

# Robot arm simulation in mobile devices using OpenGL ES

## Simulación de brazo robótico en dispositivos móviles utilizando OpenGL ES

Dino Tonatiah Gutierrez-Martinez<sup>1</sup>, Marly Yuleth Cantú-Charles<sup>1</sup>, Jesus Eduardo Uriegas-Ibarra<sup>1</sup>, and Marco Aurelio Nuño-Maganda<sup>\*1</sup>

<sup>1</sup>Universidad Politécnica de Victoria,  
Departamento de Ingeniería en Tecnologías de la Información,  
Av. Nuevas Tecnologías 5902, Parque científico y tecnológico de Tamaulipas 801, Victoria, Tamps., México, 87138.  
{1930351,1930064,1930526,mnunom}@upv.edu.mx

### Abstract

Computer simulation allows users to interact with a simulated environment, which makes possible learning without incurring the costs associated with a real environment. That is why different projects have emerged trying to replicate different real life experiences in 3D simulation. With the increase in digital activity due to the pandemic, many students have lost the experience of interacting first-hand with physical processes, such is the case of robotic arms that due to their cost and social distancing is not feasible, which is why we propose a 3D simulation of a robotic arm with which a user can interact.

**Keywords**— Simulation, Mobile App, OpenGL

### Resumen

La simulación por computadora permite la interacción de los usuarios con un ambiente simulado, lo cual hace posible el aprendizaje sin incurrir en los costos asociados a un ambiente real. Es por ello, que diferentes proyectos han surgido intentando replicar diferentes experiencias de la vida real en simulación 3D. Con el aumento de la actividad digital debido a la pandemia muchos estudiantes han perdido la experiencia de interactuar de primera mano con procesos físicos, tal es el caso de los brazos robóticos que por su costo y el distanciamiento social no es factible, es por ello que se propone

la simulación 3D de un brazo robótico con el cual un usuario puede interactuar.

**Palabras clave**— Simulación, Aplicación Móvil, OpenGL

### I. Introducción

La simulación de un brazo robótico es útil para verificar el rendimiento de los algoritmos de planificación, lo cual conlleva ventajas en cuanto a costo tiempo y seguridad. Un tipo importante de sistema robótico es el manipulador o brazo robótico. Los sistemas robóticos de este tipo fueron algunos de los primeros en alcanzar un uso generalizado en la industria. Los manipuladores robóticos se han convertido en un elemento estándar de las líneas de montaje modernas. Realizan una gran cantidad de tareas como: soldadura, pulverización, operaciones de recogida y colocación, perforación, corte y elevación.

Los manipuladores robóticos son algunos de los ejemplos más sencillos de sistemas robóticos prácticos. Su estudio ayuda a aclarar los principios subyacentes y los problemas que se encuentran al estudiar sistemas más complejos [1]. Un brazo robot está formado por una serie de cuerpos individuales o eslabones que están conectados por articulaciones. Los cuerpos individuales que componen el robot suelen tratarse como cuerpos rígidos. Las articulaciones que conectan los eslabones del robot pueden ser bastante complejas y pueden presentar una mecánica no trivial. Una articulación ideal es una interconexión entre cuerpos rígidos de un sistema robótico que sólo permite movimientos relativos específicos y predefinidos, como la traslación o la rotación [2].

\* Autor de correspondencia

En [3] realizaron una comparación de los programas de simulación específicamente orientados a la manipulación del brazos robóticos. Todas las plataformas consideradas en este estudio están orientadas a la computadora, dejando a un lado las aplicaciones móviles. En el presente artículo se muestran los resultados de implementar una aplicación móvil que simula un brazo robótico, esto con la finalidad de permitir al público en general interactuar con este, y de manera particular a estudiantes de ingenierías afines a la robótica.

