**Introducción**

Desde tiempos remotos, el hombre ha venido utilizando distintas herramientas para poder aumentar el alcance de su capacidad de manipulación. En un principio, no se trataba más que de palos utilizados para hacer caer la fruta madura de un árbol. En la actualidad existen dispositivos más complejos, como por ejemplo pértigas que, accionando un enganche o pinza a distancia, permiten alcanzar objetos que se encuentran en estanterías o lugares de almacenamiento de difícil acceso.

En los sistemas de tele operación de robots la intervención del operador humano muchas veces es imprescindible, especialmente en entornos no estructurados y dinámicos en los cuales los problemas de percepción y planificación automática son muy complejos.

En muchos casos, el operador está físicamente separado del robot, existiendo un sistema de telecomunicaciones entre los dispositivos que utiliza directamente el operador y el sistema de control local del robot.

La comunicación a través de internet puede ser tanto en una vía como en esquemas bilaterales con realimentación, (ortega2004) para esto se han planteado algunas propuestas.

Estas propuestas han tenido gran acogida al resolver problemas de seguridad en la estabilidad de la comunicación, evitando así desconexiones entre los hosts desconexiones hacia el canal de comunicación, problemas en los servicios iniciados en las terminales, errores del sistema base de las aplicaciones, desconexión o a su vez suspensión de los servicios, la solución más aceptada es en la utilización de webservice a través de request http polling y administración por DOM, para mantener la consistencia de los valores se ha aplicado formatos en XML.

Aunque muchos de los problemas se han resulto con este esquema de comunicación, aún se ven grandes inconvenientes con el tiempo que toma un paquete llegar desde su host origen a su host destino, ya que los protocolos y modelo de interacción cliente servidor no fueron creados para conexiones en tiempo real y con opciones de comunicación full duplex en los cuales tanto los clientes como los servidores pueden iniciar una comunicación de forma autónoma.

Para simplificar el proceso de envío y recepción de información manteniendo conexiones consistentes y el manejo de datos en paquetes estructurados simplificados para evitar la pérdida de información o generación de basura en memoria se presenta la implementación de un servicio de Teleoperación  el cual utiliza la comunicación en tiempo real mediante el uso del protocolo de comunicación websocket y  de paquetes Json para el manejo de datos, permitiendo administrar los clientes que se conectan, y los mensajes que envían.

**Desarrollo**

1. **Metodología de Desarrollo de Software**

Al iniciar un proyecto de software es necesario realizar una planificación de cómo se van a cumplir correctamente los objetivos planteados.

El autor Goncalves plantea que la experiencia ha demostrado que los proyectos exitosos son aquellos que son administrados siguiendo una serie de procesos que permiten organizar y luego controlar el proyecto, considerando válido destacar que aquellos procesos que no sigan estos lineamientos corren un alto riesgo de fracasar. Es necesario destacar la importancia de los métodos, pero el éxito del proyecto depende más de la comunicación efectiva con los interesados, el manejo de las expectativas y las personas que participan en el proyecto.

Existen diferentes modelos y metodologías que han sido en los últimos años herramientas de apoyo para el desarrollo del software. El autor Somerville, menciona que “Modelo de desarrollo de software: es una representación simplificada del proceso para el desarrollo de software, presentada desde una perspectiva específica.

Es necesario definir que tele operación es un conjunto de tecnologías que comprenden la operación o gobierno a distancia de un dispositivo por un ser humano. Por tanto, teleoperar es la acción que realiza un ser humano de operar o gobernara distancia un dispositivo; mientras que un sistema de tele operación será aquel que permita teleoperar un dispositivo, que se denominará dispositivo tele operado.

