

**ARQUITECTURA DE TELECONTROL DE UN ROBOT MEDIANTE EL USO DE
INTERFACES GESTUALES**

JUAN CAMILO ESTEFAN ANGEL

**UNIVERSIDAD EAFIT
FACULTAD DE INGENIERÍAS
MAESTRÍA EN INGENIERÍA
MEDELLÍN, COLOMBIA
2013**

**ARQUITECTURA DE TELECONTROL DE UN ROBOT MEDIANTE EL USO DE
INTERFACES GESTUALES**

JUAN CAMILO ESTEFAN ANGEL

TRABAJO DE GRADO

**EDWIN MONTOYA
ASESOR**

**UNIVERSIDAD EAFIT
FACULTAD DE INGENIERÍAS
MAESTRÍA EN INGENIERÍA
MEDELLÍN, COLOMBIA
2013**

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Medellín, 11 de junio de 2013

AGRADECIMIENTOS

Muchas gracias a la Universidad EAFIT por prestar los Lego Mindstorms para llevar a cabo las pruebas necesarias para las validaciones del modelo planteado.

Especial agradecimiento a Edwin Montoya por toda su ayuda brindada para el correcto desarrollo del proyecto.

A mi padre, a mi madre y a mi hermano por todo el apoyo que me brindaron durante la planeación y elaboración de todo el proyecto.

Muchas gracias a Dios y a todas las personas que de alguna u otra forma estuvieron involucradas en el proyecto y ayudaron a que el mismo se desarrollara de manera satisfactoria.

TABLA DE CONTENIDO

LISTA DE FIGURAS	8
LISTA DE TABLAS	10
LISTA DE ANEXOS	11
RESUMEN.....	12
PALABRAS CLAVES.....	12
GLOSARIO	13
1. DEFINICIÓN DEL PROBLEMA.....	14
2. OBJETIVOS	16
2.1. Objetivo General.....	16
2.2. Objetivos Específicos.....	16
3. ALCANCE Y PRODUCTOS.....	17
4. METODOLOGÍA	18
5. MARCO TEÓRICO	19
5.1. Antecedentes.....	19
5.2. Conceptos importantes.....	20

5.2.1.	Aprendizaje de Máquina (Machine Learning)	20
5.2.2.	Modelo oculto de Markov (HMM)	20
5.2.3.	Reconocimiento de gestos.....	21
5.3.	Trabajos relacionados.....	27
6.	MODELO DEL SISTEMA.....	30
	Modelo propuesto	31
6.1.	Detección e interpretación de movimientos del cuerpo.....	34
6.1.1.	Movimientos modelados	35
6.1.1.1.	Algoritmo para detección y realización de movimientos	36
6.1.1.2.	Movimiento de brazos	38
6.1.1.3.	Movimiento de los pies	38
6.1.1.4.	Movimiento de la cabeza.....	39
6.2.	Robot	40
6.3.	Comunicación con el robot	41
6.4.	Video.....	43
7.	VALIDACIÓN DEL MODELO	45
7.1.	Detección e interpretación de movimientos del cuerpo.....	46
7.1.1.	Microsoft Kinect	46
7.1.2.	Visual Studio.....	47
7.1.3.	Movimientos reconocidos	48
7.1.3.1.	Movimiento de brazos	48
7.1.3.2.	Desplazamiento - Movimiento de ruedas	49

7.1.3.3.	Giros.....	50
7.1.3.4.	Señal de parada	51
7.2.	Robot	51
7.2.1.	Lego Mindstorms	51
7.3.	Comunicación con el Robot.....	53
7.3.1.	Comunicación mediante sockets	53
7.3.2.	Comunicación mediante Bluetooth	54
7.3.3.	Android	54
7.4.	Comunicación bidireccional por medio de video.....	54
8.	EVALUACIÓN DEL MODELO.....	55
9.	CONCLUSIONES	57
10.	REFERENCIAS.....	58

LISTA DE FIGURAS

Figura 1. Planteamiento del Problema.....	15
Figura 2. Esquema General Arquitectura de telecontrol de un robot mediante el uso de interfaces gestuales	17
Figura 3. Ejemplo de transición de estados en un modelo oculto de Márkov	21
Figura 4. Saludo mediante estados	22
Figura 5. Estados del gesto de un círculo.....	23
Figura 6. Gesto de círculo ejecutado 10 veces.....	24
Figura 7. Mensajes	25
Figura 8. Probabilidad de estado específico	25
Figura 9. Algoritmo de Baum-Welch	26
Figura 10. Matriz A inicial.....	26
Figura 11. Yaskawa Electric's next-generation robot operated using Kinect: SmartPal VII.....	28
Figura 12. Robot Asimo de Honda.....	29
Figura 13. Modelo Ideal	30
Figura 14. Articulaciones Modelo ideal y Modelo Propuesto	32
Figura 15. Modelo básico del Sistema	33
Figura 16. Modelo esqueleto.....	34
Figura 17. Fórmula: Porcentaje de Movimiento	35
Figura 18. Pseudocódigo: Detección y realización de Movimientos	37

Figura 19. Rango de movimiento de las manos.....	38
Figura 20. Rango de movimiento de los pies.....	39
Figura 21. Rango de movimiento de la cabeza.....	40
Figura 22. Partes del Robot.....	41
Figura 23. Mensaje de comunicación con el Robot.....	42
Figura 24. Ejemplo mensaje de comunicación con el Robot.....	43
Figura 25. Robot con dispositivo de video.....	44
Figura 26. Partes de la implementación del modelo.....	45
Figura 27. Microsoft Kinect.....	46
Figura 28. Esqueleto reconocido por Kinect.....	47
Figura 29. Señal de movimiento de brazos.....	48
Figura 30. Señal de desplazamiento.....	49
Figura 31. Señal de giro.....	50
Figura 32. Señal de parada.....	51
Figura 33. Partes del Lego Mindstorms.....	52
Figura 34. Lego Mindstorms utilizado en la validación del modelo.....	53
Figura 35. Cronograma.....	60
Figura 36. Mensaje SetOutputState.....	61
Figura 37. Mensaje Bluetooth.....	62

LISTA DE TABLAS

Tabla 1. Comparación Modelo ideal y Modelo propuesto	31
Tabla 2. Implementación del modelo	45
Tabla 3. Aciertos del modelo	55
Tabla 4. Latencia de envío de movimientos al Robot	56

LISTA DE ANEXOS

Anexo A. Cronograma	60
Anexo B. Comunicación de Lego Mindstorms mediante Bluetooth.....	61

RESUMEN

El proyecto *Arquitectura de telecontrol de un robot mediante el uso de interfaces gestuales*, presenta el diseño de un sistema que permite gestionar un robot con forma de humanoide de manera remota con señales y movimientos del cuerpo.

De esta forma es posible interactuar con el entorno y otras personas en donde se encuentra el robot con solo mover el cuerpo e incluso es posible establecer tele-presencia pues el sistema dispone de comunicación bidireccional de video. Esta interacción remota se puede realizar desde cualquier parte del mundo gracias a que el sistema está diseñado para trabajar mediante sockets.

El proyecto fue validado satisfactoriamente mediante la creación de un prototipo.

PALABRAS CLAVES

Tele-control, tele-presencia, reconocimiento, patrones, gestual, interfaces gestuales, movimiento, humanoide, robótica, robots, android, celulares, java, kinect, lego, mindstorms, skype, video, sockets, movilidad, comunicación remota

GLOSARIO

Humanoide¹: Que tiene forma o características del ser humano.

Bluetooth: Protocolo de comunicación inalámbrica para transferir datos entre dos o más dispositivos. Es ampliamente usado en dispositivos móviles.

Android: Sistema operativo creado por Google Inc. Usado en mayor parte en dispositivos móviles.

Kinect: Sistema de reconocimiento de movimientos creado por Microsoft Corporation.

Socket: Abstracción a nivel de software que permite el intercambio de flujo de datos entre dos o más dispositivos de manera fiable y ordenada.

Java: Lenguaje de programación de alto nivel orientado a objetos que es independiente de la plataforma debido a su proliferación en diversos dispositivos.

Legó Mindstorms: Conjunto de herramientas que permiten entender y crear sistemas robóticos mediante el uso de piezas y programación de acciones. Es desarrollado por la empresa de origen danés Legó.

HCI: Las interacciones humano-computador (HCI, de sus siglas en inglés) es el concepto que envuelve el estudio, planeación y diseño de la interacción entre las personas y las computadoras (por ejemplo los botones, las interfaces de usuario de un aplicativo de computador, etc.)

Movimientos finos: Movimientos voluntarios que realiza un humano con el fin de ejecutar actividades que requieran precisión y que involucran pequeños grupos musculares. Ejemplo: agarrar objetos.

Movimientos gruesos: Movimientos voluntarios que realiza un humano con el fin de ejecutar actividades que involucran grandes grupos musculares e implican velocidad y fuerza. Ejemplo: correr, saltar, etc.

¹ Humanoide. Real Academia Española. (2001). *Diccionario de la lengua española* (22.^a ed.). Consultado en <http://www.rae.es/rae.html>

1. DEFINICIÓN DEL PROBLEMA

Las personas son un recurso valioso para las empresas y sus conocimientos son determinantes para el correcto funcionamiento de éstas. Muchas veces el personal no puede estar en el lugar donde se necesita, posiblemente por factores como tiempo, lejanía o costo y su presencia se vuelve un factor primordial para la resolución de problemas.

La tele-presencia es una combinación de tecnologías que buscan el efecto de conseguir que una persona tenga la sensación de estar presente en un lugar alejado (ITU-T Technology, 2012). Para ello es preciso que la persona pueda emplear sus sentidos y obtener estímulos del lugar remoto. Es algo más que la videoconferencia puesto que, además de la imagen, también se comunica la posición, los movimientos, las acciones, la voz y se puede interactuar con documentos y otros objetos. Por tanto, la información que se intercomunica es amplia y muy rica en cuanto al tipo o clase de medio (Wikitel, 2012).

El concepto de tele-presencia es aplicable en campos como marketing, medicina, exploración de lugares con difícil acceso humano, turismo e incluso a nivel militar. Actualmente hay sistemas que aplican el concepto de tele-presencia, permitiendo realizar videoconferencias e interactuar con documentos digitales, sin embargo, la posibilidad de desplazarse de manera autónoma o interactuar con objetos físicos no es común. Un ejemplo de este tipo de sistemas es el *Polycom® RealPresence™ Desktop* (Polycom, 2012) que permite realizar videoconferencias e interactuar con objetos digitales pero sin la posibilidad de desplazarse de manera autónoma en el lugar de interés.

Teniendo en cuenta este panorama y el concepto de tele-presencia se vuelve inherente la necesidad de movilidad para poder tener una mejor interacción y perspectiva del entorno, donde las personas puedan desplazarse en un lugar determinado, realizar acciones y observar el mismo de manera remota.

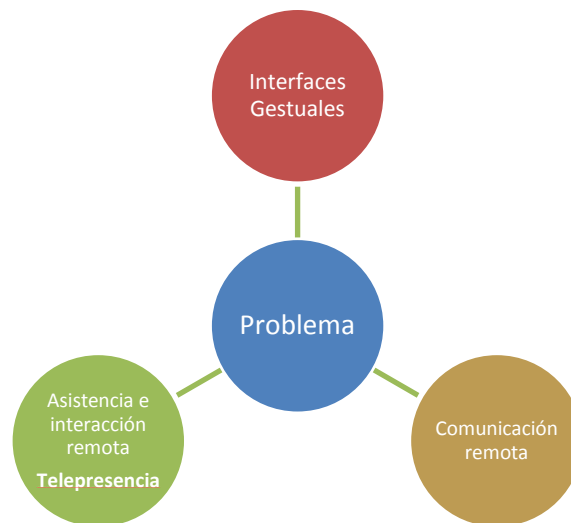
Las interacciones humano-computador (HCI, de sus siglas en inglés) es el concepto que envuelve el estudio, planeación y diseño de la interacción entre las personas y las computadoras. Generalmente las formas de poder interactuar con una máquina o dispositivo han sido con controles físicos como botones, palancas e incluso trajes. Sin embargo, la tecnología ha permitido que el mismo cuerpo sea el instrumento para poder enviar órdenes a dispositivos y este mecanismo sería ideal para usarlo en tele-presencia pues permitiría que las personas pudieran actuar con el entorno de una forma más natural.