El artículo se organiza de la siguiente manera: en la sección II, se muestra una investigación sobre algunas aplicaciones con funcionamiento similar, pretendiendo concentrar un pequeño estado del arte de aplicación móviles para la simulación de brazos robóticos. En la sección III, se describen algunas tecnologías necesarias para el desarrollo del proyecto. En la sección IV se detalla el diseño del brazo robótico virtual, así como la arquitectura de la aplicación. En la sección V se describe el desarrollo de la aplicación móvil, además de mostrar los resultados obtenidos. Finalmente, en la sección VI se describen las conclusiones y trabajo futuro.

## II. Revisión de Aplicaciones Relacionadas

### II.1. ARM Robot Simulator (3D)

Es un simulador 3D para probar el Robot industrial de 6 ejes ARM. Es posible probarlo mediante el establecimiento de la velocidad y el ángulo de cada uno de los de 6 ejes. El usuario puede establecer varias opciones como el objetivo X, Y, Z y es posible ver el punto de ajuste de X, Y, Z, así como la velocidad de los 6 ejes. Está diseñado para comportarse de manera similar al brazo robótico utilizado en la industria real [4]. Esta aplicación se muestra en la Figura 1.

### II.2. SCARA Robot Simulator (3D)

Esta aplicación es un simulador 3D para probar el Robot industrial. El usuario puede configurar la velocidad y el ángulo de cada eje. Puede establecer varias opciones como el punto de destino (X, Y, Z, T), el punto de vista (X, Y, Z, T) y la velocidad para cada eje en 6 pasos. Está diseñado para comportarse de forma similar al Robot Scara utilizado en la industria real [5]. Esta aplicación se muestra en la Figura 2.

### II.3. DELTA Robot Simulator (3D)

Es un simulador 3D para probar el Robot Delta industrial de 6 ejes. Puede establecer varias opciones como el punto de destino (X, Y, Z, T), el punto de vista (X, Y, Z, T) y la velocidad para cada eje en 6 pasos. Está diseñado para comportarse de forma similar al Robot



Figura 1: ARM Robot Simulator de VINET

Delta utilizado en la industria real. Es una opción para presentar la simulación educativa a los estudiantes que están aprendiendo sobre la industria de los robots o sobre cómo manejar y diseñar robots. El funcionamiento básico de esta aplicación se basa en los 6 valores de paso establecidos en la pestaña *Step Motion Settings* [6]. Esta aplicación se muestra en la Figura 3.

### II.4. ABB IRB 4600/40

El IRB 4600/40 es un brazo robot industrial de 6 grados de libertad (DOF, por sus siglas en inglés) desarrollado por ABB. Tiene una carga útil de 40 kg. Permite visualizar una simulación de un brazo robótico con movilidad. Mediante los controles de la izquierda es posible realizar los movimientos requeridos. Con el motor A se puede