Para el desarrollo del sistema de tele operación se han analizado las siguientes metodologías de desarrollo de software:

* 1. **Programación Extrema o XP (Extreme Programing)**

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

* 1. **Scrum**

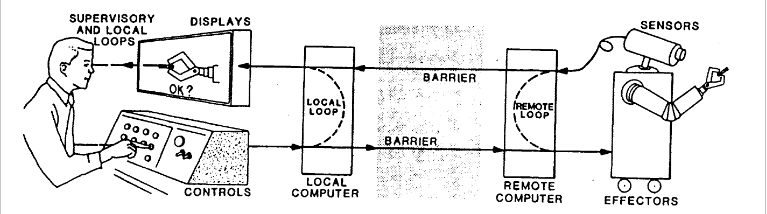
Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

En la presente investigación se utiliza la metodología Scrum, que al ser una metodología versátil permite un enfoque de gestión ágil que facilita la administración de proyectos, programas cualquier tamaño y complejidad, facilitando el flujo de información, la comunicación entre el equipo de trabajo y la entrega de valor con oportunidad a los interesados de la organización.

1. **Desarrollo del Servicio de Tele Operación**

Un servicio de tele operación consta de los siguientes elementos:

* Operador o teleoperador: es un ser humano que realiza a distancia el control de la operación. Su acción puede ir desde un control continuo hasta una intervención intermitente, con la que únicamente se ocupa de monitorizar y de indicar objetivos y planes cada cierto tiempo.
* Dispositivo teleoperado: podrá ser un manipulador, un robot, un vehículo o dispositivo similar. Es la máquina que trabaja en la zona remota y que está siendo controlada por el operador.
* Interfaz: conjunto de dispositivos que permiten la interacción del operador con el sistema de teleoperación. Se considera al manipulador maestro como parte del interfaz, así como a los monitores de vídeo, o cualquier otro dispositivo que permita al operador mandar información al sistema y recibir información del mismo.
* Control y canales comunicación: conjunto de dispositivos que modulan, transmiten y adaptan el conjunto de señales que se transmiten entre la zona remota y la local. Generalmente se contará con uno o varias unidades de procesamiento.
* Sensores: conjunto de dispositivos que recogen la información, tanto de la zona local como de la zona remota, para ser utilizada por el interfaz y el control.

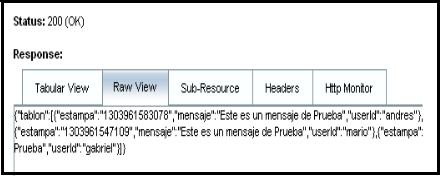
****

Esquema servicio Teleoperación

* 1. **Identificación de Clientes y Re direccionamiento**

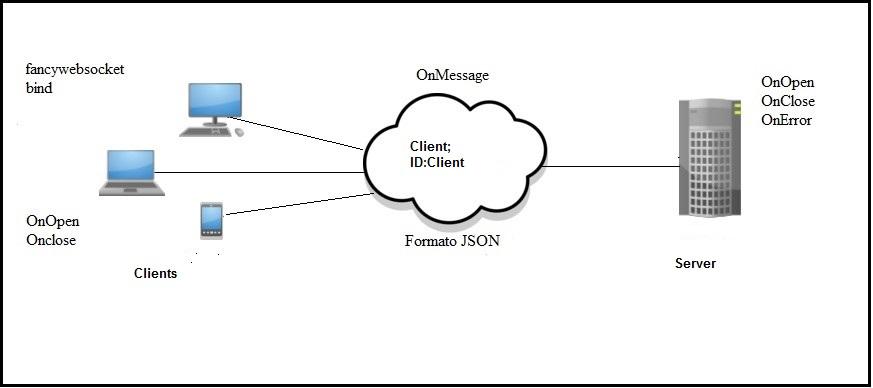
En el servicio de tele operación planteado es necesario definir el origen y destino de los datos que se envían entre los clientes y el servidor. Para esto es necesario que cada cliente se identifique utilizando un formato de datos específico que sea interpretado por el servidor, esto es necesario para lograr la redirección de paquetes de datos correspondientes a cada cliente conectado.