Las HCI han evolucionado últimamente hacia la interacción humana mediante el reconocimiento de gestos. Ejemplo de ello son los controles de interacción mediante acelerómetros como son los controles WiiMote del Nintendo Wii o los controles PS Move de Playstation 3. Pero tal vez el más destacable en este sentido sea el *Microsoft Kinect* que captura el movimiento del cuerpo y procesa esta información reconociendo patrones como gestos, símbolos o acciones corporales. *Kinect* ha vendido hasta el día de hoy más de 18 millones (Joystiq, 2012) de dispositivos y ha sido utilizado en campos como los videojuegos, la investigación, la robótica, entre otros. En la medicina cabe resaltar el sistema *Gesture-based Surgical Manipulation of a da Vinci Robot using a Kinect* que permite realizar cirugías a distancia con muy buena precisión (YouTube LLC, 2012).

Kinect tiene un gran potencial comercial e investigativo pues gracias a su excepcional arquitectura y sensores permiten explorar nuevas formas de interacción con el entorno, concepto fundamental en innovación.

El problema de este proyecto es cómo introducir interfaces gestuales para la manipulación remota de un robot en un ambiente de tele-presencia.

Figura 1. Planteamiento del Problema



2. OBJETIVOS

2.1. Objetivo General

Diseñar un sistema que permita gestionar remotamente un robot con forma de humanoide mediante el reconocimiento de gestos corporales.

2.2. Objetivos Específicos

- Realizar la caracterización del sistema de adquisición gestual que permita el telecontrol de un robot humanoide.
- Realizar la caracterización del robot con forma de humanoide a ser controlado remotamente mediante una interface gestual.
- Diseñar un sistema que permita la integración de la interacción gestual con un robot humanoide a distancia.
- Diseñar un protocolo que permita la interacción humana de manera remota
- Validar el modelo/arquitectura propuesta mediante la implementación de un prototipo.
- Evaluar el modelo/arquitectura con el prototipo realizado.

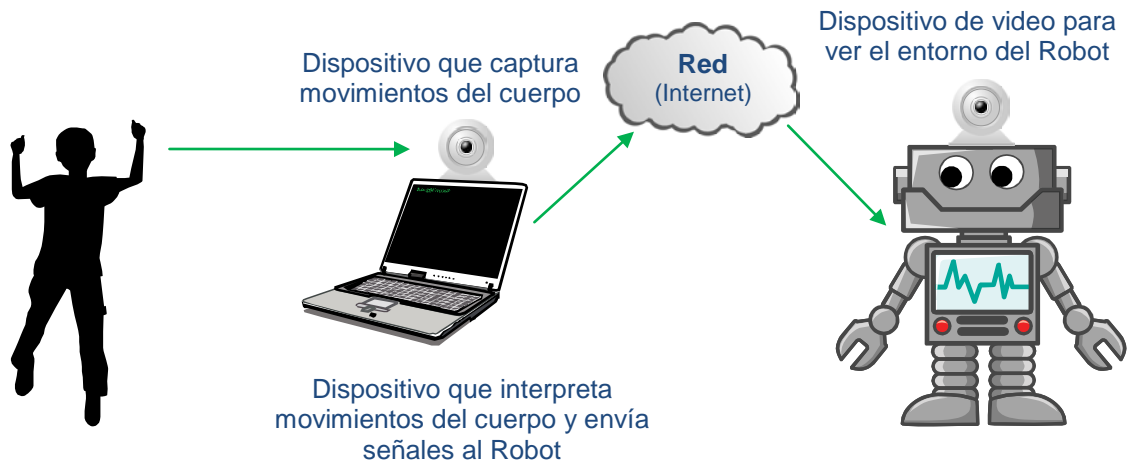
3. ALCANCE Y PRODUCTOS

El proyecto tiene como alcance el diseño y validación del sistema que permita gestionar un robot con forma de humanoide de manera remota con señales y movimientos del cuerpo, posibilitando así la interacción con el entorno en el que se encuentra.

La idea del sistema es poder generar una interacción de tele-presencia entre receptor y emisor donde puedan observarse y escucharse entre sí mediante el empleo de cualquier tipo de protocolo multimedia.

A continuación se plantea un esquema de la solución inicial del problema.

Figura 2. Esquema General Arquitectura de telecontrol de un robot mediante el uso de interfaces gestuales



4. METODOLOGÍA

Se usó el método empírico-analítico en su rama experimental que ha sido uno de los que más resultados han dado. Aplica la observación de fenómenos, que en un primer momento es sensorial. Con el pensamiento abstracto se elaboran las hipótesis y se diseña el experimento para poder reproducir el objeto de estudio, controlando así el fenómeno y llegar a una posible hipótesis válida.

En el **método empírico experimental**, (Radrigan R, 2005) el investigador interviene sobre el objeto de estudio modificando a éste directa o indirectamente para crear las condiciones necesarias que permitan revelar sus características fundamentales y sus relaciones esenciales bien sea:

- Aislando al objeto y las propiedades que estudia de la influencia de otros factores.
- Reproduciendo el objeto de estudio en condiciones controladas.

Así, los datos son sacados de la manipulación sistemática de variables en un experimento. Una diferencia clara con el método empírico en general es que éste además trata de considerar los errores de modo que una inferencia pueda ser hecha en cuanto a la causalidad del cambio observado.

Tipo de investigación:	Empírico – Experimental – Exploratorio
Campo de investigación:	Ciencias de la computación
Línea de investigación:	Telemática – Software- Hardware
Temática de investigación:	Protocolos, tele-presencia, interfaces gestuales

5. MARCO TEÓRICO

5.1. Antecedentes

Las formas de comunicación en los humanos son muy variadas. Podemos remontarnos a la prehistoria donde usábamos simples gestos y sonidos para poder comunicarnos entre sí. La evolución nos ha ayudado a podernos comunicar mejor gracias a las nuevas formas de expresión y tecnologías que vamos desarrollando. El lenguaje es una de las herramientas más comunes que usamos para comunicarnos y poco ha evolucionado desde su creación.

Con la aparición de las primeras computadoras para usuario final a principios de los años 80s nace un nuevo concepto para la comunicación de los humanos con dichos dispositivos. Este concepto se denomina HCI (Human-computer interaction, interacción humano-computador en español) y fue usado por primera vez en 1980 por Card, Moran y Newell (Card, Moran, & Newell, 1980). Las HCI abarcan diferentes clasificaciones de acuerdo a la forma de interacción que se quiera establecer con una máquina, sin embargo, para la presente investigación es de importancia las interfaces de usuario.

Dentro de las interfaces de usuario se pueden distinguir dos conceptos básicos:

- **Input (entrada):** Permite a los usuarios manipular los sistemas.
- **Output (salida):** Permite a los sistemas indicar los efectos de la manipulación de los usuarios.

De acuerdo a la forma de interacción del usuario, actualmente los tipos de HCI más comunes son:

- **GUI** (Interfaces de usuario gráficas, de sus siglas en inglés): Aceptan dispositivos de entrada como teclados y ratones. Los dispositivos de salida son comúnmente las pantallas de los computadores.
- **Touchscreens** (Pantallas sensibles al tacto): Muy usadas en teléfonos inteligentes, cajeros, puntos de venta y tabletas. El dispositivo de entrada este caso serían los dedos o un stylus (palo que permite la interacción con una pantalla). El dispositivo de salida sería una pantalla sensible al tacto.
- **Command-Line Interfaces** (interfaces de línea de comandos): Usadas principalmente por administradores de sistemas y programadores. La entrada en este caso sería un conjunto de caracteres y la salida sería una respuesta en caracteres generada en un monitor.

- **Gesture interfaces** (interfaces gestuales): Es una extensión de las interfaces de usuario gráficas, varía respecto a ellas en el sentido que la entrada en vez de ser un dispositivo como un teclado, sería en este caso, gestos corporales que son reconocidos por dispositivos como cámaras o sensores.

Generalmente las formas de poder interactuar con una máquina o dispositivo han sido con controles físicos como botones, palancas e incluso trajes. Sin embargo, la tecnología ha permitido que el mismo cuerpo sea el instrumento para poder enviar órdenes a dispositivos y este mecanismo sería ideal para usarlo en tele-presencia pues permite que las personas puedan interactuar con el entorno de una forma más natural. Las interfaces gestuales hacen parte de esta forma de interacción y es la que se va a utilizar en la presente investigación.

5.2. Conceptos importantes

Durante el desarrollo del documento hay ciertos conceptos que son de alto interés tener claros para poder entender con mayor facilidad todo el proceso argumentativo del mismo. A continuación se presentan dichos conceptos.

5.2.1. Aprendizaje de Máquina (Machine Learning)

Es una rama de la inteligencia artificial cuyo objetivo es desarrollar sistemas y técnicas que permitan a las máquinas aprender a partir de datos obtenidos. De esta forma se puede lograr que un sistema pueda tomar decisiones a partir del aprendizaje, concepto muy útil en reconocimiento de patrones.

5.2.2. Modelo oculto de Markov (HMM)

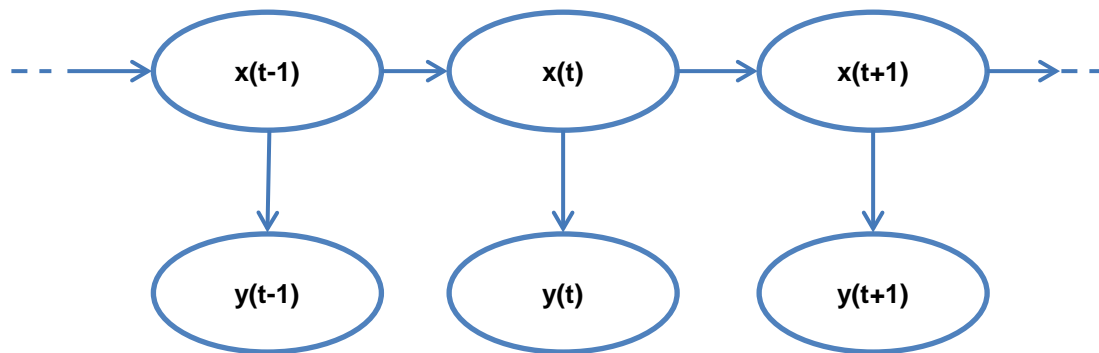
Es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de Márkov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos de dicha cadena a partir de los parámetros observables. Los parámetros extraídos se pueden emplear para llevar a cabo sucesivos análisis, por ejemplo en aplicaciones de reconocimiento de patrones como los sistemas de reconocimiento del habla (Baker, 1975). Un HMM se puede considerar como la red bayesiana dinámica más simple.

En un modelo de Márkov normal, el estado es visible directamente para el observador, por lo que las probabilidades de transición entre estados son los únicos parámetros. En un modelo oculto de Márkov, el estado no es visible directamente, sino que sólo lo son las variables influidas por el estado. Cada estado tiene una distribución de probabilidad sobre los posibles símbolos de salida. Consecuentemente, la secuencia de símbolos generada por un HMM proporciona cierta información acerca de la secuencia de estados.

El diagrama que se observa en la Figura 3 muestra la arquitectura general de un HMM. Cada óvalo representa una variable aleatoria que puede tomar determinados valores. La variable aleatoria $x(t)$ es el valor de la variable oculta en el instante de tiempo t . La variable aleatoria $y(t)$ es el valor de la variable observada en el mismo instante de tiempo t . Las flechas indican dependencias condicionales.

En la Figura 3 adicionalmente se establece que el valor de la variable oculta $x(t)$ solo depende del valor de la variable oculta $x(t-1)$. A esto se le llama propiedad de Márkov. De forma similar, el valor de la variable observada $y(t)$ solo depende del valor de la variable oculta $x(t)$.

Figura 3. Ejemplo de transición de estados en un modelo oculto de Márkov



En el próximo numeral **5.2.3 Reconocimiento de gestos**, se detallará el funcionamiento de este modelo con el caso específico del reconocimiento de un círculo.

5.2.3. Reconocimiento de gestos

Es una rama de la visión por computador con el propósito de interpretar gestos humanos por medio de algoritmos matemáticos. Estos gestos pueden ser generados por los movimientos de la mano, la cara o incluso del cuerpo.

Es posible llegar al reconocimiento de gestos por medio del empleo de cámaras y sensores aplicando algoritmos y modelos matemáticos. Uno de los modelos más usados en el reconocimiento de gestos es el HMM ya que permite modelar estados respecto al tiempo.

