

# XII CONGRESO DE MEXICANO DE ROBOTICA



## COMRob 2010

Mazatlán, México  
del 3 al 6 de noviembre

Asociación Mexicana de Robótica A.C.  
<http://www.amrob.org>

# MEMORIAS



## Simulación en Java de un Manipulador Hidráulico

Vargas E., Gorrostieta E., Pedraza C. y Sotomayor A.

**Resumen**— Este trabajo presenta la manera en cómo se diseñó e implementó un programa computacional capaz de simular los movimientos en un manipulador hidráulico de dos grados de libertad. Se describe la metodología de diseño del software que utilizaron estudiantes de informática, de grupos diferentes, para construir los elementos del sistema utilizando el lenguaje de programación java. Se muestra el modelo cinemático directo del manipulador, los elementos de instrumentación del sistema hidráulico, y los resultados de visualización gráfica del manipulador bajo una plataforma de navegación en internet. El trabajo finaliza mostrando una proyección de los sistemas digitales de monitoreo y control manual de procesos industriales tomando como base las tecnologías afines con el lenguaje de programación Java.

Keywords: Java, software, hydraulic manipulator.

**Abstract**— This paper shows the way as a computational program was designed and implemented to simulate the movements of a hydraulic manipulator of two degrees of freedom. The methodology used for students of informatics to design and implement the software is shown. The direct kinematics model of the manipulator, the results and the visualization of the manipulator on internet are presented. The paper ends by showing a projection about the digital systems to monitor and control manual of industrial process using compatible technologies with java language program.

Palabras Clave: Java, software, manipulador hidráulico.

### I. INTRODUCCIÓN.

El uso de software para la enseñanza de la robótica es un medio que permite desarrollar diversos niveles de conocimiento de la robótica. Esta labor no es nueva, se han utilizado diversos programas comerciales y de uso libre para cubrir diferentes áreas del conocimiento en las últimas décadas [1], [2] y [3]. La programación de robots, el control, la modelación y la simulación, así como el diseño electrónico y mecánico son algunas áreas del conocimiento que hacen uso de software para desarrollar un conocimiento sistemático de la robótica. De esta forma, es importante

Vargas E., Profesor e investigador en la Facultad de Informática de la Universidad Autónoma de Querétaro. México. (Tel. 442-192 12 00; fax: 442- 192 1332; e-mail: emilio@mecatronica.net).

Gorrostieta E., Profesor e investigador en la Facultad de Informática de la Universidad Autónoma de Querétaro. México. (Tel. 442-192 12 00; fax: 442- 192 1332; e-mail: efengorrostieta@gmail.com).

Pedraza C., Profesor e investigador en la Facultad de Informática de la Universidad Autónoma de Querétaro. México. (Tel. 442-192 12 00; fax: 442- 192 1332; e-mail: carlos.pedraza@uaq.mx).

Sotomayor A., Profesor e investigador en la Facultad de Informática de la Universidad Autónoma de Querétaro. México. (Tel. 442-192 12 00; fax: 442- 192 1332; e-mail: artemiosotomayor@gmail.com).

continuar con esta visión que permita adecuar los cambios tecnológicos de software para cubrir un perfil profesional que se mantenga vigente con respecto a los requerimientos del mercado. En este sentido, se ha seleccionado el lenguaje de programación Java para ser utilizado como una herramienta que permita el desarrollo de un conocimiento, tanto de programación como de robótica en los estudiantes. Por otra parte, existen una gran variedad de fenómenos vinculados con el movimiento de un robot de tipo hidráulico. Por una parte, están los propios elementos del sistema hidráulico como son la bomba, las válvulas de control y acción. Y por otra parte, se encuentra el movimiento cinemático y dinámico, así como su control. En la medida que los modelos y herramientas respondan de manera segura con la realidad de dichos fenómenos, se lograrán mejores condiciones de diseño y uso de manipuladores hidráulicos, mejorando así su comprensión y uso.

En lo referente a las herramientas de simulación, hay una amplia variedad de opciones que permiten simular modelos matemáticos complejos, como se presentan en el comportamiento de un robot hidráulico. En este trabajo se muestra el diseño e implantación de software utilizando el lenguaje de programación java, de acuerdo a la arquitectura software que se muestra en la figura 1.