En la identificación de los clientes, el formato de datos utilizado para que sea interpretado tanto por el servidor como por el cliente es Json, que es un formato ligero basado en texto, con definición de lenguaje completamente independiente, cuya sintaxis está basada en un subconjunto de funcionalidades y palabras reservadas del lenguaje de script Web: JavaScript. Debido a su sencillez es fácil generar y procesar un documento con este formato; Json usa las convenciones para el manejo de datos, que son comunes a la familia de lenguajes: C, C++, C#, Java, Perl, Python, etc. Esta característica hace de Json un lenguaje ideal para el intercambio de datos entre aplicaciones cliente servidor o aplicaciones que se ejecuten en diferentes escenarios.



Formato de datos Json

Para que el cliente se pueda conectar al servidor se utiliza la siguiente estructura en formato Json: {“cliente”: “NombreCliente”}; la cual permite validar y almacenar en memoria todos estos registros.

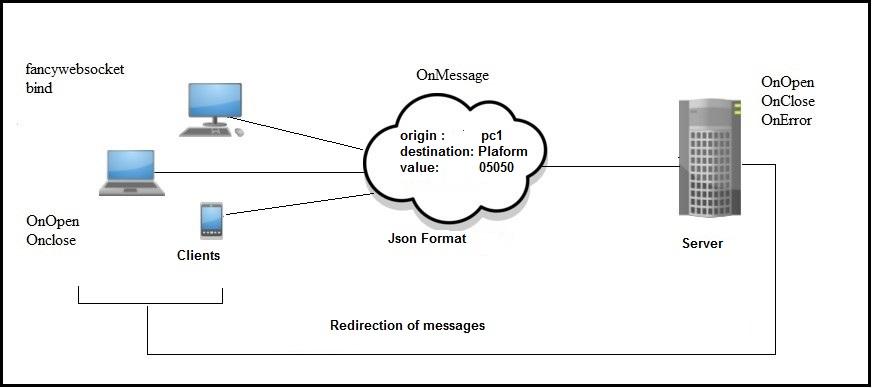


Identificación de clientes

En el Re direccionamiento es necesario que se identifique que cliente es el origen y que cliente es el destino de dicho paquete, ya que en el servidor se envía los datos en base a la identificación del cliente y al origen y destino del paquete.

El paquete es enviado por parte de los clientes al servidor debe ser escrito con la siguiente estructura:

{“origen”: “NombreOrigen”, “destino”: “NombreOrigen”, “mensaje”: “Mensaje”}



Re direccionamiento de paquetes

Este proceso permite tener el control de los datos que son enviados y los datos que reciben los clientes que estén conectados con el servidor.

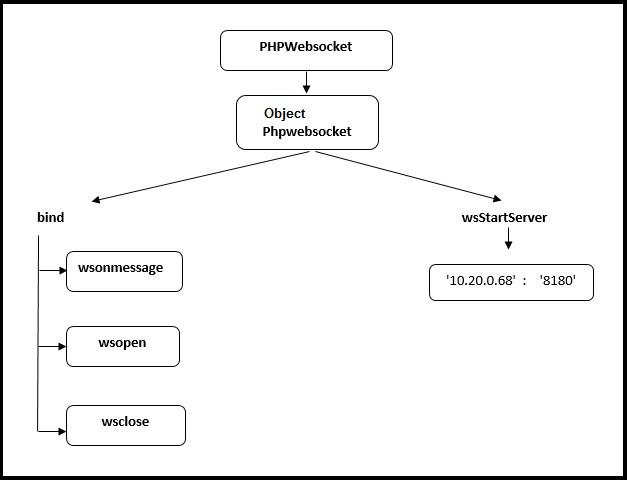
* 1. **Implementación del Servidor**

Como se trata de comunicación en tiempo real, para implementar un servidor es necesario utilizar un protocolo de comunicación que cumpla con los requerimientos de la investigación. Estos requerimientos son: comunicación en tiempo real full dúplex, retardos mínimos de tiempo, estabilidad de comunicación.

Los protocolos de comunicación son una serie de normas que usan los equipos informáticos para gestionar sus diálogos en los intercambios de información. Dos o más equipos diferentes de marcas diferentes se pueden comunicar sin problemas en el caso en que usen el mismo protocolo de comunicaciones.