El siguiente ejemplo ilustra el uso del modelo HMM para llegar al reconocimiento de un gesto:

Figura 4. Saludo mediante estados



Fuente: <http://www.creativedistraktion.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

Como se puede apreciar en la Figura 4, un saludo puede ser modelado como una secuencia de estados. Cada estado en este caso es reconocido como una posición en el espacio (coordenadas x , y , z).

En un escenario ideal estos estados pueden reconocer un saludo, pero los movimientos pueden variar. Por ejemplo una persona puede saludar con otra variación angular o con movimientos más fuertes, dificultando así su reconocimiento.

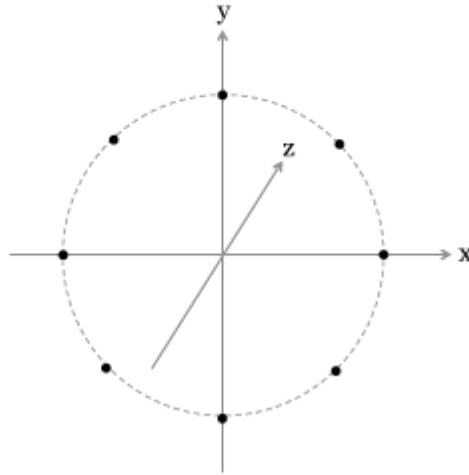
De esta forma es útil usar el modelo HMM donde se manejan probabilidades para cada estado y es posible mejorar el reconocimiento de acuerdo al aprendizaje que vaya teniendo el sistema.

Para ilustrar lo que se menciona con anterioridad es útil colocar el ejemplo de un gesto de un círculo. En la Figura 5 se puede apreciar los diferentes estados necesarios para poder reconocer el gesto de un círculo en un escenario ideal.

Figura 5. Estados del gesto de un círculo

An Ideal Circular "0" Gesture

Showing eight (8) states.



Fuente: <http://www.creativedistraktion.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

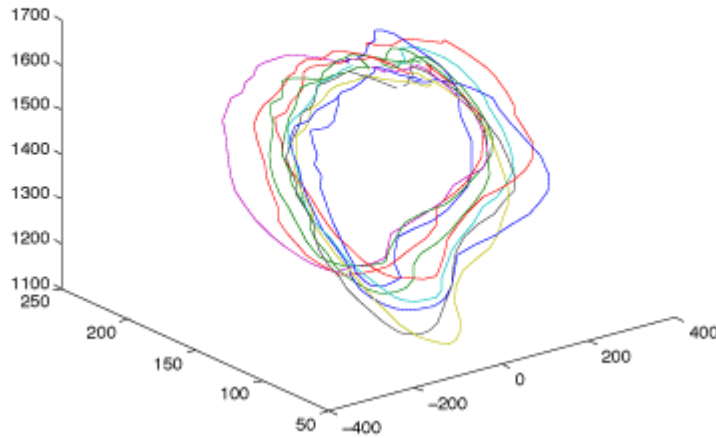
Sin embargo en la realidad hay otras variables que pueden perjudicar el correcto reconocimiento del gesto. Estas variables pueden ser: poca luz, movimientos bruscos, otros objetos interfiriendo.

A estas variables se les llama "ruido" y este mismo debe ser eliminado o ignorado con el fin de reconocer bien los gestos. Esto se logra por medio del modelo HMM y estadística. En la Figura 6 se puede apreciar cómo puede variar la realización del gesto en círculo ejecutado 10 veces.

Figura 6. Gesto de círculo ejecutado 10 veces

Noisy, Imperfect Gesture Data

Showing ten (10) circular "O" gestures in 3D space.



Fuente: <http://www.creativedistraktion.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

Para resolver esto, es necesario plantear el siguiente modelo:

Para determinar si una secuencia de observaciones se ajusta al modelo, se requiere hacer una serie de cálculos estadísticos. El primero puede ser expresado como la probabilidad de que un estado verdadero i en el paso del tiempo t con los datos:

$$Pr_t(i) = \Pr (s_t = i|O)$$

Si somos capaces de calcular esta probabilidad marginal en cada paso de tiempo en una secuencia, podemos predecir los estados ocultos y determinar en qué medida los datos se ajustan a nuestro modelo gestual. Es necesario recordar que el modelo que se está usando es una cadena de Markov, o una secuencia de eventos aleatorios, por lo que también es de interés las transiciones entre estados (Hall, 2011).

Es posible calcular las probabilidades marginales del estado de ser lo que es (de emisiones) y el cambio (transición) en cada paso de tiempo utilizando un algoritmo de paso de mensajes llamado algoritmo de adelante-atrás, donde los mensajes se calculan como:

Figura 7. Mensajes

$$\alpha_{t+1}(j) = \sum_{i=1}^N [\alpha_t(i) a_{ij}] b_j(o_{t+1})$$

$$\beta_{t+1}(j) = \sum_{i=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$

Fuente: <http://www.creativedistracton.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

La probabilidad de que el modelo esté en un estado específico i en el tiempo t y la probabilidad de transición del estado i al estado j en el tiempo t se puede estimar de la siguiente forma:

Figura 8. Probabilidad de estado específico

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{Pr(O|\lambda)}$$

$$\gamma_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{Pr(O|\lambda)}$$

Fuente: <http://www.creativedistracton.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

En última instancia, el modelo va a tabular estas probabilidades en la matriz de transición A y la matriz de emisión B . Después de que haber calculado las probabilidades marginales en el tiempo t con la fórmula de la Figura 8, se puede volver a estimar los parámetros del modelo con algoritmo de cálculo óptimo iterativo de optimización demostrado por Baum y Welch, el algoritmo de Baum-Welch, que consta de dos ecuaciones de actualización, *alfa* y *beta* (Hall, 2011).

Figura 9. Algoritmo de Baum-Welch

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$
$$\hat{b}_j(o_k) = \frac{\sum_{t: O_t = o_k} \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)}$$

Fuente: <http://www.creativedistraccion.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

Aunque es necesario inicializar los parámetros del modelo, la inicialización más importante para un modelo de gestos es la matriz de transición, porque da la oportunidad de limitar las posibles transiciones. Colocando en cero las probabilidades en los bordes de la matriz, es decir, las probabilidades de saltar entre estados distantes en el modelo, se puede restringir el modelo a la secuencia de estados adyacentes que constituyen un gesto. En otras palabras, se elige una antes de restringir la matriz de transición de tal manera que las transiciones de estado pueden ocurrir solamente en una dirección y entre dos estados adyacentes, también llamado un HMM de izquierda a derecha. La matriz A definida con lo anteriormente dicho se verá por lo tanto así: (Hall, 2011)

Figura 10. Matriz A inicial

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Fuente: <http://www.creativedistraccion.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>

De esta forma se puede garantizar que el reconocimiento podrá ser lo más cercano posible a los estados ideales planteados teniendo en cuenta las posibles variaciones que se puedan presentar.

5.3. Trabajos relacionados

Observando el estado del arte de proyectos que integren un sistema de tele-presencia con el reconocimiento de gestos y el uso de un robot se pueden encontrar bastantes, sin embargo se pueden destacar los siguientes por su similitud con el desarrollado en este documento:

- **Yaskawa Electric's next-generation robot operated using Kinect: SmartPal VII** (DigInfoTV, 2012): Es un proyecto de origen japonés llevado a cabo por la empresa *Yaskawa Electric* que es una de las líderes en la producción de robots industriales en el mundo. Es un robot que es operado remotamente mediante el sistema de captura de movimiento *Kinect*. Está orientado inicialmente para realizar tareas de cuidado de los niños del hogar, algo similar a una niñera remota. Hasta el momento no hay planes de liberarlo comercialmente ni costos aproximados. En la Figura 11 se muestra una imagen de un prototipo funcional del proyecto.
- **NXT Avatar** (The NXT Step, 2012): Es un proyecto que utiliza la tecnología Lego Mindstorms y un celular iPhone 4 para realizar tele-presencia. La gestión remota se realiza mediante un software que ofrece una interfaz gráfica de usuario compuesta por botones para gestionar el robot.

Figura 11. Yaskawa Electric's next-generation robot operated using Kinect: SmartPal VII



Fuente: <http://www.diginfo.tv/v/11-0248-r-en.php>

Si se ignora todo el sistema y se observa solo la parte robótica se puede destacar el robot de *Honda* llamado *Asimo* (Honda, 2013). Tal vez sea el robot con forma de humanoide más avanzado que existe pues cuenta con más de 20 años de desarrollo (Honda, 2013) y sigue en constante evolución. Tiene las articulaciones más comunes de los humanos: cada dedo, hombros, manos, piernas, rodillas, etc. Se caracteriza por poder caminar de forma natural como un humano, realizar actividades delicadas de tacto como servir bebidas e incluso es capaz de subir y bajar escaleras. En la Figura 12 se puede apreciar una foto de este robot.

Figura 12. Robot Asimo de Honda



Fuente: Honda. <http://asimo.honda.com>

Como se puede ver con los anteriores proyectos, el desarrollado en este documento destaca en ser un sistema económico en su forma de implementación. Se ofrece la posibilidad de gestión remota mediante el sistema de captura de movimientos de *Kinect*, es decir, solo con el cuerpo se gestiona el robot y se ofrece la posibilidad de interacción con el entorno gracias al uso de la videoconferencia. Además se garantiza su uso a nivel global por utilizar sockets como medio de comunicación.

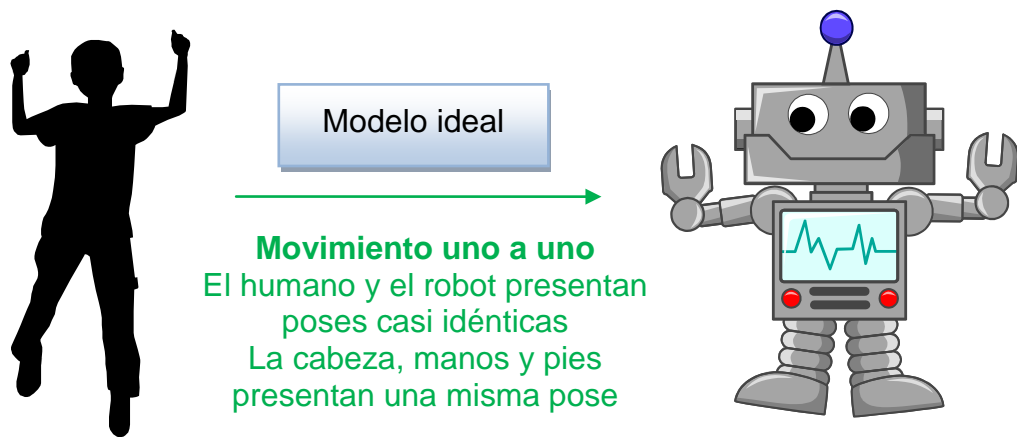
6. MODELO DEL SISTEMA

El presente proyecto tiene como objetivo diseñar un sistema que permita gestionar remotamente un robot con forma de humanoide mediante el reconocimiento de gestos corporales. Para llevar a cabo esta idea es necesario disponer de dispositivos que permitan el reconocimiento de dichos gestos para luego ser interpretados y enviados a un robot para que se pueda producir los movimientos necesarios y establecer la tele-presencia en el lugar remoto deseado.

Cuando se habla de un modelo ideal, sería aquel donde cada articulación del cuerpo que produce [movimientos finos](#) y [gruesos](#) fuera reconocida en el espacio tridimensional y pudiera ser traducido en movimientos de un robot, estableciendo un movimiento uno a uno.

La Figura 13 ilustra un ejemplo de este modelo ideal.

Figura 13. Modelo Ideal



Como se aprecia en la figura anterior, el robot realiza un movimiento uno a uno de los gestos corporales. Se puede hablar de un modelo ideal cuando se dispone de un robot actuador que tenga las mismas articulaciones de un cuerpo humano con el fin de propiciar el mayor realismo y naturalidad a los movimientos reconocidos del cuerpo. También es necesario en un modelo ideal que todo movimiento realizado por el cuerpo sea enviado al robot sin realizar variaciones en su forma con el fin de propiciar un movimiento lo más similar posible a la realidad.

Modelo propuesto

El modelo que se plantea en este documento, a diferencia de un modelo ideal, tiene como base reconocer los movimientos del cuerpo en un plano 2D y de ciertos puntos del cuerpo (más adelante se describirán), restricciones que impiden reconocer algunos movimientos. En la Tabla 1 se muestra un comparativo entre el modelo ideal y el planteado.