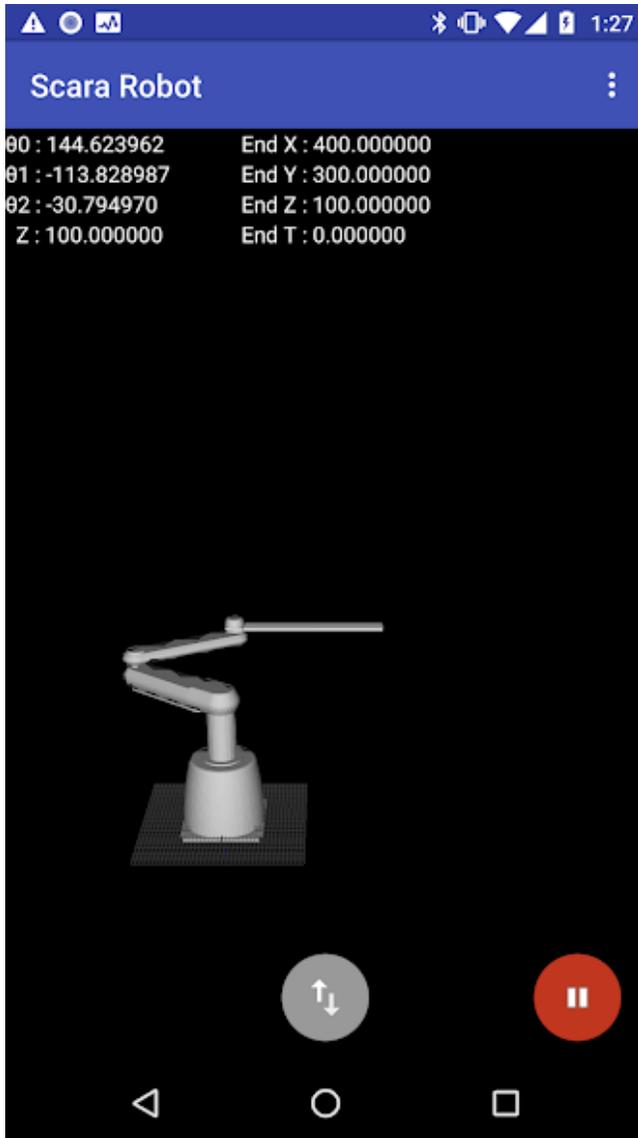


Figura 2: SCARA Robot Simulator de VINET

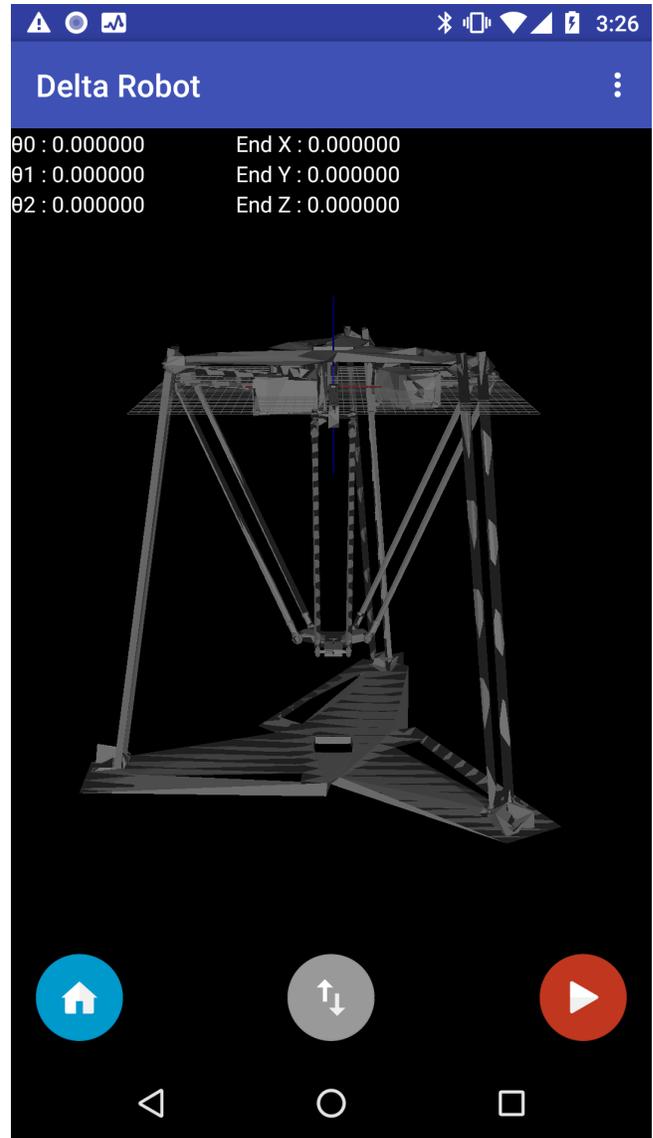


Figura 3: DELTA Robot Simulator de VINET

girar el brazo 360° en su eje. Mediante el motor B es posible subir o bajar el brazo desde su unión 1. Gracias al motor C se puede mover el brazo hacia arriba o hacia abajo desde la unión 2. Con el motor D se puede rotar la palma y los dedos se pueden mover mediante el motor E [7]. Esta aplicación se muestra en la Figura 4.

