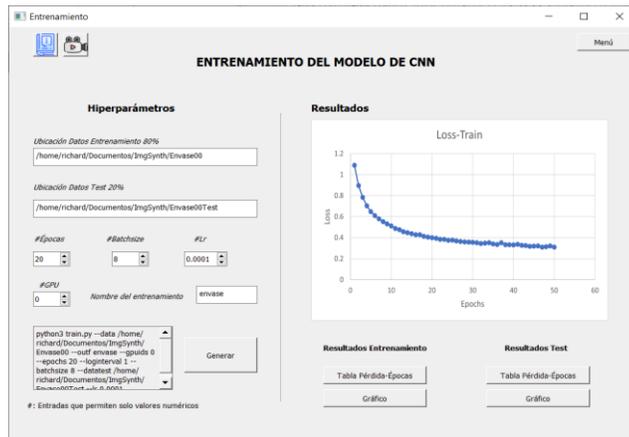
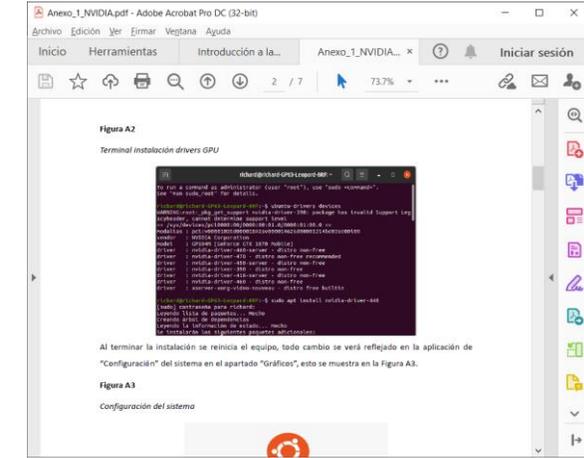
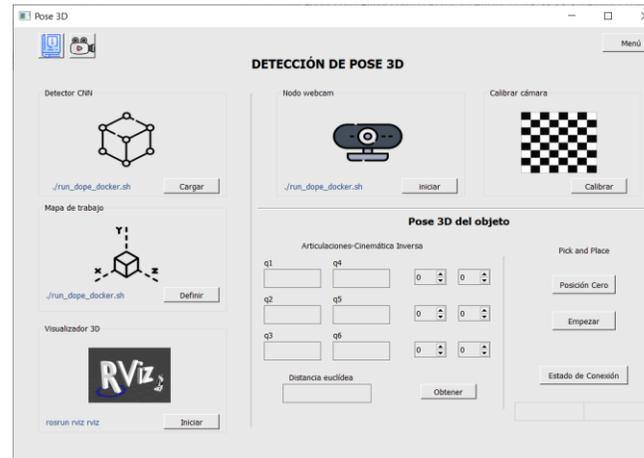
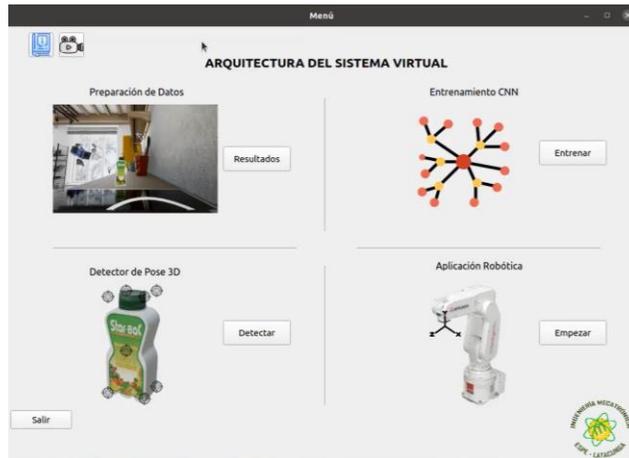
ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