Tabla 1. Comparación Modelo ideal y Modelo propuesto

Modelo ideal	Modelo propuesto	Observaciones
Reconocimiento en 3D	Reconocimiento en un plano 2D	No es posible reconocer movimientos finos ni profundidad
Robot con todas las articulaciones del cuerpo	Robot con ciertas articulaciones del cuerpo	Solo algunas partes del cuerpo se podrían mover en el robot. En el caso de este modelo: cabeza, pies y manos
Movimientos finos y gruesos	Movimientos gruesos	Solo es posible establecer movimientos gruesos en el modelo planteado

Las articulaciones que se tienen en cuenta en el modelo propuesto son las siguientes:

- Cabeza
- Manos
- Hombros
- Caderas
- Pies

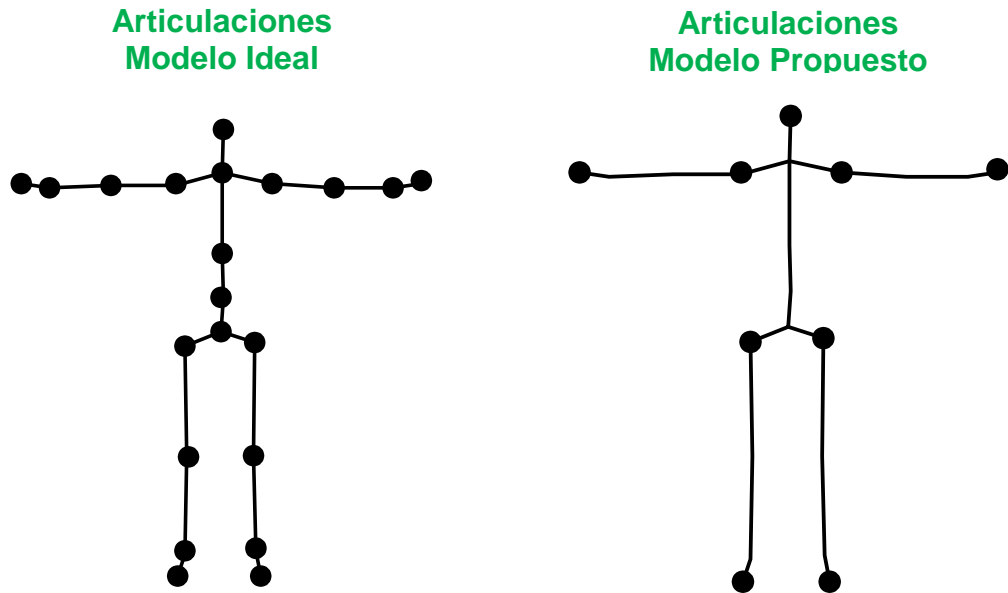
Con estas articulaciones es posible definir los siguientes movimientos:

- Movimiento de Cabeza
- Movimiento de Pies
- Movimiento de Brazos

En el numeral **6.1.1 Movimientos modelados** se detalla cómo se reconocen estos movimientos.

La Figura 14 muestra un comparativo entre el modelo propuesto y el modelo ideal respecto a las articulaciones tenidas en cuenta en ambos casos y dibujadas en un esqueleto humano.

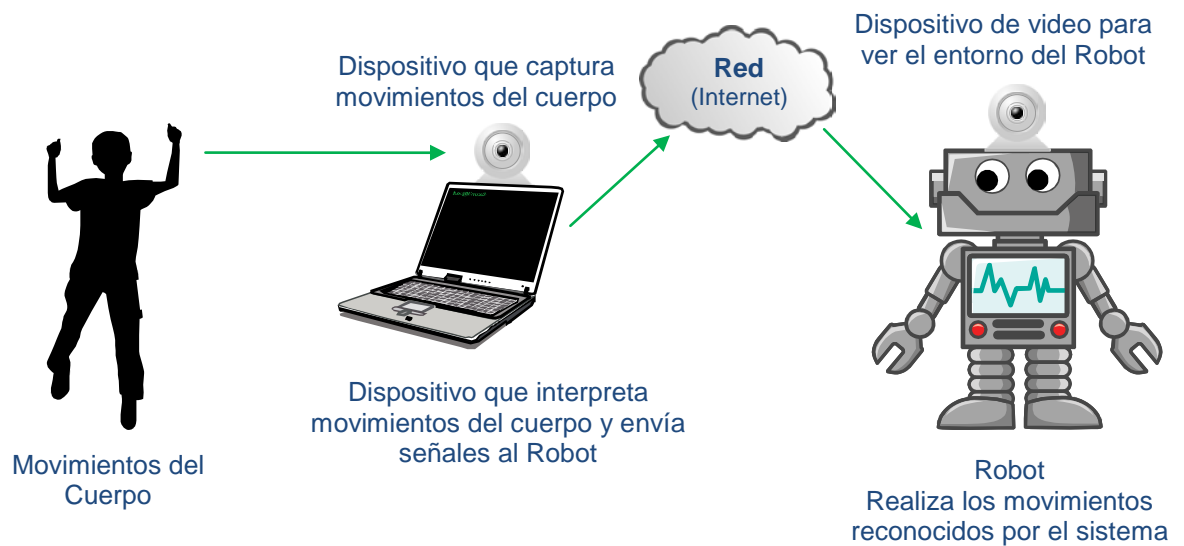
Figura 14. Articulaciones Modelo ideal y Modelo Propuesto



Teniendo como base lo anteriormente expuesto, quedaría faltando las partes que permiten que el modelo propuesto sea funcional y pueda ser llevado hasta un robot. Para ello es necesario establecer un sistema que permita el reconocimiento e interpretación de los movimientos del cuerpo para que después sean enviados al robot y sean procesados con el fin de producir el movimiento deseado.

En la Figura 15 se puede apreciar un gráfico con el modelo básico del sistema teniendo en cuenta las partes que se describieron anteriormente.

Figura 15. Modelo básico del Sistema



El sistema ha sido desglosado en las siguientes partes para facilitar mejor su comprensión:

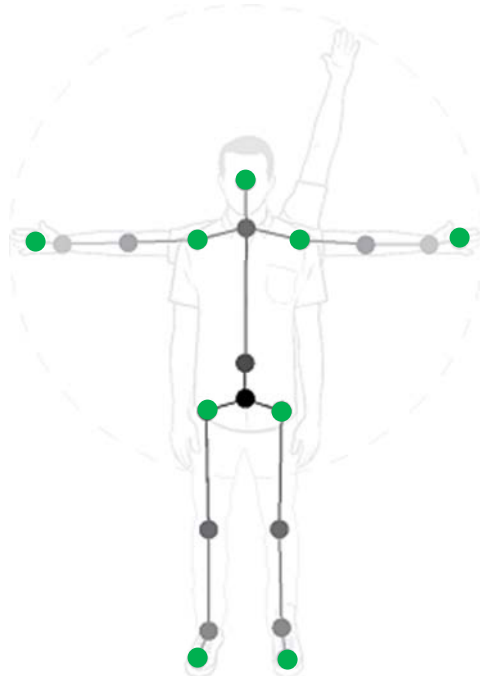
- Detección e interpretación de movimientos del cuerpo
- Robot
- Comunicación con el robot
- Video

6.1. Detección e interpretación de movimientos del cuerpo

La detección de movimientos del cuerpo se realiza mediante el reconocimiento visual de patrones con puntos de referencia en el cuerpo.

Este tipo de reconocimiento se conoce como Modelo Esqueleto (Pavlovic, Sharma, & Huang, 1997) y puede apreciarse en la Figura 16.

Figura 16. Modelo esqueleto



Fuente: Joint Orientation. Kinect Documentation. <http://msdn.microsoft.com/en-us/library/hh973073.aspx>

Este modelo utiliza como referencia el cuerpo para poder reconocer las principales articulaciones en el mismo, conllevando esto a reconocer varios gestos en un plano 2D.

Por ejemplo es posible reconocer gestos como movimiento de brazos, movimientos de pies y hasta saludos.

Las ventajas de este modelo son las siguientes:

- Reconocimiento de la posición (x, y) de cada articulación en todo momento lo que facilita los cálculos para reconocer los gestos
- La calibración de la cámara que visualiza el esqueleto es mínima. Por ejemplo si la persona la cual se está capturando se desplaza o cambia de sitio, no requiere re-calibración el sistema como si puede suceder con otros métodos.

Restricciones:

- Algunas articulaciones no son detectables, por ejemplo, las falanges de las manos.
- Al ser un reconocimiento 2D, la profundidad no es detectable. Esto implica que movimientos en el eje (z) no son reconocibles. Por ejemplo el movimiento de un brazo en profundidad no sería reconocible.

Sin embargo para este modelo, solo es necesario el reconocimiento en un plano 2D y solo se tiene en cuenta los puntos correspondientes a la cabeza, manos y pies.

6.1.1. Movimientos modelados

Al tener en cuenta el modelo esqueleto descrito anteriormente, es posible establecer ciertos movimientos que son reconocidos para ser llevados al movimiento del robot.

Como es posible saber la ubicación en las coordenadas (x, y) de cada articulación del cuerpo en todo momento, es posible trabajar mediante posiciones absolutas y porcentuales los movimientos que se desean reconocer si se establecen límites de movimiento.

Para establecer el porcentaje de movimiento m en un determinado momento se resuelve mediante la siguiente fórmula:

Figura 17. Fórmula: Porcentaje de Movimiento

$$m = \frac{(p - li) * 100}{ls - li}$$

Dónde:

- m : porcentaje de movimiento
- p : posición actual
- li : límite inferior
- ls : límite superior

Por ejemplo se desea saber el porcentaje de movimiento del brazo izquierdo en un momento dado. Para ello es necesario establecer los límites de este movimiento, en este caso se va a asumir que el hombro izquierdo y la cadera izquierda son estos límites. Entonces de acuerdo a la fórmula anterior, se define:

- m : porcentaje de movimiento, valor que se desea conocer
- p : posición actual del brazo izquierdo. Por ejemplo: 0.27
- li : posición actual de la cadera izquierda. Por ejemplo: 0.22
- ls : posición actual del hombro izquierdo. Por ejemplo: 0.30

Resolviendo la fórmula:

$$m = \frac{(0.27 - 0.22) * 100}{0.30 - 0.21} = 55.55\%$$

El porcentaje del movimiento del brazo izquierdo en este momento fue del 55.55%. De esta forma se pueden realizar los cálculos de los movimientos establecidos en este modelo.

6.1.1.1. Algoritmo para detección y realización de movimientos

Teniendo en cuenta la fórmula de la Figura 17, es posible establecer un algoritmo que permita reconocer la variación de movimientos en el tiempo y así poder enviar este reconocimiento al robot.

El algoritmo parte de la premisa de estar siempre calculando el porcentaje de movimiento de cada parte del cuerpo del modelo (cabeza, brazos y pies), almacena este porcentaje en un vector para poder estar comparando cada vez la variación de movimiento y si esta variación supera un *delta* definido se considera como un movimiento válido y se procede con el envío de la señal de movimiento al robot.

En la Figura 18 se muestra un pseudocódigo con el algoritmo descrito en esta sección.