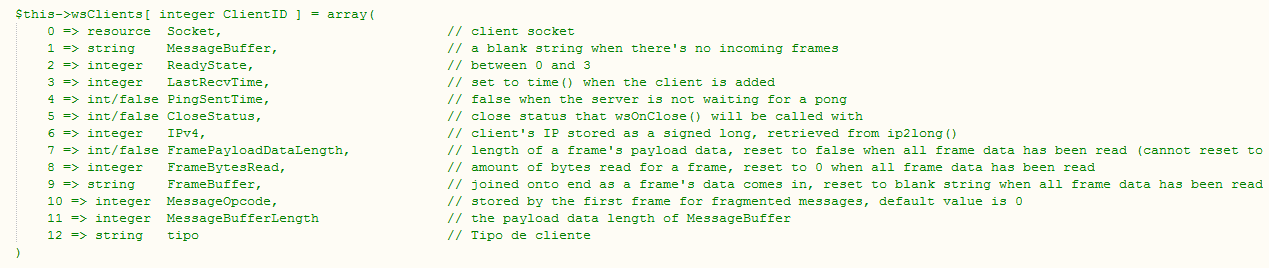
El protocolo de comunicación que se utiliza para implementar el servidor es el protocolo Websocket, que es una tecnología que proporciona un canal de comunicación bidireccional y full-dúplex. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor.

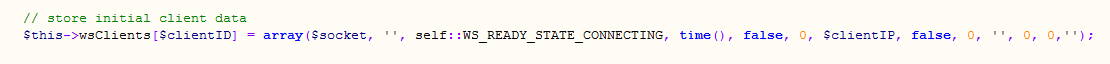
El servidor está desarrollado en PHP, utilizando la librería PHPwebsocket que es una librería libre, que contiene métodos, constantes y variables de contexto global para abrir, cerrar la conexión y manipular los mensajes que son enviados desde los clientes al servidor.



Esquema PHPwebsocket

Al tratarse de una librería libre se puede editar su contenido, para el re direccionamiento, se añadió un campo al vector predefinido que viene en la librería, este campo permite almacenar el tipo de cliente que se conecta, almacenando el nombre, origen, destino y en base a esta información poder realizar el re direccionamiento de los paquetes entre los clientes que estén conectados en el servidor.





Vector: Tipo Cliente

En el servidor se crean los métodos wsOnMessage, wsOnOpen, wsOnClose que son los que administran los mensajes que llegan al servidor, permiten abrir y cerrar la comunicación.

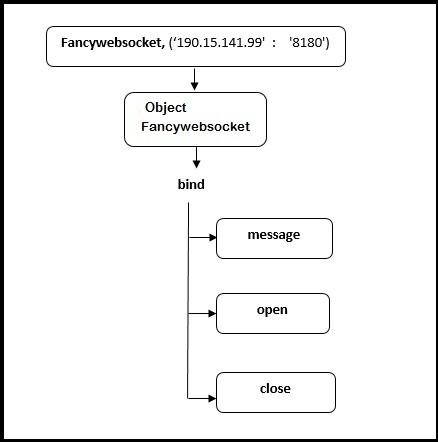
La función wsonmessage es en donde se realiza el re direccionamiento de paquetes, primero interpreta el paquete que está en formato Json y se realiza el envío de paquete basándose en el origen y destino de los paquetes que llegan.

Una vez se tienen las funciones anteriormente mencionadas se debe especificar la dirección Ip y el puerto al que deben conectarse los clientes, al tratarse de un servidor perteneciente a Cedia, se trabaja con: Ip interna 10.20.0.68, Ip Externa: 190.15.141.99, puerto: 8180.

* 1. **Cliente Web**

Una vez se tiene el proceso de tele operación se procede a realizar un cliente web que muestra los datos en tiempo real que están siendo enviados, y un modelo 3D que simula el movimiento del cliente que está siendo tele operado.