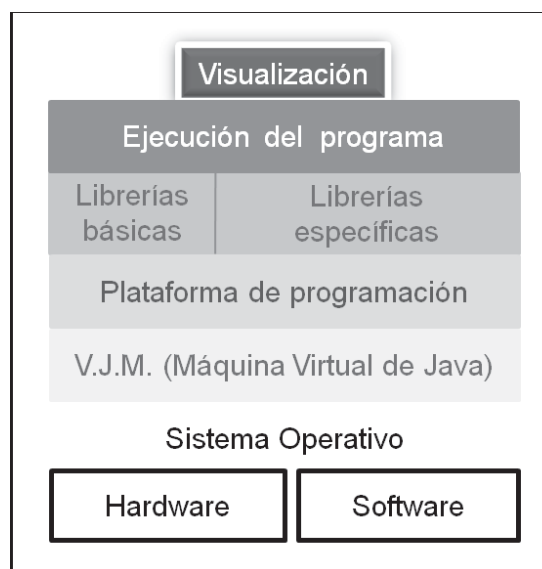


Figura 1. Arquitectura del funcionamiento de java.

Se seleccionó java porque este lenguaje permite crear programas que se pueden visualizar en diversos sistemas operativos, sus datos funcionan como objetos, amplia capacidad de interconexión TCP/IP, su gran aceptación en la comunidad internacional de internet, el lenguaje y

plataformas de programación se distribuyen de forma gratuita, su seguridad multiusuario, y también por la posibilidad de desarrollar aplicaciones que se pueden visualizar en los navegadores más conocidos, así como en dispositivos móviles [4] y [5].

## II. METODOLOGÍA.

Uno de los aspectos más relevantes vinculados con el desarrollo de un proyecto de software, es la determinación de tareas y responsabilidades asociadas con las actividades que realizan los integrantes de un equipo de desarrollo de software. En este sentido, la metodología de trabajo juega un papel fundamental para organizar y administrar de forma adecuada el proyecto de software. La Figura 2 muestra un diagrama conceptual de la metodología que se utilizó para realizar los diferentes programas java que se muestran en éste trabajo. Es importante mencionar que ésta metodología no es la primera vez que se utiliza, ya que forma parte de un modelo de trabajo más amplio de los autores asociado con el diseño y desarrollo de máquinas mecatrónicas [6] y [7].

El primer elemento a considerar en dicha metodología es una definición clara y precisa de las necesidades que tiene el cliente, lo que en la práctica nos permite definir las especificaciones técnicas que debe de cubrir el software a fin de responder satisfactoriamente dichas necesidades. Es fundamental determinar el objetivo general del proyecto, con el propósito de tener una guía que nos permita evitar desvíos que pudieran distraer recursos en tareas poco claras o innecesarias. Así mismo, la determinación y consecución de objetivos específicos nos permiten disgregar los problemas del proyecto, y una administración adecuada de personas y recursos. Una vez realizadas estas actividades, la metodología nos lleva a diseñar dos aspectos fundamentales del programa computacional: a) El modelo matemático, y b) La interfaz de usuario. Ambas actividades están muy relacionadas, y requieren de una atención especial a fin de buscar soluciones que permitan una implementación de forma eficiente. Es común revisar el diseño de la interfaz, incluso con la participación del cliente, valorando su sencillez y facilidad de uso. En esta etapa se logran "prototipos" que muestran una idea inicial de cómo funcionará el programa, y su interacción con el usuario. Aprobadas la etapa de diseño, la siguiente etapa importante es la programación en código de los prototipos del modelo, y el prototipo de interfaz. Por lo regular, la programación y las pruebas de código llevan a una depuración del programa de forma paralela e integral. Esto es, se comparte información y resultados de los códigos de programa con los integrantes del grupo de desarrollo, a fin de que se obtenga un programa completo que responda al diseño deseado.

Una vez que se han superado las pruebas del código en un mismo programa, se considera que el software está funcionando conforme al diseño establecido. Es recomendable comparar el uso de dicho programa con el objetivo del proyecto de software y las necesidades del cliente, con el propósito de asegurar que se logran las expectativas de uso. Bajo estas consideraciones, y en concordancia con el proyecto académico desarrollado por alumnos de informática, se diseñaron e implementaron

funciones diversas que muestran un comportamiento de cada elemento del robot, esto con el propósito de integrar dichos elementos en una misma plataforma de ejecución.