CONTROL

COMUNICACIÓN DE SISTEMAS



INTERFAZ GRÁFICA DE USUARIO

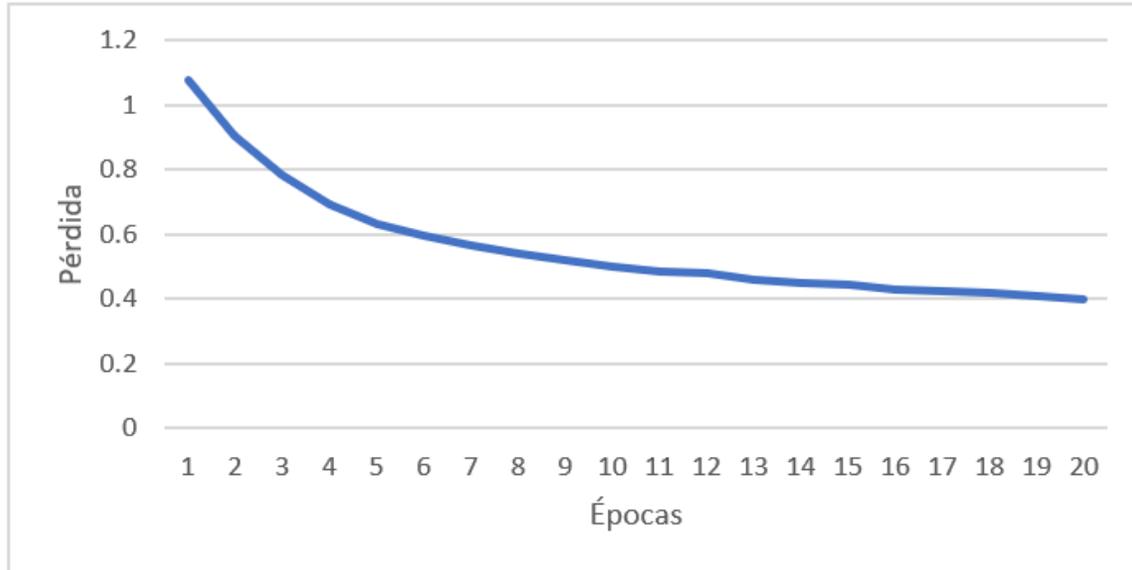


PRUEBAS Y RESULTADOS

ENTRENAMIENTO Y PRUEBA DE DATOS

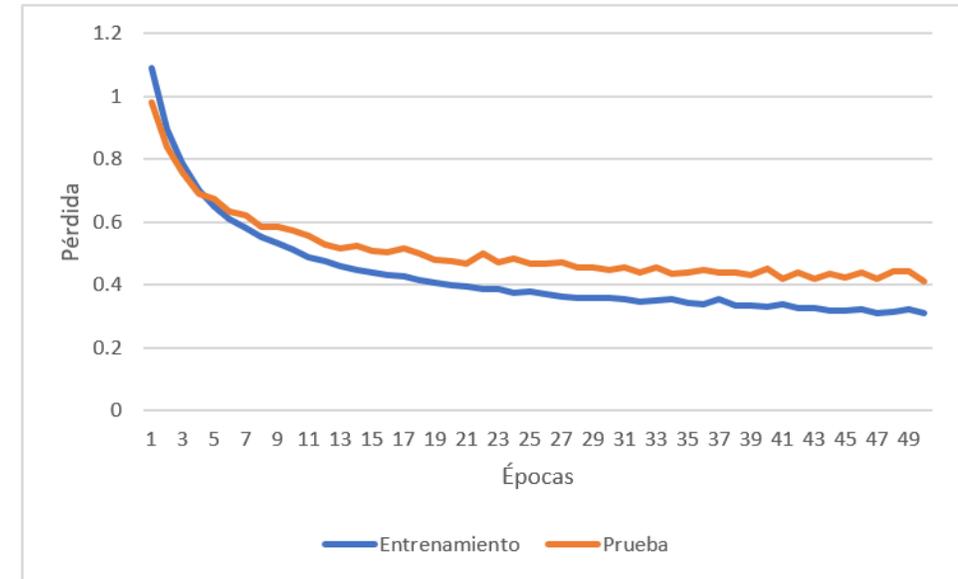
Entrenamiento 80%: 3.200 imágenes

Prueba 20%: 800 imágenes



Tamaño de lote: 16

Taza de aprendizaje: 0.0001



Tamaño de lote: 8

Taza de aprendizaje: 0.0001

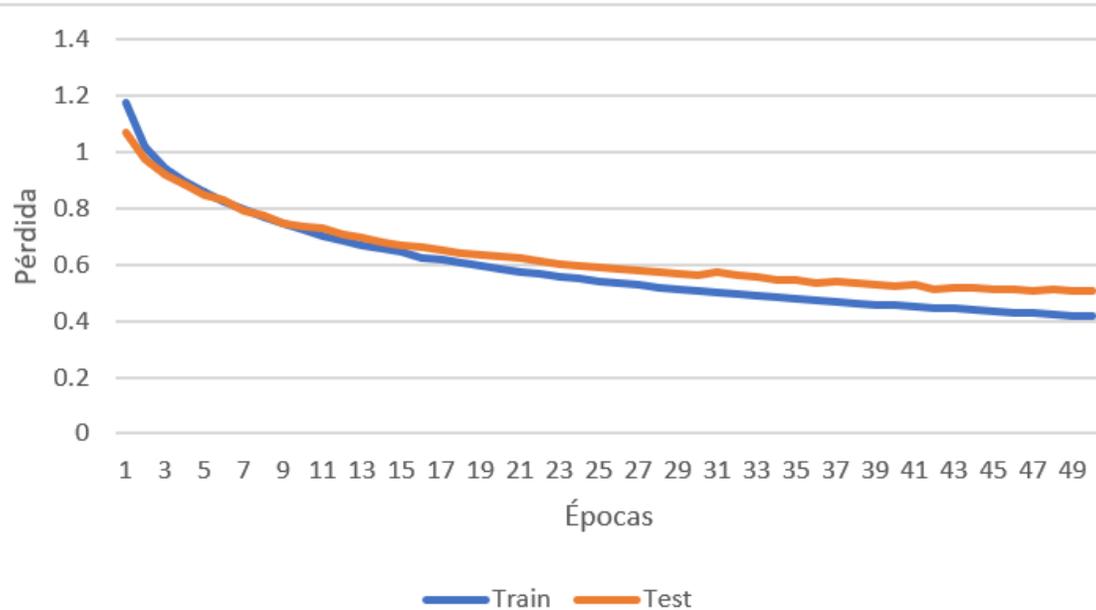
GPU: NVIDIA Geforce GTX 1070-8GB



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

PRUEBAS Y RESULTADOS

ENTRENAMIENTO Y PRUEBA DE DATOS



Tamaño de lote: 8
Taza de aprendizaje: 0.00001



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

PRUEBAS Y RESULTADOS

DETECCIÓN DE LA POSE 3D

```
/roo...  
9 puntos válidos encontrados  
9 puntos válidos encontrados
```

```
richa...  
frame_id: "image"  
pose:  
  position:  
    x: 0.00880528117543654  
    y: 0.13663274029100939  
    z: 64.11887128342129  
  orientation:  