En la parte del cliente web también es necesario utilizar el protocolo de comunicación websocket, esto se consigue utilizando la librería fancywebsocket que es una librería libre escrita en JavaScript, que contiene las funciones open, close, para la abrir y cerrar la conexión, message, para administrar los paquetes de datos, además se deben especificar la dirección Ip y el puerto que están abiertos en el servidor para conocer a donde debe conectarse el cliente web.

  
Esquema Fancywebsocket

Para la visualización de los datos se utiliza JavaScript que es un lenguaje de programación interpretado orientado al lado del cliente implementado como parte de un navegador web y Jquery que es una biblioteca JavaScript que permite manipular ordenadamente los elementos de una página web.

Desde el servidor se envían todos los paquetes que le llegan, a un cliente web que de la misma manera se identificará y esperará que le lleguen paquetes, este cliente contiene los métodos para desempaquetar los mensajes que llegan en formato Json, permitiendo manipular lo que se va a mostrar en tiempo real en el cliente web.

* 1. **Streaming de Audio y Video**

La tecnología de streaming es un servicio que ha sido pensado para emisiones de Audio/Video en directo a través de Internet. Se utiliza para optimizar la descarga y reproducción de archivos de audio y video que suelen tener un cierto peso, es decir es la forma con la que se puede compartir contenido multimedia desde una página web, sin necesidad de descargarlo al ordenador.

Una transmisión de streaming nunca queda almacenada en el equipo del usuario, razón por la cual la hace un poco más segura para evitar plagio o que capturen el archivo de audio/video.

Se utilizó la Api WebRTC para implementar streaming de audio y video, Las API de WebRTC están diseñadas para permitir que las aplicaciones Javascript puedan crear conexiones en tiempo real con los canales de audio, vídeo y/o datos directamente entre usuarios a través de sus navegadores, o para servidores que soportan los protocolos de WebRTC.

1. Cliente Streaming

El cliente streaming se encarga de realizar la codificación de los datos de tal forma de que sean lo suficientemente “livianos” para transmitirse rápidamente sin perder calidad. Codecs como AAC plus y mp3 para el Audio y H264 o VP6 en video son los más comunes.

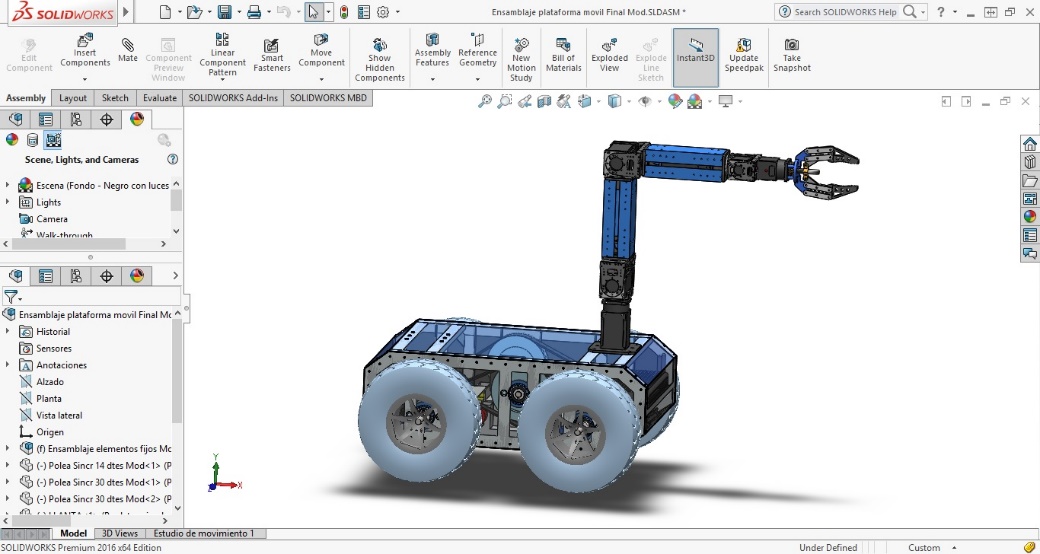
1. Servidor Streaming

Es este quien se encarga de recibir la señal desde el emisor, codificar o decodificar según sea el caso, y redistribuir dichos datos a tantos usuarios como se soliciten desde la página web del usuario.