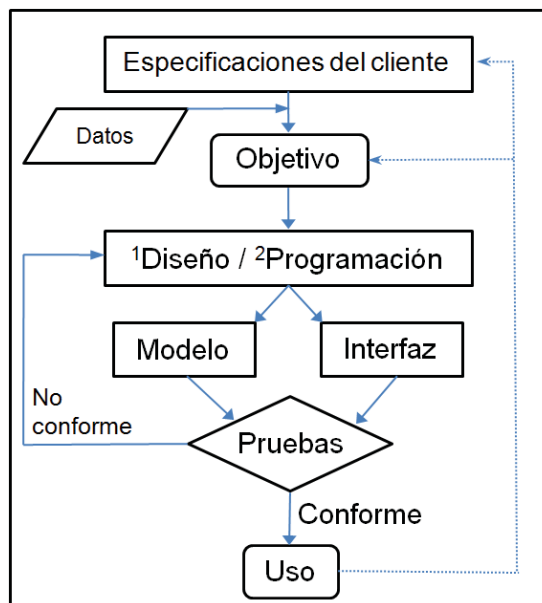


Figura 2. Diagrama de la metodología de diseño.

## III. ALCANCES.

Debido a que existe una amplia variedad de manipuladores, válvulas, bombas, actuadores y sensores, resulta fuera del alcance de este trabajo mostrar una herramienta general que permita simular un manipulador de dos grados de libertad de tipo genérico. Lejos de suponer que el software servirá para visualizar y controlar el estado del manipulador, lo que en la práctica resulta muy conveniente y en esta etapa no se pretende aún lograr, el presente trabajo se orientó en desarrollar una serie de programas que permiten conocer de forma visual la condición o estado específico de un elemento del manipulador hidráulico. De esta forma se pretende confirmar la funcionalidad de un sistema basado en programas java que permitan monitorear el cambio de estado del manipulador. Así mismo, las condiciones reales que se presentan durante la acción del manipulador son simuladas en esta primera etapa de desarrollo a fin de facilitar la implantación y prueba de código de dichos elementos. No obstante, es posible conocer valores como presión, temperatura, velocidad o fuerza mediante sensores adecuados a cada variable, acondicionando su señal para que diversos dispositivos permitan llevar el registro de sus valores a una computadora. Por ejemplo, el módulo SNAP que presenta la compañía sueca Imsys es una plataforma completa tipo hardware y software que incluye múltiples interfaces estándar de tipo industrial para dispositivos locales, así como los protocolos TCP/IP y Ethernet 10/100 Mbit/s para acceso remoto [8].



#### IV. CASO 1. MEDIDOR ANALÓGICO.

Una forma útil y práctica de visualizar la condición de una variable en los procesos hidráulicos industriales es a través de medidores de tipo analógico. Este tipo de indicadores permiten al usuario una interpretación instantánea del valor de la variable, así como una comparación entre el valor actual y el valor máximo medible.



Figura 3. Medidor analógico utilizado en procesos hidráulicos industriales.

El diseño de un objeto medidor como el que se presenta en la Figura 3 en java requiere una definición vinculada con sus parámetros gráficos (tamaño, posición, color, nombre, valor de la variable) a fin de representar dicho objeto de forma dinámica. Incluso se pueden definir colores específicos que muestren cierto grado de peligrosidad, así como agregar sonidos específicos que indiquen al usuario alguna condición de alerta en la variable. El sistema de monitoreo basado en programación java permite la captura de los valores de la variable vinculados con el tiempo, y registrar ambos datos en un archivo. Esto último resulta particularmente útil cuando se desea evaluar dicha variable y su comportamiento en el proceso, o bien hacer un análisis de tipo estadístico

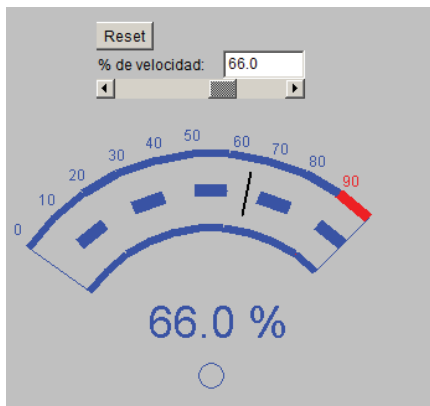


Figura 4. Medidor analógico construido en java.