Figura 18. Pseudocódigo: Detección y realización de Movimientos

```
/* Sea PorcentajeActual un vector de 5 posiciones donde se almacenará el
porcentaje actual de cada parte del cuerpo:
    PorcentajeActual [1] -> Cabeza
    PorcentajeActual [2] -> Brazo izquierdo
    PorcentajeActual [3] -> Brazo derecho
    PorcentajeActual [4] -> Pie izquierdo
    PorcentajeActual [5] -> Pie derecho
*/

PorcentajeActual[1 ... 5] = 0 // Se inicializan todas en 0
Delta = 5 // Permite establecer un nivel de tolerancia para la detección
del movimiento. 5 representaría una variación del 5% del movimiento

Mientras verdadero Entonces
    // Este procedimiento sería la fórmula definida en la Figura 17
    m = CalcularPorcentajeMovimiento(ParteDelCuerpo)

    Si m > PosicionActual[ParteDelCuerpo] Entonces
        PorcentajeMovimiento = m - PorcentajeActual [ParteDelCuerpo]

        Si PorcentajeMovimiento >= Delta Entonces
            RealizarMovimiento(ParteDelCuerpo, HaciaArriba,
PorcentajeMovimiento)
        FinSi

    Sino Entonces
        PorcentajeMovimiento = PorcentajeActual [ParteDelCuerpo] - m

        Si PorcentajeMovimiento >= Delta Entonces
            RealizarMovimiento(ParteDelCuerpo, HaciaAbajo,
PorcentajeMovimiento)
        FinSi

    FinSi

    PorcentajeActual[ParteDelCuerpo] = m

FinMientras
```

A continuación se describen los movimientos que son válidos para este modelo. La fórmula planteada con anterioridad es aplicable cada uno:

6.1.1.2. Movimiento de brazos

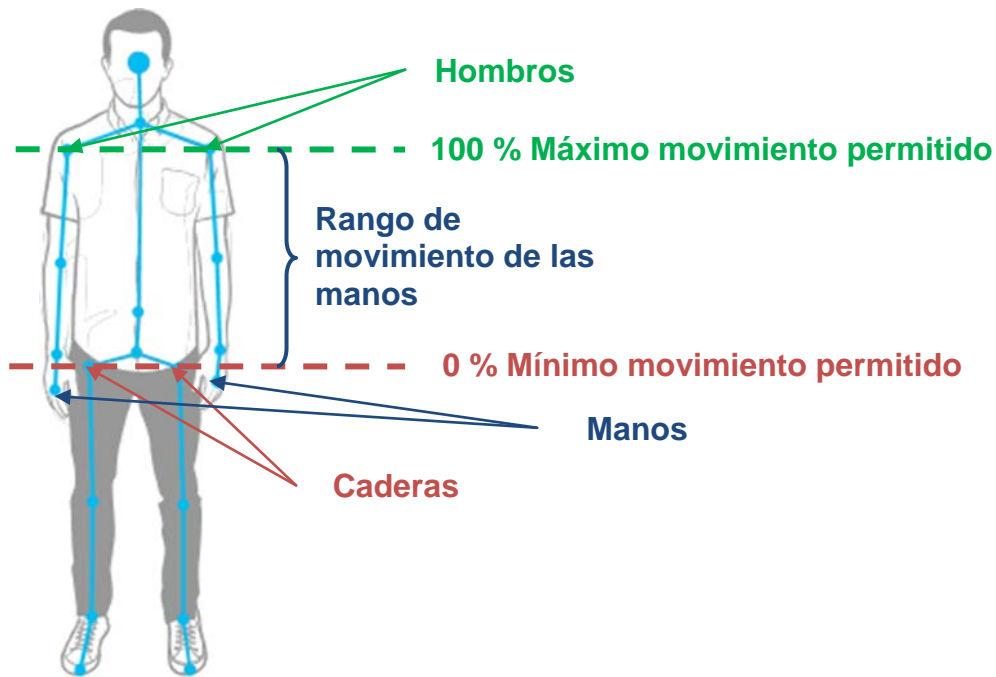
Para el movimiento de brazos se establece como puntos de referencia los hombros, caderas y manos del modelo planteado.

Los hombros son el límite superior de movimiento de las manos, siendo éste el 100% del rango permitido.

Las caderas son el límite inferior de las manos, siendo el 0% del movimiento permitido.

En la Figura 19 se ilustra un gráfico donde es posible visualizar el rango permitido para el movimiento de brazos.

Figura 19. Rango de movimiento de las manos



Si las manos ejercen un movimiento por fuera de este rango, no será reconocido por el modelo.

Este movimiento solo reconoce el desplazamiento de la mano en el eje (y).

6.1.1.3. Movimiento de los pies

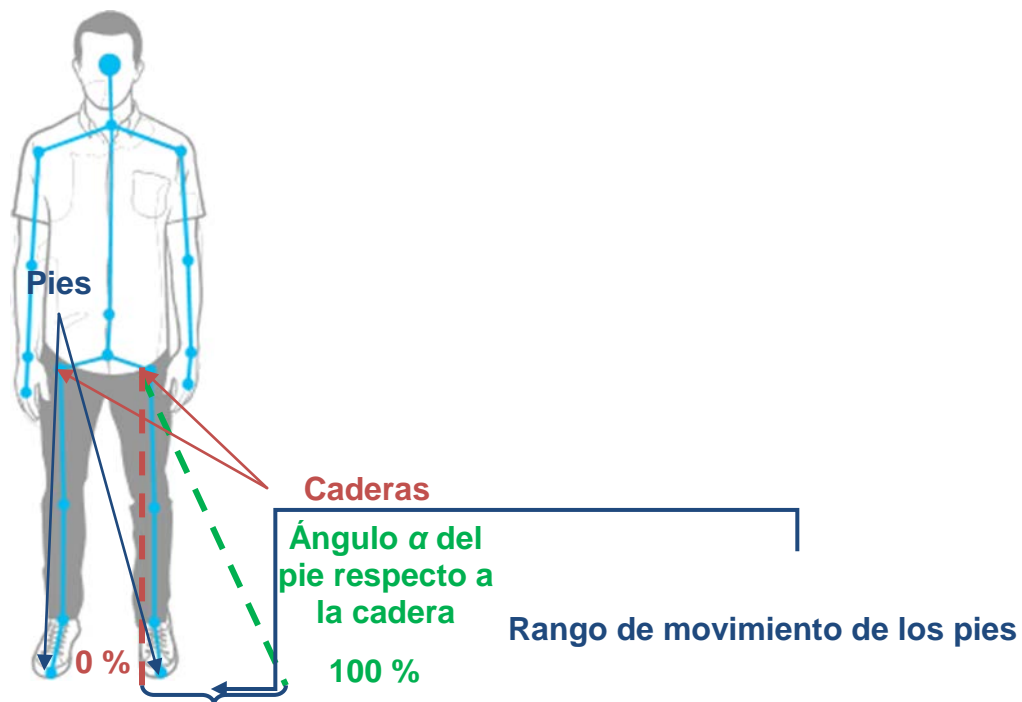
Para el movimiento de los pies se establece como puntos de referencia las caderas y los pies del modelo planteado.

Las caderas son el límite inferior de los pies, siendo el 0% del movimiento permitido.

Se asume como el 100% del movimiento permitido, un ángulo α respecto a las caderas. Se sugiere que el ángulo α sea de 45° pues es el movimiento máximo que se realiza en el desplazamiento de un pie por lo general.

En la Figura 20 se ilustra un gráfico donde es posible visualizar el rango permitido para el movimiento de pies.

Figura 20. Rango de movimiento de los pies



Como es posible saber la posición de las caderas y de los pies en todo momento, es posible calcular el ángulo α entre ambos.

Este movimiento solo reconoce el desplazamiento de los pies en el eje (x) y por lo tanto ejercicios como caminar nos son posibles en primera instancia.

6.1.1.4. Movimiento de la cabeza

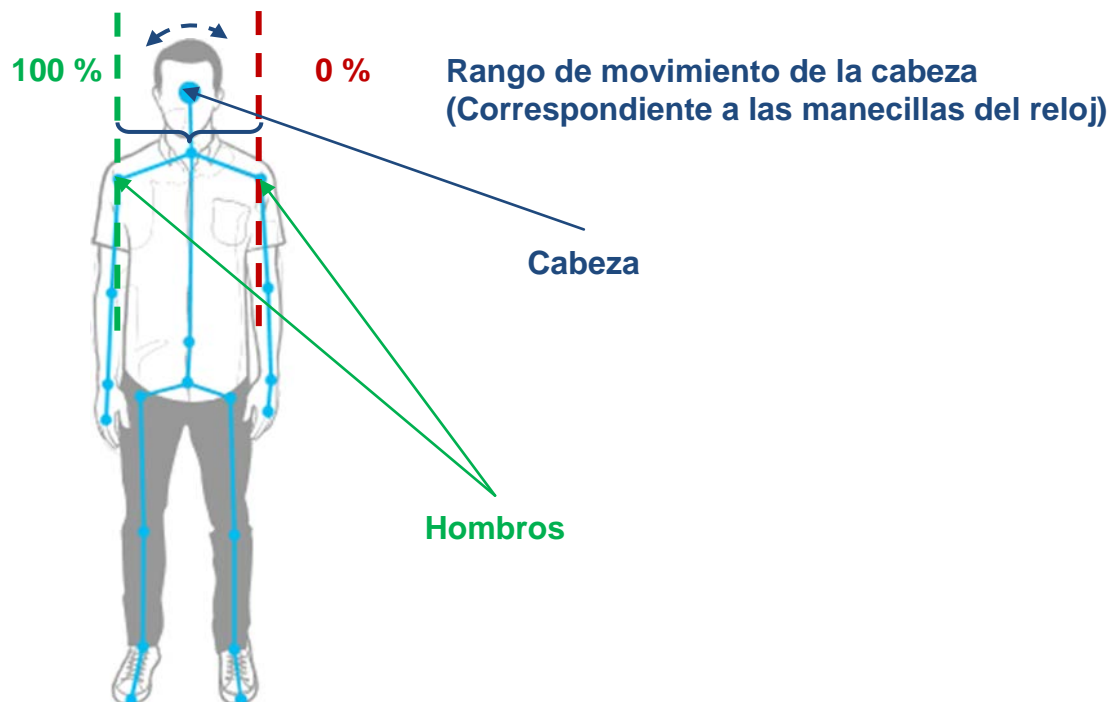
Para el movimiento de la cabeza se establece como puntos de referencia los hombros y la cabeza del modelo planteado.

Los hombros son los límites superior e inferior a la vez de la cabeza siendo esta última perpendicular a los hombros.

Se asume que el 0% del límite del movimiento es el hombro izquierdo y el 100% el hombro derecho correspondiendo a un movimiento de acuerdo a las manecillas del reloj.

En la Figura 21 se ilustra un gráfico donde es posible visualizar el rango permitido para el movimiento de la cabeza.

Figura 21. Rango de movimiento de la cabeza



Este movimiento solo reconoce el desplazamiento de la cabeza en el eje (x).

6.2. Robot

El robot planteado en este sistema corresponde a uno con forma de humanoide el cual debe disponer de las siguientes partes:

- Cabeza
- Brazos
- Pies

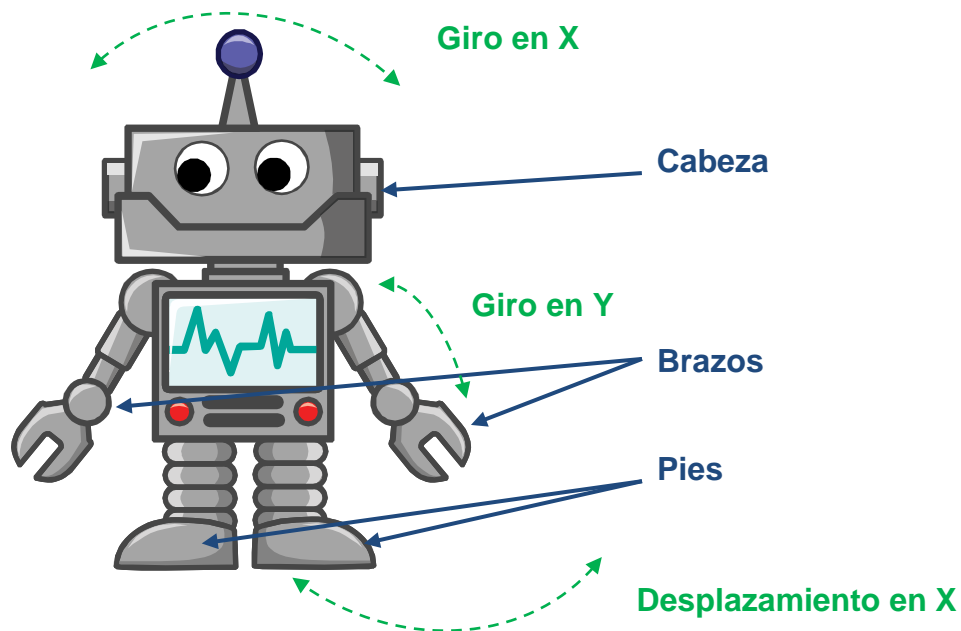
Adicionalmente para realizar las funciones de movimiento y poder interpretar las señales que se envían mediante los movimientos del cuerpo, es necesario que el robot disponga de:

- **Unidad de procesamiento:** Procesador que se encargará de interpretar las señales que le son transmitidas.
- **Puerto de comunicaciones:** Dispositivo que recibe las señales que se envían al reconocer los movimientos del cuerpo.
- **Actuadores:** Motores que permiten que las partes del robot se muevan.

La cabeza debe permitir girar en el eje (x), mientras que los brazos deben permitir el giro en el eje (y). Los pies por su lado deben permitir un desplazamiento en el eje (x).

En la Figura 22 se ilustra un gráfico con las partes del robot y sus movimientos.