### II.5. KUKA's youBot

El youBot es un brazo robótico móvil desarrollado por KUKA. Su brazo tiene cinco grados de libertad y cuenta con una pinza lineal. Su base está diseñada con cuatro ruedas Mecanum que permiten un movimiento omnidireccional. Estas ruedas están modeladas de forma eficiente utilizando la fricción asimétrica. Es posible controlar los movimientos que el brazo realiza con los controles que

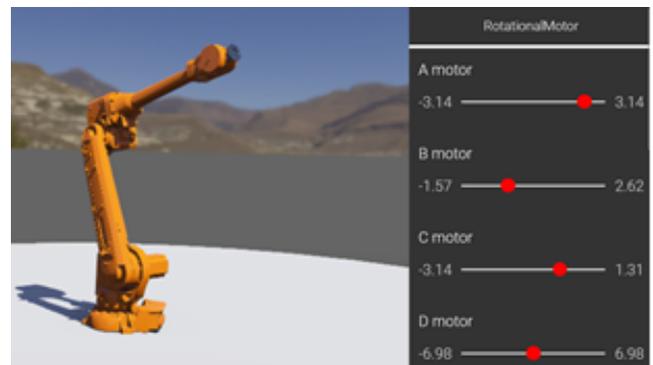


Figura 4: Simulación de ABB' IRB 4600/40

están en el panel derecho. Entre los movimientos permitidos encontramos: girar en su propio eje, moverse en sentido de derecha a izquierda y viceversa, mover el “codo” hacia los lados, mover su “mano” en sentido horizontal, girar la palma y mover las llantas con las que cuenta [8]. Esta aplicación se muestra en la Figura 5.

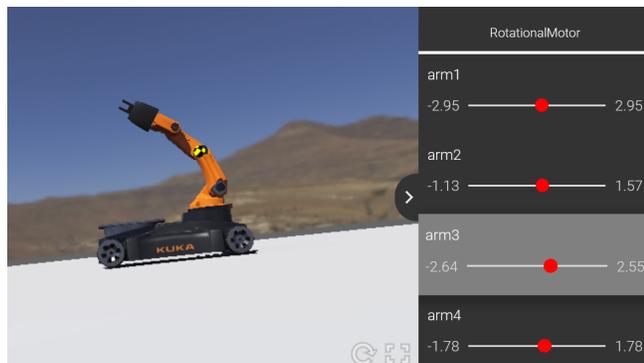


Figura 5: Simulación de KUKA'S youBot

La principal ventaja de la aplicación que simula un brazo robot en comparación con aquellas propuestas en el estado de arte consiste en que el desarrollo aquí descrito es de código abierto, lo que significa que cualquiera que solicite el código fuente podrá realizar las adecuaciones que considere necesarias.

### III. Tecnologías requeridas

En esta sección revisaremos las principales tecnologías requeridas para el desarrollo de este proyecto.

#### III.1. Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles. Lo que lo hace diferente de otros sistemas operativos es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma. Es de código abierto, lo que significa que cualquiera puede descargar el código fuente, inspeccionarlo, compilarlo e incluso cambiarlo.

Android permite programar aplicaciones utilizando el lenguaje de programación Java. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como las llamadas, la agenda, el GPS, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.

Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hacen que una de las cosas más importantes de este sistema operativo sea la cantidad de aplicaciones disponibles, que extienden casi sin límites la experiencia del usuario [9].

#### III.2. OpenGL

Open Graphics Library es una API multiplataforma de gráficos que especifica una interfaz de software estándar para hardware de procesamiento de gráficos 2D y 3D.

La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como: puntos, líneas y triángulos. Se usa ampliamente en: CAD (Computer-Aided Design), realidad virtual, representación científica, visualización de información y simulación de vuelo, así como en desarrollo de videojuegos.

Proporciona un conjunto común de comandos que se pueden usar para administrar gráficos en diferentes aplicaciones y en múltiples plataformas [10].

#### III.3. Blender

Es una suite de creación de contenido 3D totalmente integrada que ofrece una amplia gama de herramientas esenciales que incluyen: modelado, renderizado, animación y rigging, edición de video, VFX, composición, texturizado y muchos tipos de simulaciones.