Técnicamente cualquier PC podría ser servidor de streaming, sin embargo, dado que son estos equipos quienes procesan todos los datos de audio y video requieren gran capacidad de procesamiento. Generalmente son servidores XEON QuadCore con hasta 8 núcleos. Por otra parte, también son los encargados de la distribución hacia todo Internet. por tanto, deben tener un gran ancho de banda y transferencia para poder recibir cientos o miles de solicitudes simultáneas.

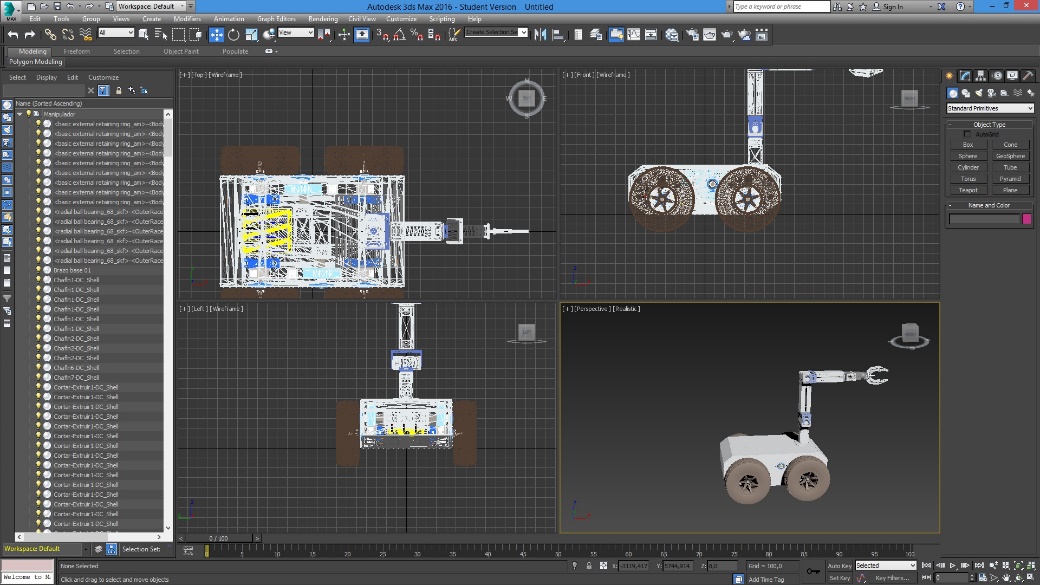
* 1. **Modelo 3D**

El modelo generalmente se obtiene a partir de un diseño CAD en SolidWorks, el cual asegura el funcionamiento cinemático sin problemas. Es necesario guardarlo como un archivo de ensamblaje (\*.ASM).



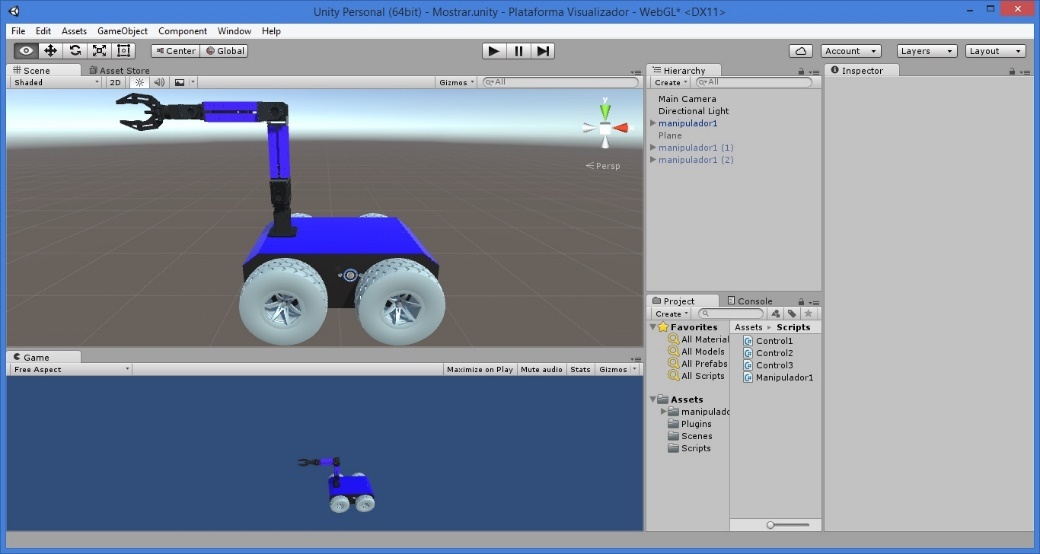
Modelo 3D del manipulador móvil en SolidWorks