    x: -0.007844582384869118  
    y: 0.9988490028818806  
    z: -0.008031328660607972  
    w: 0.046632925374944234  
--
```

$$\text{Error Porcentual \%} = \frac{|\text{Valor Real} - \text{Valor Estimado}|}{\text{Valor Real}} * 100$$

$$\text{Distancia Euclidiana} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2}$$



PRUEBAS Y RESULTADOS

DETECCIÓN DE LA POSE 3D

Posición

N° Prueba	Distancia Euclidiana Real (cm)	Ubicación Euclidiana Estimada (cm)	Error absoluto (cm)
1	46.0800575	45.79581778	0.28423972
2	53.7303536	55.28395098	1.55359737
3	56.2987646	57.88133662	1.58257198
4	56.0775534	57.16241311	1.08485967
5	56.9073106	57.53517261	0.62786201
25	58.0810778	59.14799413	1.06691631
Error Porcentual		1.6947 %	

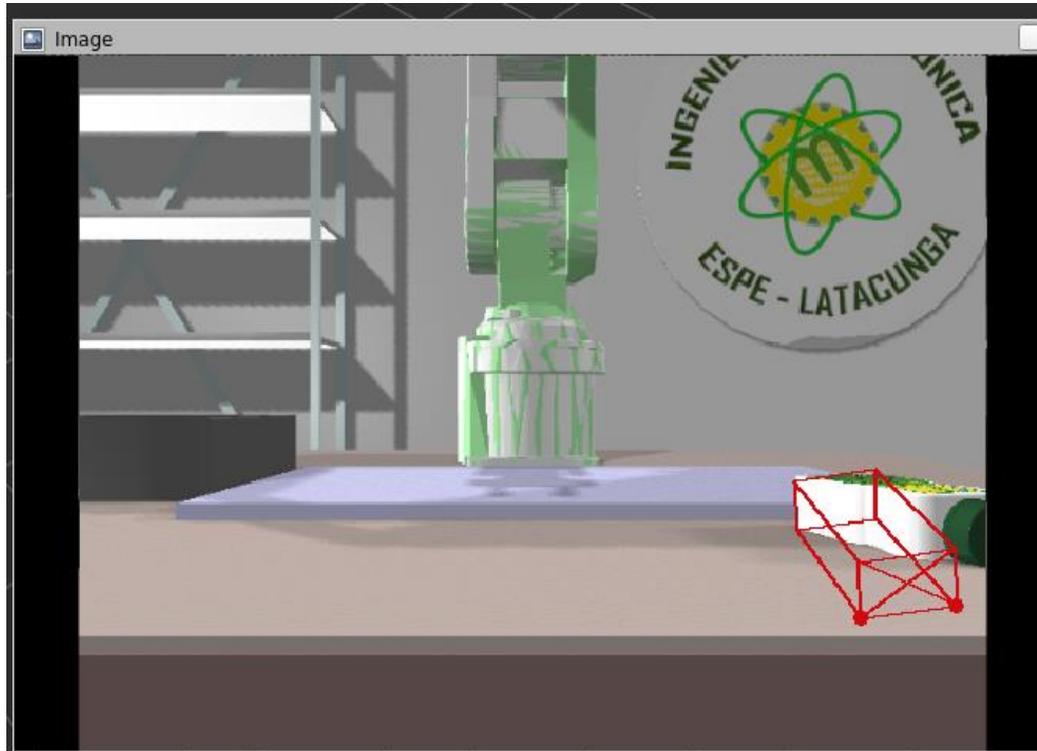
Orientación

N° Prueba	Alpha α	Beta α	Gamma γ
1	5.74	0.3969	2.58849
2	7.11	0.332	1.27056
3	0.951	2.421	2.23284
4	3.98	3.26	5.23890
5	4.138	3.05	5.38948
25	0.69	2.01	3.97134
Promedio %	4.106	3.8409	1.9932