Es multiplataforma, con una interfaz gráfica de usuario (GUI, por sus siglas en inglés) OpenGL que es uniforme en todas las plataformas principales. Es una arquitectura 3D que permite un flujo de trabajo de creación rápido y eficiente [11, 12].

#### III.4. Simulación a computadora

La simulación a computadora intenta modelizar sistemas reales o hipotéticos mediante un ordenador de forma que su funcionamiento puede ser estudiado con la finalidad de predecir su comportamiento [13].

Las simulaciones por computadora abarcan desde programas informáticos cuya ejecución dura unos minutos hasta conjuntos de ordenadores conectados en red cuya ejecución dura horas, e incluso hay simulaciones que se extienden varios días.

Permite anticiparse al proceso real, validarlo y obtener su mejor configuración. Se emplea en el diseño de nueva tecnología para llegar a comprender mejor su funcionamiento.

### IV. Sistema propuesto

#### IV.1. Diseño del objeto gráfico

Para el brazo robótico se utilizó la herramienta Blender, que como se mencionó anteriormente, es una herramienta de creación de contenido 3D, debido a que es el más común y es gratuito, se optó por crear el diseño del brazo robótico en dicho software. Para llegar al diseño actual del brazo se propuso varios sketches hechos en

papel basándonos en brazos robóticos industriales para seleccionar el que mejor se adaptara a este proyecto, teniendo bocetos simples, detallados y complejos, dando como resultado el diseño actual, teniendo un diseño semi-complejo, pero que se adapta mejor a la aplicación móvil. En la Figura 6, se muestra el diseño del brazo efectuado en la herramienta Blender.



Figura 6: Modelo del brazo robótico

#### IV.2. Maquetación de la aplicación

Gráficamente se desea hacer la aplicación lo más intuitiva posible, por lo cual, primero se desarrolló un modelo donde cada movimiento se realizaría mediante gestos *touch*, sin embargo, se llegó a la conclusión que estos no pueden ser modelados de una manera eficiente debido a que son difíciles de entender en un principio y son más complicados de implementar dadas las características del proyecto. Por ello, se decidió implementar una barra de botones en la parte inferior de la aplicación, la cual permite al usuario controlar cada movimiento disponible.

La Figura 7 muestra los botones diseñados para la aplicación, como se observa el objetivo de los mismos es hacer más intuitivo el funcionamiento de cada botón mediante imágenes en vez de texto.

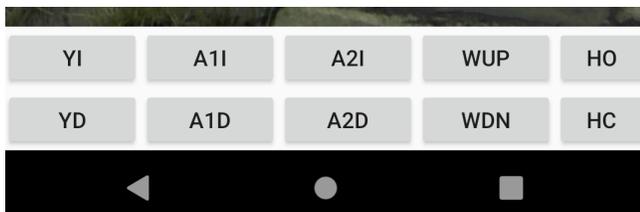


Figura 7: Diseño de botones de control

Se decidió crear los botones utilizando Photoshop, ya que este software otorga más control sobre el diseño a realizar que otro tipo de software.

Por otra parte, la estructura de la aplicación es la que se observa en la Figura 8, como se observa los botones de control se encuentran en la parte inferior de la pantalla,

existe un título en la parte superior, mientras que al centro se encuentra la simulación.