Las principales variables definidas en la construcción del medidor analógico son las siguientes:

- xg, yg - Posición de la base del medidor en el plano XY (píxeles).
- valor\_p - Representa el valor en porcentaje de la variable (número real).
- valor\_t - Variable de tipo texto que muestra el valor de la variable
- theta, Posición angular en grados de la aguja indicadora.

div - Representa el número de divisiones del indicador (10 en el ejemplo).

#### V. CASO 2. INTERRUPTOR DE USO GENERAL.

Otro elemento ampliamente utilizado en los sistemas hidráulicos industriales es el interruptor. Existe una gran variedad de tipos, tamaños y diseños. Este tipo de elementos permite de forma manual encender o apagar un subsistema del proceso hidráulico, y en general activar o desactivar algún dispositivo.



Figura 5. Interruptor de tipo industrial.

La representación del interruptor por medio de software se puede realizar considerando una variedad de formas. En el caso que nos ocupa las variables relevantes definidas están asociadas al tamaño del interruptor, su posición, el texto y la acción a realizar.

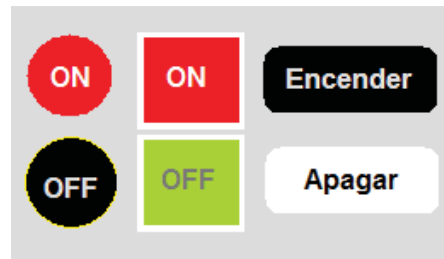


Figura 6. Interruptores construidos en java.

El diseño del interruptor y su programación se efectuó considerando que el texto se presenta de forma horizontal, así como su orientación. Sin embargo, es posible modificar sin dificultad el código para visualizar el interruptor y su texto con una cierta orientación angular si así se requiere.

#### VI. CASO 3. ELECTROVÁLVULA UNIDIRECCIONAL.

En lo que respecta a la electroválvula unidireccional de tipo industrial, ésta se diseñó considerando dos diferencias: a) Construcción gráfica y b) Imagen gráfica. En el primer diseño se construyó la válvula con base en líneas y figuras regulares, de forma similar a los objetos anteriores que se han presentado en este trabajo. En el segundo diseño se recupera un archivo tipo gif de la imagen de la válvula, previamente guardada en la computadora, y se coloca como una imagen en un lugar específico de la pantalla. Esta manipulación lo permite el método *drawImage* de java.

Los parámetros de éste método son:

- img*, es la imagen que se desea mostrar.
- x,y* son las coordenadas para posicionar la esquina superior izquierda en la pantalla de la computadora.
- width y height*, dimensiones del rectángulo en donde se colocará la imagen.
- observer*, objeto que se ha modificar tan pronto como la imagen se haya situado en su posición.



Figura 7. Electroválvula de tipo industrial.

Para el diseño de la electroválvula y su programación se efectuó considerando, además de los parámetros de visualización mencionados en los anteriores elementos, un estado lógico de un bit, que representa lo siguiente. Si el bit es 1, (ON) la válvula está energizada, la válvula se considera abierta. El fluido pasa por el cuerpo de la válvula. Si el bit es 0, (OFF) la válvula se encuentra des-energizada, la válvula se encuentra cerrada. El fluido no pasa por el cuerpo de la válvula. No hay flujo.

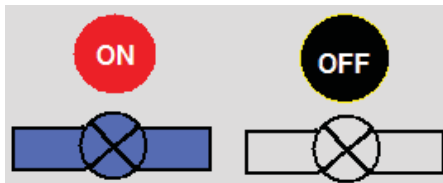


Figura 8. Estados de la electroválvula construida en java.

Es importante mencionar que no se modeló el flujo de corriente eléctrica en la electroválvula, ni mucho menos la fuerza electromagnética que mueve el vástago del cuerpo de la válvula, ya que para fines de monitoreo y control en los procesos, únicamente se requiere conocer el estado de éste tipo de elementos, y analizar la consecuencia de dicho estado en el proceso. Como parámetros de modelado, se consideró el gasto volumétrico del fluido que pasa por la válvula, el diámetro interno de entrada y salida de la válvula, así como su estado lógico.