PRUEBAS Y RESULTADOS

DETECCIÓN DE LA POSE 3D



PRUEBAS Y RESULTADOS

DETECCIÓN DE LA POSE 3D



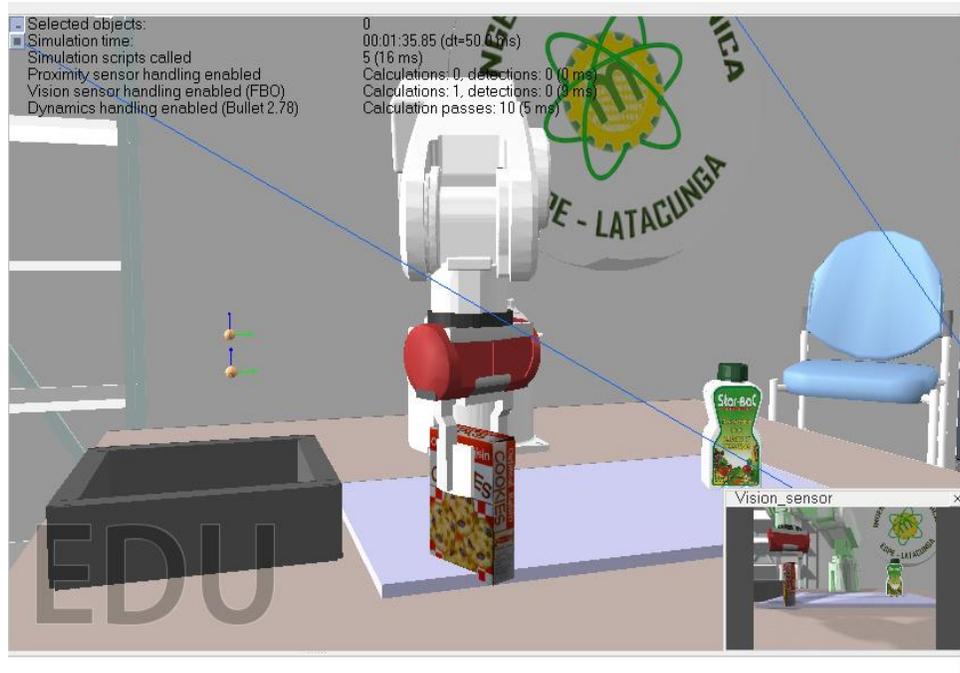
Objeto pre-entrenado

N° Prueba	Distancia Euclidiana Real (cm)	Ubicación Euclidiana Estimada (cm)	Error absoluto (cm)
1	54.83400222	54.78247458	0.09397024
2	53.42795242	53.52446496	0.18064054
3	51.84480109	50.84711416	1.92437218
4	57.84793341	58.6829941	1.44354455
5	46.95299245	47.34731485	0.83982379
6	64.12007018	64.8033055	1.0655561
7	57.80672193	58.31234624	0.87468083
8	50.8922106	50.67148238	0.43371711
9	59.20261987	60.03168745	1.40039001
10	63.30876874	63.81435919	0.79861046
Error			0.924669 %
Porcentual			