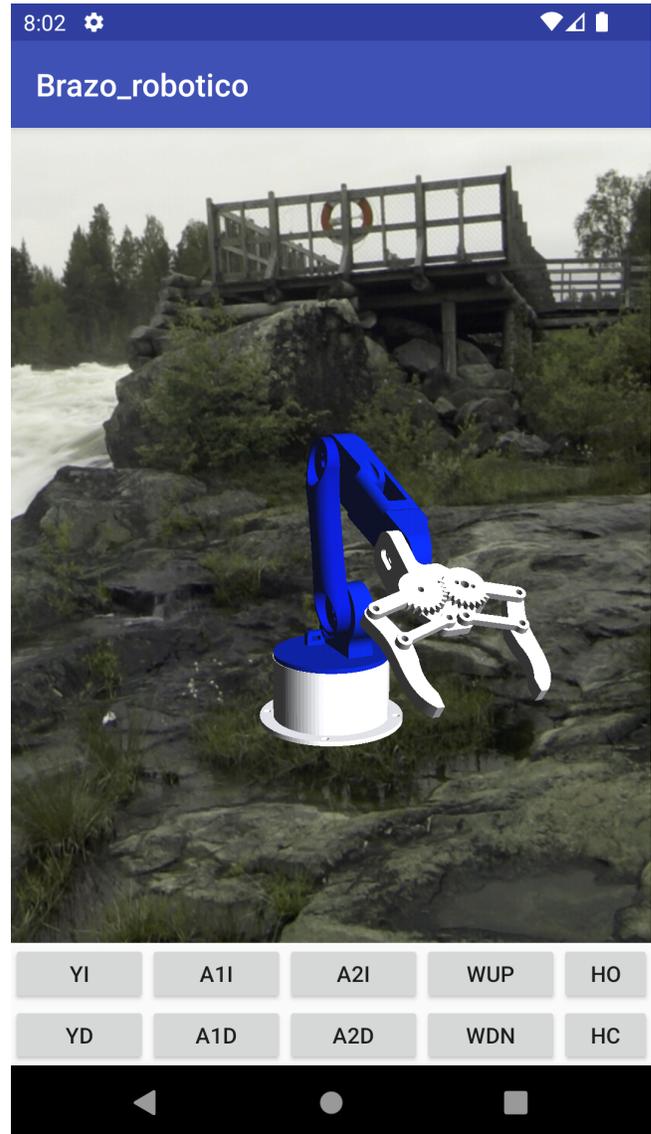


Figura 8: Maquetación de la Aplicación

#### V. Resultados

##### V.1. Librerías gráficas para el modelado del brazo

En la Figura 9 se muestra el funcionamiento de OpenGL, como se observa, se trata de un proceso laborioso, en el que cada imagen que se renderiza necesita diseñarse de manera lógica mediante un uso extensivo de matrices para embeber funcionalidad en el sistema; por otra parte, el desarrollo del proyecto se vuelve más sencillo mediante el uso de objetos diseñados en Blender, es por ello que se decidió utilizar una librería.

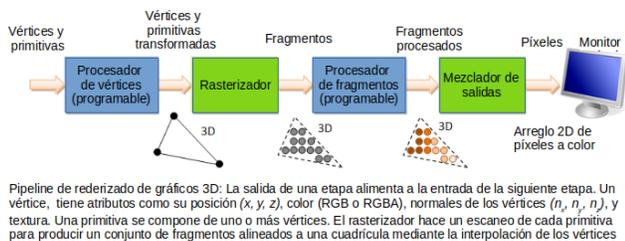


Figura 9: Canalización de Renderización [14]

Rajawali es una biblioteca de código abierto para el renderizado 3D OpenGL en Android 3. Esta librería se ha utilizado para aplicaciones de visualización de gráficos tanto en 2D como de 3D, y es un motor para Android basado en OpenGL ES 2.0/3.0 que proporciona un acceso sencillo a la potencia y eficiencia de OpenGL para sistemas embebidos [15]. Esta librería se utiliza principalmente por su flexibilidad, así como por su facilidad de uso, esencial para un proyecto con limitaciones en cuanto al tiempo de desarrollo. Esta librería permite el trabajo con archivos *.obj*, de esta forma se pueden cargar las distintas piezas que conforman el brazo robótico en formato *obj*.

### V.2. Implementación de movilidad del brazo

Para modelar el brazo robot, se analizó el modelo y se dividió en diferentes segmentos. A cada segmento se le asignó un *archivo obj*, resultado de modelado en la herramienta Blender. Otra tarea que procedió a realizarse es la coordinación en cuanto al movimiento de los componentes del brazo. A partir de videos del brazo que se requiere modelar, se procedió a delimitar los movimientos posibles de cada pieza del brazo, de tal forma que se imite al brazo original en la medida de las posibilidades en cuanto al desarrollo de éste proyecto.