### VII. MANIPULADOR HIDRÁULICO.

El manipulador hidráulico de dos grados de libertad que se ha considerado como ejemplo para probar la consistencia de la información bajo una plataforma de internet considera un movimiento rotacional en la base del manipulador, y un

movimiento lineal sobre el primer eslabón del manipulador. La Figura 9 muestra la representación gráfica del manipulador.

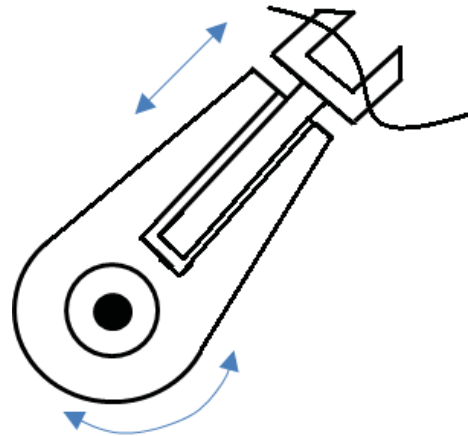


Figura 9. Modelo 2D del manipulador hidráulico.

El modelo cinemático directo del manipulador, considera la posición del órgano terminal del robot modelada como una función que depende de la posición de cada actuador, y que se encuentra en concordancia con los elementos hidráulicos del manipulador accionados de forma manual, esto a fin de controlar la posición del manipulador mediante la presión (fuerza) y el flujo del fluido hidráulico (velocidad), se cuantifica el tiempo de apertura y cierre de las electroválvulas hidráulicas.

A fin de simular el movimiento del manipulador, se consideró que la masa de los eslabones es constante, el fluido hidráulico es incompresible, el movimiento se efectúa a temperatura constante, y que el flujo y la presión no presenta oscilaciones en sus valores nominales, principalmente.

De esta forma, una vez definidos objetos principales que componen al manipulador, se construyó el manipulador integrando dichos objetos.

Dos electroválvulas 5/3 son utilizadas para mover el manipulador, mismas que fueron modeladas mediante condiciones lógicas de botones de control, se utilizan dos botones por cada electroválvula. La Figura 10 muestra la representación del accionamiento hidráulico básico para cada actuador.

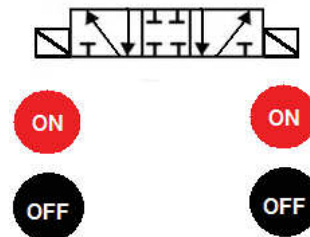


Figura 10. Representación del accionamiento hidráulico.

El modelo de la posición del cilindro hidráulico asociado al flujo constante de fluido generado por una bomba de

engranes, nos permite describir la posición (x) del cilindro como:

$$x = \int_{t=0}^{t=tf} \frac{\Omega}{A} dt \quad \dots(1)$$

Donde  $t=0$  es el tiempo de inicio del accionamiento en el cilindro,  $tf$  es el tiempo en que se detiene el accionamiento en el cilindro,  $\Omega$  representa el gasto del fluido hidráulico y A es el área transversal del émbolo del cilindro. Por lo que al integrar la ecuación 1, la posición del cilindro resulta ser:

$$x = \frac{\Omega tf}{A} + C \quad \dots(2)$$

Siendo C la constante de integración que representa una posición inicial al momento de iniciar el accionamiento del cilindro. Con respecto al actuador rotativo hidráulico, existen diferentes tipos y modelos que permiten generar un movimiento giratorio. Uno de los diseños más simples se basa en el accionamiento de un engrane sobre una cremallera, de ésta forma la velocidad tangencial del engrane es igualada a la velocidad lineal de la cremallera. Por lo que la posición angular ( $\theta$ ) del actuador rotativo se basa en el modelo de posición del actuador lineal. La ecuación 3 define la posición angular:

$$\theta = \frac{\Omega tf}{A r} + D \quad \dots(3)$$

Siendo r el radio tangencial efectivo entre el engrane y la cremallera, y D una constante que se relaciona con la posición angular inicial al momento de accionar el actuador rotativo.