Figura 22. Partes del Robot



El robot adicionalmente debe disponer de algún mecanismo de comunicación para poder enviar las órdenes que se han detectado en el modelo descrito.

A continuación se detalla esta comunicación.

6.3. Comunicación con el robot

La comunicación con el Robot se realiza mediante la definición de un protocolo de comunicación utilizando una comunicación vía Sockets, Bluetooth, RFID o algún otro protocolo. Se aconseja que el protocolo de comunicación que se escoja tenga

control de errores (por ejemplo TCP) debido a que el protocolo que se va a definir no los controla.

El protocolo asume y define lo que se anuncia continuación:

- El protocolo permite enviar mensajes de movimiento al Robot para que él los interprete y los realice.
- Se asume que solo se realiza la comunicación mediante un único Robot y un único dispositivo que envía los gestos corporales reconocidos.
- Los mensajes que se envíen no requieren acuse de recibo.
- Los mensajes pueden perderse.
- Los mensajes que se pierdan no se reenvían ni se reordenan.

Los mensajes del protocolo son definidos como una cadena de strings variable y que está formada por tres partes que se separan por comas cada una:

Figura 23. Mensaje de comunicación con el Robot

Parte Cuerpo	Sentido	% Movimiento
--------------	---------	--------------

Mensaje en cadena de strings:

ParteCuerpo,Sentido,Movimiento

Dónde:

Parte Cuerpo

- Brazolzquierdo
- BrazoDerecho
- Pielzquierdo
- PieDerecho
- Cabeza

Sentido

- Arriba
- Abajo
- Izquierda
- Derecha

% Movimiento

- 0% - 100%

A continuación se puede apreciar un ejemplo de esta comunicación:

Figura 24. Ejemplo mensaje de comunicación con el Robot

Brazolzquierdo	Arriba	20%
----------------	--------	-----

Mensaje en cadena de strings:

Brazolzquierdo,Arriba,20

Descripción del movimiento

En este caso el Robot realizará un movimiento del brazo izquierdo con sentido hacia arriba en un 20% de su capacidad motora

Con este formato de mensaje es posible implementar todos los movimientos planteados en este modelo.

6.4. Video

La comunicación de video es utilizada como fuente de retroalimentación para los movimientos del robot. Cuando se realiza un movimiento mediante el modelo planteado no es posible tener una retroalimentación del mismo, por eso es importante disponer de una cámara que muestre en todo momento lo que el robot realiza para así tener un mejor control de los movimientos.

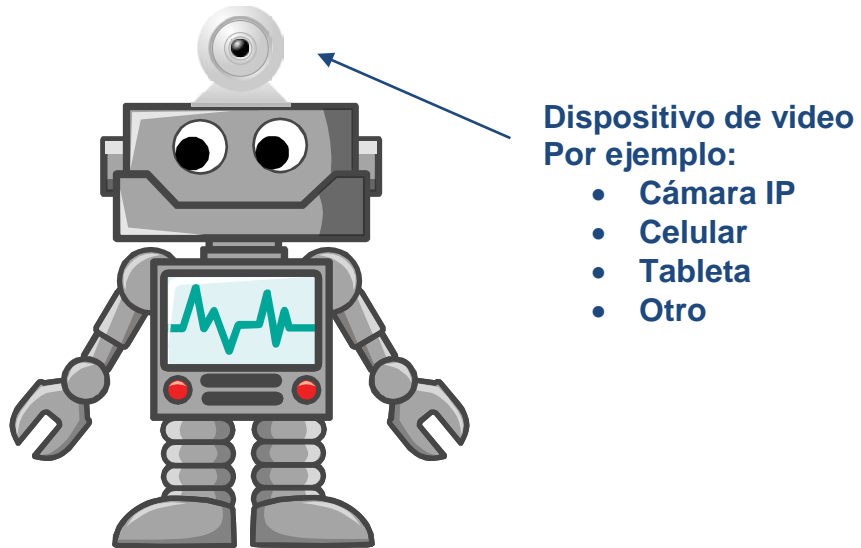
Debido a que hay muchos sistemas en la actualidad que permiten la comunicación por medio de video no es relevante profundizar en este tema.

Algunas opciones son cualquier tipo de comunicación mediante protocolos h264 o sistemas de comunicación de libre uso como son Skype, Gtalk, Viber, etc.

Los dispositivos que se pueden emplear en este apartado pueden ser: cámaras IP, dispositivos móviles como celulares o tabletas, etc.

El robot debe tener un espacio reservado para posicionar este dispositivo. En la Figura 25 se muestra un gráfico con una sugerencia de esta posición.

Figura 25. Robot con dispositivo de video



Estrictamente para el modelo, solo es necesario el video unidireccional, sin embargo, es importante disponer de video bidireccional para poder interactuar con otras personas. De esta forma se podrá visualizar el entorno del robot y también se podrá interactuar con otras personas ejerciendo así el concepto de tele-control y tele-presencia al tiempo.

7. VALIDACIÓN DEL MODELO

El modelo descrito anteriormente establece el uso de brazos, pies y cabeza; sin embargo para efectos de la implementación del mismo solo se usó la parte de los brazos y la de los de pies fue ligeramente modificado para acomodarse al robot escogido para esta implementación.

Para comprender mejor la implementación del modelo es importante separarlo en las siguientes partes de acuerdo al modelo planteado:

Tabla 2. Implementación del modelo

Modelo planteado	Dispositivo / Tecnología usada en la implementación del modelo planteado
Detección e interpretación de movimientos del cuerpo	Microsoft Kinect programado con C#
Robot	Lego Mindstorms
Comunicación con el robot	Android + Sockets + Bluetooth
Comunicación bidireccional por medio de video	Skype

En la Figura 26 se ilustran las partes de la implementación del modelo.

Figura 26. Partes de la implementación del modelo



Según el planteamiento mostrado en el gráfico, se puede observar una persona emisora realizando movimientos. Éstos son interpretados por un dispositivo de captura de movimientos (*Microsoft Kinect*). Este dispositivo está conectado a un PC para que puedan interpretarse y ser enviados estos movimientos mediante sockets (por internet o wifi) a un dispositivo celular (con sistema operativo Android) y éste a su vez envía la señal mediante *bluetooth* al robot humanoide, en nuestro caso un robot construido con *Lego Mindstorms* para que se pueda producir el movimiento de manera física.

Ya que el robot tiene en su equipamiento un celular, con él podremos realizar toda la interacción multimedia ya que éste propicia una cámara frontal para observar el escenario donde se encuentre el robot y a su vez podrá mostrar la persona receptora que está a distancia permitiendo así la tele-presencia.

7.1. Detección e interpretación de movimientos del cuerpo

La adquisición de estos tipos de dispositivos de reconocimiento de gestos por lo general han sido muy costosos y de difícil acceso comercial. Sin embargo desde que *Microsoft* liberó *Kinect* al público se ha visto una proliferación de esta tecnología a nivel investigativo gracias a su bajo costo y sencillez de usar.

7.1.1. Microsoft Kinect

Microsoft Kinect es un dispositivo para controlar videojuegos con el movimiento del cuerpo desarrollado por Microsoft para la videoconsola Xbox 360. Fue lanzado al público en el 2011. *Kinect* permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con ella o con algún periférico adicional. Adicionalmente el dispositivo reconoce gestos, comandos de voz, objetos e imágenes. En la Figura 27 se aprecia una imagen de este dispositivo.

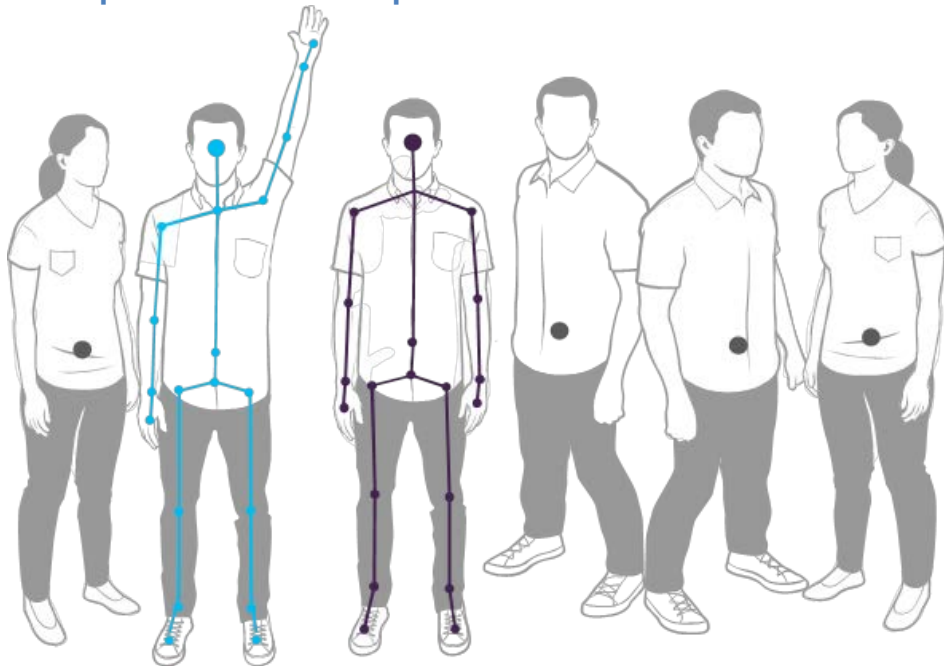
Figura 27. Microsoft Kinect



Fuente: <http://commons.wikimedia.org/wiki/File:Xbox-360-Kinect-Standalone.png>

Kinect ofrece dentro de su kit de desarrollo distintas formas de identificar objetos y cuerpos dentro de las cuales se destaca la opción llamada *Skeletal Tracking* o *Seguimiento del Esqueleto* que es la ideal para el modelo planteado. Este tipo de detección reconoce básicamente 20 puntos en el cuerpo:

Figura 28. Esqueleto reconocido por Kinect



Fuente: <http://msdn.microsoft.com/en-us/library/hh973074.aspx>

Como podemos observar en la Figura 28, los puntos que son reconocidos por *Kinect* no incluyen detalles como por ejemplo los dedos de las manos, las expresiones faciales o el movimiento del cuello.

Estas limitaciones no permiten que se haga reconocimiento de gestos con los dedos de las manos, o expresiones faciales. Por ejemplo no podríamos reconocer con facilidad gestos como un pulgar hacia arriba o expresiones faciales como la felicidad.

Sin embargo para nuestro modelo la forma de detección que propicia *Kinect* es totalmente apropiada.

7.1.2. Visual Studio

Es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) producido por *Microsoft Corporation* y que trabaja generalmente bajo sistemas operativos *Windows*. Permite la creación de aplicaciones de escritorio, móviles y también

web. Soporta diferentes lenguajes de programación entre los que se incluye: C++, C#, Visual Basic y J#.

Kinect al ser un dispositivo desarrollado por *Microsoft*, su API está soportada bajo la plataforma *.Net Framework* la cual permite que se desarrolle en él mediante el uso de *Visual Studio*. Para llevar a cabo este proyecto se usó el lenguaje de programación C# realizando una aplicación de escritorio del tipo *Windows Presentation Foundation (WPF)* y con *Visual Studio 2010 Express Edition*.

Gracias a este aplicativo es posible reconocer los gestos que se generan con el cuerpo y que se envían mediante *sockets* a un dispositivo celular.

7.1.3. Movimientos reconocidos

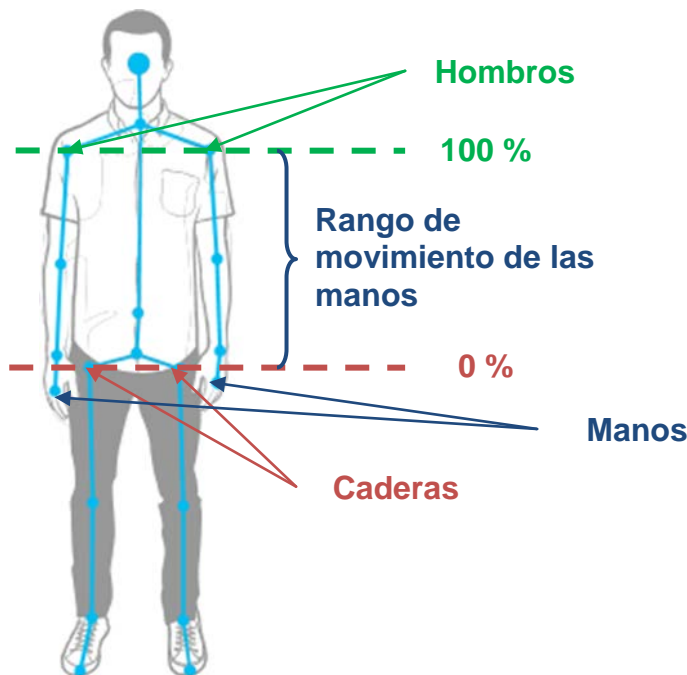
7.1.3.1. Movimiento de brazos

El movimiento de brazos se definió de acuerdo a como lo propone el modelo planteado.