El brazo robot propuesto involucra 5 grados de libertad: la base, el brazo, el codo, la muñeca (subir y bajar), rotación de la muñeca y la apertura y cerradura de los dedos (gripper). El diagrama mostrado en la Figura 12 solo fue utilizado como referencia para el modelado de la mayoría de los componentes del brazo. Los dedos no forman una articulación independiente, sino que forman parte del mismo sistema de apertura y cerradura simultánea. Para efectos de esta simulación, no fueron incluidos pesos ni materiales de los componentes, por lo que no se consideraron aspectos de fuerzas y cargas.

Mediante el uso del sistema de coordenadas se puede ir colocando cada pieza mediante prueba y error, de forma que se llega a tener el brazo renderizado pero sin movimiento. Para el movimiento del brazo se hace uso del modelo de jerarquía presentado en [16], el cual se basa en la naturaleza del movimiento de los brazos humanos, como se observa en la Figura 12, existen varias articu-



Figura 10: Renderizado aleatorio de las piezas del brazo robótico

laciones en un brazo humano, donde cada articulación mueve una parte del brazo, en la articulación del hombro (shoulder joint) al realizarse movimiento se mueve el brazo superior, inferior, la mano y los dedos, mientras que si se mueve la articulación del hombro (elbow joint) el brazo superior queda estático, moviéndose así solo el brazo inferior, la mano y los dedos, y así ocurre con el resto de las articulaciones.

Como podrá inferir el lector, existe una jerarquía en las articulaciones, donde la articulación del brazo es la que está más arriba de la jerarquía ya que permite mover el brazo entero, mientras que la articulación del dedo está en la parte de abajo, ya que permite mover solo un dedo.

La anterior lógica se implementó en el brazo robóti-



Figura 11: Brazo armado renderizado

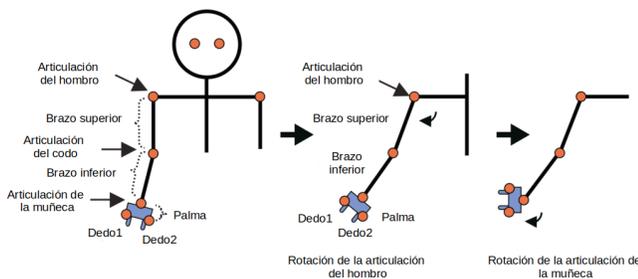


Figura 12: Estructura y movimiento desde el hombro hasta los dedos basado en el diseño de Matsuda [16]

co para que este funcionara de manera correcta, de no seguirse dicha lógica el movimiento del objeto no sería posible.

La aplicación finalizada se muestra en la Figura 13, como se observa, la aplicación implementa un skybox de una fábrica, lo cual permite hacer más realista la simulación. Las articulaciones del brazo solo pueden ser manipuladas por los botones de la interfaz, por lo que no es necesario ningún método que requiera de la pantalla Touch.



Figura 13: Aplicación Móvil de Brazo Robótico

La aplicación final cuenta con los siguientes botones, cuya función de cada botón es la siguiente:

1. Rotar la base hacia la izquierda (YI)
2. Rotar la base hacia la derecha (YD)
3. Mover brazo 1 a la izquierda (A1I)
4. Mover brazo 1 a la derecha (A1D)
5. Mover brazo 2 a la izquierda (A2I)
6. Mover brazo 2 a la derecha (A2D)

7. Mover muñeca hacia arriba (WUP)
8. Mover muñeca hacia abajo (WDN)
9. Abrir el gripper (HO)
10. Cerrar el gripper (HC)

## VI. Conclusiones y Trabajo Futuro

### VI.1. Conclusiones

Se logró la implementación de una aplicación móvil que permite al usuario la interacción con un brazo robótico.

La principal desventaja de este desarrollo, es que no permite la simulación de más objetos, mientras que la interfaz gráfica no es del todo intuitiva.