### VIII. MODELO CINEMÁTICO DIRECTO.

El modelo cinemático directo determina la posición del órgano terminal del manipulador con respecto a un sistema de coordenadas de referencia. Para el caso que nos ocupa, el ángulo  $\theta$  (actuador angular hidráulico) y el desplazamiento b (actuador lineal hidráulico) se consideran conocidos para determinar la posición (x, y) del órgano terminal del manipulador, ambos parámetros se consideran funciones dependientes del tiempo. La Figura 11 muestra la disposición del manipulador y sus variables cinemáticas asociadas con su posición. La posición (x, y) del órgano terminal del manipulador se determina mediante la siguiente ecuación:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (a+b) \cos(\theta) \\ (a+b) \sin(\theta) \end{bmatrix} \quad \dots(4)$$

Siendo:

- a = Parámetro constante que define la distancia efectiva entre el eje de giro del mecanismo del manipulador y la base del vástago del actuador lineal hidráulico (pistón).
- b = Distancia efectiva entre la base del vástago del actuador lineal hidráulico y el órgano terminal del manipulador.

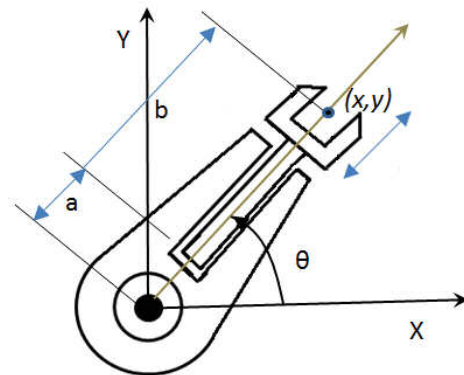


Figura 11. Parámetros del modelo cinemático directo.

Derivando la ecuación (4) se obtiene la velocidad del órgano terminal:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -(a+b) \sin(\theta) * \frac{d\theta}{dt} + \dot{b} \cos(\theta) \\ (a+b) \cos(\theta) * \frac{d\theta}{dt} + \dot{b} \sin(\theta) \end{bmatrix} \quad \dots(5)$$

La aceleración del órgano terminal queda definida por la expresión:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \ddot{b} \cos(\theta) - 2\dot{b} \sin(\theta) \dot{\theta} + (a+b)[- \dot{\theta}^2 \cos(\theta) - \ddot{\theta} \sin(\theta)] \\ \ddot{b} \sin(\theta) + 2\dot{b} \cos(\theta) \dot{\theta} + (a+b)[- \dot{\theta}^2 \sin(\theta) + \ddot{\theta} \cos(\theta)] \end{bmatrix} \quad \dots(6)$$

Por otra parte, con relación a la integración de las válvulas, los medidores analógicos y el modelo cinemático directo, la Figura 12 muestra la visualización del proceso de accionamiento del manipulador que se analizó bajo esta perspectiva de visualización.

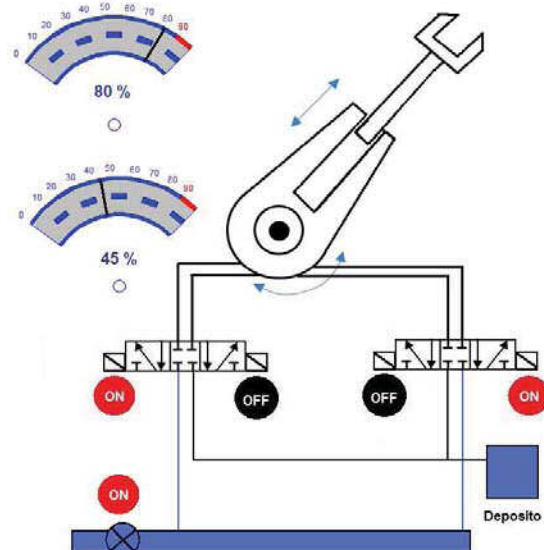


Figura 12. Visualización del accionamiento.