Se establece como puntos de referencia los hombros, caderas y manos.

Los hombros son el límite superior de movimiento de las manos, siendo éste el 100% del rango permitido. Las caderas son el límite inferior de las manos, siendo el 0% del movimiento permitido.

Figura 29. Señal de movimiento de brazos

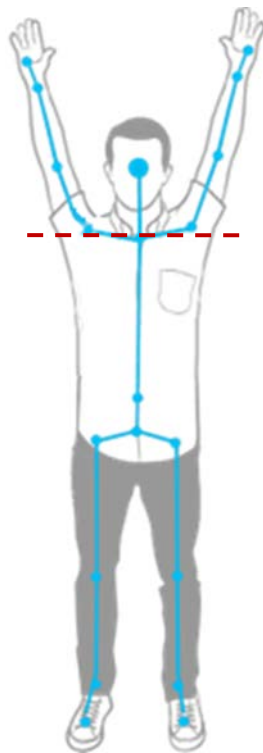


7.1.3.2. Desplazamiento - Movimiento de ruedas

Debido a que el robot escogido para esta validación no presenta piernas sino ruedas, el modelo planteado se modificó con el fin de posibilitar el desplazamiento del robot.

Para ello se estableció la señal de tener ambos brazos flexionados hacia arriba (por encima de los hombros) para que el robot pueda girar sus ruedas y de esta forma ejercer un desplazamiento en el entorno.

Figura 30. Señal de desplazamiento



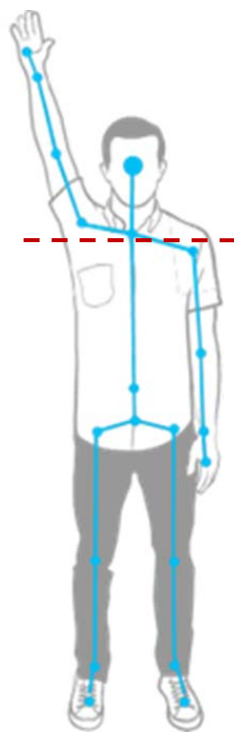
Ambas manos deben estar flexionadas hacia arriba y por encima de la línea imaginaria de los hombros

7.1.3.3. Giros

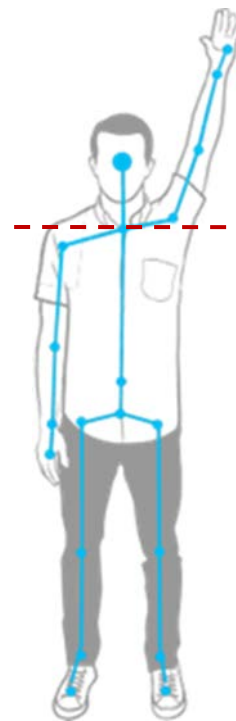
Para poder ejercer un desplazamiento en el entorno en todas las direcciones y debido a que el robot utilizado en la validación posee ruedas, se hace necesario establecer señales de giro.

Para llevar a cabo los giros se estableció la señal de tener el brazo flexionado hacia la izquierda o derecha (según a donde se desee girar) superando la línea imaginaria de los hombros.

Figura 31. Señal de giro



Giro a la derecha

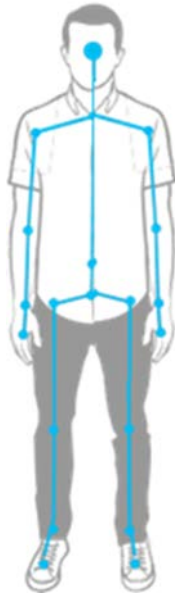


Giro a la izquierda

7.1.3.4. Señal de parada

Para poder detener el robot es necesario establecer una señal de parada. Se definió tener ambas manos flexionadas hacia abajo para llevar a cabo esta tarea.

Figura 32. Señal de parada



Ambas manos deben estar flexionadas hacia abajo

7.2. Robot

Para la implementación del modelo se escogió el Lego Mindstorms debido a su facilidad de uso, economía y buen acople al modelo.

A continuación se muestran detalles de esta tecnología:

7.2.1. Lego Mindstorms

Lego Mindstorms es un conjunto de herramientas que permiten entender y crear sistemas robóticos mediante el uso de piezas y programación de acciones.

Fue comercializado por primera vez en 1998 y ha sido muy usado en el mundo académico.

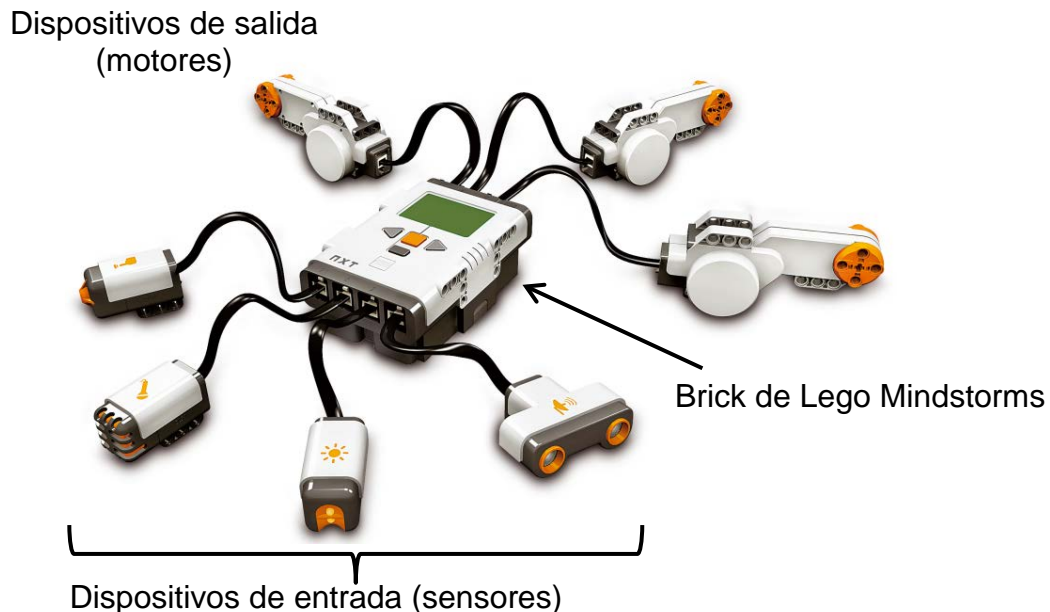
Podemos dividir los módulos que vienen con el kit en dos principales componentes:

- Dispositivos de entrada (sensores): Los sensores permiten interactuar con el medio para obtener datos y reaccionar a los mismos de acuerdo a la programación que se establezca. Algunos sensores con los que viene el kit de fábrica son: sensor infra-rojo, sensor de luz y el sensor de ruido.
- Dispositivos de salida (motores): Permiten establecer el movimiento de motores para llevar a cabo acciones tales como mover brazos o ruedas.

La parte lógica de los kits de **Legó Mindstorms** son gestionados por dispositivos llamados bricks. A éste se conectan todos los sensores y motores para que los pueda procesar y llevar a cabo las acciones que se programen en ellos. Es importante resaltar que cada brick tiene un límite de 3 puertos para dispositivos de salida y 4 puertos para los de entrada.

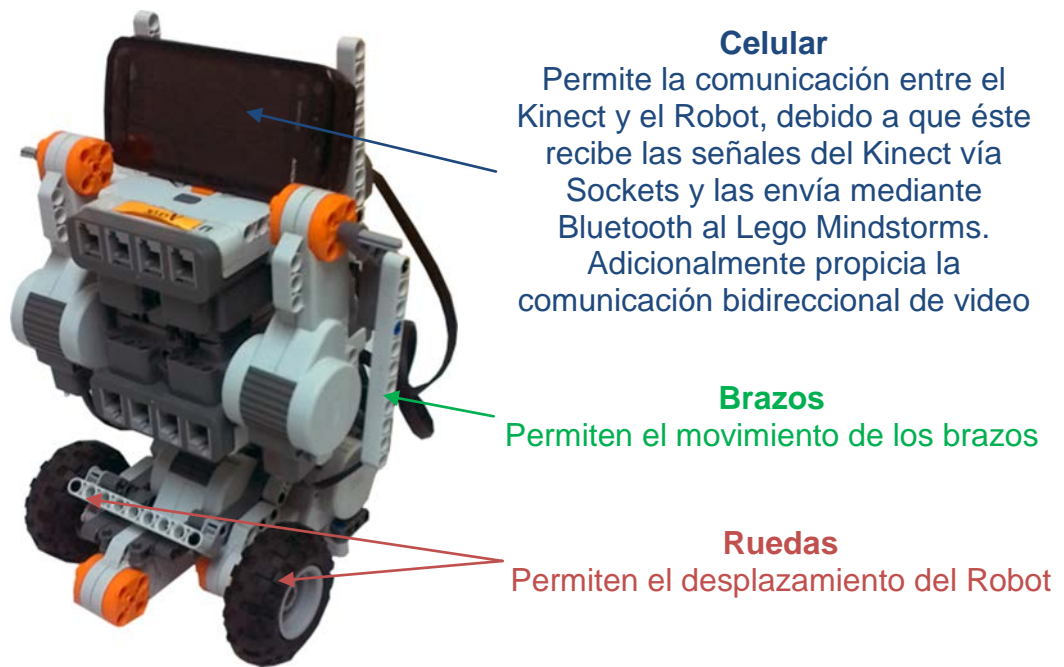
En el presente proyecto es necesario usar 2 bricks debido a que es necesario conectar 4 motores: 2 para las ruedas de desplazamiento y 2 para los brazos del robot.

Figura 33. Partes del Legó Mindstorms



En la Figura 34 se puede apreciar el robot creado mediante el uso de Legó Mindstorms que se utilizó para la validación de este modelo. En él se pueden apreciar las diferentes partes que la componen

Figura 34. Lego Mindstorms utilizado en la validación del modelo



7.3. Comunicación con el Robot

Los bricks de **Lego Mindstorms** vienen con tecnología *bluetooth* incorporada lo que permite gestionar los mismos desde dispositivos que soporten esta tecnología.

Debido a que un dispositivo celular va a estar en la parte superior del robot, puede enviarle señales *bluetooth* para producir los movimientos en el robot.

En el **Anexo B. Comunicación de Lego Mindstorms mediante Bluetooth** se puede encontrar información de cómo se realiza esta comunicación.

7.3.1. Comunicación mediante sockets

Es un concepto de intercambio de flujo de datos que generalmente se da en dispositivos de cómputo con el fin de que se puedan comunicar. Este intercambio permite que haya comunicación en una sola vía o bidireccional.

Para llevar a cabo esta comunicación es necesario implementar protocolos que propicien la apropiada comunicación entre dispositivos y por lo general se llevan a cabo conociendo sus direcciones IP y puertos.

Esta tipo comunicación se utilizó para enviar las señales reconocidas por *Kinect* hasta el dispositivo celular para que luego este mismo las envíe al *Robot*.

7.3.2. Comunicación mediante Bluetooth

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia, eliminando así la necesidad de cables.

Generalmente los dispositivos que hacen uso de Bluetooth son los teléfonos inteligentes (smartphones) ya que por su naturaleza de movilidad son un gran medio para transmitir datos sin cables.

Este tipo de comunicación es usado para la comunicación entre el dispositivo celular y el *Robot* para que este último pueda realizar los movimientos que le son enviados.

7.3.3. Android

Android es un sistema operativo enfocado a dispositivos móviles creado por Google. Es bastante utilizado en el mundo y permite que se desarrolle en él sin costo alguno, además propicia grandes herramientas para llevar a cabo ideas de aplicaciones.

El aplicativo que se desarrolló permite que el dispositivo celular reciba las señales que provienen del *Kinect* mediante sockets y las envía al robot por medio de Bluetooth. El entorno de desarrollo utilizado fue *Eclipse*.