Se importa el modelo en formato \*.asm en el programa 3ds Max, con el objetivo de establecer jerarquías de cada uno de los elementos que conforman el modelo 3D. La organización de los componentes mediante jerarquías asegura la correcta compatibilidad de movimiento cinemático que tendrá el modelo, dependiendo del funcionamiento del robot.



Modelo 3D del manipulador móvil en 3DMax

Después de este proceso, se debe exportar el modelo en formato \*.fbx, para posteriormente importarlo en Unity, El proceso de agregar texturas al modelo se lo realiza dentro del entorno de Unity con el fin de observar en tiempo real el estado del modelo y que los recursos empleados sean compatibles con Unity.

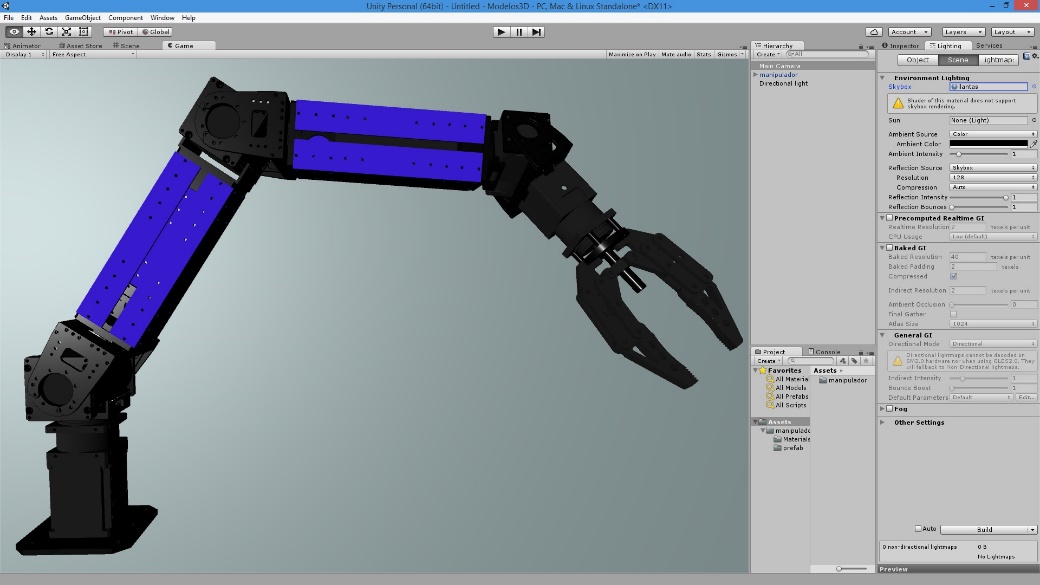


**Figura 17.** Modelo 3D del manipulador móvil en Unity.

Para darle movimiento al modelo 3D, tal y como lo haría un modelo en la vida real, se analizará su modelo cinemático. En la mayoría de modelos mecánicos y robóticos existen dos tipos de movimiento predominantes el movimiento por punto de rotación y el movimiento por rigging.