## IX. RESULTADOS.

El diseño y la implantación del software que simula el accionamiento de los elementos del manipulador considerado en este trabajo muestran una forma de integración de elementos mecánicos, eléctricos e hidráulicos bajo una plataforma multiusuario que muestre la simulación de cambios de forma interactiva con los usuarios. La experiencia lograda permite abordar elementos que permitan mayor integración, ya que se reconoce que se han modelado y simulador unos cuantos elementos del robot que están asociados a su movimiento manual. Esto sin tomar en cuenta los modelos energéticos y termodinámicos propios de un sistema hidráulico. Por otra parte, desde un punto de vista de la enseñanza de la robótica, visualizar dicho accionamiento mediante una interface gráfica desde cualquier navegador de internet facilita el aprendizaje de conceptos asociados a la robótica y manipulación virtual.

## X. CONCLUSIONES.

La construcción de simuladores para analizar el movimiento de mecanismos articulados no es algo nuevo [1], [2] y [3]. Sin embargo, el uso de un lenguaje de programación como java muestra la posibilidad en desarrollar programas que permitan visualizar el movimiento de un proceso de manipulación de robots utilizando plataformas de navegación en internet, así como tecnología móvil (teléfonos celulares, PDA, entre otros dispositivos). Los resultados logrados en diferentes grupos de alumnos muestran la portabilidad y la accesibilidad de forma segura en sistemas multiusuario. A raíz de la experiencia docente con alumnos de informática, se concluye que la implantación de programas de simulación de mecanismos articulados utilizando el lenguaje de programación Java, permite una mejor comprensión de fenómenos físicos, así como una interacción en internet que propicie la comparación de algoritmos y su depuración. En el futuro se pretende trabajar en los siguientes puntos:

- Integrar el uso de sensores en el software de visualización.
- Generar reportes vinculados con la administración de la manipulación con el robot.
- Integrar tecnología móvil en el software de visualización (teléfonos celulares).
- Modelar la cinemática inversa del robot.

Por otra parte, desde el punto de vista didáctico se reconoce que el desarrollo de éste tipo de herramientas permite en los estudiantes aprender y aplicar la metodología de diseño de software aquí presentada. Esto a fin de lograr habilidades en el desarrollo de software que les permita en el futuro explorar nuevas aplicaciones de java orientadas a generar beneficios en la sociedad.

## XI. REFERENCIAS

- [1] Esquembre F., "Easy Java Simulations: a software tool to create scientific simulations in Java", Computer Physics Communications, Volume 156, Issue 2, 1 January 2004, Pages 199-204.

- [2] J. Schumacher, D. Welch, D. Raymond, "Teaching introductory programming, problem solving and information technology with robots at West Point," *Frontiers in Education, Annual*, vol. 2, pp. F1B-2-7vol.2, Frontiers in Education Conference, 2001. 31st Annual, 2001.
- [3] Z.M. Qiu, Y.P. Chen, Z.D. Zhou, S.K. Ong and A.Y.C. Nee, "Multi-User NC Machining Simulation over the WWW", *The International Journal of Advanced Manufacturing Technology* Volume 18, Number 1, 1-6, DOI: 10.1007/PL00003949.
- [4] A. F. Quintas, "Java 2. Manual de usuario y tutorial" 4ª edición actualizada a la versión J2SE 5, Alfaomega y Ra-Ma, Cd. de México, México, 2006.
- [5] F. Esquembre, "Creación de Simulaciones Interactivas en Java", Pearson-Prentice Hall, Madrid, España, 2005
- [6] Vargas-Soto E. , "Teaching Mechatronics with Real Projects and Integral Vision", Clute Institute for Academic Research, Proc. of the Teaching and Learning Conference, ISSN 1539-8757, Salzburg, Austria, Junio 23-26, 2008.
- [7] Moreno R., Cabrera J. S., Hernández F., Vargas E. y Pedraza J.C., "Software de Simulación de un Robot Manipulador Marca Mitsubishi RV-2AJ ", 4to. Congreso Nacional de Mecatrónica, Asociación Mexicana de Mecatrónica A.C., Universidad Tecnológica de Coahuila, ISBN: 970-9702-01-7, pp. 129-132, Noviembre 3, 4 y 5 del 2005, Ramos Arizpe, Coahuila.
- [8] Imsys AB, Suecia, fecha: 10/Junio/2010, 11h30, sitio en internet: <http://www.imsystech.com/products/im1810-snf.htm>