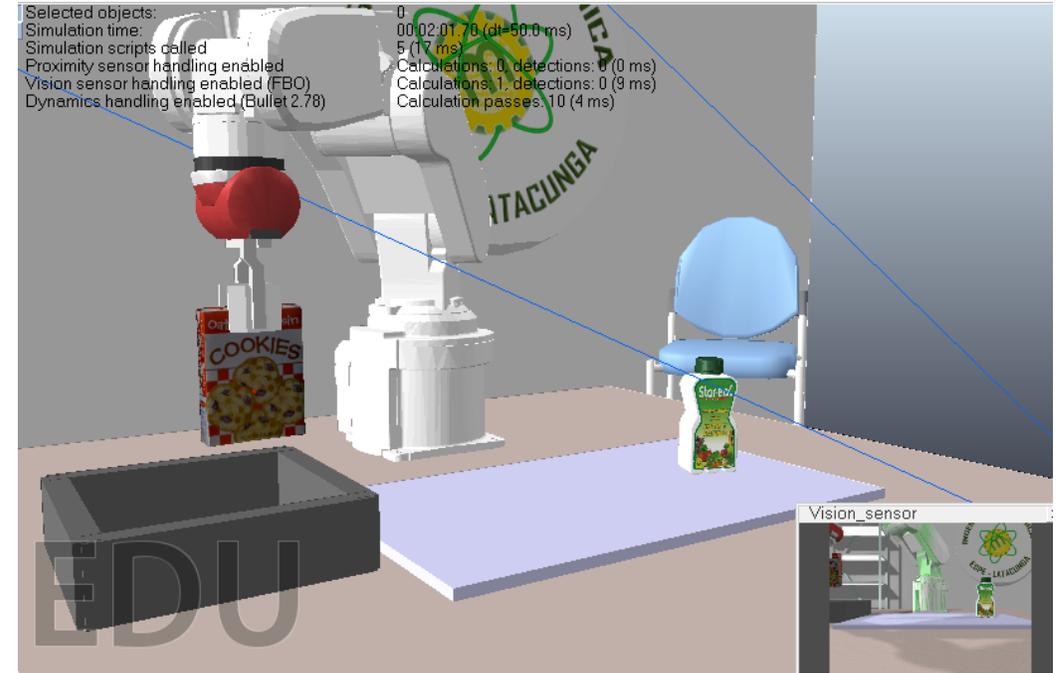


PRUEBAS Y RESULTADOS

APLICACIÓN ROBÓTICA



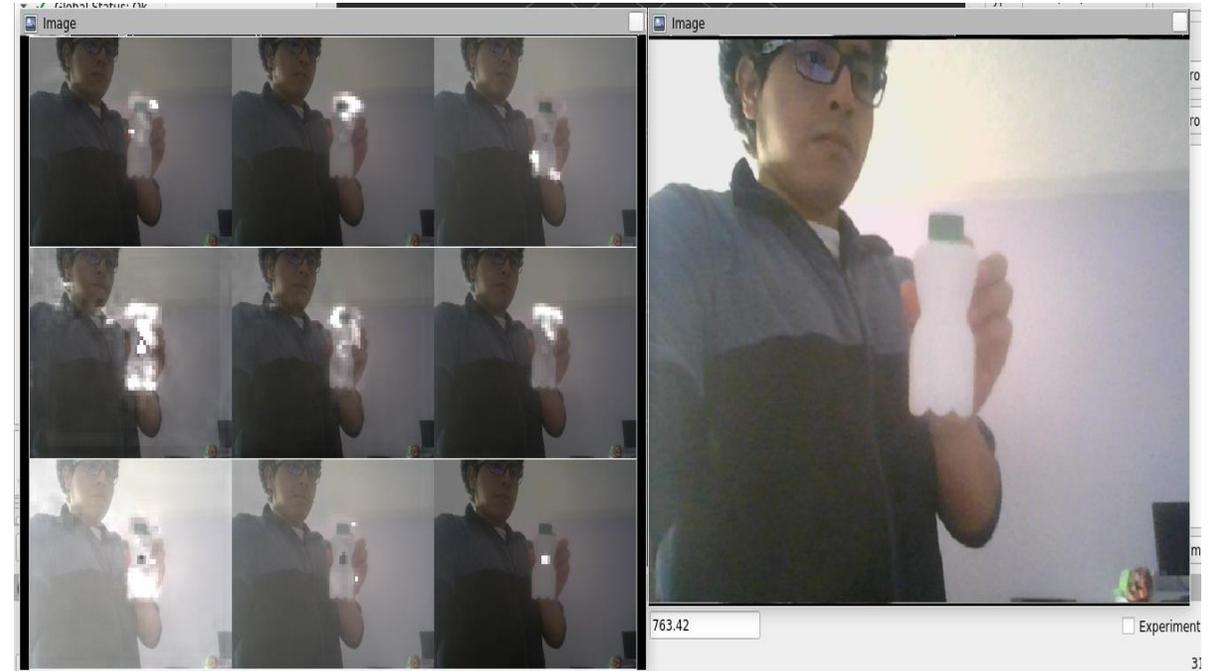
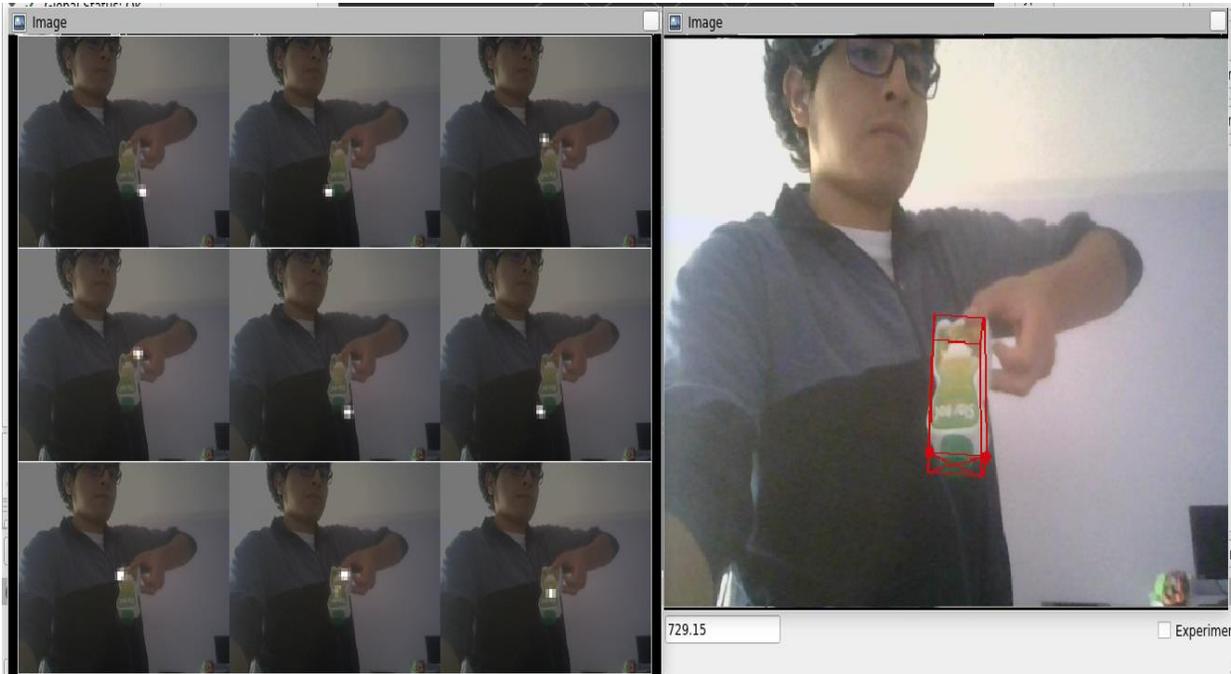
Recoger