* **Movimiento por Punto de Rotación**

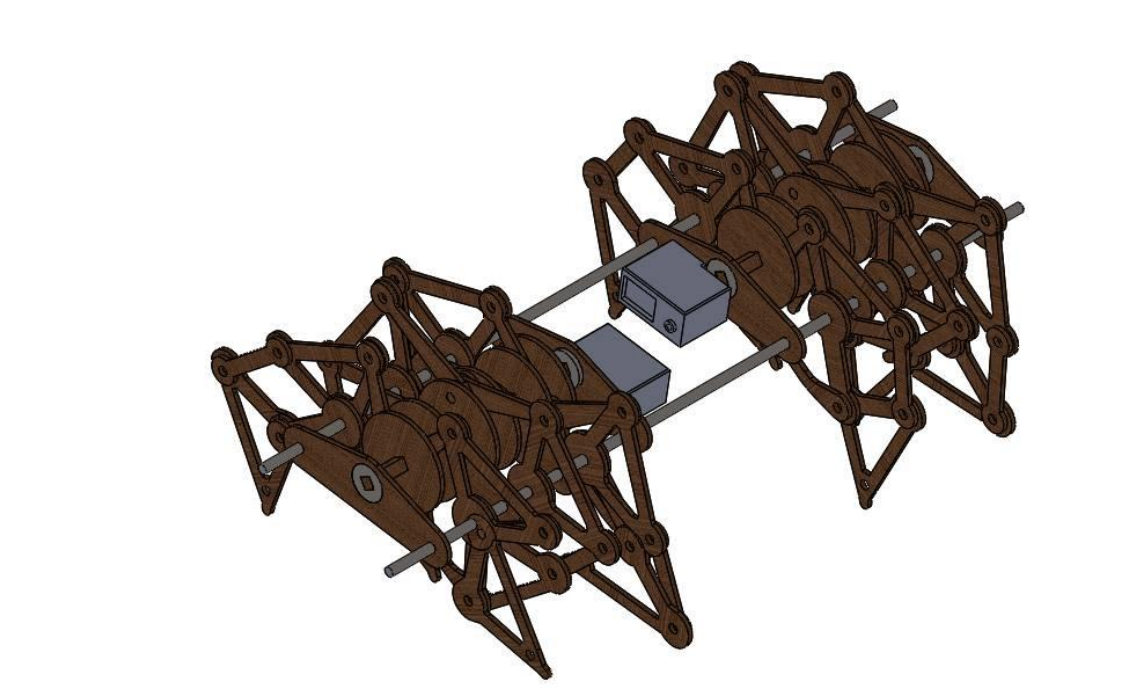
Este tipo de movimiento se caracteriza por mover independientemente cada articulación del modelo 3D, sin tener en cuenta el movimiento de otra articulación, aunque se tiene que tomar en cuenta que el script que controla la totalidad del modelo tomara en cuenta todos y cada uno de los puntos de rotación para lograr el objetivo del modelo 3D (locomoción, tarea de control, entre otros). Dentro de este tipo de movimiento se encuentran los robots que tienen motores en cada articulación como (humanoides, brazos robóticos, vehículos y similares).



Movimiento del brazo robótico por Punto de Rotación

#### **Movimiento por Rigging**

Este tipo de movimiento se usa para lograr el modelo cinemático de robots que se mueven mediante transmisión de movimiento desde un motor pasando por articulaciones hasta llegar a sus extremos operativos. El caso más representativo para agregar este tipo de movimiento es el mecanismo Theo Jansen, el cual mediante un motor es capaz de mover tres pares de patas por lado,



Movimiento por Rigging

Una vez se tiene listo el modelo 3m se lo exporta en formato WebGl, para que sea compatible con web, es decir reproducir el video 3d desde una página Web.

El formato WebGl contiene los archivos JavaScript, css, html necesarios para poder utilizar las funciones creadas desde unity para dar movimiento al modelo 3D. Al modelo 3d se le añade la funcionalidad de Websocket para que pueda reflejar los mismos movimientos que realiza el robot real, esto se logra al enviar los mismos datos al robot real y al modelo 3d creado.

REFERENCIAS