### VI.2. Trabajo Futuro

Actualmente, se puede interactuar con la simulación mediante el uso de una interfaz gráfica, sin embargo, sería deseable que en vez de una interfaz gráfica exista una simulación utilizando realidad aumentada, lo cual sería más intuitivo y realista para el usuario.

## Referencias

- [1] M. W. Spong y M. Vidyasagar. *Robot Dynamics and Control*. New York: John Wiley y Sons, 1989.
- [2] A.J. Kurdila y P. Ben-Tzvi. *Dynamics and Control of Robotic Systems*. Wiley, 2019. ISBN: 9781119524830. URL: <https://books.google.com.mx/books?id=T00qDwAAQBAJ>.
- [3] Florent P. Audonnet, Andrew Hamilton y Gerardo Aragon-Camarasa. *A Systematic Comparison of Simulation Software for Robotic Arm Manipulation using ROS2*. 2022. DOI: 10.48550/ARXIV.2204.06433. URL: <https://arxiv.org/abs/2204.06433>.
- [4] VINEIT. *ARM ROBOT SIMULATOR (3D) - Apps on Google Play*. Jun. de 2019. URL: [https://play.google.com/store/apps/details?id=com.vineit.armrobot&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.vineit.armrobot&hl=en_US&gl=US).
- [5] VINEIT. *SCARA ROBOT SIMULATOR (3D) - Apps on Google Play*. Feb. de 2019. URL: [https://play.google.com/store/apps/details?id=com.vineit.scararobot&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.vineit.scararobot&hl=en_US&gl=US).
- [6] VINEIT. *DELTA ROBOT SIMULATOR (3D) - Apps on Google Play*. Feb. de 2019. URL: [https://play.google.com/store/apps/details?id=com.vineit.deltarobot&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.vineit.deltarobot&hl=en_US&gl=US).
- [7] ABB' IRB 4600/40. 2020. URL: <https://cyberbotics.com/>.
- [8] KUKA's youBot. 2020. URL: <https://cyberbotics.com/>.
- [9] Alejandro Nieto Gonzalez. *¿Qué es Android?* Feb. de 2011. URL: <https://www.xatakandroid.com/sistema-operativo/que-es-android>.
- [10] Carlos Eduardo Oviedo de la Torre y col. «OpenGL en dispositivos móviles». En: (2004).
- [11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2022. URL: <http://www.blender.org>.
- [12] Blender. *Introducción — Blender Manual*. Ago. de 2022. URL: [https://docs.blender.org/manual/es/dev/getting\\_started/about/introduction.html#:~:text=Blender%20es%20una%20suite%20de%20creaci%C3%B3n%20de%20contenido%203D%20totalmente,y%20muchos%20tipos%20de%20Simulaciones..](https://docs.blender.org/manual/es/dev/getting_started/about/introduction.html#:~:text=Blender%20es%20una%20suite%20de%20creaci%C3%B3n%20de%20contenido%203D%20totalmente,y%20muchos%20tipos%20de%20Simulaciones..)
- [13] Facultad de Informática de Barcelona. *Simulación*. 2017. URL: <https://www.fib.upc.edu/retro-informatica/avui/simulacio.html#:~:text=La%20simulaci%C3%B3n%20por%20ordenador%20intenta,la%20evoluci%C3%B3n%20de%20la%20Inform%C3%A1tica..>
- [14] Chua Hock-Chuan. *3D Graphics with OpenGL*. 2022. URL: [https://www3.ntu.edu.sg/home/ehchua/programming/opengl/CG\\_BasicsTheory.html](https://www3.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html) (visitado 19-08-2020).
- [15] *Rajawali Official Website*. 2022. URL: <https://rajawali.github.io/Rajawali> (visitado 19-08-2020).
- [16] K. Matsuda y R. Lea. *WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL*. OpenGL. Pearson Education, 2013. ISBN: 9780133364927. URL: <https://books.google.com.mx/books?id=3c-jmWkLNwUC>.