Colocar

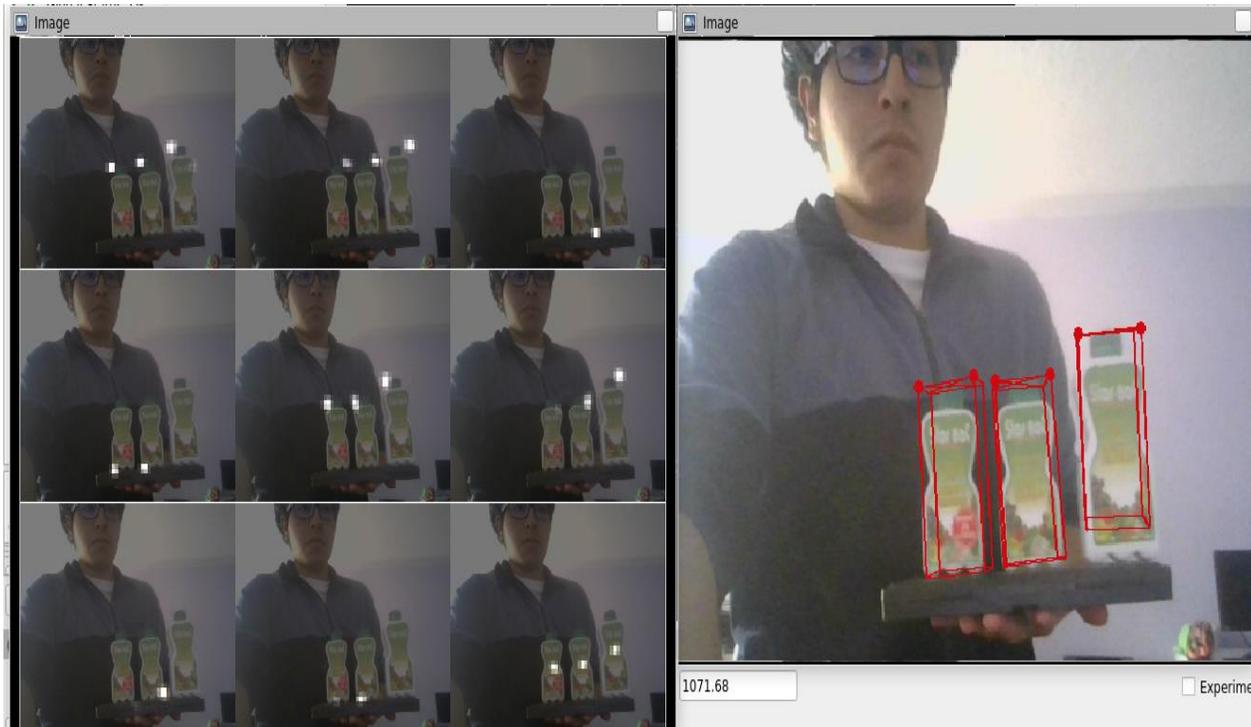
PRUEBAS Y RESULTADOS

CÁMARA WEB



PRUEBAS Y RESULTADOS

CÁMARA WEB

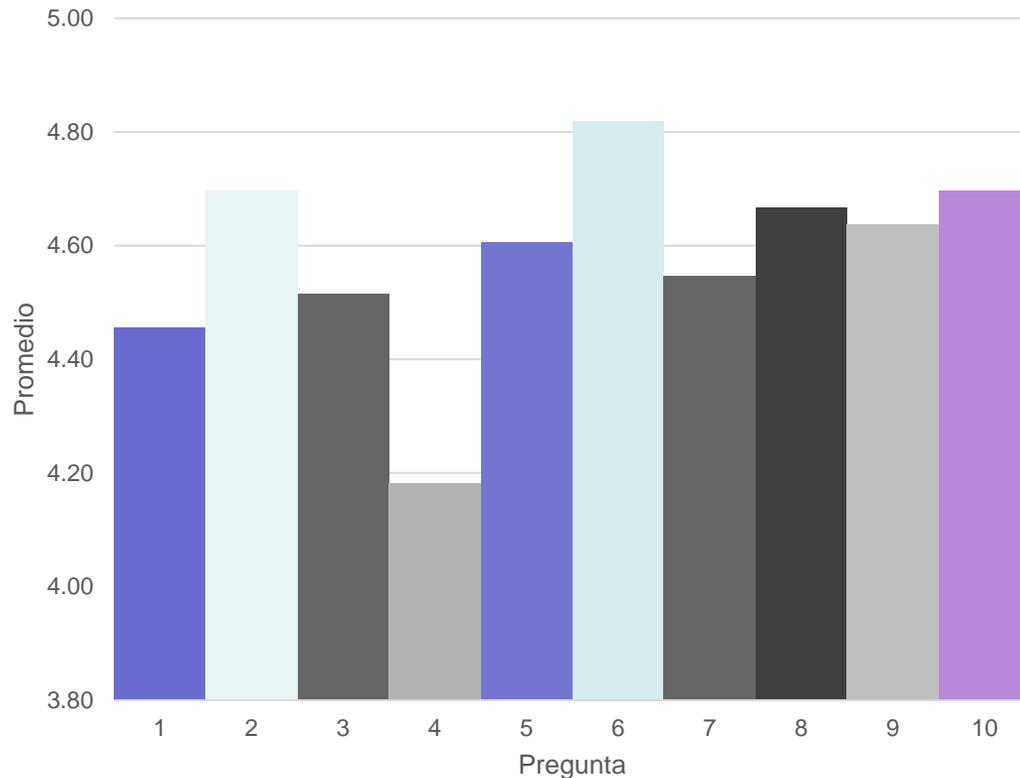


Distancia euclidiana (cm)	Error porcentual
51.243052	2.486104
51.58948	3.17896
51.959	3.918
52.41839	4.83678
50.7576	1.5152
52.0051	4.0102
51.8965	3.793
51.23265	2.4653
50.98231	1.96462
52.68035	5.3607
Error porcentual promedio	3.3528864%



PRUEBAS Y RESULTADOS

ANÁLISIS DE USABILIDAD



Preguntas

1. La información de las pantallas es clara y eficiente para su comprensión
2. Los colores empleados respetan su comodidad visual
3. La elección de menús es presentada con botones claros y accesibles
4. El programa permite moverse con facilidad entre etapas
5. La calidad técnica de los videos y manuales te han parecido adecuados
6. La información brindada resulta nueva y de interés
7. No se han presentado errores en la aplicación
8. El sistema virtual se adaptaría a sus necesidades en asignaturas de niveles superiores
9. Recomendaría el sistema virtual a otros estudiantes
10. Indique el grado de satisfacción con el sistema virtual en general



PRUEBAS Y RESULTADOS

VALIDACIÓN DE HIPÓTESIS

Método de Chi-Cuadrado

$$\chi^2 = \sum \frac{(fo - Ei)^2}{Ei}$$

$$4.043 \geq 3.841$$

Aplicación robótica ejecutada	Envase	Caja de galletas	TOTAL
Si	24	29	53
No	6	1	7
TOTAL	30	30	60

La condición de aceptación de hipótesis fue cumplida, con lo que resulta en una afirmación que el diseño e implementación de un sistema virtual basado en técnicas de aprendizaje profundo ayudará a la detección de la pose 3D de un objeto para aplicaciones robóticas.



CONCLUSIONES

- En este trabajo se diseñó e implementó un sistema virtual para la detección de la pose 3D mediante una arquitectura basada en redes neuronales convolucionales, en un conjunto de datos sintéticos fotorrealistas y aleatorización de dominios de un objeto personalizado, hacia un entorno virtual para el desarrollo de la aplicación robótica de recoger y colocar; la etapa más importante fue establecer la comunicación entre sistemas y la generación de la interfaz gráfica de usuario de los procesos vinculados.
- Para el diseño de algoritmo de detección se partió de la recopilación de información y conceptos sobre el aprendizaje profundo el cual permite un nivel de abstracción mayor de característica por las convoluciones realizadas en las capas ocultas; además de realizarse una búsqueda de las aplicaciones robóticas que afrontan el problema de pose 3D siendo la navegación, realidad aumentada y en robots fijos la manipulación de objetos, se determinó esta acción por la versatilidad en definir las acciones de control.