7.4. Comunicación bidireccional por medio de video

La comunicación bidireccional de video permite tener retroalimentación de las acciones del robot y también permite interactuar con otras personas para realizar tareas de tele-presencia.

Esta comunicación se realizó mediante el uso del sistema de videoconferencia Microsoft Skype.

8. EVALUACIÓN DEL MODELO

Los resultados obtenidos fueron en general muy satisfactorios.

Los movimientos que se realizan con mucha rapidez, el Kinect tiende a no reconocerlos bien.

En la Tabla 3 se muestra los resultados obtenidos de acuerdo a 50 movimientos realizados:

Tabla 3. Aciertos del modelo

Movimiento	Aciertos	Fallidos	% de Aciertos
Movimiento Brazo Derecho hacia abajo	45	5	90%
Movimiento Brazo Derecho hacia arriba	46	4	92%
Movimiento Brazo Izquierdo hacia abajo	41	9	82%
Movimiento Brazo Izquierdo hacia arriba	42	8	84%
Giro Mano Derecha	50	0	100%
Giro Mano Izquierda	49	1	98%
Ambas Manos hacia arriba	49	1	98%
Promedio de Aciertos			92%

Como se puede observar, todas las pruebas realizadas superan el **90% de aciertos en la detección del movimiento**, lo que garantiza un muy buen desempeño del modelo.

Los movimientos que presentaron más problemas son los asociados con los movimientos de brazos. Esto puede suceder porque el movimiento debe realizarse con mucho cuidado para que el algoritmo no lo confunda con el movimiento de giro de mano.

Respecto a la latencia que se presenta para el envío de señales al robot se presenta la Tabla 4 que se puede ver a continuación:

Tabla 4. Latencia de envío de movimientos al Robot

Movimiento	Tiempo (ms)
1	1021
2	542
3	152
4	158
5	170
6	209
7	140
8	190
9	220
10	203
Tiempo tomado bajo una red WiFi	
Promedio	301

Como se puede apreciar en la Tabla 4, al principio el envío de señales al robot presenta una latencia de poco más de un segundo, pero luego esta latencia tiende a la baja. Como promedio se tiene **301 milisegundos** lo que corresponde a un buen tiempo de respuesta.

Es importante tener en cuenta que el tiempo tomado en esta tabla corresponde desde cuando se realiza el movimiento hasta que llega al robot, es decir, considera el proceso de detección mediante *Kinect*, el envío de la señal mediante sockets al dispositivo móvil y finalmente su entrega al robot mediante *Bluetooth*.

9. CONCLUSIONES

El proyecto desarrollado permitió diseñar y validar un sistema de gestión de un robot con forma de humanoide mediante el reconocimiento de gestos corporales. Para la caracterización del sistema se partió de un modelo ideal donde se establecía un robot y un sistema de reconocimiento gestual idóneo para luego presentar el modelo propuesto y realizar comparaciones entre ambos.

El modelo propuesto fue validado con la ayuda de dispositivos como *Microsoft Kinect*, *Android* y *Lego Mindstorms*. De acuerdo a esta validación fue posible establecer que el desempeño del mismo fue satisfactorio pues en promedio más del **90% de los movimientos fueron reconocidos con éxito** y tan solo hay una **latencia promedio de 301 milisegundos**. Adicionalmente el prototipo permitió la interacción con otras personas en el mismo entorno del robot gracias a su cámara bidireccional posibilitando así establecer tele-presencia de manera exitosa.

Las limitaciones en la validación del modelo más que todo se presentaron con el uso del *Lego Mindstorms* como dispositivo actuador debido a los pocos grados de libertad que ofrece el mismo.

La validación del modelo permitió identificar las siguientes aplicaciones:

- **Tele-presencia:** Al permitir el desplazamiento del robot de manera remota y disponer el mismo de una cámara, la tele-presencia es totalmente aplicable para poder interactuar con otras personas en un mismo sitio.
- **Ayuda a discapacitados:** El robot podría ayudar a gente discapacitada, especialmente a personas con limitantes de movilidad en sus miembros inferiores. Por ejemplo se podría usar el sistema con las manos para poder desplazarse remotamente en algún sitio con la ayuda del robot.
- **Asistencia en labores sencillas:** El robot controlado a distancia podría permitir realizar labores sencillas como presionar botones o mover objetos livianos remotamente.

10. REFERENCIAS

- Baker, J. (1975). The DRAGON system--An overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24–29.
- Card, S. K., Moran, T. P., & Newell, A. (Julio de 1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM* 23 (7): 396–410.
- DigInfoTV. (2012). *Yaskawa Electric's next-generation robot operated using Kinect: SmartPal VII*. Recuperado el 22 de Abril de 2013, de Yaskawa Electric's next-generation robot operated using Kinect: SmartPal VII: <http://www.diginfo.tv/v/11-0248-r-en.php>
- Hall, J. C. (2011). *How to Do Gesture Recognition With Kinect Using Hidden Markov Models (HMMs)*. Recuperado el 30 de Abril de 2013, de Creative Distraction: <http://www.creativedistraction.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/>
- Honda. (2013). *History: Honda's Asimo Website*. Recuperado el 22 de Abril de 2013, de Honda's Asimo Website: <http://asimo.honda.com/asimo-history/>
- ITU-T Techonology. (February de 2012). Obtenido de TelePresence: High performance video-conferencing ITU-T Briefing Reports (November 2007): http://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000020002PDFE.pdf
- Joystiq. (February de 2012). Obtenido de Xbox 360 now at 66 million worldwide, Kinect at 18 million: <http://www.joystiq.com/2012/01/09/xbox-360-now-at-66-million-worldwide-kinect-at-18-million/>
- Pavlovic, V., Sharma, R., & Huang, T. S. (July de 1997). Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 19, NO. .
- Polycom. (February de 2012). Obtenido de Polycom® RealPresence™ Desktop - Business-grade desktop visual collaboration solutions: http://www.polycom.com/products/telepresence_video/telepresence_solutions/personal_telepresence/index.html
- Radrigan R, M. (2005). *Metodología de la Investigación*.

The LEGO Group. (2006). *Lego Mindstorms NXT Direct Commands*.

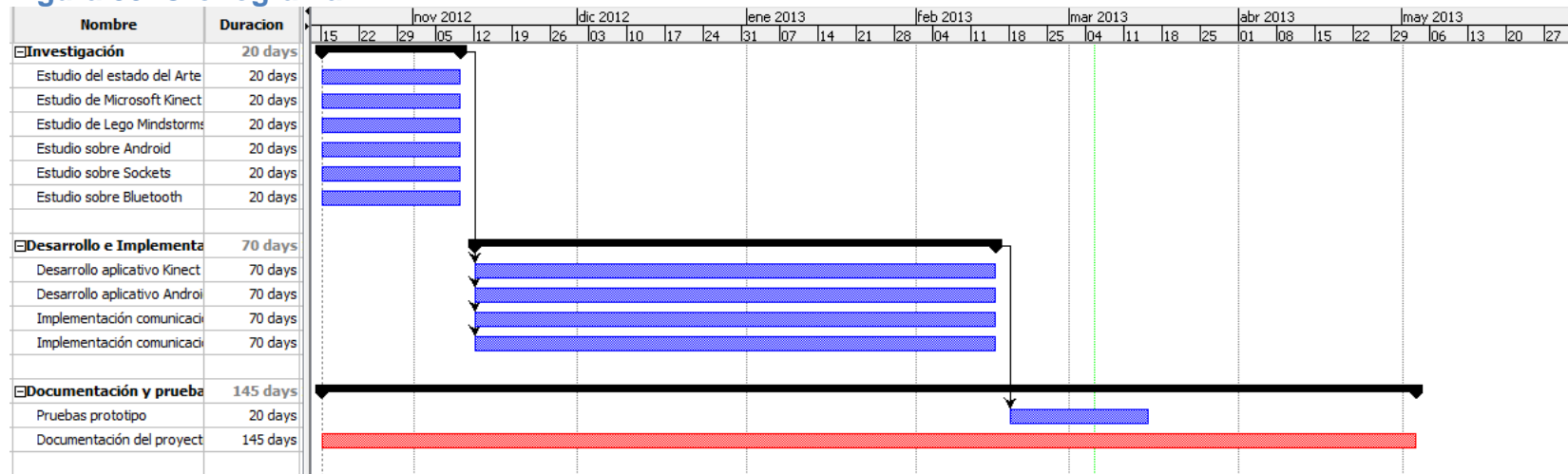
The NXT Step. (2012). *NXTAVATAR a Tele-presence robot*. Recuperado el 22 de Abril de 2013, de The NXT Step: <http://thenxtstep.blogspot.com/2012/01/nxtavatar-is-tele-presence-robot-which.html>

Wikitel. (February de 2012). Obtenido de Servicios de Telepresencia: http://wikitel.info/wiki/Servicios_de_Telepresencia#cite_note-0

YouTube LLC. (February de 2012). Obtenido de Gesture-based Surgical Manipulation of a da Vinci Robot using a Kinect: http://www.youtube.com/watch?feature=player_detailpage&v=YsY_A0kLh-g

Anexo A. Cronograma

Figura 35. Cronograma



Actividades:

- **Investigación (Octubre 15 – Noviembre 9, 2012)**
 - Estudio del estado del Arte
 - Estudio de Microsoft Kinect
 - Estudio de Lego Mindstorms
 - Estudio sobre Android
 - Estudio sobre Sockets
 - Estudio sobre Bluetooth
- **Desarrollo e Implementación (Noviembre 12, 2012 – Febrero 15, 2013)**
 - Desarrollo aplicativo Kinect y Android
 - Implementación comunicación Bluetooth y Sockets
- **Documentación y Pruebas (Octubre 15, 2012 – Mayo 3, 2013)**
 - Pruebas prototipo (Febrero 18 – Marzo 15, 2013)
 - Documentación del proyecto (Octubre 15, 2012 – Mayo 3, 2013)

Anexo B. Comunicación de Lego Mindstorms mediante Bluetooth

Lego Mindstorms soporta diferentes órdenes mediante Bluetooth. Es posible iniciar programas instalados en el brick, reproducir sonidos o incluso obtener datos de los sensores. Sin embargo en nuestro caso lo que nos interesa es maniobrar los diferentes motores que nos provee el kit para poder llevar a cabo los movimientos de los brazos y de las ruedas para podernos desplazar e interactuar con el entorno.

Para ello es necesario entender el mecanismo de envío de mensajes del tipo SetOutputState que es como Lego denominó en su documentación de Bluetooth a los mensajes para poder maniobrar los puertos de motores de los bricks.

A continuación se muestra un diagrama con los bytes necesarios a enviar para llevar a cabo la tarea:

Figura 36. Mensaje SetOutputState

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10-13
0x0C	0x00	0x80	Command-set motor output state.	Output Port	Power set point	Mode	Regulation mode	Turn Ratio	Run State	Tacho Limit

Dónde:

- **Byte 0 y 1:** Siempre deberán estar en los valores 0x0C y 0x00 pues así se inicia el mensaje
- **Byte 2:** Si se desea obtener respuesta del envío del mensaje se debe usar 0x00, sino se utiliza 0x80.
- **Output port:** Puede variar de 0 a 2 decimal. Indica el puerto al cual se le va enviar la señal. Cada brick de Lego Mindstorms solo tiene 3 puertos de salida para los motores, de aquí proviene el límite de 0 a 2.
- **Power set point:** Valores entre -100 a 100 decimal. Permite indicar la potencia con la cual el motor estará actuando.
- **Mode:** Enciende o apaga el motor especificado. 0x01 encendido. 0x02 apagado
- **Regulation mode:** Permite la sincronización de los motores.
- **Turn Ratio:** Valores en -100 a 100 sbyte. Permite establecer la velocidad de giro de los motores.
- **RunState:** Valores en -100 a 100 ubyte.

- **TachoLimit:** Permite indicar el límite de movimiento del motor. Se coloca 0 cuando se desea que el motor funcione indefinidamente.

A continuación se puede apreciar un ejemplo de un mensaje:

Ejemplo Mensaje Bluetooth

Para mover las ruedas del robot hacia adelante, el mensaje a enviar sería:

Figura 37. Mensaje Bluetooth

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10-13
0x0C	0x00	0x80	0x04	0x01	0x32	0x05	0x02	0x00	0x20	0x00 0x00 0x00 0x00

Si se desea mayor detalle en este protocolo, se puede consultar el documento **Legó Mindstorms NXT Bluetooth Direct Commands** (The LEGO Group, 2006).