CONCLUSIONES

- La generación de 4000 imágenes sintéticas en ambientes fotorrealistas y aleatorización de dominio, con una variedad de datos a nivel de renderizado y construcción de escena han demostrado que el entrenamiento únicamente de datos sintéticos brinda buenos resultados en la detección de pose 3D dentro del ambiente virtual de CoppeliaSim como en imágenes del mundo exterior.
- En el desarrollo del algoritmo de detección de posición y orientación se utilizó 9 mapas de creencia y 8 mapas de afinidad para correlacionar las ubicaciones de vértices en 2D hacia los proyectados por el cuboide 3D del envase; en conjunto de la biblioteca OpenCV se desarrolló el algoritmo de perspectiva PnP, conocido los parámetros intrínsecos de cámara y las dimensiones del objeto para predecir su pose.



CONCLUSIONES

- El algoritmo de control trabaja bajo la biblioteca Sympy de cálculo simbólico para obtener una representación de las matrices de transformación homogéneas descritas por los parámetros DH del robot Mitsubishi, en el cálculo de la cinemática inversa; esta misma se resuelve por métodos numéricos en tiempo real al recibir las configuraciones de posición y orientación final del objeto, es así que el método presentó dificultades para encontrar puntos válidos en algunos casos.
- La comunicación entre etapas del sistema virtual se realizó mediante la integración del meta sistema operativo ROS, con una tasa de refresco cada 0.5213 segundos se establece la comunicación de datos a través de publicadores y suscriptores.



CONCLUSIONES

- Las pruebas de funcionamiento del sistema virtual se realizaron por medio de ubicaciones al azar dentro del alcance máximo del manipulador. El resultado de 25 pruebas arrojó un error porcentual de 1.6947% en traslación medida por la distancia euclidiana de la cámara hacia el envase, mientras en orientación un error promedio porcentual de 3.0136% de todos los ejes, siendo 1.99% el valor menor sobre el ángulo gamma; estos resultados frente al objeto pre-entrenado de error porcentual 0.924%. Esto se ve influenciado por la distancia que existe entre ambos elementos, obteniendo datos erróneos si la distancia es mayor a 50-60 cm o una orientación que no permita el agarre del objeto; por tal razón la aplicación robótica no se completa aun cuando se encuentre en una ubicación libre de singularidades.



CONCLUSIONES

- El test de usabilidad aplicado al grupo de estudiantes determinó un alto grado de satisfacción del sistema virtual sobre la interfaz gráfica desarrollada al abarcar temas nuevos y gran interés para aplicaciones en niveles superiores con una puntuación general de 4.58 sobre 5 puntos; además la funcionalidad se ha realizado con el estadístico Chi-cuadrado donde el resultado calculado 3.841 validó la hipótesis planteada sobre el valor teórico.
- La implementación del sistema virtual integra en su totalidad herramientas de software libre como Blender 3D, Unreal Engine 4, CoppeliaSim Edu, Python con bibliotecas Sympy y ROS; con el propósito de dotar de flexibilidad y libertad de estudio de los elementos integrados, este conocimiento de nuevas tecnologías ayuda a complementar el aprendizaje individual y colectivo.



RECOMENDACIONES

- Se debe mantener en consideración al momento de la generación de modelos personalizados, que los objetos que muestran características transparentes generan confusión a la red y son difíciles de extraer características dada la pequeña reflectancia que produce la luz al llegar a su superficie.
- En caso de ver una limitación en hardware de cálculo computacional es recomendable el uso de plataformas de entrenamiento como Google Colaboratory, este permite la escritura y ejecución de código Python en Jupyter Notebook alojado en el navegador; además, de alojar los datos sobre Google Drive para el entrenamiento remoto en GPUs de mayor capacidad de memoria gráfica.



RECOMENDACIONES

- Dentro del generador de imágenes sintéticas es preciso que las imágenes capturadas muestren visibles los vértices del cuboide limitador, dado que la validación del modelo no relaciona oclusiones entre objetos.
- Evite las orientaciones que produzcan el bloque de Cargan en ángulos de Euler, ocasionan un efecto contraproducente en la ejecución de la aplicación robótica; se recomienda mantener el trabajo de ángulos por cuaterniones entregado por el detector de pose 3D.
- En virtud del sistema desarrollado, se sugiere la exploración de otros entornos de simulación como Unreal Engine, Webots, Gazebo para brindar más herramientas de prueba y puesta en escena de estos modelos de aprendizaje profundo.



RECOMENDACIONES

- Para un acercamiento más práctico del proyecto, se sugiere llevar una adquisición de datos personalizada e integrar la detección real de la pose 3D con el Robot Mitsubishi; enfocada a robots fijos y móviles en aplicaciones que abarcan este problema.
- Explorar el tema de contenedores e hilos en Python para unificar los procesos que intervienen en el sistema virtual, esto con el propósito de evitar el ingreso manual de líneas de código al terminal.



***GRACIAS POR SU
ATENCIÓN***



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA