

**Universidad de las Ciencias Informáticas**  
**Facultad 2**



**Título: “Desarrollo de una librería de funciones para el control de un robot manipulador.”**

**Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.**

**Autores:**

Yordan Ernesto Estrada Rodríguez  
Yunior Ernesto Guerrero Chacón

**Tutor:**

Ing. Javier Alexander León Martínez

**Ciudad de La Habana**  
**Junio del 2008**

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Yordan Ernesto Estrada Rodríguez

---

Firma del Autor

Yunior Ernesto Guerrero Chacón

---

Firma del Autor

Ing. Javier Alexander León Martínez

---

Firma del Tutor

## **Datos de Contacto**

Javier Alexander León Martínez

- ✓ Graduado de Ingeniería automática.
- ✓ Graduado en el Instituto Superior Politécnico José Antonio Echeverría durante el año 2003.
- ✓ Profesor de la asignatura máquinas computadoras de la Universidad de Ciencias Informáticas (UCI) desde el año 2003.
- ✓ Investiga en los temas de robótica, Inteligencia Artificial aplicada al control, sistemas de mediciones de campos magnéticos y desarrollo de hardware para la automatización.

Correo electrónico: [jleon@uci.cu](mailto:jleon@uci.cu).

## **Agradecimientos**

*Este trabajo de diploma fue posible gracias a la ayuda de algunas personas que contribuyeron y colaboraron de una manera u otra con la realización del mismo, y a los cuales les agradecemos por su apoyo incondicional.*

*Es imposible dejar de mencionar la valiosa colaboración para la feliz realización de este trabajo, al Dr. Abelardo del Pozo Quintero, al MsC. Guillermo Álvarez Bestard, al Tec. Miguel Ángel Machirán Simón, al Ing. Rafael Acosta, al Dr. Roberto Rodríguez Morales, y demás trabajadores e investigadores del Instituto de Cibernética, Matemática y Física (ICIMAF), que nos acogieron a todos como parte de su equipo.*

*Consideramos también de inestimable valor la colaboración del Profesor MsC. Eugenio César Sánchez, del Departamento de Automática y Computación del Instituto Superior Politécnico “José Antonio Echeverría” (ISPJAE), así como a los profesores MsC. Yadira Ruiz Constanten, Ing. Yenly Pérez Núñez, Ing. Lourdes Escalona Peral, Ing. Siovel Rodríguez Morales, Ing. Yalíce Gámez Batista, Dra. Sayda Coello González y a nuestra Tania Del Sol Viera, sin dejar de mencionar a nuestros compañeros Daniel García González, Cindy Santos Salgado, Aliuska Domínguez Rosales y Eliecer Peña Reyes, todos de nuestra Universidad de Ciencias Informáticas (UCI). Consideramos necesario agradecer además al Ing. Remberto León Martínez y al Ing. Javier González Orozco.*

*En especial queremos agradecer a nuestro tutor, Ing. Javier Alexander León Martínez por ser no solo un tutor, sino un amigo y el autor principal de este trabajo, que con paciencia y dedicación nos ha encaminado en nuestra vida profesional.*

## Dedicatoria

*A mis padres Denia Z. Rodríguez Martí y Jorge E. Estrada Sifones por el amor con que me han enseñado a caminar en la vida y por el apoyo que me han brindado en todo momento.*

*A mis tíos, abuelos y familia en general por demostrarme que todo es posible si de verdad se desea y a quienes les debo mucho.*

*A Cindy y su familia por apoyarme tanto y darme la fuerza para seguir adelante en todas las situaciones difíciles.*

*A todos mis amigos por su apoyo incondicional.*

*A todos aquellos que se preocupan por mí y que de una forma u otra han estado allí para mí cuando los he necesitado.*

*A la Revolución por darme la posibilidad de realizar mis sueños.*

*Yordan Ernesto Estrada Rodríguez*

*A todos mis amigos y compañeros...*

*A mi familia especialmente a mis padres por su apoyo...*

*A todos aquellos que siempre han estado a mi lado ayudándome a ser mejor cada día...*

*A los que no han estado y hubieran querido estar...*

*A los que me ayudaron a ser mejor...*

*A la Revolución y todos aquellos que han hecho posible este proyecto...*

*Yunior Ernesto Guerrero Chacón*

## **Resumen**

El presente trabajo muestra el desarrollo de una librería de funciones para el control secuencial del movimiento de un brazo manipulador de seis grados de libertad. En el mismo, se le da solución a la tarea cinemática inversa, permitiendo además la comunicación vía RS-232 con el hardware diseñado para ello.

Fue utilizado un método trigonométrico como modelo matemático, para dar solución a la tarea cinemática inversa. Se muestra además el ciclo de vida del proceso de desarrollo del software completo, guiado por la metodología RUP, obteniéndose una librería de funciones capaz de dar respuesta al problema planteado.

### **PALABRAS CLAVES:**

Robot manipulador; Cinemático inverso; Librería (DLL); Driver; Control inteligente

**Contenido**

DECLARACIÓN DE AUTORÍA..... II

DATOS DE CONTACTO..... II

AGRADECIMIENTOS..... III

DEDICATORIA.....IV

RESUMEN.....V

PALABRAS CLAVES:.....V

INTRODUCCIÓN..... 1

1 CAPITULO 1 FUNDAMENTACIÓN TEÓRICA..... 4

1.1 Introducción:..... 4

1.2 Desarrollo histórico..... 4

1.3 Teoría y conceptos..... 9

1.3.1 Definiciones de robot:..... 9

1.3.2 Definición de robótica..... 10

1.3.3 Cinemática..... 10

1.3.4 Dinámica..... 10

1.3.5 Grados de libertad..... 11

1.3.6 Movilidad..... 11

1.3.7 Articulaciones..... 12

1.3.8 Eslabones..... 12

1.3.9 Estructuras de Robots..... 12

1.3.10 Protocolo de comunicación..... 13

1.4 Tecnología..... 14

1.4.1 Selección del sistema..... 15

1.4.2 Compiladores para el lenguaje C#..... 15

1.5 Metodología..... 17

1.5.1 Características del RUP..... 17

1.6	Lenguaje de modelado.....	19
1.6.1	Historia de UML.....	19
1.6.2	¿Qué es UML? .....	20
1.6.3	Ventajas.....	20
1.7	Conclusiones .....	20
2	CAPITULO 2 ANÁLISIS MATEMÁTICO DEL ROBOT MANIPULADOR.....	21
2.1	Introducción.....	21
2.2	Cinemática del robot. ....	21
2.2.1	Síntesis Cinemática. ....	21
2.3	Método de Denavit y Hartenberg (D-H). ....	22
2.3.1	Algoritmo de Denavit – Hartenberg. ....	23
2.4	Solución algebraica matricial. (14).....	24
2.5	Problema cinemático directo. (14).....	27
2.6	Problema cinemático inverso. (6).....	31
2.6.1	Solución de la tarea inversa. (14).....	31
2.7	Problema cinemático en la práctica.....	42
2.8	Conclusiones.....	47
3	CAPITULO 3 CARACTERÍSTICAS DEL SISTEMA. ....	48
3.1	Introducción:.....	48
3.2	Negocio (Modelo de dominio): .....	48
3.2.1	Conceptos:.....	49
3.2.2	Requerimientos del sistema: .....	50
3.3	Modelo del Sistema.....	51
3.3.1	Definición de los actores.....	52
3.3.2	Descripción de los Casos de Uso del Sistema .....	54
3.4	Conclusiones.....	58
4	CAPITULO 4 DISEÑO DEL SISTEMA.....	59
4.1	Introducción.....	59
4.2	Diseño .....	59



Diagramas de Interacción.....	59
4.3 Diagrama de clases del diseño.....	59
4.4 Diagrama de despliegue .....	63
4.5 Diagrama de Componentes .....	64
4.6 Patrones GRASP usados. ....	65
4.6.1 Bajo Acoplamiento. ....	65
4.6.2 Experto.....	66
4.6.3 Controlador.....	66
4.7 Conclusiones.....	67
5 CAPITULO 5 ESTUDIO DE FACTIBILIDAD .....	68
5.1 Introducción.....	68
5.2 Puntos de Casos de Uso .....	68
5.3 Cálculo de Puntos de Casos de Uso ajustados .....	69
5.4 Cálculos del Proyecto .....	74
5.5 Beneficios tangibles e intangibles. Debe cuantificarse los tangibles.....	74
5.6 Análisis de costo .....	74
5.7 Conclusiones.....	75
CONCLUSIONES .....	76
RECOMENDACIONES.....	77
ANEXO 1 DIAGRAMAS DE INTERACCIÓN .....	78
TRABAJOS CITADOS .....	82
GLOSARIO .....	84

### **Introducción**

En el contexto actual, la noción de robótica atiende a una idea de estructura mecánica universal capaz de adaptarse, como el hombre, a muy diversos tipos de acciones y en las que concurren, en mayor o menor medida según los casos, las características de movilidad, gobernabilidad, autonomía y polivalencia. La robótica abarca una amplia gama de dispositivos con cualidades físicas y funcionales muy diversas, asociadas a la estructura mecánica particular de los mismos, a sus características operativas y al campo de aplicación para el que se ha concebido.

Todos estos factores están íntimamente relacionados, de forma que la configuración y el comportamiento de un robot condicionan su adecuación para un campo de aplicación específico. La robótica se apoya en gran medida en los progresos de la electrónica y la informática, así como en nuevas disciplinas como el reconocimiento de patrones y la inteligencia artificial; en este sentido la robótica cuenta con valiosos recursos a su alcance: electrónica, servomecanismos, controladores, sensores, equipos de comunicación, entre otros. Las investigaciones actuales se orientan especialmente a la construcción de máquinas capaces de trabajar en medios parcialmente desordenados y de responder con eficacia ante situaciones no totalmente previstas, o sea que el robot sea capaz de relacionarse con el entorno a través de sensores y de tomar decisiones en tiempo real.

Tanto la robótica como la informática, han tenido en los últimos tiempos un desarrollo ascendente creándose una dependencia tal, que ya no se concibe la primera sin la otra. La intervención de la informática en la creación de artefactos robóticos más inteligentes, funcionales y manuales ha constituido un significativo paso de avance en esta ciencia.

La constante necesidad de aumentar la flexibilidad y los campos de aplicación de los sistemas de robótica ha motivado la implementación y desarrollo de software con estos fines. Mediante el software se ha logrado que los robots puedan llevar a cabo tareas más complejas con mayor facilidad y rapidez.

En la actualidad la robótica es una ciencia de vanguardia en el mundo, y sus resultados de investigación y desarrollo tributan a corto y mediano plazo en resultados económicos. Esta es la razón fundamental del éxito de esta tecnología en las últimas décadas.

En Cuba, el tema de la robótica se vio seriamente afectado por la crisis económica de los años posteriores a 1990. Sin embargo, en los últimos años se aprecia un renacer del mismo en centros de investigación y universidades del país.

En nuestra universidad se viene realizando un grupo de trabajos como parte de la línea de desarrollo de hardware, robótica y control inteligente, del Grupo de Investigación de Automática Aplicada, al cual está suscrito al presente trabajo.

Para la realización de la investigación a la cual tributa sus resultados este trabajo, se recuperó un pequeño brazo robótico de seis grados de libertad, con la colaboración del Departamento de Control Automático del Instituto de Cibernética, Matemática y Física (ICIMAF<sup>i</sup>), y se diseñó la interfaz de hardware apropiada para el control directo del manipulador. Del mismo modo, se hace necesario desarrollar una biblioteca de funciones que, de manera flexible, permita resolver el problema cinemático inverso del robot manipulador, y que viabilice la comunicación con el nuevo hardware de interfaz para cualquier aplicación que lo requiera.

Se realizó el estudio de los métodos matemáticos que describen la cinemática de artefactos robóticos; desarrollando una librería de funciones que permita gobernar el movimiento de un robot manipulador, a partir de la solución del problema cinemático inverso, para la configuración mecánica específica del mismo.

Luego el problema científico es:

¿Cómo desarrollar el software necesario para el control del movimiento de un robot manipulador de seis grados de libertad?

Hacia la solución del problema planteado se dirige este trabajo de diploma; tomando como objeto de estudio el control de un robot manipulador de seis grados de libertad. Con ello se traza como objetivo general: crear un software para el control secuencial del movimiento de un robot manipulador de seis grados de libertad; y como objetivos específicos:

- ✓ Implementar una biblioteca que brinde estas funcionalidades.
- ✓ Implementar una aplicación que permita validar estos resultados.

Con el objetivo de guiar el curso de esta investigación se hace necesario dar respuesta a las siguientes preguntas científicas:

¿Qué algoritmos matemáticos hay detrás del movimiento cinemático de un robot?

¿Qué protocolo flexibiliza la comunicación software – hardware?

Se trazan como tareas de investigación:

- ✓ Realizar un estudio de la cinemática asociada a artefactos robóticos.
- ✓ Realizar un estudio de las especificaciones técnicas y el protocolo de comunicación con el hardware de interfaz diseñado para interactuar con el manipulador.
- ✓ Realizar una aplicación de Prueba para validar estos resultados.

De modo que como posible resultado de este trabajo, se tenga una biblioteca de funciones que permita efectuar el control secuencial del movimiento de un robot manipulador de seis grados de libertad.

### 1 Capítulo 1 Fundamentación teórica.

#### 1.1 Introducción:

Durante el desarrollo del software es preciso realizar una serie de estudios tanto conceptuales como informativos que garanticen el avance del mismo, en búsqueda de cumplir los requisitos requeridos se realiza el análisis y estudio de las tecnologías precursoras y actuales con el objetivo de perfeccionar el software.

#### 1.2 Desarrollo histórico.

La historia de la robótica ha estado unida a la construcción de "artefactos", que trataban de materializar el deseo humano de crear seres a su semejanza y que lo descargasen del trabajo. El ingeniero español Leonardo Torres Quevedo (que construyó el primer mando a distancia para su torpedo autónomo mediante telegrafía sin hilo, el ajedrecista automático, el primer transbordador aéreo y otros muchos ingenios) acuñó el término "automática" en relación con la teoría de la automatización de tareas tradicionalmente asociadas a los humanos. (1)

Si algún autor ha influido de manera considerable en la concepción del universo de los robots de ficción, éste ha sido sin duda alguna Isaac Asimov. Muchos otros, desde luego, han escrito sobre robots, pero ninguno ha relatado tan minuciosamente las actitudes y posibilidades de estas máquinas como lo hizo él. Tanto es así, que el Oxford English Dictionary reconoce a Asimov como inventor de la palabra "robótica".

Cuando tenía 22 años, Asimov escribió su cuarto relato corto sobre robots titulado "El círculo vicioso". En boca de unos de sus personajes planteó lo que consideraba axiomas básicos para el funcionamiento de un robot. Los llamó las tres reglas fundamentales de la robótica:

- ✓ Ningún robot puede hacer daño a un ser humano, o permitir que se le haga daño por no actuar.
- ✓ Un robot debe obedecer las órdenes dadas por un ser humano, excepto si estas órdenes entran en conflicto con la primera ley.
- ✓ Un robot debe proteger su propia existencia en la medida en que esta protección no sea incompatible con las leyes anteriores.

En definitiva, las famosas leyes de Asimov son aplicables a un universo donde los robots son seres inteligentes, pero quedan relegadas a una cartilla de parvulario al enfrentarse con la dura realidad.

El comienzo de la revolución industrial proporcionó un medio potencialmente poderoso de mover las máquinas: “el vapor”, las cuales fueron mejoradas por el regulador centrífugo de bolas de Sir James Watt, introducido en 1787.

El control a lazo cerrado, el desarrollo de herramientas especializadas y la división del trabajo en tareas que pudieran realizar las máquinas fueron ingredientes esenciales de la Revolución Industrial del siglo XVIII. A medida que mejoraba la tecnología, se desarrollaron máquinas más especializadas. Sin embargo, ninguna de estas máquinas tenía la versatilidad del brazo humano, y no podría alcanzar objetos alejados y colocarlos en la posición deseada.

El desarrollo del brazo artificial multiarticulado, o manipulador, llevó al robot moderno. El inventor estadounidense George Devol desarrolló en 1954 un brazo primitivo que se podía programar para realizar tareas específicas. En 1975, el ingeniero mecánico estadounidense Victor Scheinman desarrolló un brazo manipulador polivalente conocido como Brazo Manipulador Universal Programable (PUMA). El PUMA (Ver Fig1. y 2) es capaz de mover un objeto y colocarlo en cualquier posición a su alcance. El concepto multiarticulado del PUMA es la base de la mayoría de los robots actuales.

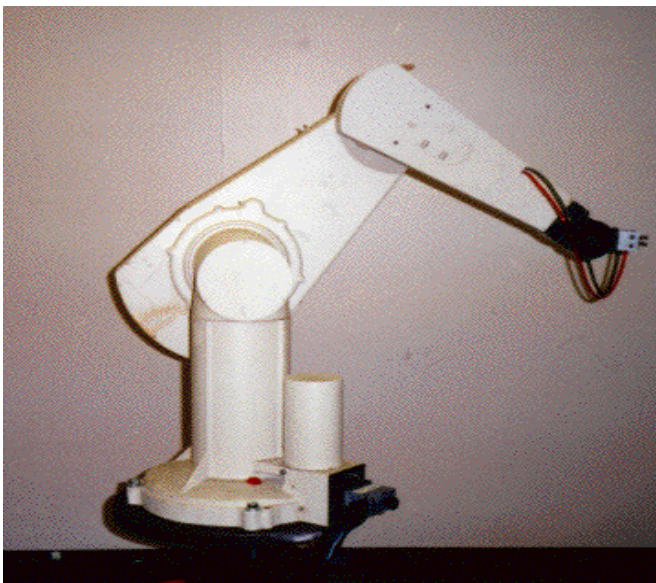


Fig.1: PUMA 560



Fig. 2: PUMA 500

En el año 1979 los japoneses introducían en las cadenas de montaje su robot SCARA (Selective Compliant Assembly Robot Arm) los cuales están dotados de libertad total de movimientos en los ejes X e Y pero limitados severamente en sus desplazamientos en el eje Z. (2) Es decir, se comportan de forma parecida al brazo humano, permitiendo ubicar el extremo de la mano en cualquier posición pero siempre sobre el plano horizontal. En el eje vertical sólo realizan manipulaciones simples que

habitualmente consisten en presionar y desplazarse unos pocos centímetros. Debido a estas características son muy usados en la fabricación de electrónica de consumo y en la clasificación de artículos para su empaquetado.

El manipulador de Stanford (Ver Fig. 3), fue uno de los primeros brazos robóticos con cinco uniones giratorias (R) y una prismática (P), y cuenta con seis grados de libertad total. Al extremo del manipulador está el efector final del robot, encargado de interactuar directamente con los objetos del entorno. Puede tratarse de un desatornillador o cualquier otra herramienta, un soplete para soldar, un aspersor de pintura o unas pinzas. La complejidad de éstas es muy variada. El más común es la mano mecánica, derivada de la mano humana. Los efectores se activan mediante señales eléctricas, hidráulicas o neumáticas. Algunos de ellos sólo aceptan señales de encendido/apagado, otros aceptan valores escalares (por ejemplo: girar a la derecha  $3^\circ$ ) y en algunos ámbitos del diseño de robots se cuenta con bibliotecas de subrutinas de nivel superior, o con lenguajes para especificar acciones que se convierten en señales primitivas. (3)

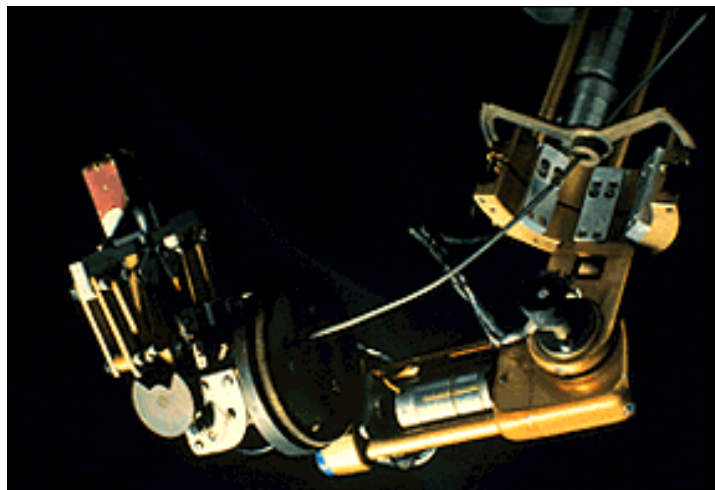


Fig. 3: Manipulador de Stanford

Otro típico brazo manipulador es el UNIMATE (Ver Fig. 4 y 5), el cual, es un robot hidráulico, de estructura espacial de tipo polar, que en su versión grande (UNIMATE 4000) es utilizado sobre todo en labores de forja, fundición y soldadura. Las versiones más pequeñas (UNIMATE 1000 y 2000) se utilizan en alimentación de maquinaria y manipulación.



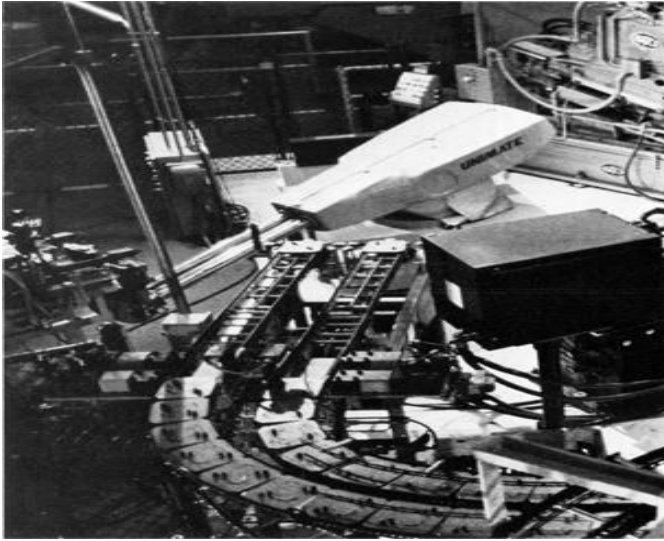


Fig. 4:UNIMATE

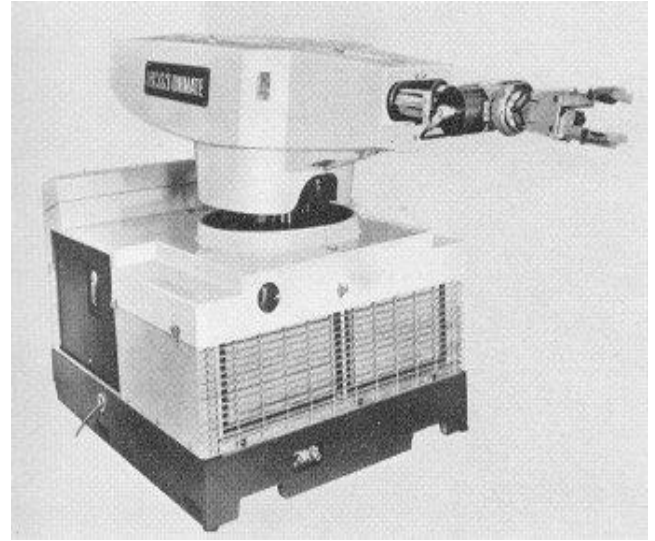


Fig. 5:UNIMATE

El ASEA 2000 (Ver Fig. 6) es un robot de 6 Grados de Libertad que puede encontrarse en aplicaciones de soldadura en la industria. Su tamaño puede considerarse mediano. Su espacio de trabajo es de aproximadamente 1.4 metros de radio. Los actuadores del robot son motores de corriente alterna y puede mover sin problemas una carga de 10 kg. En sus desplazamientos, consigue una precisión y repetitividad elevada. Puede ser programado de forma on-line por una consola, o de forma general, desde un PC a través de una conexión por puerto serie. Su lenguaje de programación es el ARLA (Asea Robot Lenguaje). Este lenguaje de programación de los robots ASEA a principios de los 90, fue sustituido por el RAPID en las nuevas controladoras. Se ha equipado además al robot con una pinza neumática con su correspondiente compresor.





Fig. 6: ASEA 2000

Los robots industriales incluyen el principio de secuenciación y retroalimentación, para proporcionar movimientos rápidos y precisos. La razón de por qué la aparición de robots es relativamente reciente reside en que esta viene precedida de la evolución de no solo máquinas computadoras rápidas y seguras que forman el corazón del robot y garantizan tanto el control como la reprogramabilidad, sino del desarrollo paralelo de sensores y su consecuente procesamiento sensorial.

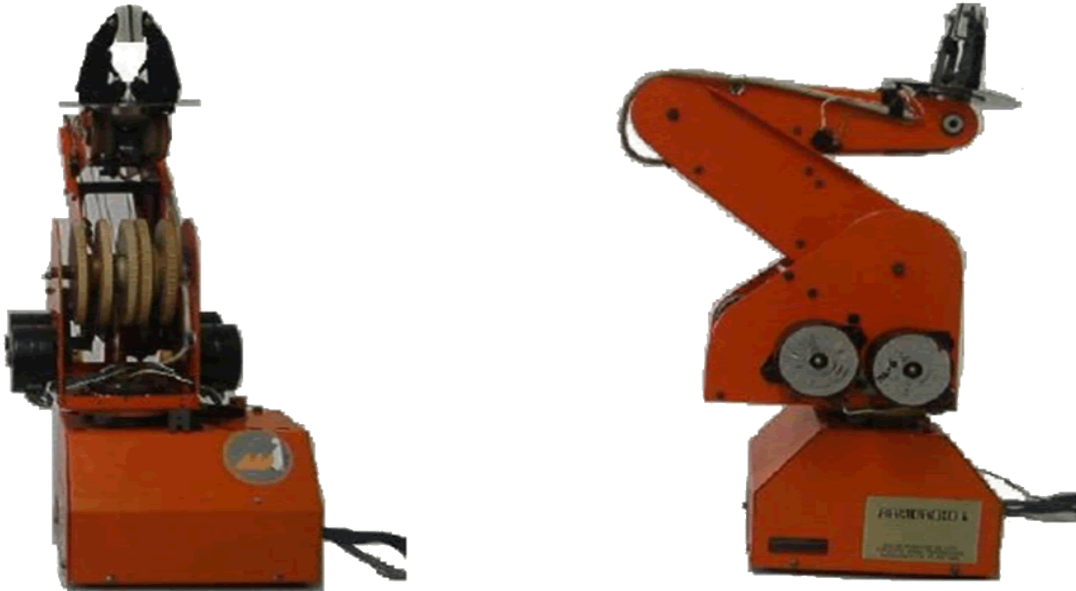


Fig. 7 Armdroid1

### 1.3 Teoría y conceptos

#### 1.3.1 Definiciones de robot:

La definición adoptada por el Instituto Norteamericano de Robótica aceptada internacionalmente para Robot es: "Manipulador multifuncional y reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables que permiten llevar a cabo diversas tareas". (4)

La anterior definición puede reducirse groseramente para su manejo como: Manipulador multifuncional programable. Si se busca en otras fuentes especializadas o diccionarios será posible encontrar: "Aparato automático que realiza funciones normalmente ejecutadas por los hombres. Máquina con forma humana. El término "robot" se debe a Karel Capek, quien lo utilizó en 1917 por primera vez, para denominar a unas máquinas construidas por el hombre y dotadas de inteligencia. Deriva de "robotnik" que define al esclavo de trabajo". (4)

### 1.3.2 Definición de robótica

Una vez introducido el concepto de robot es posible plantear la definición de la ciencia que estudia este tipo de dispositivos, la cual se denomina "Robótica" y ha evolucionado rápidamente en las últimas décadas. Es posible aproximarse a una definición de Robótica como:

"El diseño, fabricación y utilización de máquinas automáticas programables con el fin de realizar tareas repetitivas como el ensamble de automóviles, aparatos, etc. y otras actividades. Básicamente, la robótica se ocupa de todo lo concerniente a los robots, lo cual incluye el control de motores, mecanismos automáticos neumáticos, sensores, sistemas de cómputos, etc."

De esta definición es posible concluir que en la robótica se aúnan para un mismo fin varias disciplinas concluyentes, pero diferentes, como son la Mecánica, la Electrónica, la Automática, la Informática, etc. (4)

### 1.3.3 Cinemática

La cinemática de robots industriales y brazos manipuladores hace referencia al estudio de su movimiento respecto a un sistema de coordenadas fijo (o global) en función del tiempo, independientemente de las causas que lo produzca (sean fuerzas o momentos). El problema cinemático consiste en estudiar la geometría del robot en función del tiempo. (5)

#### 1.3.3.1 Problema cinemático directo:

Consiste en obtener la posición y orientación de la mano de sujeción en un brazo manipulador respecto de un sistema de referencia global, conociendo el vector  $\{q\}$  que contiene los parámetros del robot.

$$\{q\}^T = \{q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6\} \quad (6)$$

#### 1.3.3.2 Problema cinemático inverso:

Este problema consiste en la obtención de las posiciones de todos los elementos del manipulador, cuando se conoce la posición y orientación de la mano de sujeción (es decir, cuando se conoce la matriz  $[T_6]^{ii}$ ). Por lo que respecta a este, su finalidad es la obtención del vector  $\{q\}$  cuando se conoce la posición y orientación de la mano de sujeción. (5)

### 1.3.4 Dinámica

Desde el punto de vista de la TMM<sup>iii</sup> (Teoría de Mecanismos y máquinas), un sistema mecánico está formado por un conjunto de elementos, también mecánicos, entre los que existe un movimiento relativo definido.

La finalidad del análisis dinámico es llegar a conocer la naturaleza y magnitud de las fuerzas y movimientos e los elementos del sistema mecánico (incluido el elemento fijo o soporte). (5)

### **1.3.4.1 Los pasos a seguir en un análisis dinámico: (5)**

1º:

Elegir el modelo matemático más adecuado al sistema real. La elección deberá hacerse teniendo en cuenta consideraciones de precisión económicas.

2º:

Realizar hipótesis significativas, que idealizan aún más el modelo elegido y lo hacen calculable por los métodos de análisis conocidos.

3º:

Elegir un método de análisis dinámico adecuado al problema que se pretende resolver. Dentro de este aparato, se distinguen, a su vez, dos facetas:

- a) Planteamiento de las ecuaciones dinámicas del sistema (Newton, D'alambert, Lagrange, Trabajos Virtuales, Hamilton).
- b) Resolución de la ecuaciones, que serán, en general, ecuaciones diferenciales no lineales, por lo que no se puede esperar una solución en forma cerrada para ellas. Por ello, la resolución de las ecuaciones obliga a recurrir, en ocasiones, a métodos numéricos implementados en un ordenador.

### **1.3.5 Grados de libertad**

Se denomina grados de libertad de un mecanismo, al número de parámetros que es necesario conocer para determinar su posición. (5)

### **1.3.6 Movilidad**

Normalmente los robots están contruidos a partir de una serie de eslabones rígidos conectados por juntas o articulaciones. El tipo correcto de articulación define como puede moverse un eslabón en relación al otro. Sin embargo hay alternativas al enfoque de las series eslabón - articulación – eslabón. (5)

### 1.3.7 Articulaciones

Son comunes dos tipos de articulaciones: la prismática y la giratoria. Una articulación prismática, también conocida como articulación deslizante, posibilita a un eslabón deslizarse en línea recta sobre otro. Una articulación giratoria, si se considera el caso de un grado de libertad, toma la forma de una bisagra entre un eslabón y el próximo. Dos o más articulaciones de éstas puede combinarse estrechamente.

### 1.3.8 Eslabones

Con objeto de lograr la respuesta más rápida posible para un movimiento dado y un sistema de accionamiento, los eslabones que forman la estructura deben de mantenerse lo más ligeros posibles. Los eslabones deben también tan rígidos como sea posible. En la práctica hay que considerar muchos otros factores tales como el coste, las necesidades para alojar los accionadores, árboles de transmisión y cajas de engranaje, el comportamiento vibracional, el comportamiento no elástico tal como el pandeo, así como la necesidad de alcanzar un espacio de trabajo determinado.

### 1.3.9 Estructuras de Robots

#### 1.3.9.1 Robots móviles:

Tienen por definición algún medio de desplazarse tal como ruedas u orugas. Sin embargo, se considera más a un robot de base fija con las características adicionales de ser capaces de desplazar la base. Tales medios incluyen vehículos de oruga, la locomoción bípeda y múltipoda<sup>iv</sup>. La movilidad permite al robot desplazarse de uno a otro lugar de trabajo o trasladar objetos a distancias largas sin necesidad de un sistema de transporte especial. En definitiva los robots de base fija convencionales se han montado sobre raíles elevados para darles mayor movilidad, (normalmente en una sola dirección). (5)

#### 1.3.9.2 Robots cartesianos (de coordenadas rectangulares):

Las primeras tres articulaciones son del tipo prismático proporcionando un espacio de trabajo en forma de caja. Hay tres ejes de las juntas que son ortogonales situado a lo largo de ejes x, y, z de un sistema de coordenadas cartesianas. Las posiciones de las articulaciones son por lo tanto idénticas a las coordenadas de la posición del efector final, haciendo que este robot sea de los más sencillos de controlar. (5)

#### 1.3.9.3 Robots giratorios:

Todas las articulaciones son de tipo giratorio. Se le denomina antropomórfico debido a las similitudes entre su estructura y el brazo humano. Estos robots tienen un gran espacio de trabajo y son muy

populares, pero su control es mucho más complejo que el robot cartesiano, debido a su análisis dinámico.

Robots mezclados giratorios- prismáticos: La característica principal es el uso de dos o tres articulaciones giratorias con ejes verticales. La complejidad de su manejo se sitúa entre la del robot cartesiano y el robot completamente giratorio. (5)

### **1.3.9.4 Robots distribuidos:**

En lugar de utilizar un robot rápido con seis grados de libertad para realizar una tarea complicada, puede ser más económico dividir la tarea en una serie de operaciones llevadas a cabo en paralelo por robots más sencillos y baratos. Incluso algunos dispositivos pueden tener un solo grado de libertad. (5)

### **1.3.9.5 Robots paralelos:**

Estos, tienen los eslabones dispuestos en paralelo en lugar de en serie. Normalmente tienen un espacio de trabajo pequeño, pero sus errores de posicionamiento ya no son acumulativos como en un robot de eslabones en serie. La principal desventaja de los mecanismos puramente paralelos es su limitado campo de movimiento giratorio, además de que su tratamiento matemático es complejo en comparación con sus homólogos series. (5)

### **1.3.9.6 Robots de eslabones flexibles:**

Todos los eslabones de los robots se flexionarán en algún grado bajo cargas estáticas o dinámicas. La aplicación más corriente de eslabones deliberadamente flexibles se encuentra en el diseño de dedos de agarre. (5)

### **1.3.10 Protocolo de comunicación**

Los protocolos son reglas de comunicación que permiten el flujo de información entre sistemas que manejan lenguajes distintos. Por ello, para que exista una correcta comunicación entre el software y el hardware, es necesario establecer un lenguaje común que favorezca el entendimiento de ambas partes.

Luego de un estudio realizado en pos de implementar dicho protocolo, se concluyó que se utilizaría un lenguaje simple, donde los comandos que reconoce el hardware serán identificados por los ASC II de las letras que se numerarán a continuación.

Comando	Significado	Descripción
R	Reiniciar	Enviado desde la PC y ordena al brazo manipulador volver a su posición inicial.
C	Cerrar	Enviado desde la PC y ordena al brazo manipulador cerrar la mano.
A	Abrir	Enviado desde la PC y ordena al brazo manipulador abrir la mano.
0	Recibido	Enviado desde el hardware al sistema como chequeo de recepción.
1	Terminó	Enviado desde el hardware al sistema, y significa que ha terminado una acción específica que se ordenó.
E	Enviar	Enviado desde la PC y prepara a la interfaz para la recepción de los datos para el movimiento.

El envío de los datos para el movimiento se realiza luego del envío del comando “E” y se efectúa de uno en uno siempre esperando la confirmación de recepción. Estos son enviados siguiendo siempre el mismo orden.

### 1.4 Tecnología

C# (pronunciado "si Sharp" o C sostenido) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA<sup>v</sup> e ISO<sup>vi</sup>. (7)

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes (más notablemente de Delphi y Java). C# fue diseñado para combinar el control de lenguajes de bajo nivel como C y la velocidad de programación de lenguajes de alto nivel como Visual Basic. (7)

C# significa, "do sostenido" (C corresponde a do en la terminología musical anglo-sajona). El símbolo # viene de sobreponer "++" sobre "++" y eliminar las separaciones, indicando así su descendencia de C++.

C#, como parte de la plataforma .NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 acabó la beta y salió la versión 2.0 del lenguaje que incluye mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos

parciales y tipos anulables. Ya existe la versión 3.0 de C# en fase de beta destacando los tipos implícitos y el LINQ (Language Integrated Query). (7)

Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Aunque aún no existen, es posible implementar compiladores que no generen programas para dicha plataforma, sino para una plataforma diferente como Linux, Win32 o UNIX. (7)

### 1.4.1 Selección del sistema.

#### 1.4.1.1 Principales Ventajas

El sistema es desarrollado en C# principalmente a petición del usuario; pero aún así, es necesario dar a conocer las ventajas que este lenguaje de programación ofrece, dentro de algunos ejemplos se pueden mencionar:

- ✓ Compila a código intermedio (CIL) independiente del lenguaje en que haya sido escrita la aplicación e independiente de la máquina donde vaya a ejecutarse
- ✓ Recolección de basura de forma automática.
- ✓ Eliminación del uso punteros, en C# no se necesitan.
- ✓ No hay que preocuparse por archivos de cabecera ".h".
- ✓ No importa el orden en que hayan sido definidas las clases ni las funciones.
- ✓ No hay necesidad de declarar funciones y clases antes de definir las.
- ✓ No existen las dependencias circulares.
- ✓ Soporta definición de clases dentro de otras.
- ✓ No existen funciones, ni variables globales, todo pertenece a una clase.
- ✓ Todos los valores son inicializados antes de ser usados (automáticamente por defecto, o manualmente desde constructores estáticos).
- ✓ No se pueden utilizar valores no booleanos (enteros, coma flotante...) para condicionales. Es mucho más limpio y menos propenso a errores.
- ✓ Puede ejecutarse en una sandbox restringida. (8)

#### 1.4.2 Compiladores para el lenguaje C#:

- ✓ Microsoft.NET framework SDK incluye un compilador de C#, pero no un IDE.
- ✓ Microsoft Visual C#, IDE por excelencia de este lenguaje, versión 2002, 2003, 2005 y 2008.
- ✓ #develop, es un IDE libre para C# bajo licencia LGPL, muy similar a Microsoft Visual C#.



- ✓ Mono, es una implementación GPL de todo el entorno .NET desarrollado por Novell. Como parte de esta implementación se incluye un compilador de C#.
- ✓ Delphi 2006, de Borland Software Corporation.
- ✓ DotGNU Portable.NET, de la Free Software Foundation. (7)

### 1.4.2.1 Selección del framework

A continuación se resumen las ventajas más importantes que proporciona el Framework 2.0 de .Net:

- ✓ Código administrado: El CLR realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.
- ✓ Interoperabilidad multilenguaje: El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio (MSIL).
- ✓ Compilación just-in-time: El compilador JIT incluido en el Framework compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.
- ✓ Garbage collector: El CLR proporciona un sistema automático de administración de memoria denominado recolector de basura (garbage collector). El CLR detecta cuándo el programa deja de utilizar la memoria y la libera automáticamente. De esta forma el programador no tiene por que liberar la memoria de forma explícita aunque también sea posible hacerlo manualmente (mediante el método `Dispose()` se libera el objeto para que el recolector de basura lo elimine de memoria).
- ✓ Seguridad de acceso al código: Se puede especificar que una pieza de código tenga permisos de lectura de archivos pero no de escritura. Es posible aplicar distintos niveles de seguridad al código, de forma que se puede ejecutar código procedente del Web sin tener que preocuparse si esto va a estropear el sistema.
- ✓ Despliegue: Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El Framework realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones. (9)

Además de las ventajas expuestas se puede decir que para la comunicación mediante el protocolo RS232 brinda ventajas ya que implementa la clase `SerialPort` donde se definen métodos para interactuar con él de forma rápida y fácil.

### 1.5 Metodología

El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de Rational, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

El nombre Proceso Unificado se usa para describir el proceso genérico que incluye aquellos elementos que son comunes a la mayoría de los refinamientos existentes. También permite evitar problemas legales ya que Proceso Unificado de Rational o RUP son marcas registradas por IBM (desde su compra de Rational Software Corporation en 2003). El primer libro sobre el tema se denominó, en su versión española, El Proceso Unificado de Desarrollo de Software (ISBN 84-7829-036-2) y fue publicado en 1999 por Ivar Jacobson, Grady Booch y James Rumbaugh, conocidos también por ser los desarrolladores del UML, el Lenguaje Unificado de Modelado. Desde entonces los autores que publican libros sobre el tema y que no están afiliados a Rational utilizan el término Proceso Unificado, mientras que los autores que pertenecen a Rational favorecen el nombre de Proceso Unificado de Rational. (10)

#### 1.5.1 Características del RUP

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto. (10)

Diagrama ilustrando como el énfasis relativo en las distintas disciplinas cambia a lo largo del proyecto.

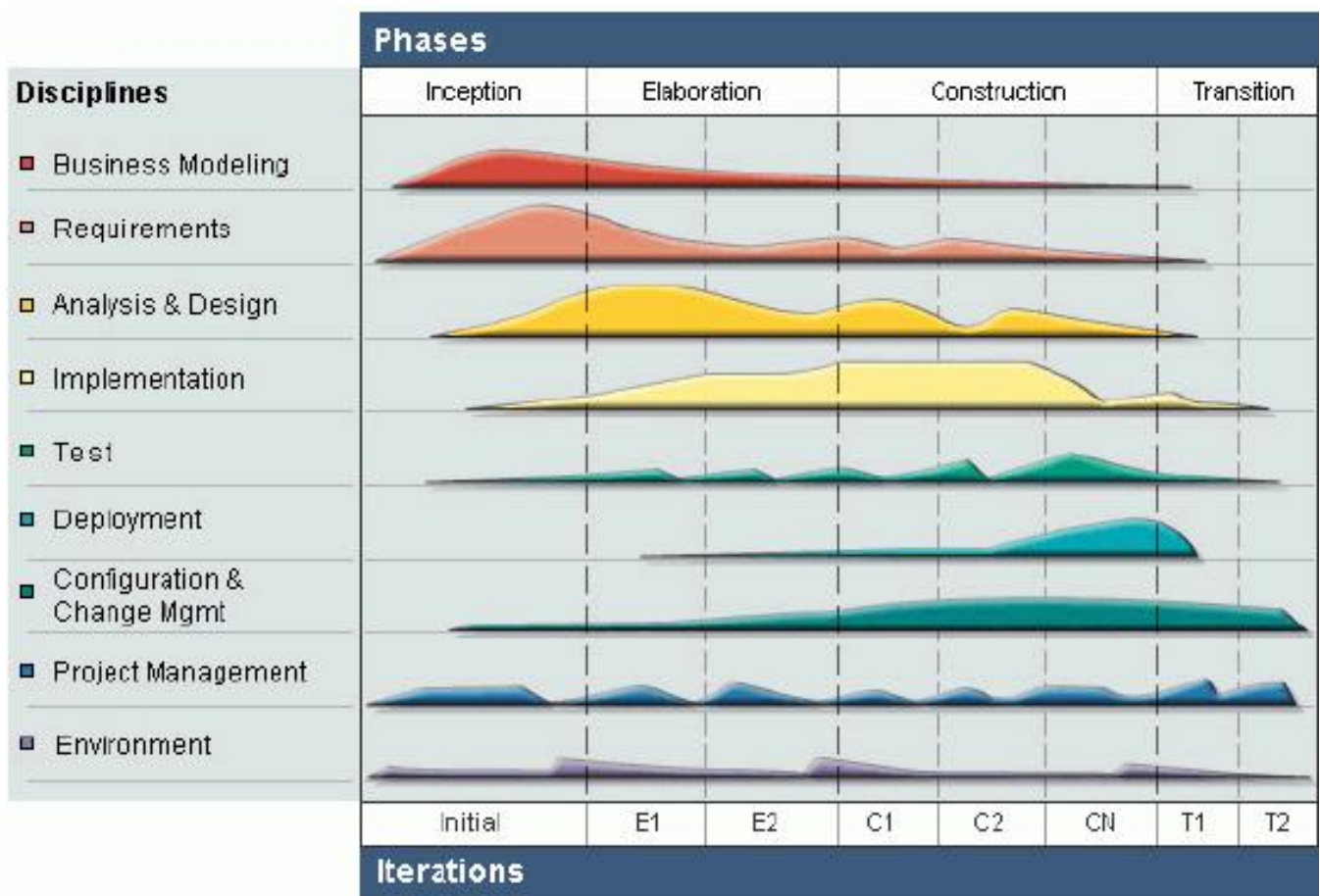


Fig. 8 Diagrama ilustrando como el énfasis relativo en las distintas disciplinas cambia a lo largo del proyecto

### 1.5.1.1 Dirigido por los casos de uso

En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. el proceso dirigido por casos de uso es el RUP. (10)

### **1.5.1.2 Centrado en la arquitectura**

El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc. (10)

### **1.5.1.3 Enfocado en los riesgos**

El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero. (10)

Se llega a la conclusión de que la metodología RUP es la más indicada para el desarrollo del sistema.

## **1.6 Lenguaje de modelado**

Teniendo en cuenta que la metodología escogida propone como lenguaje de modelado al UML se valora las principales cualidades y ventajas del mismo.

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc., hasta la implementación y configuración con los diagramas de des-pliegue. (11)

### **1.6.1 Historia de UML**

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch (dos reputados investigadores en el área de metodología del software). El objetivo de ambos era unificar dos métodos que habían desarrollado: el método Booch y el OMT (Object Modeling Tool). El primer borrador apareció en octubre de 1995. En esa misma época otro reputado investigador, Jacobson, se unió a Rational y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Además, este lenguaje se abrió a la colaboración de otras empresas para que aportaran sus ideas. Todas estas colaboraciones condujeron a la definición de la primera versión de UML. (11)

### 1.6.2 ¿Qué es UML?

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y una regla para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este lenguaje indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. (11)

### 1.6.3 Ventajas

UML es además un método formal de modelado. Esto aporta las siguientes ventajas:

- ✓ Mayor rigor en la especificación.
- ✓ Permite realizar una verificación y validación del modelo realizado.
- ✓ Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto. (11)

## 1.7 Conclusiones

El objetivo de este capítulo fue el realizar un análisis de la tecnología a utilizar, de la herramienta para modelar el problema, los principales lenguajes de programación y plataformas de desarrollo. Luego de realizado el proceso de investigación, se decidió que el sistema se implementará utilizando como lenguaje C#, plataforma .NET utilizando metodología RUP y UML como lenguaje de modelación.

## 2 Capítulo 2 Análisis matemático del robot manipulador.

### 2.1 Introducción

En este capítulo se introduce la teoría en la cual se sostiene el modelo matemático del sistema. Se enuncian las soluciones tanto del problema Cinemático directo como del Inverso, además de los ajustes necesarios para conectar la teoría con la realidad.

### 2.2 Cinemática del robot.

Consiste en el estudio de su movimiento con respecto a un sistema de referencia. O sea, no es más que una descripción analítica del movimiento espacial del robot como una función del tiempo.

Relaciona la posición y orientación del extremo del robot (localización) con los valores de sus coordenadas articulares.

#### 2.2.1 Síntesis Cinemática.

Para dar solución al problema cinemático existen distintos métodos de solución: geométricos, trigonométricos, algebraicos e iterativos.

Los métodos trigonométricos, al igual que los geométricos brindan ecuaciones cerradas simples como solución a la cinemática, lo que los convierte en los métodos más sencillos de aplicar, sin embargo, sólo son aplicables cuando la geometría del robot manipulador es relativamente simple (por lo general hasta 3 grados de libertad).

Los métodos algebraicos por su parte, son algo más independientes de la geometría del robot, y permiten la obtención de ecuaciones relativamente poco complejas. Sin embargo, los métodos iterativos, poseen la enorme ventaja de ignorar por completo la geometría del robot manipulador, pero a costo de una inmensa pérdida de tiempo con respecto a los demás métodos, debido a la alta complejidad de los cálculos que realizan.

#### ✓ Problema cinemático directo:

Determinar la posición y orientación del extremo del robot, con respecto a un sistema de coordenadas de referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot. (12)

✓ Problema cinemático inverso:

Determinar la configuración que deben adoptar las articulaciones del robot para alcanzar una posición y orientación del extremo conocidas.

✓ Modelo diferencial (matriz Jacobina):

Parte de la cinemática que intenta obtener las relaciones entre las velocidades del movimiento de las articulaciones y las del extremo del robot.

### 2.3 Método de Denavit y Hartenberg (D-H).

Permite el paso de un eslabón al siguiente mediante 4 transformaciones básicas, que dependen exclusivamente de las características constructivas del robot. Las transformaciones básicas que relacionan el sistema de referencia del elemento  $i$  con el sistema del elemento  $i-1$  son:

1. Rotación  $\theta_i$  alrededor del eje  $z_{i-1}$ .
2. Traslación  $d_i$  a lo largo del eje  $z_{i-1}$ .
3. Traslación  $a_i$  a lo largo del eje  $x_i$ .
4. Rotación alrededor del eje  $x_i$ , un ángulo  $\alpha_i$ .

$$\begin{aligned}
 {}^{i-1}A_i &= T(z, \theta_i)T(0,0, d_i)T(a_i, 0,0)T(x, \alpha_i) = \\
 &\begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &\begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & \alpha_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & \alpha_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)
 \end{aligned}$$

### 2.3.1 Algoritmo de Denavit – Hartenberg.

- ✓ Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- ✓ Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n.
- ✓ Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- ✓ Para i de 0 a n-1 situar el eje  $z_i$  sobre el eje de la articulación i+1.
- ✓ Situar el origen del sistema de la base  $\{S_0\}$  en cualquier punto del eje  $z_0$ . Los ejes  $x_0$  e  $y_0$  se situarán de modo que formen un sistema dextrógiro con  $z_0$ .
- ✓ Para i de 1 a n-1, situar el sistema  $\{S_i\}$  (solidario al eslabón i) en la intersección del eje  $z_i$  con la línea normal común a  $z_{i-1}$  y  $z_i$ . Si ambos ejes se cortasen se situaría  $\{S_i\}$  en el punto de corte. Si fuesen paralelos  $\{S_i\}$  se situaría en la articulación i+1.
- ✓ Para i de 1 a n-1, situar  $x_i$  en la línea normal común a  $z_{i-1}$  y  $z_i$ .
- ✓ 8) Para i de 1 a n-1, situar  $y_i$  de modo que forme un sistema dextrógiro con  $x_i$  y  $z_i$ .
- ✓ Situar el sistema  $\{S_n\}$  en el extremo del robot de modo que  $z_n$  coincida con la dirección de  $z_{n-1}$  y  $x_n$  sea normal a  $z_{n-1}$  y  $z_n$ .
- ✓ Obtener  $\theta_i$  como el ángulo que hay que girar en torno a  $z_{i-1}$  para que  $x_{i-1}$  y  $x_i$  queden paralelos.
- ✓ Obtener  $d_i$  como la distancia, medida a lo largo de  $z_{i-1}$ , que habría que desplazar  $\{S_{i-1}\}$  para que  $x_i$  y  $x_{i-1}$  quedasen alineados.
- ✓ Obtener  $a_i$  como la distancia medida a lo largo de  $x_i$ , que ahora coincidiría con  $x_{i-1}$ , que habría que desplazar el nuevo  $\{S_{i-1}\}$  para que su origen coincidiese con  $\{S_i\}$ .
- ✓ Obtener  $\alpha_i$  como el ángulo que habría que girar en torno a  $x_i$ , que ahora coincidiría con  $x_{i-1}$ , para que el nuevo  $\{S_{i-1}\}$  coincidiese totalmente con  $\{S_i\}$ .
- ✓ Obtener las matrices de transformación  ${}^{i-1}A_i$ .
- ✓ Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot:
 
$$T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n$$
- ✓ La matriz T define la orientación (sub matriz de rotación) y posición (sub matriz de traslación) del extremo referido a la base en función de la n coordenada articulares.



Los cuatro parámetros de D-H ( $\theta_i$ ,  $d_i$ ,  $a_i$ ,  $\alpha_i$ ) dependen únicamente de las características geométricas de cada eslabón y de las articulaciones que le unen con el anterior y el siguiente. (6)

### 2.4 Solución algebraica matricial. (14)

En el trabajo de Denavit y Hartenberg (1955) referido a los mecanismos, se definieron las relaciones entre los eslabones y las articulaciones y se crearon las matrices de transformación homogénea para representar las relaciones entre dos articulaciones sucesivas enlazadas por un eslabón, mediante la asignación de sistemas de coordenadas cartesianas en cada articulación.

Esta teoría fue aplicada en los robots por Paul et al (1981<sup>a</sup>), donde se obtienen las matrices de transformación homogénea para cada articulación mediante la expresión:

$$A_i = \text{Rot}(Z_{i-1}, \theta_i) * \text{Trasl}(Z_{i-1}, d_i) * \text{Trasl}(X_{i-1}, a_i) * \text{Rot}(X_i, \alpha_i) \quad (3.4)$$

Se utiliza una nueva representación de los sistemas de coordenadas cartesianas en las articulaciones para determinar las matrices de transformación homogénea por la expresión (Pozo, 1988):

$$A_i = \text{Rot}(Z_{i-1}, \vartheta_i) * \text{Trasl}(Z_{i-1}, L_i) \quad (3.5)$$

Esto significa que se asignará el eje Z a lo largo del eslabón y no que coincida con el eje de la articulación como se definió en el artículo de Paul et al. (1981<sup>a</sup>). Por lo tanto en la ecuación (1.2) se hará una rotación del sistema (i-1) un ángulo  $\vartheta_i$ , alrededor del eje  $Z_{i-1}$  hasta hacer coincidir los ejes  $Z_{i-1}$  y  $Z_i$ . Después se traslada el sistema (i-1) una distancia  $L_i$  a lo largo del eslabón hasta hacer coincidir los sistemas de coordenadas (i-1) e (i).

Los ejes  $X_i$  correspondientes a los diferentes sistemas de coordenadas cartesianas se mantienen paralelos entre sí. Los ejes  $Y_i$  de los diferentes sistemas de coordenadas cartesianas se calculan por la regla de la mano derecha.

Las matrices de transformación homogénea se forman aplicando la ecuación (1.2) para  $i=1,2,\dots,6$ .

$$\begin{aligned}
 A_1 &= \text{Rot}(Z_0, \theta_1) * \text{Trasl}(0, 0, 0) \\
 A_2 &= \text{Rot}(Y_1, \theta_2) * \text{Trasl}(0, 0, L_2) \\
 A_3 &= \text{Rot}(Y_2, \alpha_3) * \text{Trasl}(0, 0, L_3) \\
 A_4 &= \text{Rot}(X_3, \theta_4) * \text{Trasl}(0, 0, L_4) \\
 A_5 &= \text{Rot}(Y_4, \theta_5) * \text{Trasl}(0, 0, L_5) \\
 A_6 &= \text{Rot}(Z_5, \theta_6) * \text{Trasl}(0, 0, 0)
 \end{aligned}
 \tag{3.6}$$

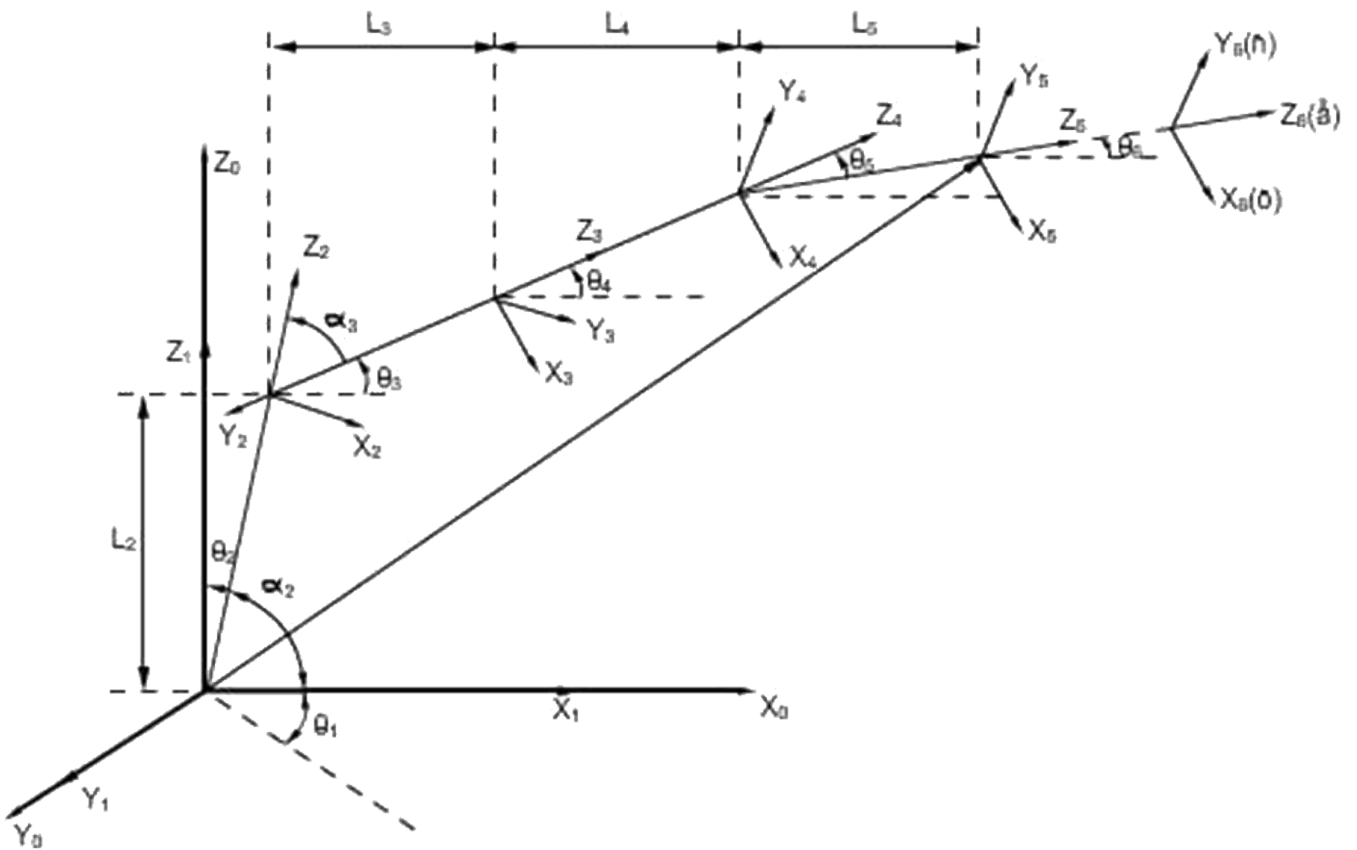


Fig. 9: Esquema Cinemático.

Se observa que en las articulaciones 1 y 6 los ejes  $Z_{i-1}$  y  $Z_i$  al ser paralelos se produce la rotación sobre ellos mismos. En las articulaciones 2, 3, 4 y 5 los ejes  $Z_{i-1}$  y  $Z_i$  correspondientes a cada articulación

crean un ángulo entre si. En la articulación 3, el ángulo que se genera es diferente de los ángulos de coordenadas generalizadas y hay que expresarlo en función de estos mismos. Por lo tanto, de la

Fig. 9.

$$\alpha_2 = 90 - \theta_2 = \alpha_3 + \theta_3 \quad (3.7)$$

$$\alpha_3 = 90 - (\theta_2 + \theta_3) = 90 - \theta_{23} \quad (3.8)$$

Donde:

$$\theta_{23} = \theta_2 + \theta_3$$

Representando las matrices de rotación y traslación de la ecuación (3.6) por matrices de transformación homogénea se obtienen las matrices:

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

$$A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_2 & 0 & s_2 & s_2 L_2 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & c_2 L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

$$A_3 = \begin{bmatrix} c\alpha_3 & 0 & s\alpha_3 & 0 \\ 0 & 1 & 0 & 0 \\ -s\alpha_3 & 0 & c\alpha_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{23} & 0 & c_{23} & c_{23} L_3 \\ 0 & 1 & 0 & 0 \\ -c_{23} & 0 & s_{23} & s_{23} L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_4 & -s_4 & 0 \\ 0 & s_4 & c_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_4 & -s_4 & s_4 L_4 \\ 0 & s_4 & c_4 & c_4 L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ 0 & 1 & 0 & 0 \\ -s_5 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_5 & 0 & s_5 & s_5 L_5 \\ 0 & 1 & 0 & 0 \\ -s_5 & 0 & c_5 & c_5 L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

Donde:

$$s_i = \sin \theta_i \quad y \quad c_i = \cos \theta_i$$

$$s_{23} = \sin \theta_{23} \quad y \quad c_{23} = \cos \theta_{23}$$

## 2.5 Problema cinemático directo. (14)

Teniendo todas las coordenadas cartesianas definidas y las matrices  $A_i$  de transformación homogénea que relacionan las articulaciones sucesivas, es posible obtener las coordenadas absolutas de la mano o agarre referidas a la base del robot, para un juego de valores de las coordenadas generalizadas ( $q$ ), esto es, la solución de la tarea directa.

Ahora bien, la relación de las coordenadas absolutas del robot con respecto a las coordenadas  $n-1$ , esta dado por  $U_n$ , donde:

$$U_n = A_n * A_{n+1} * \dots * A_6 \quad (3.15)$$

Entonces el sistema de coordenadas de la mano o agarre con respecto a la base, se expresa como la matriz  $T_6$  y está dada por la matriz  $U_1$ , o sea:

$$T_6 = U_1 = A_1 * A_2 * A_3 * A_4 * A_5 * A_6 \quad (3.16)$$

Con el fin de poder hallar el producto de matrices expresada en la ecuación (3.1) en cada articulación, se expresará esta en sentido inverso con el objetivo de usar los productos intermedios después; por lo tanto:

$$U_n(q) = A_n(q) * U_{n+1}(q) \quad (3.17)$$

Por lo que desarrollando la ecuación 3.17

$$U_6 = A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

$$U_5 = A_5 * U_6 = \begin{bmatrix} c_5 c_6 & -c_5 c_6 & s_5 & s_5 L_5 \\ s_6 & c_6 & 0 & 0 \\ -s_5 c_6 & s_5 s_6 & c_5 & c_5 L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

$$U_4 = A_4 * U_5 = \begin{bmatrix} c_5 c_6 & -c_5 c_6 & s_5 & s_5 L_5 \\ s_4 s_5 c_6 + c_4 s_6 & -s_4 s_5 s_6 + c_4 c_6 & -s_4 c_5 & -s_4 (c_5 L_5 + L_4) \\ -c_4 s_5 c_6 + s_4 s_6 & c_4 s_5 s_6 + s_4 c_6 & c_4 c_5 & c_4 (c_5 L_5 + L_4) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

$$U_3 = A_3 * U_4 = \begin{bmatrix} s_{23} c_5 c_6 + c_{23} (-c_4 s_5 s_6 + s_4 s_6) & -s_{23} c_5 s_6 + c_{23} (c_4 s_5 s_6 + s_4 c_6) \\ s_4 c_5 c_6 + c_4 s_6 & -s_4 s_5 s_6 + c_4 c_6 \\ -s_{23} c_5 c_6 + s_{23} (-c_4 s_5 s_6 + s_4 s_6) & c_{23} c_5 s_6 + s_{23} (c_4 s_5 s_6 + s_4 c_6) \\ 0 & 0 \\ s_{23} s_5 + c_{23} c_4 c_5 & s_{23} s_5 L_5 + c_{23} [c_4 (c_5 L_5 + L_4) + L_3] \\ -s_4 c_5 & -s_4 (c_5 L_5 + L_4) \\ -c_{23} s_5 + s_{23} c_4 c_5 & c_{23} c_5 L_5 + s_{23} [c_4 (c_5 L_5 + L_4) + L_3] \\ 0 & 1 \end{bmatrix} \quad (3.21)$$

$$U_2 = A_2 * U_3 = \begin{bmatrix} s_3 c_5 c_6 + c_3 (-c_4 s_5 c_6 + s_4 s_6) & -s_3 c_5 s_6 + c_3 (c_4 s_5 s_6 + s_4 c_6) \\ s_4 s_5 s_6 + c_4 s_6 & -s_4 s_5 s_6 + c_4 c_6 \\ -c_3 c_5 c_6 + s_3 (-c_4 s_5 c_6 + s_4 s_6) & c_3 c_5 s_6 + s_3 (c_4 s_5 s_6 + s_4 c_6) \\ 0 & 0 \\ s_3 s_5 + c_3 c_4 c_5 & s_3 s_5 L_5 + c_3 [c_4 (c_5 L_5 + L_4) + L_3] + s_2 L_2 \\ -s_4 c_5 & -s_4 (c_5 L_5 + L_4) \\ -c_3 s_5 + s_3 c_4 c_5 & -c_3 s_5 L_5 + s_3 [c_4 (c_5 L_5 + L_4) + L_3] + c_2 L_5 \\ 0 & 1 \end{bmatrix} \quad (3.22)$$

$$U_1 = A_1 * U_2 = T_6 = \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

Donde:

$$\begin{aligned} n_x &= c_4 [s_3 c_5 c_6 + c_3 (-c_4 s_5 c_6 + s_4 s_6)] - s_1 (s_4 s_5 c_6 + c_4 s_6) \\ n_y &= s_1 [s_3 c_5 c_6 + c_3 (-c_4 s_5 c_6 + s_4 s_6)] + c_1 (s_4 s_5 c_6 + c_4 s_6) \\ n_z &= s_1 [s_3 c_5 c_6 + c_3 (-c_4 s_5 c_6 + s_4 s_6)] + c_1 (s_4 s_5 c_6 + c_4 s_6) \\ o_x &= -c_3 c_5 c_6 + s_3 (-c_4 s_5 c_6 + s_4 s_6) \\ o_y &= c_1 [-s_3 c_5 s_6 + s_3 (c_4 s_5 s_6 + s_4 c_6)] - s_1 (-s_4 s_5 s_6 + c_4 c_6) \\ o_z &= s_1 [-s_3 c_5 s_6 + s_3 (c_4 s_5 s_6 + s_4 c_6)] + c_1 (-s_4 s_5 s_6 + c_4 c_6) \\ a_x &= c_3 c_5 s_6 + s_3 (c_4 s_5 s_6 + s_4 c_6) \\ a_y &= c_1 (c_3 c_4 c_5 + s_3 s_5) + s_1 s_4 c_5 \\ a_z &= s_1 (c_3 c_4 c_5 + s_3 s_5) + c_1 s_4 c_5 \\ p_x &= -c_3 s_5 + s_3 c_4 c_5 \\ p_y &= c_1 [s_3 s_5 L_5 + c_3 c_4 (c_5 L_5 + L_4) + c_3 L_3 + s_2 L_2] + s_1 s_4 (c_5 L_5 + L_4) \\ p_z &= s_1 [s_3 s_5 L_5 + c_3 c_4 (c_5 L_5 + L_4) + c_3 L_3 + s_2 L_2] - c_1 s_4 (c_5 L_5 + L_4) \\ p_z &= -c_3 s_5 L_5 + s_3 c_4 (c_5 L_5 + L_4) + s_3 L_3 + c_2 L_2 \end{aligned} \quad (3.24)$$

La solución de la tarea directa requiere efectuar 50 multiplicaciones, 25 sumas y 12 llamadas a funciones trigonométricas. En total se realizan 87 operaciones matemáticas.

En la mano o agarre del robot se le asocia un sistema de coordenadas cartesianas adicional a los otros sistemas de coordenadas cartesianas asignados a cada articulación (Pozo et al., 1988). Este sistema lo conforman 3 vectores unitarios (n, o, a).

El vector a es un vector unitario perpendicular a la palma de la mano alejándose de esta.

El vector o es paralelo a la palma de la mano alejándose de esta y ortogonal al vector a.

El vector n es un vector normal que se obtiene del producto vectorial (a \* o).

Estos tres vectores son los que forman la orientación de la mano o agarre (Fig. 10).

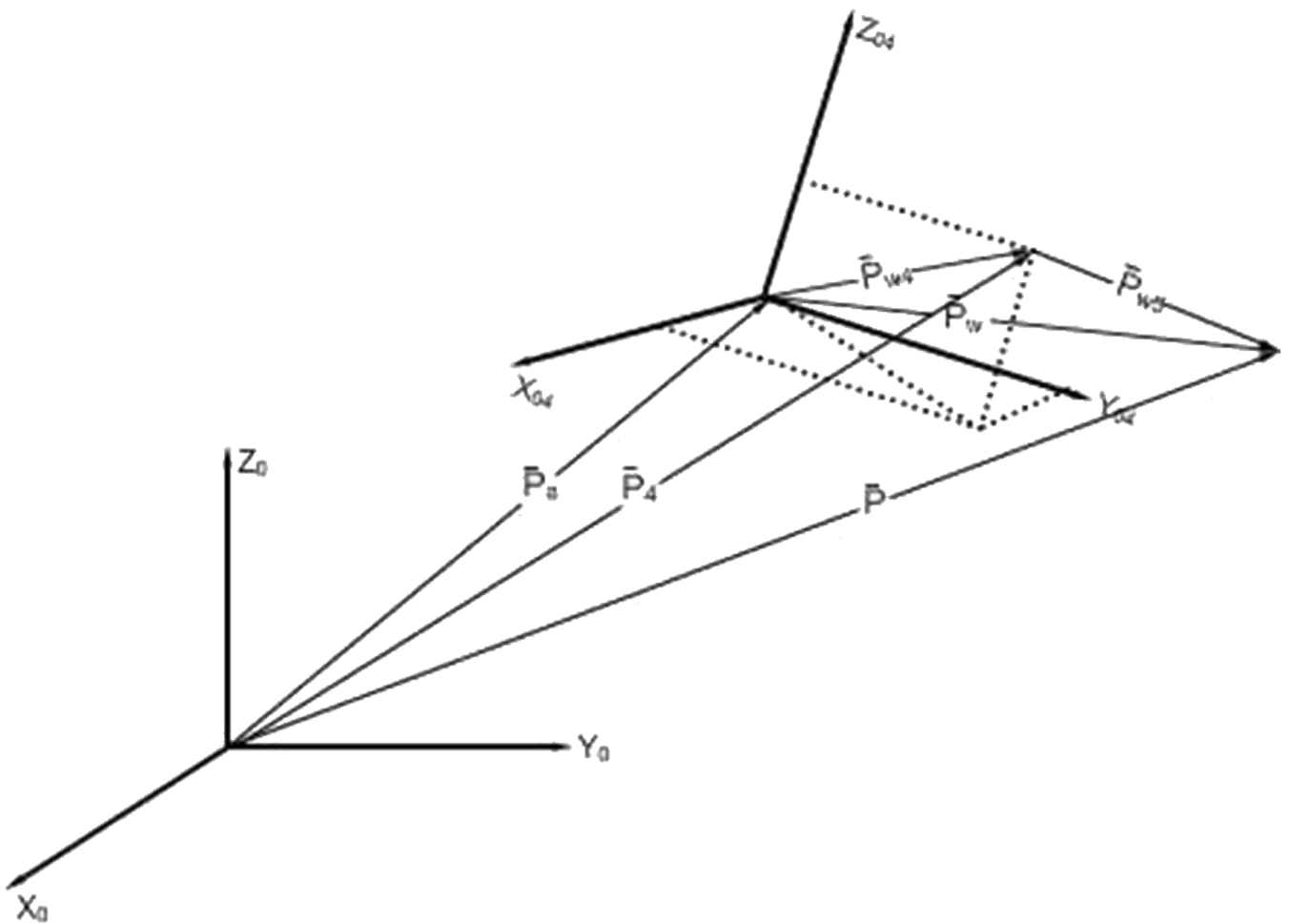


Fig. 10: Relación entre los vectores  $P_a$ ,  $P_4$  y  $P_{w4}$ .

Ahora, si se relaciona este sistema de coordenadas fijas en la mano o agarre con el sistema de coordenadas cartesianas de la base, resulta posible obtener un vector  $p$ , el cual determina la posición de la mano o agarre con respecto a la base (Fig. 9).

Estos 4 vectores se pueden relacionar en una matriz de transformación homogénea de 4 x 4 (similar a la ecuación (3.23)) compuesta de una sub matriz de orientación  $R$  y un vector de posición  $P$ , o sea:

$$T_6 = \begin{bmatrix} R_{3 \times 3} & \vdots & P_{3 \times 1} \\ \dots & \vdots & \dots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \quad (3.11)$$

Por tanto, se podría concluir que los elementos  $\underline{n}$ ,  $\underline{o}$  y  $\underline{a}$  de la ecuación (3.23) son equivalentes a los elementos de la sub matriz R en (3.25) y el elemento p de la ecuación (3.23) corresponde al vector P de la ecuación (3.25).

### 2.6 Problema cinemático inverso. (6)

El objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot,  $q = [q_1 \ q_2 \dots \ q_m]^T$ , para que su extremo se posicione y oriente según una determinada localización espacial. La resolución no es sistemática: Depende de la configuración del robot y pueden existir soluciones múltiples. Se debe intentar conseguir una solución cerrada, es decir, encontrar una relación matemática explícita de la forma:

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma)$$

$$k = 1 \dots n \text{ (GDL)}$$

Este tipo de solución presenta las siguientes ventajas:

- ✓ Posibilidad de resolución en tiempo real (seguimiento de trayectorias).
- ✓ Posibilidad de incluir restricciones que garanticen la mejor solución (por ejemplo, límite en los recorridos articulares).
- ✓ Posibilidad de simplificaciones.
- ✓ Problema: No siempre existe.

#### 2.6.1 Solución de la tarea inversa. (14)

Con el fin del control del robot, la tarea inversa es de suma importancia, debido a que los valores de las coordenadas generalizadas deben ser entregados al algoritmo de control a partir de los valores de la posición y de los puntos cartesianos de la trayectoria.

La solución simultánea de las ecuaciones en (3.24) dan los valores de las coordenadas generalizadas ( $q$ ). Sin embargo, este método además de ser muy laborioso conduce a indeterminaciones debido a que existe más de una solución para cada una de las coordenadas generalizadas (12 ecuaciones con 6 incógnitas).

Después de un estudio pormenorizado y de aplicaciones de métodos que no condujeron a resultados satisfactorios, se llegó a la conclusión de dividir el problema total en dos partes y aplicar a cada uno de ellos un método distinto.



Primeramente se separa el brazo de la mano o agarre, esto es, se calcula  $\theta_1$ ,  $\theta_2$  y  $\theta_3$  por un método trigonométrico algebraico. (Ranky et al., 1985).

En las Figs. 11 y 12 se muestran las relaciones entre los vectores del brazo y la mano de agarre.  $P$  es el vector de posición desde las coordenadas de la base al centro de la mano. El vector  $\underline{A}$  representa la dirección del eje Z en el sistema de la mano o agarre;  $P_a$  es el vector medido desde el origen al punto de conexión del brazo y la muñeca, o sea, la articulación 3, mientras que  $P_w$  es un vector que se define de la mano o agarre, teniendo una dirección diferente al vector  $\underline{A}$  y una longitud medida desde el punto en la conexión del brazo y la muñeca al centro de la mano. Todo lo planteado aquí es referido al sistema de coordenadas de la base.

Examinando la traslación y rotación del robot, el primer paso es el análisis de la traslación. Estos resultados conducirán a determinar los ángulos del brazo  $\theta_1$ ,  $\theta_2$  y  $\theta_3$ .

La traslación es:

$$[\text{Trasl. Total}] = [\text{Trasl. brazo}] + [\text{Trasl. mano}] \quad (3.26)$$

O sea:

$$P = P_a + P_w \quad (3.27)$$

$$\begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} = \begin{bmatrix} P_{ax} \\ P_{ay} \\ P_{az} \end{bmatrix} + \begin{bmatrix} P_{wx} \\ P_{wy} \\ P_{wz} \end{bmatrix} \quad (3.28)$$

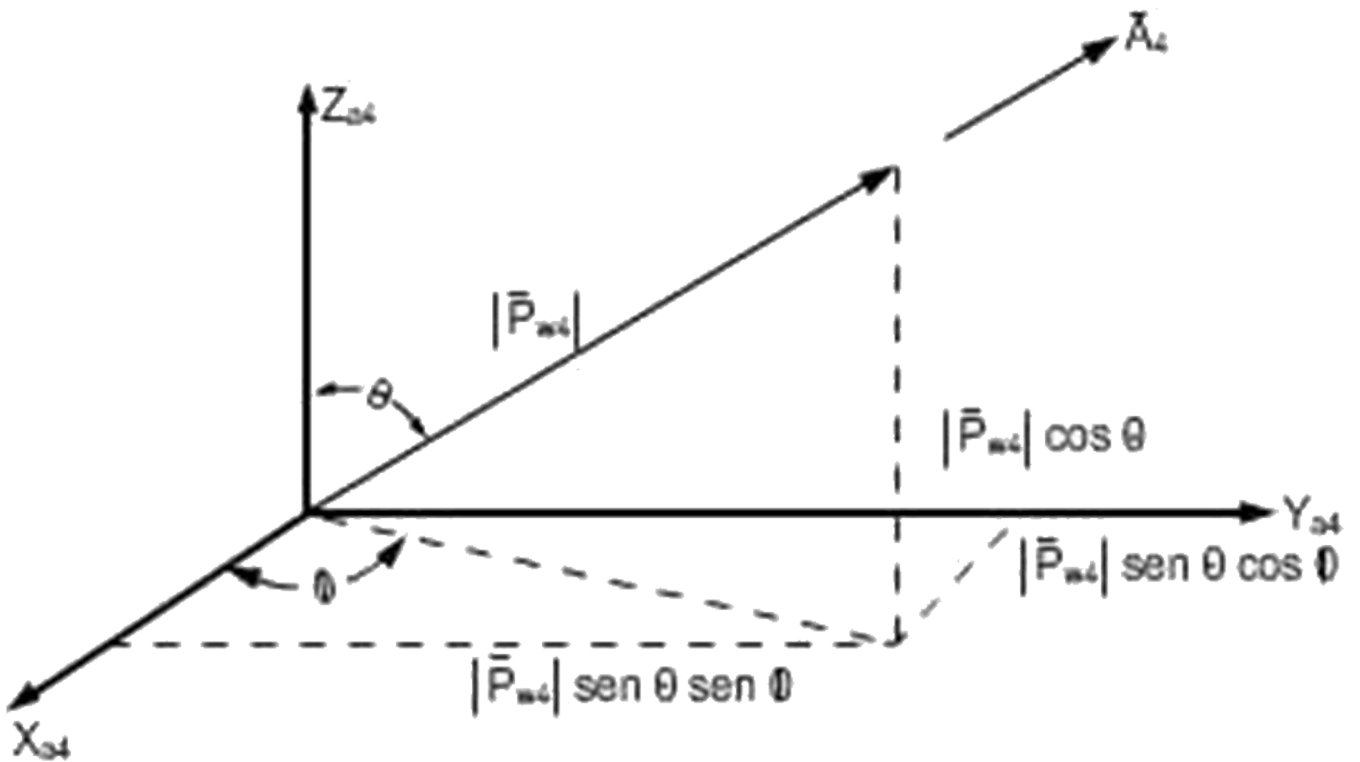


Fig. 11: Relaciones para la solución de  $P_{w4}$  en términos de  $A_4$ ,  $\phi$  y  $\theta$ .

En la Fig. 12 esta representada la relación entre los vectores  $P$ ,  $P_a$  y  $P_w$ . Los componentes de los tres vectores de  $P_w$  pueden ser expresados en términos de dirección del vector  $A$  y de la magnitud de  $P_w$ .

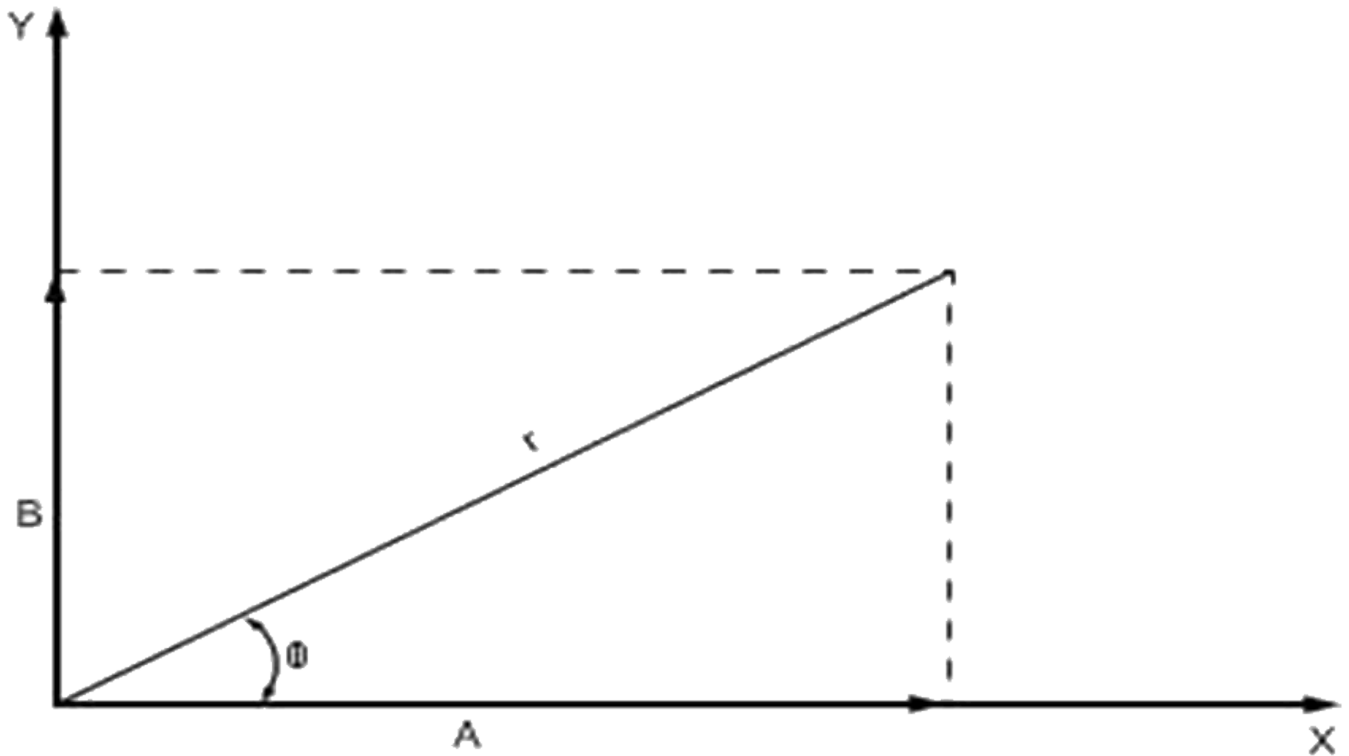


Fig. 12: Relaciones trigonométricas entre A, B, r y  $\theta$ .

De la ecuación (3.27) el vector de traslación del brazo es:

$$P_a = P - P_w \quad (3.29)$$

$$P_w = P_{w_4} - P_{w_5} \quad (3.30)$$

Así mismo de la Fig. 12 se puede observar que:

$$P_a = P_4 - P_{w_4} \quad (3.31)$$

Ahora bien, de la Fig. 11 se puede determinar la magnitud de  $P_{w_4}$  y los ángulos azimut  $\varnothing$  y polar  $\theta$  del vector  $\underline{A}$  en el cuarto sistema de coordenadas cartesianas con respecto a la base, o sea:

$$\varnothing = \tan^{-1} \left[ \frac{A_{y_4}}{A_{x_4}} \right] \quad (3.32)$$

$$\Theta = \tan^{-1} \left[ \frac{\sqrt{Ax_4^2 + Ay_4^2}}{Az_4} \right] \quad (3.33)$$

$$Pw_{4x} = |Pw_4| \sin \Theta \cos \emptyset \quad (3.34)$$

$$Pw_{4y} = |Pw_4| \sin \Theta \sin \emptyset \quad (3.35)$$

$$Pw_{4z} = |Pw_4| \cos \Theta \quad (3.36)$$

Donde:

$$|Pw_4| = L_4 \quad (3.37)$$

Y sustituyendo en la ecuación (3.31) :

$$\begin{bmatrix} Pax \\ Pay \\ Paz \end{bmatrix} = \begin{bmatrix} Px_4 \\ Py_4 \\ Pz_4 \end{bmatrix} + \begin{bmatrix} Pwx_4 \\ Pwy_4 \\ Pwz_4 \end{bmatrix} \quad (3.38)$$

Esta ecuación es conocida numéricamente, pues  $P_4$  y  $Pw_4$  están determinados.

Por otro lado:

$${}^oT\alpha = {}^oT_3 = A_1 * A_2 * A_3 \quad (3.39)$$

$${}^oT\alpha = \begin{bmatrix} c_1 s_3 & -s_1 & c_1 c_3 & c_1 (s_2 L_2 + c_3 L_3) \\ s_1 s_3 & c_1 & s_1 c_3 & s_1 (s_2 L_2 + c_3 L_3) \\ -c_3 & 0 & s_3 & c_2 L_2 + s_3 L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.40)$$

Y como resulta de interés el vector de posición dentro de esta matriz de transformación homogénea:

$$Pa = {}^oT\alpha * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 (s_2 L_2 + c_3 L_3) \\ s_1 (s_2 L_2 + c_3 L_3) \\ c_2 L_2 + s_3 L_3 \\ 1 \end{bmatrix} \quad (3.41)$$

Igualando los componentes del vector  $Pa$  con los términos de la ecuación (3.41)

$$c_1(s_2L_2 + c_3L_3) = Pax \quad (3.42)$$

$$s_1(s_2L_2 + c_3L_3) = Pay \quad (3.43)$$

$$c_2L_2 + s_3L_3 = Paz \quad (3.44)$$

Dividiendo la ecuación (3.43) entre la ecuación (3.42), se obtiene  $\theta_1$ , o sea:

$$\theta_1 = \tan^{-1} \left[ \frac{Pay}{Pax} \right] \quad (3.45)$$

Ahora elevando al cuadrado, sumando y simplificando las ecuaciones (3.42) y (3.43)

$$[c_1(s_2L_2 + c_3L_3)]^2 = Pax^2 \quad (3.46)$$

$$[s_1(s_2L_2 + c_3L_3)]^2 = Pay^2 \quad (3.47)$$

-----

$$(s_2L_2 + c_3L_3)^2 = Pax^2 + Pay^2$$

Y

$$s_2L_2 + c_3L_3 = \sqrt{Pax^2 + Pay^2} = A \quad (3.48)$$

Por lo que:

$$c_3 = (A - s_2L_2)/L_3 \quad (3.49)$$

Y despejando  $s_3$  en la ecuación (3.44)

$$s_3 = (Paz - c_2L_2)/L_3 \quad (3.50)$$

Aplicando la identidad  $s_3^2 + c_3^2 = 1$  Entonces:

$$\left[ Paz - c_2 L_2 / L_3 \right]^2 + \left[ A - s_2 L_2 / L_3 \right]^2 = 1 \quad (3.51)$$

Desarrollando la ecuación (3.51)

$$(Paz)^2 - 2Pazc_2L_2 + (c_2L_2)^2 + A^2 - 2As_2L_2 + (s_2L_2)^2 = (L_3)^2 \quad (3.52)$$

$$-2As_2L_2 - 2Pazc_2L_2 = (L_3)^2 - (L_2)^2 - (Paz)^2 - A^2 \quad (3.53)$$

$$-As_2 - Pazc_2 = \frac{[(L_3)^2 - (L_2)^2 - (Paz)^2 - (A)^2]}{2L_2} \quad (3.54)$$

Para simplificar se define  $D = \frac{[(L_3)^2 - (L_2)^2 - (Paz)^2 - (A)^2]}{2L_2}$  entonces:

$$-As_2 - Pazc_2 = D \quad (3.55)$$

La ecuación (3.55) tiene una sola variable y para una ecuación de esta forma

$$-A \sin \theta + B \cos \theta = D \quad (3.56)$$

donde:  $B = -Paz$

La solución puede ser obtenida como sigue:

$$A = r \cos \varnothing \quad (3.57)$$

$$A = r \cos \varnothing \quad (3.58)$$

entonces:

$$r = \sqrt{A^2 + B^2} \quad (3.59)$$

$$\varnothing = \tan^{-1} B/A \quad (3.60)$$

La Fig. 12 muestra las relaciones trigonométricas. Sustituyendo para A y B en la ecuación (3.56):

$$-r \cos \phi \sin \theta + r \sin \phi \cos \theta = D \quad (3.61)$$

Dividiendo por r y reagrupando

$$\sin \phi \cos \theta - \cos \phi \sin \theta = D/r \quad (3.62)$$

El miembro izquierdo de la ecuación (3.62) es  $\sin(\phi - \theta)$ ;

así:

$$\sin(\phi - \theta) = D/r \quad (3.63)$$

Puesto que  $\sin^2 \alpha + \cos^2 \alpha = 1$

$$\cos(\phi - \theta) = \pm \sqrt{1 - D^2/r^2} \quad (3.64)$$

Y por tanto:

$$\tan(\phi - \theta) = \frac{D/r}{\pm \sqrt{1 - D^2/r^2}} \quad (3.65)$$

o

$$\tan(\phi - \theta) = \pm D / \sqrt{r^2 - D^2} \quad (3.66)$$

Sustituyendo para r de la ecuación (3.59):

$$\tan(\varphi - \theta) = \pm D / \sqrt{A^2 + B^2 - D^2} \quad (3.67)$$

entonces:

$$\varphi - \theta = \tan^{-1} \left[ \pm D / \sqrt{A^2 + B^2 - D^2} \right] \quad (3.68)$$

Ahora sustituyendo  $\varphi$  de la ecuación (3.60) en (3.68)

$$\theta = \tan^{-1} \left[ B/A \right] - \tan^{-1} \left[ \pm D / \sqrt{A^2 + B^2 - D^2} \right] \quad (3.69)$$

Entonces para la ecuación (3.55) la solución de la ecuación (3.69) se determina  $\theta_2$ , o sea:

$$\theta_2 = \tan^{-1} \left[ B/A \right] - \tan^{-1} \left[ \pm D / \sqrt{A^2 + B^2 - D^2} \right] \quad (3.70)$$

Y dividiendo la ecuación (3.50) entre la ecuación (3.49), se obtiene  $\theta_3$ , por lo que:

$$\theta_3 = \tan^{-1} \left[ Paz - c_2 L_2 / A - s_2 L_2 \right] \quad (3.71)$$

Ahora bien, para determinar los ángulos  $\theta_4$ ,  $\theta_5$  y  $\theta_6$  se empleará un método desarrollado por Paul et al (1981<sup>a</sup>).

Hay n-1 ecuaciones matriciales ( $U_i$ ) obtenidas en la tarea directa y pre multiplicando por  $A_i^{-1}$  ( $i = 1, \dots, n-1$ ) la matriz  $T_n$ .

$$A_1^{-1} * T_6 = U_2 \quad (3.72)$$

$$A_2^{-1} * A_1^{-1} * T_6 = U_3 \quad (3.73)$$

$$A_3^{-1} * A_2^{-1} * A_1^{-1} * T_6 = U_4 \quad (3.74)$$

$$A_4^{-1} * A_3^{-1} * A_2^{-1} * A_1^{-1} * T_6 = U_5 \quad (3.75)$$

$$A_5^{-1} * A_4^{-1} * A_3^{-1} * A_2^{-1} * A_1^{-1} * T_6 = U_6 \quad (3.76)$$



Estas ecuaciones se pueden igualar a las obtenidas en las ecuaciones (3.18 – 3.22). Por lo tanto, de la ecuación (3.72)

$$A_1^{-1} * T_6 = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.77)$$

La inversa de la matriz de transformación homogénea es simple de obtener. El producto de estas dos matrices es:

$$A_1^{-1} * T_6 = \begin{bmatrix} F_{11}(n) & F_{11}(o) & F_{11}(a) & F_{11}(p) \\ F_{12}(n) & F_{12}(o) & F_{12}(a) & F_{12}(p) \\ F_{13}(n) & F_{13}(o) & F_{13}(a) & F_{13}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix} = U_2 \quad (3.78)$$

donde:

$$F_{11} = c_1x + s_1y \quad (3.79)$$

$$F_{12} = -s_1x + c_1y \quad (3.80)$$

$$F_{13} = z \quad (3.81)$$

(x, y, z) se refiere a los componentes de los vectores dados como argumentos a  $F_{11}$ ,  $F_{12}$  y  $F_{13}$ , por ejemplo:

$$F_{11}(p) = c_1px + s_1py \quad (3.82)$$

Como se han determinado los tres primeros ángulos por otro método, es por lo que en la ecuación (3.78) no es posible evaluar nada. La práctica aconseja resolver de la primera ecuación la primera coordenada generalizada  $q_1$  y si es posible alguna más, y así sucesivamente. En este caso será necesario analizar  $U_4$ , pero primero hay que calcular  $U_3$ , o sea, evaluando la ecuación (3.73)

$$A_2^{-1} * U_2 = \begin{bmatrix} F_{21}(n) & F_{21}(o) & F_{21}(a) & F_{21}(p) \\ F_{22}(n) & F_{22}(o) & F_{22}(a) & F_{22}(p) \\ F_{23}(n) & F_{23}(o) & F_{23}(a) & F_{23}(p) - L_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = U_3 \quad (3.83)$$

donde:

$$F_{21} = c_2(c_1x + s_1y) - s_2z = c_2F_{11} - s_2F_{13} \quad (3.84)$$

$$F_{22} = -s_1x + c_1y = F_{12} \quad (3.85)$$

$$F_{23} = s_2(c_1x + s_1y) + c_2z = s_2F_{11} + c_2F_{13} \quad (3.86)$$

Y evaluando la ecuación (3.74)

$$A_3^{-1} * U_3 = \begin{bmatrix} F_{31}(n) & F_{31}(o) & F_{31}(a) & F_{31}(p) \\ F_{32}(n) & F_{32}(o) & F_{32}(a) & F_{32}(p) \\ F_{33}(n) & F_{33}(o) & F_{33}(a) & F_{33}(p) - s_{23}L_2 - L_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = U_4 \quad (3.87)$$

donde:

$$F_{31} = s_3(c_1x + s_1y) - c_3z = s_3F_{11} - c_3F_{13} \quad (3.88)$$

$$F_{32} = -s_1x + c_1y = F_{22} \quad (3.89)$$

$$F_{33} = c_3(c_1x + s_1y) + s_3z = c_3F_{11} + s_3F_{13} \quad (3.90)$$

Igualando los elementos (2,4) y (3,4) de  $U_4$ , se calcula  $\theta_4$ , por lo que:

$$s_4(c_5L_5 + L_4) = -s_1px + c_1py = F_{32}(p) \quad (3.91)$$

$$c_4(c_5L_5 + L_4) = c_3F_{11}(p) + s_3F_{13}(p) - s_{23}L_2 - L_3 = F_{33}(p) - s_{23}L_2 - L_3 \quad (3.92)$$

$$\theta_4 = \tan^{-1} \left[ \frac{-F_{32}(p)}{F_{33}(p) - s_{23}L_2 - L_3} \right] \quad (3.93)$$

Y por ultimo evaluando la ecuación (3.75), se obtienen los ángulos  $\theta_5$  y  $\theta_6$ , o sea:

$$A_4^{-1} * U_4 = \begin{bmatrix} F_{41}(n) & F_{41}(o) & F_{41}(a) & F_{41}(p) - c_{23}L_2 \\ F_{42}(n) & F_{42}(o) & F_{42}(a) & F_{42}(p) - s_4(s_{23}L_2 + L_3) \\ F_{43}(n) & F_{43}(o) & F_{43}(a) & F_{43}(p) - c_4(s_{23}L_2 + L_3) - L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = U_5 \quad (3.94)$$

donde:

$$F_{41} = s_3(c_1x + c_1y) - c_3z = F_{31} \quad (3.95)$$

$$F_{42} = c_4(-s_1x + c_1y) + s_4[c_3(c_1x + s_1y) + s_3z] = c_4F_{32} + s_4F_{33} \quad (3.96)$$

$$F_{43} = s_4(-s_1x + c_1y) + c_4[c_3(c_1x + s_1y) + s_3z] = -s_4F_{32} + c_4F_{33} \quad (3.97)$$

Si de  $U_5$  se igualan los elementos (1,3) y (3,3) se determina  $U_5$  y los elementos (2,1) y (2,2) para hallar  $U_6$

$$s_5 = F_{41}(a) \quad (3.98)$$

$$c_5 = -s_4F_{32}(a) + c_4F_{33}(a) = F_{43}(a) \quad (3.99)$$

$$\theta_5 = \tan^{-1} F_{41}(a) / F_{43}(a) \quad (3.100)$$

y

$$s_6 = c_4F_{32}(n) + s_4F_{33}(n) = F_{42}(n) \quad (3.101)$$

$$c_6 = c_4F_{32}(o) + s_4F_{33}(o) = F_{42}(o) \quad (3.102)$$

$$\theta_6 = \tan^{-1} F_{42}(n) / F_{42}(o) \quad (3.103)$$

Es necesario aclarar que en el calculo de  $\theta_4$  se hubieran podido coger los elementos (2,3) y (3,3) de  $U_4$ , pero esto trae como consecuencia que la configuración del robot se convierta en degenerado (no corresponde a su origen) para  $\theta_5 = 90^\circ$  y  $\theta_5 = -90^\circ$ , y como consecuencia habría que sumar o restar  $180^\circ$  a  $\theta_4$ , lo que redundaría en más comparaciones y operaciones computacionales.

## 2.7 Problema cinemático en la práctica.

Muy a menudo sucede que en la práctica, los cálculos y ecuaciones definidas teóricamente no se corresponden de manera satisfactoria con la realidad, y eso se debe a consideraciones mecánicas del brazo manipulador principalmente, que son necesarias tener en cuenta. Es por ello que se dedica este epígrafe a comentar las acciones que fueron realizadas con vistas a lograr conectar la teoría con la realidad.

Primeramente es necesario destacar que, debido a que el robot manipulador con el que se ha trabajado es accionado de forma secuencial por motores de paso, la tarea directa deja de ser un objetivo principal en esta teoría, siendo necesaria solamente para asignar valores a variables que posteriormente serán utilizadas.

Por otro lado, la teoría fue desarrollada específicamente para otro robot manipulador que, aún con características similares al presentado, posee sus particularidades. La diferencia principal está dada en el número de sensores que posee el brazo manipulador con el cual se trabajó, pues, en teoría, el desarrollo matemático fue pensado para un robot con sensores en la muñeca mediante los cuales se podría obtener datos sobre tres parámetros específicamente, sin los cuales se dificulta la exactitud del movimiento del mismo. Este robot manipulador no posee estos dispositivos, y debido a la carencia del hardware necesario para incorporarlos, fue preciso encontrar una solución alternativa para resolver el problema que se plantea.

Por tanto, contando con la información que brinda el brazo manipulador acerca de su posición en el espacio, fue necesario dirigirse por un método trigonométrico alternativo. Primeramente se toma el ángulo de la pinza o muñeca con respecto a la horizontal como constante, con el objetivo de posibilitar una mejor visión de la cámara (sensor con el que se cuenta en este caso en particular para la ubicación del objetivo a alcanzar).

Luego, ya teniendo esta información del ángulo final, el punto en el espacio  $(x, y, z)$  y la distancia desde el punto inicial  $(0, 0, 0)$  hasta el punto en el espacio que se desea alcanzar es posible encontrar una solución para los demás ángulos de la siguiente forma:

Se tiene que el ángulo  $\alpha_1$  es conocido  $=30^\circ$ , además se conoce que  $W_f = \sqrt{x^2 + y^2}$  y como es conocido  $(x, y, z)$  final, el valor de  $W_f$  es numéricamente conocido. Se tiene también los valores de los brazos del manipulador, los cuales están representados en la Fig. 14 como  $B_1, B_2, B_3$  y  $B_4$ . Luego se calcula el valor de  $L_1$ :

$$L_1 = (B_4 * \sin \alpha_1) + W_f$$

Y luego se calcula el valor de  $L_2$ :

$$L_2 = B_4 * \cos \alpha_1$$

Ahora, con el objetivo de calcular el valor de  $\alpha_2$  se hace necesario calcular cada una de sus partes:

$$\alpha_{2a} = 90^\circ - \alpha_1$$

Seguidamente se encuentra el valor de la recta  $d_1$ , que no es más que la distancia existente entre el punto de origen del sistema hasta la última articulación del brazo manipulador.

$$d_1 = \sqrt{(W_f - L_2)^2 + L_1^2}$$

Una vez conocida la distancia  $d_1$  se calcula el valor del ángulo  $\alpha_{5a}$  y  $\alpha_{5b}$  respectivamente.

$$\alpha_{5a} = \sin^{-1} L_1 / d_1$$

$$\alpha_{5b} = 90^\circ - \alpha_{5a}$$

Con el objetivo de facilitar la búsqueda de los demás ángulos, se calcula la recta  $d_2$ , la cual es la distancia entre la primera y la última articulación del robot manipulador.

Según la Ley de Los Cosenos:

$$d_2 = \sqrt{d_1^2 + B_1^2 - 2 * d_1 * B_1 * \cos \alpha_{5b}}$$

Ahora, dándole solución a los demás ángulos que forman a  $\alpha_2$ :

$$\alpha_{2b} = 90^\circ - \alpha_{5a}$$

Ahora despejando en la Ley de los Cosenos:

$$\alpha_{2c} = \cos^{-1} \left( \frac{d_1^2 + d_2^2 - B_1^2}{2 * d_1 * d_2} \right)$$

$$\alpha_{2d} = \cos^{-1} \left( \frac{B_3^2 + d_2^2 - B_2^2}{2 * B_3 * d_2} \right)$$

Ahora, para buscar el valor del  $\alpha_2$ , se suman todos los ángulos que lo conforman.

$$\alpha_2 = \alpha_{2a} + \alpha_{2b} + \alpha_{2c} + \alpha_{2d}$$

Encontrando  $\alpha_3$ :

$$\alpha_3 = \cos^{-1} \left( \frac{B_3^2 + B_2^2 - d_2^2}{2 * B_3 * B_2} \right)$$

Por último se calculan los ángulos que conforman a  $\alpha_4$  y se suman.

$$\alpha_{4a} = 180^\circ - (\alpha_{5b} + \alpha_{2c})$$

$$\alpha_{4b} = 180^\circ - (\alpha_3 + \alpha_{2d})$$

$$\alpha_4 = \alpha_{4a} + \alpha_{4b}$$

Finalmente se calcula  $\alpha_6$  el cual representa el primer grado de libertad del sistema como se ve en la Fig. 13 del epígrafe.

$$\alpha_6 = \tan^{-1} y/x$$

Ahora, para ajustar los ángulos encontrados al grado de libertad que posee el Armdroid1 en la realidad, es necesario hacer un análisis técnico de dicho robot manipulador y llevarlo a la teoría, por lo que luego de realizado dicho análisis, queda que:

$$\alpha_1 = 30^\circ$$

$$\alpha_2 = 270^\circ - \alpha_2$$

$$\alpha_3 = \alpha_3 - 30^\circ$$

$$\alpha_4 = 225^\circ - \alpha_4$$

$$\alpha_5 = 90^\circ$$

$$\alpha_6 = \alpha_6$$

Donde quedan establecidos límites mecánicos que acotan de forma tanto inferior como superior los ángulos encontrados:

$$0^\circ \leq \alpha_2 \leq 180^\circ$$

$$30^\circ \leq \alpha_3 \leq 180^\circ$$

$$30^\circ \leq \alpha_4 \leq 225^\circ$$

$$0^\circ \leq \alpha_6 \leq 180^\circ$$

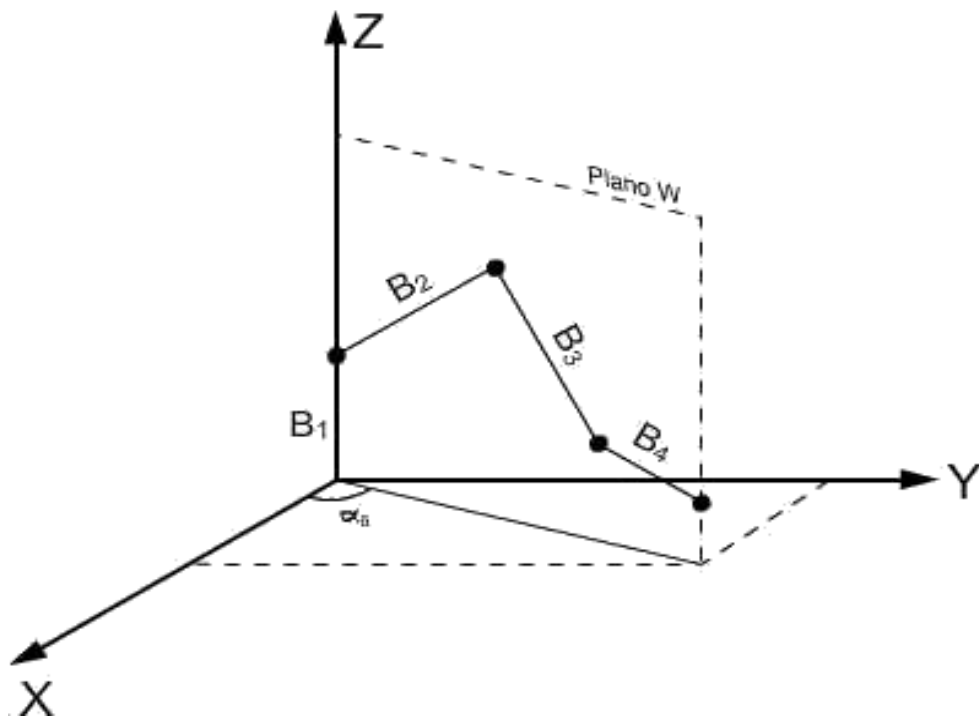


Fig. 13 Representación del Brazo manipulador en las tres dimensiones.

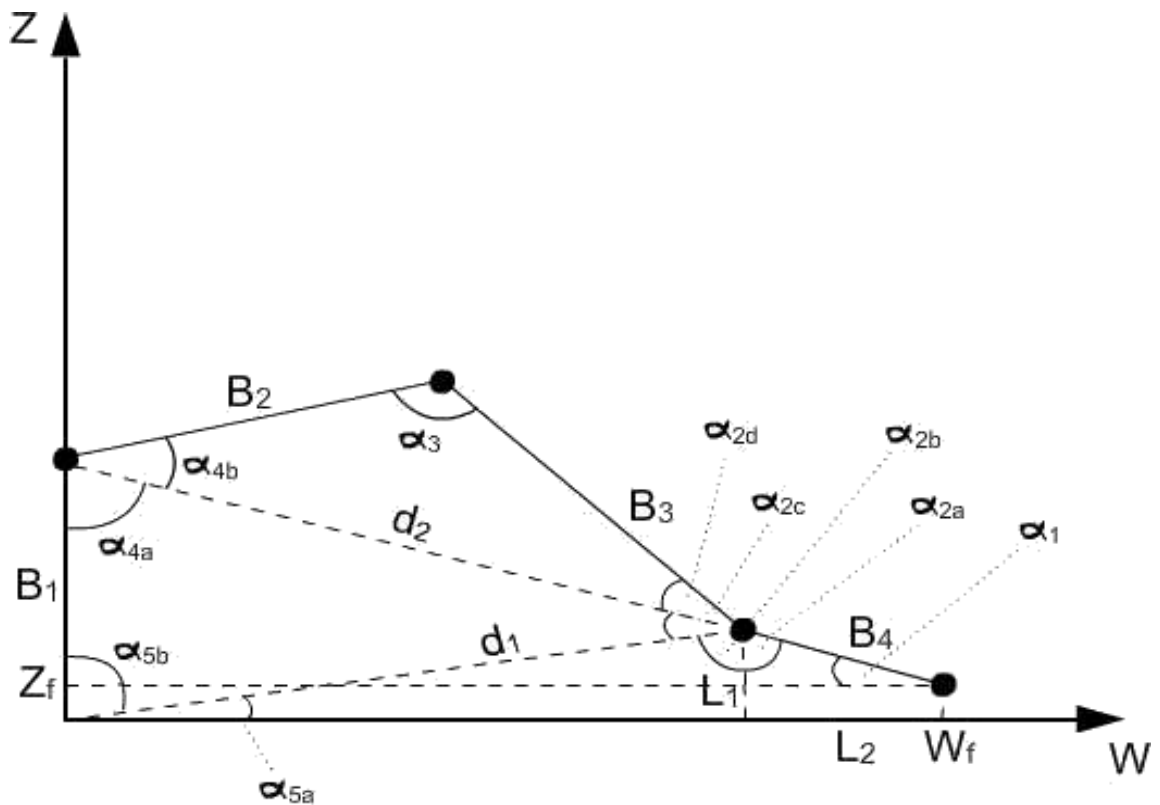


Fig. 14 Representación del brazo manipulador en el plano  $W$ .

## 2.8 Conclusiones

En el transcurso de este capítulo se dio a conocer la teoría matemática en la cual se basa el sistema, y el desarrollo cinemático que se tuvo en cuenta para la implementación del mismo, también se realizó una investigación con la intención del esclarecimiento del problema cinemático inverso.



### 3 Capítulo 3 Características del sistema.

#### 3.1 Introducción:

En este capítulo se realiza el análisis del proceso de desarrollo del software. Este proceso de desarrollo se centró en la metodología RUP, haciendo uso del UML para la modelación de los artefactos. Como herramienta CASE para asistir el proceso de desarrollo fue utilizado el Rational Rose.

Se presenta el modelo de dominio, como alternativa al modelo de negocio, para entender el contexto en que se ubica el sistema.

Se presentan los requerimientos funcionales y no funcionales especificados por el usuario. Se describen los casos de uso y la propuesta del sistema a desarrollar.

#### 3.2 Negocio (Modelo de dominio):

Dada la relación entre los conceptos fundamentales del sistema se obtiene el siguiente modelo de dominio:

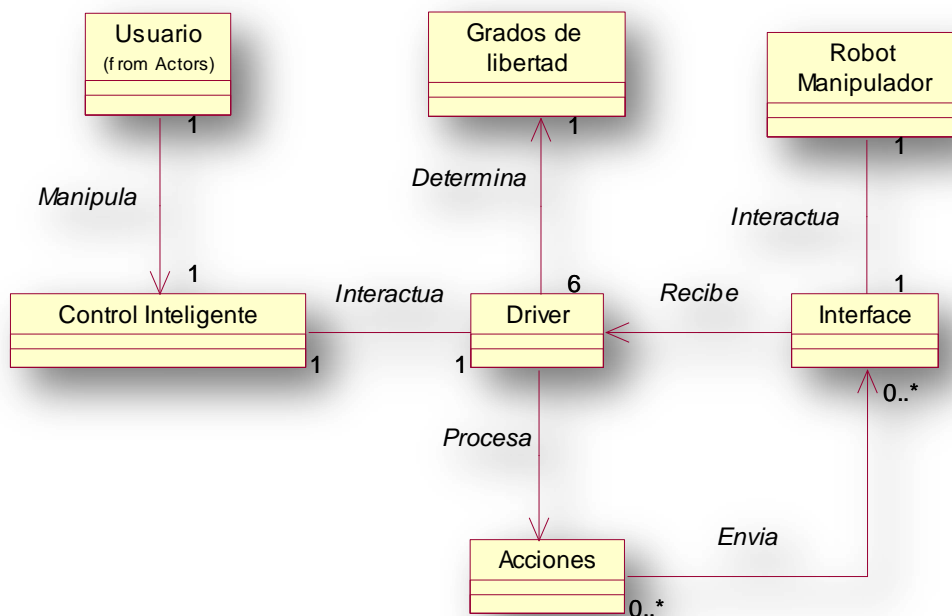


Fig. 15 Modelo de Dominio

### 3.2.1 Conceptos:

✓ **Usuario:**

Persona encargada de la configuración y/o supervisión de la aplicación.

✓ **Control Inteligente:**

Sistema que tiene la habilidad para actuar de forma apropiada en un entorno incierto y que hace uso del sistema.

✓ **Grados de Libertad:**

Se denomina grados de libertad de un mecanismo, al número de parámetros que es necesario conocer para determinar su posición. Con este concepto se hace referencia a la obtención de los ángulos de las articulaciones.

✓ **Driver:**

Un driver técnicamente, no es más que un software o programa que sirve de intermediario entre un dispositivo de hardware y el sistema operativo. Su finalidad y diseño es la de permitir extraer el máximo de las funcionalidades del dispositivo para el cual ha sido diseñado (15). Archivo o archivos que permiten que un sistema operativo entienda y maneje diversos periféricos y hardwares que se conectan o forman parte de la computadora. Sistemas operativos como Windows suelen tener una gran base de controladores para poder ser compatibles con diversos hardwares, pero muchas veces es necesario instalar otros controladores para poder hacerlos funcionar correctamente. (16)

✓ **Acciones:**

Secuencia de instrucciones enviadas (procesadas) por el driver hacia la interfaz del sistema. Con ello se hace referencia a los comandos de bajo nivel los cuales la interfaz puede interpretar.

✓ **Interface:**

Hardware creado específicamente para el control del movimiento de un robot manipulador.

✓ **Robot Manipulador:**

Un robot, es un agente artificial mecánico o virtual. Es una máquina usada para realizar un trabajo automáticamente y que es controlada por una computadora.

Si bien la palabra robot puede utilizarse para agentes físicos y agentes virtuales de software, estos últimos son llamados "bots" para diferenciarlos de los otros.

En general, un robot, para ser considerado como tal, debería presentar algunas de estas propiedades:

- ✓ No es natural, sino que ha sido creado artificialmente.
- ✓ Puede sentir su entorno.
- ✓ Puede manipular cosas de su entorno.
- ✓ Tiene cierta inteligencia o habilidad para tomar decisiones basadas en el ambiente o en una secuencia pre-programada automática.
- ✓ Es programable.
- ✓ Puede moverse en uno o más ejes de rotación o traslación.
- ✓ Puede realizar movimientos coordinados. (16)

### 3.2.2 Requerimientos del sistema:

#### 3.2.2.1 Requisitos funcionales:

##### R<sub>1</sub>: Inicializar robot manipulador.

R<sub>1.1</sub>: Inicializar valores de las longitudes de los brazos por defecto.

R<sub>1.2</sub>: Reconfigurar los valores de las longitudes de los brazos del manipulador.

R<sub>1.3</sub>: Validar las dimensiones de los brazos del robot.

##### R<sub>2</sub>: Comunicar PC-Robot

R<sub>2.1</sub>: Codificar del datagrama para el envío.

R<sub>2.2</sub>: Descodificar del datagrama recibido.

##### R<sub>3</sub>: Tratamiento de Errores

R<sub>3.1</sub>: Comprobación de errores de comunicación.

R<sub>3.2</sub>: Tratamiento de errores de comunicación.

##### R<sub>4</sub>: Determinar libertades.

R<sub>4.1</sub>: Determinación del modelo matemático óptimo.

R<sub>4.2</sub>: Optimización de la velocidad.

R<sub>4.3</sub>: Tratamiento de errores de hardware.

### 3.2.2.2 *Requisitos no funcionales*

Los requerimientos no funcionales no son más que las propiedades o cualidades que el producto debe tener, estos se clasifican por categorías, ya sea de software, hardware, apariencia o interfaz interna, seguridad, portabilidad y de usabilidad. Los mismos se identifican a continuación:

#### **Requisitos de software:**

- ✓ El sistema operativo a utilizar es Windows XP, se necesitan tener otros programas instalados como .NET framework 2.0.

#### **Requisitos de Usabilidad**

- ✓ El driver ha de ser utilizado por personas con conocimientos básicos de robótica.
- ✓ Permitir fácil manejo, con la capacidad de brindar confianza y seguridad en su uso.

#### **Requisitos de Hardware:**

- ✓ Como requisitos de hardware se requiere un disco duro de 10 GB o superior, 128 megabytes (MB) de memoria RAM o más, procesador de tipo Pentium III a 1.0 GHz o superior, con interfaz RS-232.
- ✓ Se requiere de una interfaz de hardware capaz de controlar hasta 6 motores paso a paso.
- ✓ Es necesario un robot manipulador de 6 grados de libertad.
- ✓ El sistema deberá comunicarse a través del puerto COM1 utilizando el protocolo RS32.

### **3.3 Modelo del Sistema**

Los actores representan los usuarios del sistema y otras aplicaciones que interactúan con él, es decir, representan terceros fuera del sistema que guardan relación con el mismo. Estos suelen corresponderse con trabajadores o actores del negocio. Debido a las características particulares del sistema solamente hay un actor, el Usuario. Los casos de usos definidos se relacionan exclusivamente con este actor.

### 3.3.1 Definición de los actores.

Tabla 3.1: Actores.

Actores	Justificación
<b>Usuario</b>	Se encarga de entrar los datos para el sistema, pedir información y realizar las acciones.

Tabla 3.2: Casos de Uso.

Nombre CU	Propósito	RF
<b>CU_Inicializar Robot Manipulador</b>	Permite Configurar los valores al Robot manipulador	R <sub>1</sub>
<b>CU_Comunicar PC-Robot</b>	Permite el envío de los datos calculados a robot manipulador y procesas los errores que puedan ocurrir.	R <sub>2</sub> ,R <sub>4</sub>
<b>CU_Tratar Errores.</b>	Permite el tratar de los errores ocurridos durante la transmisión.	R <sub>3</sub>
<b>CU_Mover Robot.</b>	Permite realizar los cálculos para determinar los ángulos necesarios de las articulaciones.	R <sub>4</sub>

Diagrama de casos de uso del sistema.

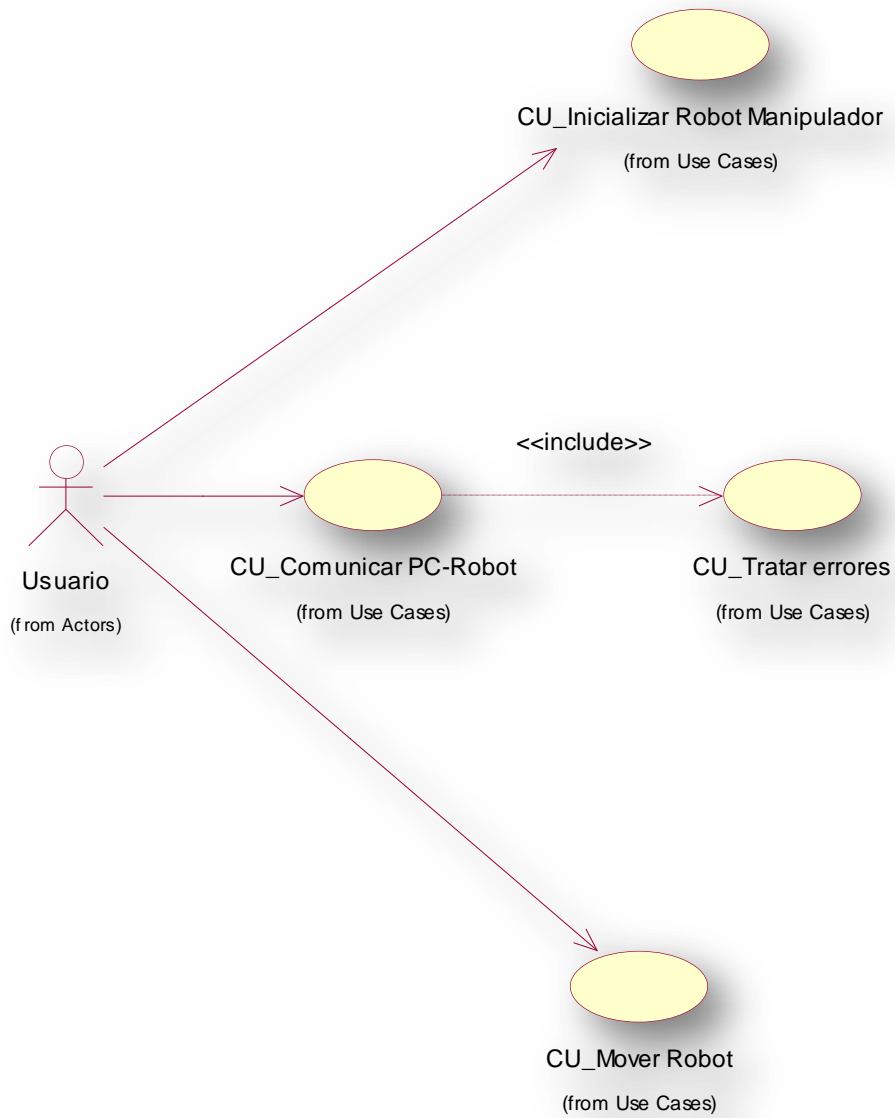


Fig. 16 Modelo de Casos de Uso del sistema.

**3.3.2 Descripción de los Casos de Uso del Sistema**

<b>CUS1</b>	CU_Inicializar Robot Manipulador
<b>Actores</b>	Usuario
<b>Propósito</b>	Configurar las dimensiones de los brazos del robot manipulador y llevar el mismo a la posición inicial.
<b>Resumen:</b>	El caso de uso se inicia una vez que se inicialice el sistema y cuando el usuario desee llevar al robot manipuladora la posición inicial.
<b>Casos de uso asociados</b>	
<b>Referencias</b>	R <sub>1</sub>
<b>Precondiciones</b>	El usuario debe entrar las dimensiones de los brazos o debe ordenar al robot manipulador ir a la posición inicial.
<b>Pos condiciones</b>	Se debe configurar las dimensiones de los brazos, o el robot manipulador toma la posición inicial.
<b>Flujo Normal de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. Entra las dimensiones de los brazos	1.1 Revisa los valores que se han entrado. 1.2 Se cambian las dimensiones. 1.3 Se devuelve un mensaje de que se ha realizado con éxito la operación.
2. Se ordena al robot ir a la posición inicial.	2.1 Se le envía al driver el comando de buscar la posición inicial. 2.2 El driver envía a la interfaz la orden de llevar al robot a la posición inicial. 2.3 El manipulador envía un mensaje de que ha llegado a su posición inicial.
<b>Flujo alternativo“”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
Prototipo	

<b>CUS2</b>	CU_Comunicar PC-Robot
<b>Actores</b>	Usuario
<b>Propósito</b>	Establecer una comunicación entre el usuario y el robot manipulador para enviar los datos obtenidos.
<b>Resumen:</b>	El caso de uso se inicia una vez que el usuario haya enviado una solicitud de movimiento, y esta información sea necesaria hacerla llegar al robot manipulador.
<b>Casos de uso asociados</b>	CUS3
<b>Referencias</b>	R <sub>2</sub> ,R <sub>4</sub>
<b>Precondiciones</b>	El usuario debe tener la información que va enviar, debe existir un canal físico entre el robot manipulador y la interfaz y esta a su vez con la computadora.
<b>Pos condiciones</b>	Se obtiene una comunicación entre el usuario y el robot manipulador.
<b>Flujo Normal de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. Enviar la información por el canal.	1.1 El driver comprueba la conectividad. 1.2 El driver codifica la información. 1.3 El driver envía la información codificada. 1.4 El driver espera mensaje de confirmación de recepción. 1.5 El driver obtiene la información de estado enviado por el robot manipulador.
<b>Flujo alternativo“”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. Enviar la información por el canal.	1.1 El sistema debe esperar para el siguiente envío por el mensaje de confirmación de recepción. 1.2 En caso de error este es tratado.
Prototipo	



<b>CUS3</b>	CU_Tratar Errores.<<include>>	
<b>Actores</b>	Usuario	
<b>Propósito</b>	Tratar los errores y evitar que los datos utilizados por el robot manipulador estén incorrectos.	
<b>Resumen:</b>	El caso de uso se inicia una vez que el usuario haya enviado o recibido algún dato.	
<b>Casos de uso asociados</b>		
<b>Referencias</b>	R <sub>3</sub>	
<b>Precondiciones</b>	El usuario debe haber enviado algún dato.	
<b>Pos condiciones</b>	Se obtiene información sobre los errores ocurridos en el proceso de comunicación con el robot manipulador.	
	<b>Flujo Normal de eventos</b>	
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1. Envío de información por el canal.	1.1 El driver obtiene la información de estado enviado por el robot manipulador. 1.2 Comprueba errores en los datos de haber errores envía un mensaje de error de envío e incrementa el contador de errores de envío. 1.3 Comprueba el número de errores de envíos si supera a 10 envía un error de hardware y receta el contador de errores.
	<b>Flujo alternativo“”</b>	
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<b>Prototipo</b>		

<b>CUS4</b>	CU_Mover Robot.
<b>Actores</b>	Usuario
<b>Propósito</b>	Determinar los parámetros (ángulos) necesarios para posicionar la mano de sujeción y tomar el objeto deseado. Resuelve la tarea cinemática inversa.
<b>Resumen:</b>	El caso de uso se inicia una vez que el usuario envía la solicitud de mover el brazo.
<b>Casos de uso asociados</b>	
<b>Referencias</b>	R <sub>4</sub>
<b>Precondiciones</b>	El usuario debe haber entrado las coordenadas del objeto deseado.
<b>Pos condiciones</b>	Se obtienen los ángulos de las articulaciones del robot.
<b>Flujo Normal de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. Solicitud de mover el brazo.	1.1 Determina si las coordenadas están dentro del área de trabajo. 1.2 El driver determina mediante la tarea cinemática inversa los ángulos del robot. 1.3 Considera el posible error introducido por el hardware.
<b>Flujo alternativo“”</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
Prototipo	

### **3.4 Conclusiones.**

En el capítulo se muestran las características particulares que debe presentar el software para cumplir con los requerimientos especificados por el cliente. Se realiza el modelo del dominio, se definen los requisitos funcionales y no funcionales que debe cumplir y finalmente se elaboró el diagrama de casos de uso del sistema.

### **4 Capítulo 4 Diseño del sistema.**

#### **4.1 Introducción**

En este capítulo se define como se va a desarrollar la aplicación, a través de los artefactos del diseño, aplicando los principios del mismo.

#### **4.2 Diseño**

##### **Diagramas de Interacción**

Los diagramas de interacción no son más que una descripción del modo en el que cada operación detectada en los diagramas de secuencia lleva a cabo sus responsabilidades y modifica el estado del sistema. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia.

El tipo de diagrama seleccionado para construir los diagramas de interacción fue el de Secuencia, debido a que muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente. (17)

El diagrama de secuencias es el núcleo de un modelo dinámico, y muestra todos los cursos alternos que pueden tomar los casos de uso. Los diagramas de secuencias se componen de 4 elementos que son: el curso de acción, los objetos, los mensajes y los métodos.

#### **4.3 Diagrama de clases del diseño**

El diagrama de clases del sistema contiene la colección de los elementos estáticos y dinámicos del sistema, como clases, tipos y sus relaciones, conectados unos a otros y a sus contenidos.

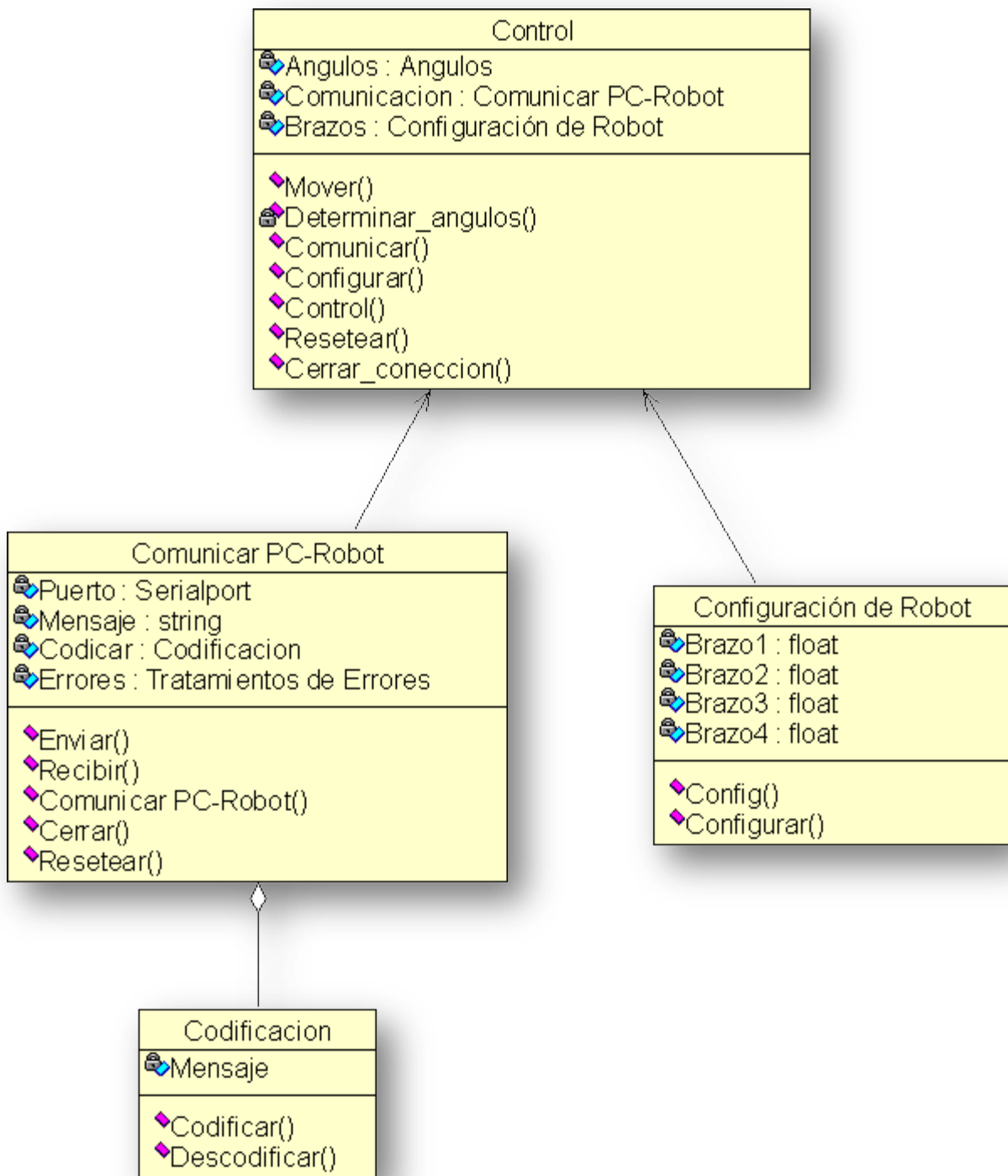


Fig. 17 Diagrama de clases de diseño CU\_Inicializar Robot Manipulador

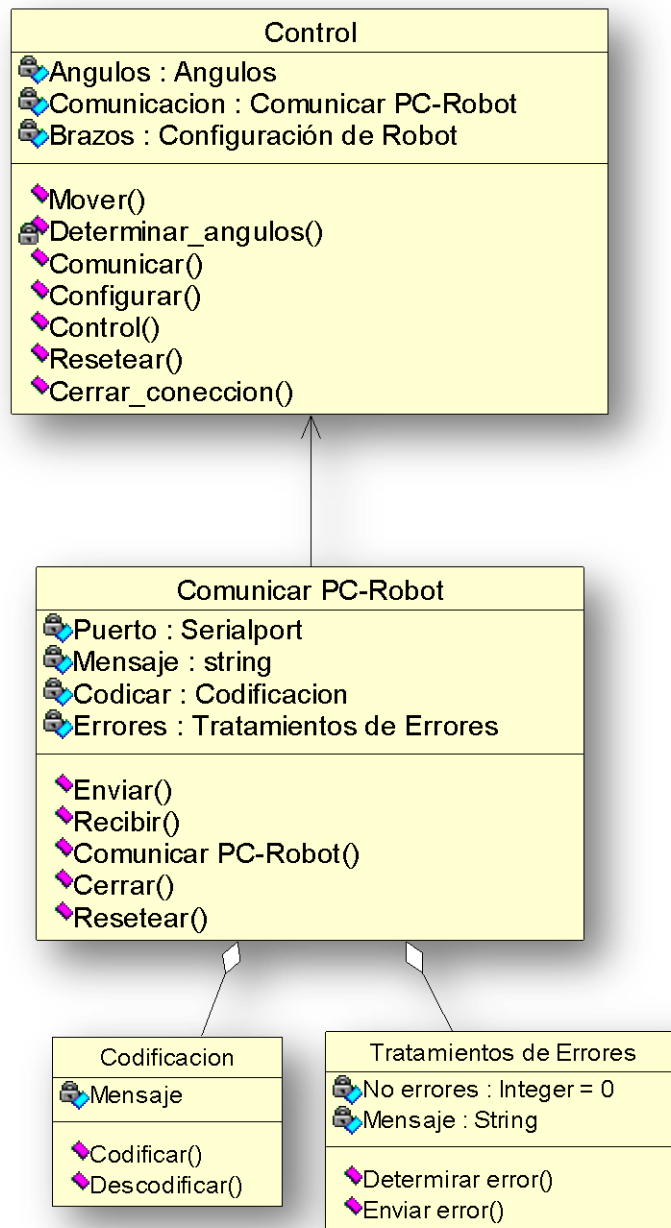


Fig. 18 Diagrama de clases de diseño CU\_Comunicar PC-Robot

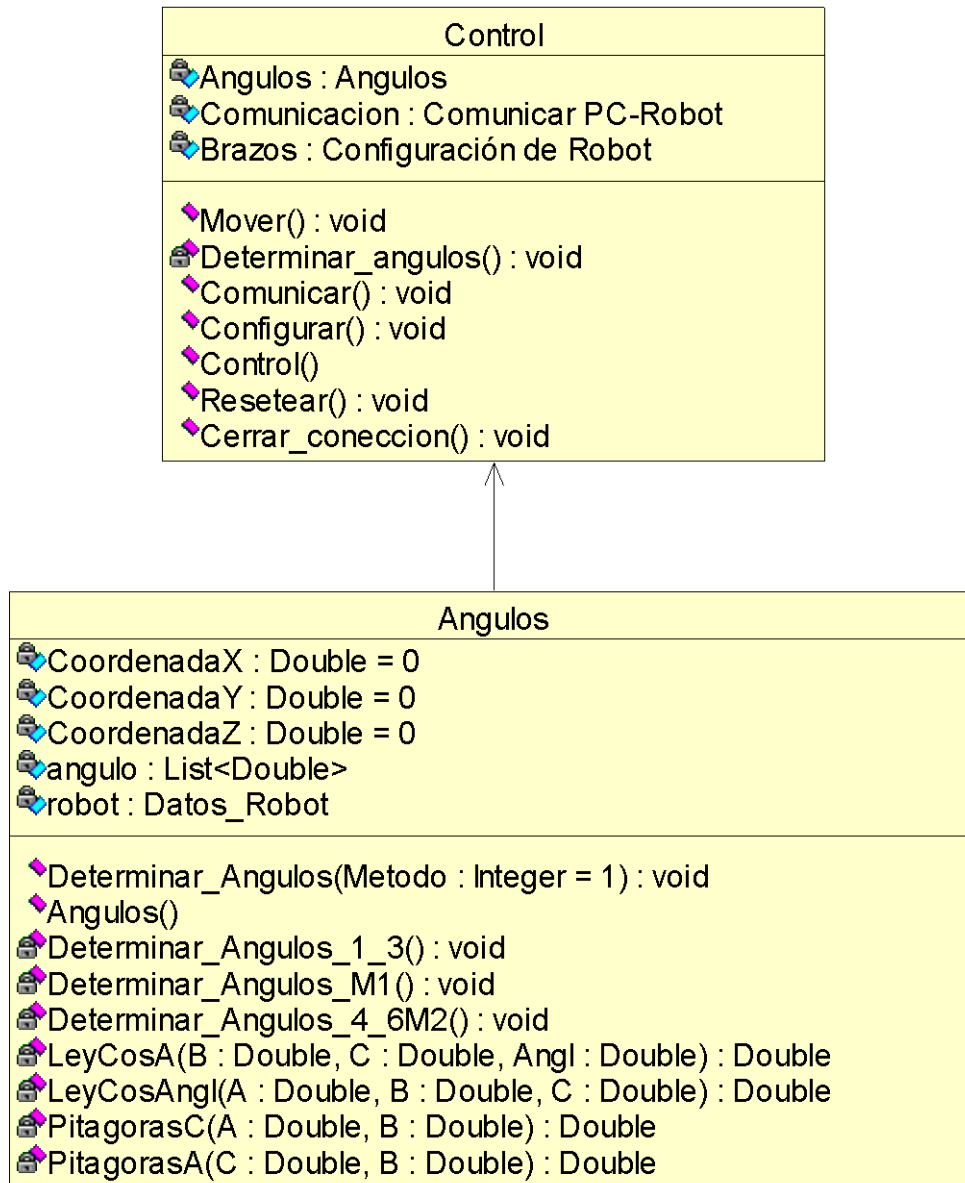


Fig. 19 Diagrama de clases de diseño CU\_Mover Robot

4.4 Diagrama de despliegue

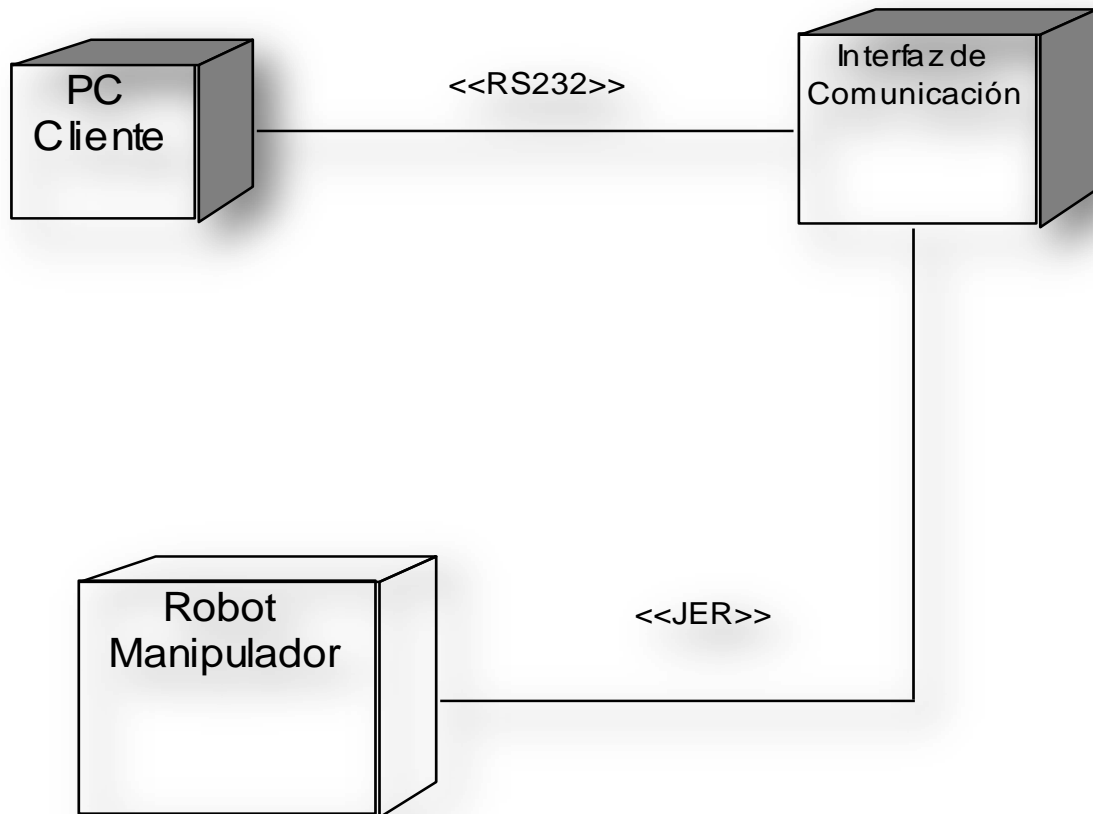


Fig. 20 Diagrama de despliegue



4.5 Diagrama de Componentes

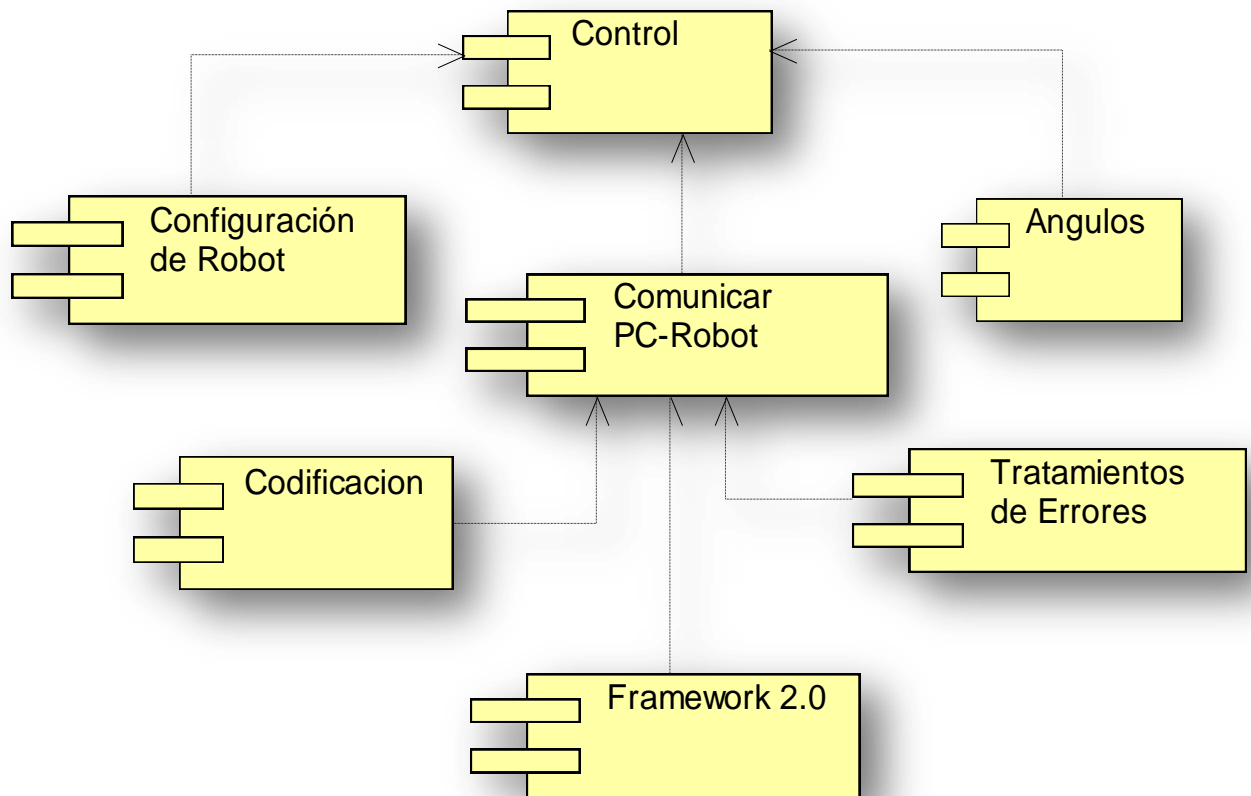


Fig. 21 Diagrama de componentes

### 4.6 Patrones GRASP <sup>vii</sup>usados.

En la terminología de objetos, el patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas <sup>1</sup>. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas.

Expresado lo anterior con palabras más simples, el patrón es una pareja problema / solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas. (18)

¿Qué son los patrones GRASP?

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrón. (18)

#### 4.6.1 Bajo Acoplamiento.

**Solución:** Asignar una responsabilidad para mantener bajo acoplamiento.

**Problema:** ¿Cómo dar soporte a una dependencia escasa y a un aumento de reutilización?

El acoplamiento es una medida de la fuerza con que una clase esta conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras. (18)

Debe haber pocas dependencias entre las clases. Si todas las clases dependen de todas ¿cuánto software es posible extraer de un modo independiente y reutilización en otro proyecto? Para determinar el nivel de acoplamiento de clases, son muy buenos los diagramas de colaboración de UML. Uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda. Siempre hay que considerar las ventajas de la delegación respecto de la herencia. (19)

---

<sup>1</sup> La notación formal de los patrones nació con los patrones arquitectónicos de Christopher Alexander [AIS77]. En los años 80, Kent Beck y Ward Cunningham hicieron su aplicación al software [Beck94, Coplien95].

### 4.6.2 Experto.

**Solución:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

**Problema:** ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos?

Un modelo de clase puede definir docenas y hasta cientos de clases de software, y una aplicación tal vez requiera el cumplimiento de cientos o miles de responsabilidades. Durante el diseño orientado a objetos, cuando se definen las interacciones entre los objetos, fueron tomadas decisiones sobre la asignación de responsabilidades a las clases. Si se hace en forma adecuada, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y se presenta la oportunidad de reutilizar los componentes en futuras aplicaciones. (18)

La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase, contiene toda la información necesaria para realizar la labor que tiene encomendada. Hay que tener en cuenta que esto es aplicable mientras se esté considerando los mismos aspectos del sistema:

- ✓ Lógica de negocio
- ✓ Persistencia a la base de datos
- ✓ Interfaz de usuario (19)

### 4.6.3 Controlador.

**Solución:** Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:

- ✓ el “sistema” global (controlador de fachada).
- ✓ la empresa u organización global (controlador de fachada).
- ✓ algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas).
- ✓ un manejador artificial de todos los eventos del sistema de un caso de uso, generalmente denominados “Manejador <NombreCasodeUso>”(controlador de casos de uso)

**Problema:** ¿Quién debería encargarse de atender un evento del sistema?

Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. Se asocia a operaciones del sistema: las que emite en respuesta del sistema.

Un Controlador es un objeto de interfaz no destinada a usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. (18)

Asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema (19)

### 4.7 Conclusiones

En este capítulo se presentó el diseño del sistema, donde se incluyeron los diagramas del sistema necesarios para lograr un mejor desarrollo del sistema planteado; además se incluyeron los patrones GRASP usados para la modelación del mismo.

## 5 Capítulo 5 Estudio de factibilidad

### 5.1 Introducción

Para la realización de un sistema es muy importante hacer un análisis del costo del mismo, por tanto en el presente capítulo realizan un estudio sobre la factibilidad del sistema. Para ello se aplica la técnica de estimación por caso de uso que servirá para calcular el costo, tiempo de desarrollo, el esfuerzo y la cantidad de personas que se necesitan para desarrollar el sistema.

### 5.2 Puntos de Casos de Uso

- ✓ Cálculo de Puntos de Casos de Uso sin ajustar.

Se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

- ✓ Factor de Peso de los Actores sin ajustar (UAW).

Los criterios se muestran en la siguiente tabla:

Tipo de actor	Descripción	Factor de Peso
Simple	Otro Sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación(API, Application Programming Interface)	1
Medio	Otro sistema que interactúa con el sistema mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 5.1: Peso según tipo de actor

El usuario del sistema es un tipo de actor simple, ya que es otro sistema que interactúa con el sistema a desarrollar, por tanto el factor de peso es igual a:

$$UAW=1*1=1$$

- ✓ Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Factor de Peso
<b>Simple</b>	El Caso de Uso contiene de 1 a 3 transacciones	<b>5</b>
<b>Medio</b>	El Caso de Uso contiene de 4 a 7 transacciones	<b>10</b>
<b>Complejo</b>	El Caso de Uso contiene 8 o más transacciones	<b>15</b>

Tabla 5.2: Peso según tipo de caso de uso

Todos los casos de uso del sistema tienden a ser simples, pues tienen menos de 4 transacciones cada uno. Por lo que se tienen 4 casos de uso de tipo simple (peso 5), con lo cual el factor de peso de los casos de uso sin ajustar resulta:

$$UUCP = UAW + UUCW$$

$$UUCP = 1 + 20 = 21$$

### 5.3 Cálculo de Puntos de Casos de Uso ajustados

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$\mathbf{UCP = UUCP \times TCF \times EF = 21 \times 1.1 \times 1,145 = 26,4495}$$

Donde:

**UCP:** Puntos de Casos de Uso ajustados

**UUCP:** Puntos de Casos de Uso sin ajustar

**TCF:** Factor de complejidad técnica

**EF:** Factor de ambiente

Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la tabla 17 se muestra el significado y el peso de cada uno de éstos factores:

<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>
T1	Sistema distribuido	2
T2	Objetivos de performance o tiempo de respuesta	1
T3	Eficiencia del usuario final	1
T4	Procesamiento interno complejo	1
T5	El código reutilizable	1
T6	Facilidad de instalación	0.5
T7	Facilidad de uso	0.5
T8	Portabilidad	2
T9	Facilidad de cambio	1
T10	Concurrencia	1
T11	Incluye objetivos especiales de seguridad	1
T12	Provee acceso directo a terceras partes	1
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1

Tabla 5.3: Peso según factor

El factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$\mathbf{TCF = 0.6 + 0.01 \times \Sigma (\text{Peso} \times \text{Valor asignado}) = 0.6 + 0.01 \times 50 = 1.1}$$

T1- Peso = 2, VA = 0

T2- Peso = 1, VA = 4

T3- Peso = 1, VA = 4

T4- Peso = 1, VA = 5

T5- Peso = 1, VA = 4

T6- Peso = 0.5, VA = 5

T7- Peso = 0.5, VA = 5

T8- Peso = 2, VA = 5

T9- Peso = 1, VA = 4

T10- Peso = 1, VA = 4

T11- Peso = 1, VA = 0

T12- Peso = 1, VA = 5

T13- Peso = 1, VA = 5

Factor de ambiente (EF)

En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores.



<b>Factor</b>	<b>Descripción</b>	<b>Peso</b>
<b>E1</b>	Familiaridad con el modelo de proyecto utilizado	<b>1.5</b>
<b>E2</b>	Experiencia en la aplicación	<b>0.5</b>
<b>E3</b>	Experiencia en orientación a objetos	<b>1</b>
<b>E4</b>	Capacidad del analista líder	<b>0.5</b>
<b>E5</b>	Motivación	<b>1</b>
<b>E6</b>	Estabilidad de los requerimientos	<b>2</b>
<b>E7</b>	Personal Part-time	<b>-1</b>
<b>E8</b>	Dificultad del lenguaje de programación	<b>-1</b>

‘Tabla 5.4 Factor ambiente

Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso} \times \text{Valor asignado}) = 1.4 - 0.03 \times 8.5 = 1.145$$

De los Puntos de Casos de Uso a la estimación del esfuerzo

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.

- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

### **CF = 28 horas-hombre/Punto de Casos de Uso**

E1	Valor	1.5	Peso	0
E2	Valor	0.5	Peso	1
E3	Valor	1	Peso	4
E4	Valor	0.5	Peso	2
E5	Valor	1	Peso	5
E6	Valor	2	Peso	0
E7	Valor	-1	Peso	0
E8	Valor	-1	Peso	2

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF = 26,4495 \times 28 = 740,586 \text{ horas-hombre}$$

Donde

**E:** esfuerzo estimado en horas-hombre

**UCP:** Puntos de Casos de Uso ajustados

**CF:** factor de conversión

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Se asume que se trabaja **6** horas diarias durante **20** días, al mes serían 120 horas.

$$ET = 740,586 \text{ horas-hombre} = 6,17155 \text{ mes} - \text{ hombre.}$$

**ET:** esfuerzo total

**TD:** tiempo de desarrollo

**CH:** cantidad de hombres

**SM:** salario medio

**CH = 2 hombres**

**SM = \$ 175**

**TD = ET / CH**

$$TD = 6,17155 / 2 = 3,1 \text{ meses} \approx 4 \text{ meses}$$

$$C = TD \times CH \times SM$$

$$C = 4 \times 2 \times 175 = \$ 1400$$

#### 5.4 Cálculos del Proyecto

Cálculo de	Valor
Esfuerzo	6,946 mes – hombre
Tiempo de desarrollo	4 meses
Cantidad de hombres	2
Salario medio	\$ 175
Costo	\$1400

**Tabla 5.5: Resultados del proyecto**

#### 5.5 Beneficios tangibles e intangibles. Debe cuantificarse los tangibles.

El Sistema tiene como objetivo principal solucionar el problema cinemático inverso para brazos manipuladores de 6 grados de libertad.

El beneficio fundamental de este sistema es permitir el movimiento de robots manipulador de 6 grados de libertad mediante un driver que sirva de intermediario entre el sistema de control y el robot manipulador.

#### 5.6 Análisis de costo

Antes de desarrollar una aplicación es muy importante que primero se haga un análisis de su costo, ya que este dirá si los beneficios que ella reportará se corresponden con su precio y de acuerdo a esto se considerará si se continua o no con su desarrollo.

Este sistema puede ser empleado como driver de los robots de producción en serie ensamblado de las industrias.

Analizando el costo del proyecto, los numerosos beneficios que reporta, detallados con anterioridad, se puede concluir que su implementación es realmente factible.

### **5.7 Conclusiones**

En este capítulo se describió el estudio de factibilidad realizado al sistema propuesto a través de la técnica de puntos de casos de uso, mediante el cual se obtuvo el costo y tiempo de desarrollo, cantidad de personas, es decir datos con los que es posible medir la viabilidad del proyecto. Se analizaron además los beneficios que el mismo reportará al ser implantado.

Por todo esto se llegó a la conclusión de que es factible implementar el sistema propuesto.

### **Conclusiones**

En este trabajo se ha realizado un análisis de la tecnología a utilizar, así como de la herramienta para modelar el problema, y los principales lenguajes de programación y plataformas de desarrollo. Se desplegó la teoría matemática que modela el problema planteado.

Con la intención de facilitar el entendimiento del sistema, se creó el modelo de dominio y se definieron los requisitos funcionales y no funcionales del mismo. Se obtuvo además el diagrama de casos de uso correspondiente y fueron identificados los patrones GRASP necesarios para la modelación.

Finalmente se llevó a cabo un estudio de la factibilidad del sistema mediante la técnica de puntos de casos de usos obteniendo el costo, el tiempo de desarrollo y los beneficios que reportará el sistema al ser implementado; concluyendo que sería factible el desarrollo e implantación del sistema propuesto.

### **Recomendaciones**

Se recomienda la Incorporación de sensores en la mano del brazo manipulador para poder obtener la orientación de la misma en la posición inicio y así implementar completamente el modelo matemático propuesto.

ANEXO 1 Diagramas de Interacción

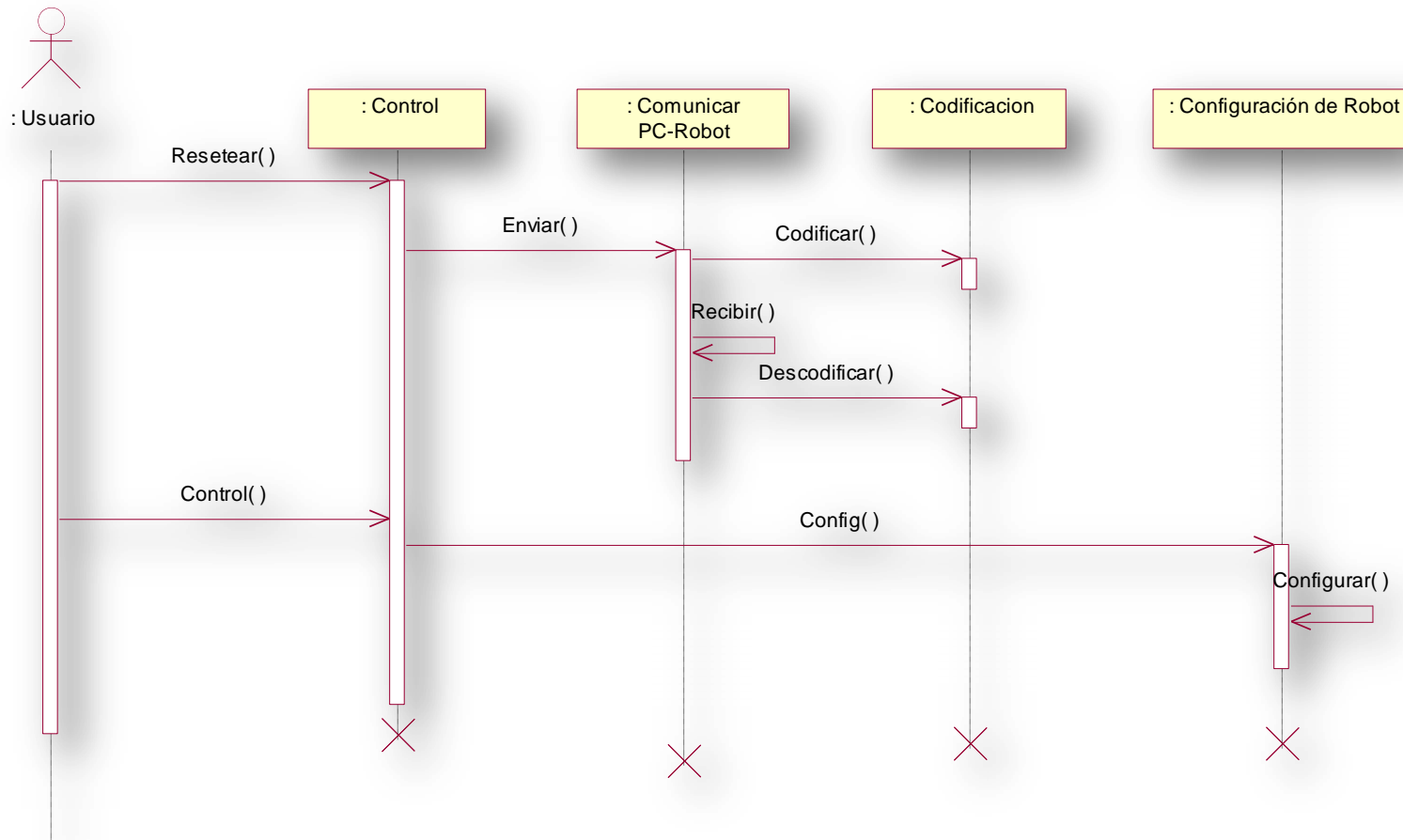


Fig. 1 Anexo1 Diagrama de Interacción CU\_Inicializar Robot Manipulador

## ANEXO 1 DIAGRAMAS DE INTERACCIÓN

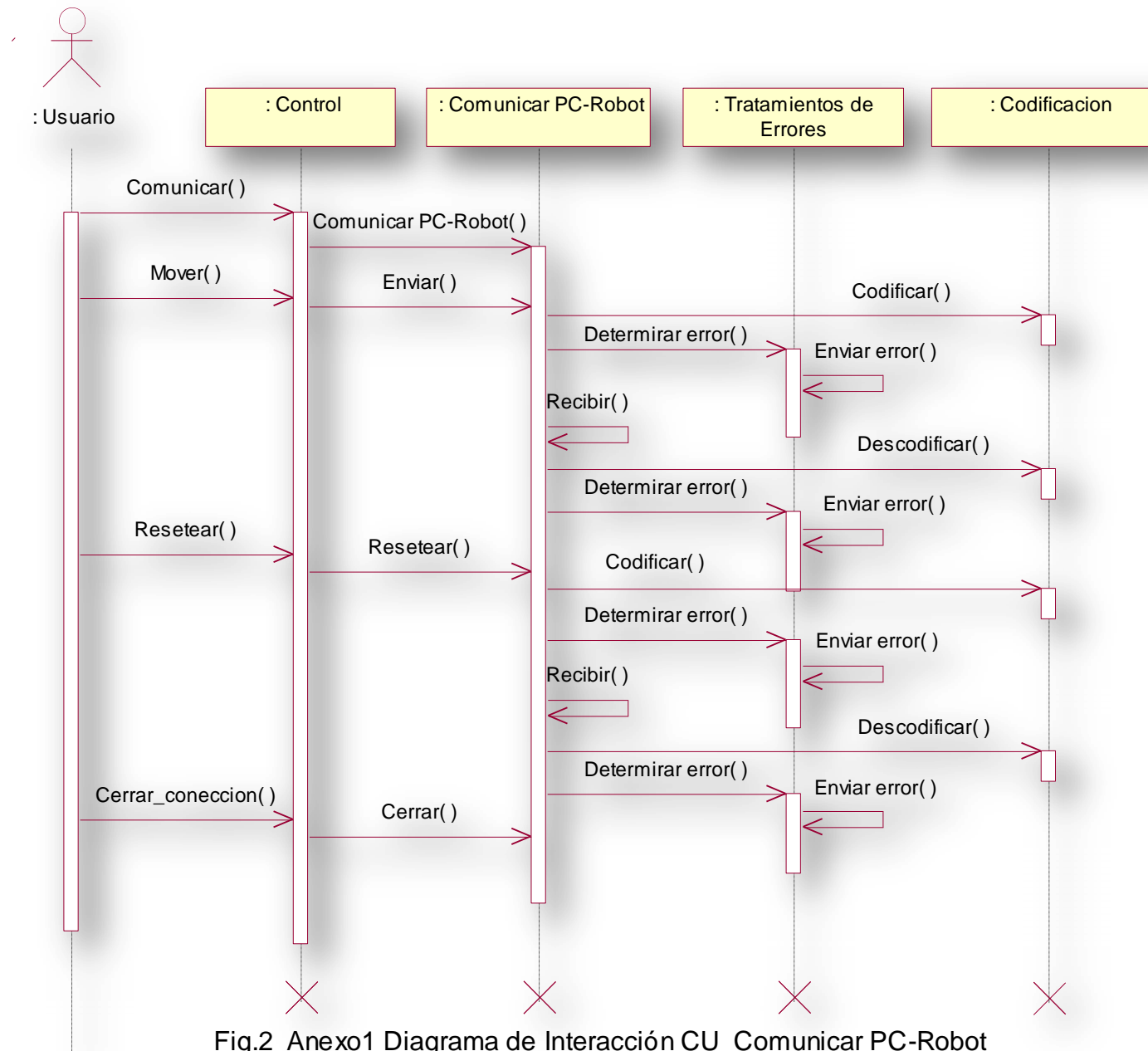


Fig.2 Anexo1 Diagrama de Interacción CU\_Comunicar PC-Robot



## ANEXO 1 DIAGRAMAS DE INTERACCIÓN

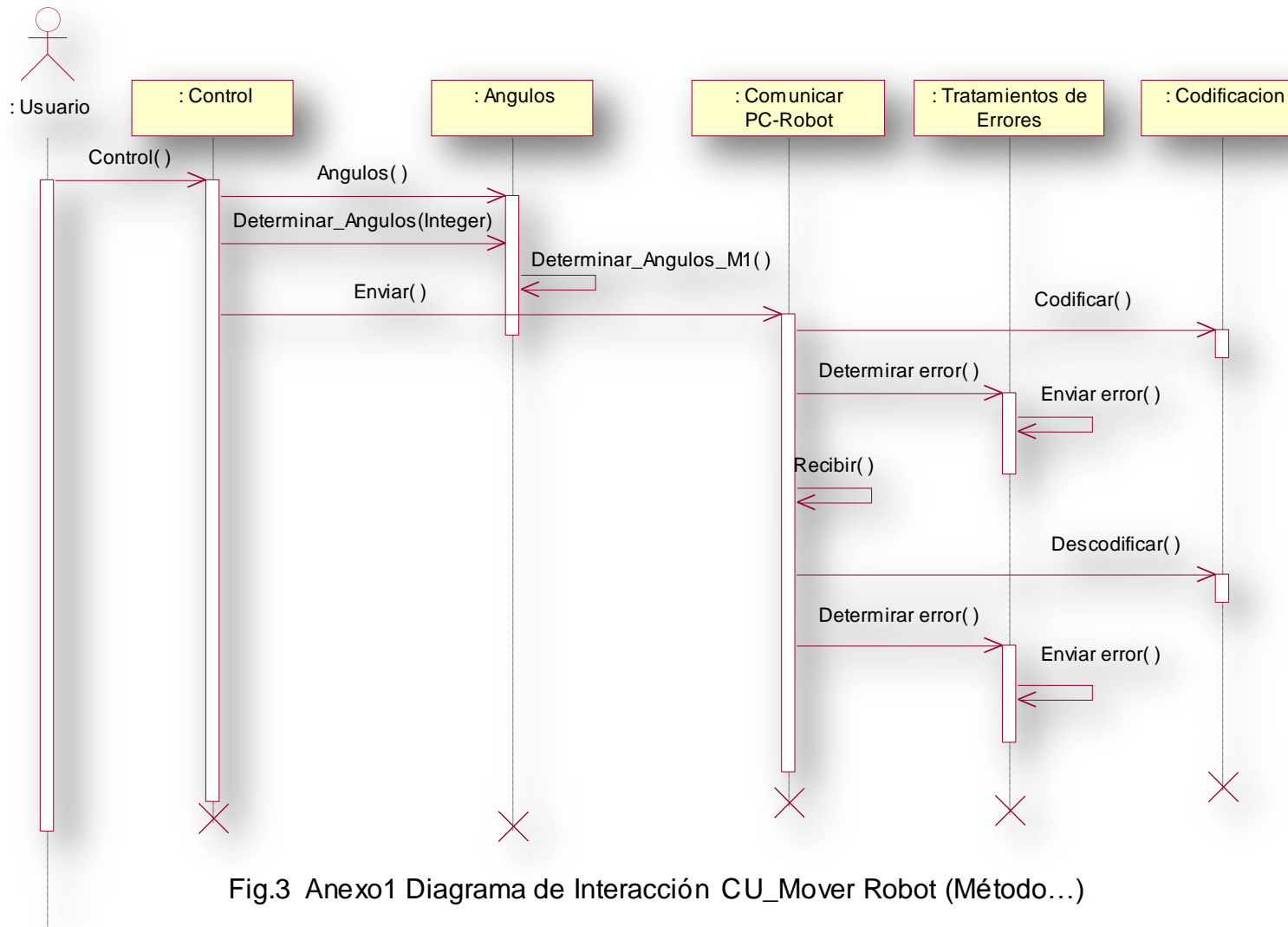


Fig.3 Anexo1 Diagrama de Interacción CU\_Mover Robot (Método...)

## ANEXO 1 DIAGRAMAS DE INTERACCIÓN

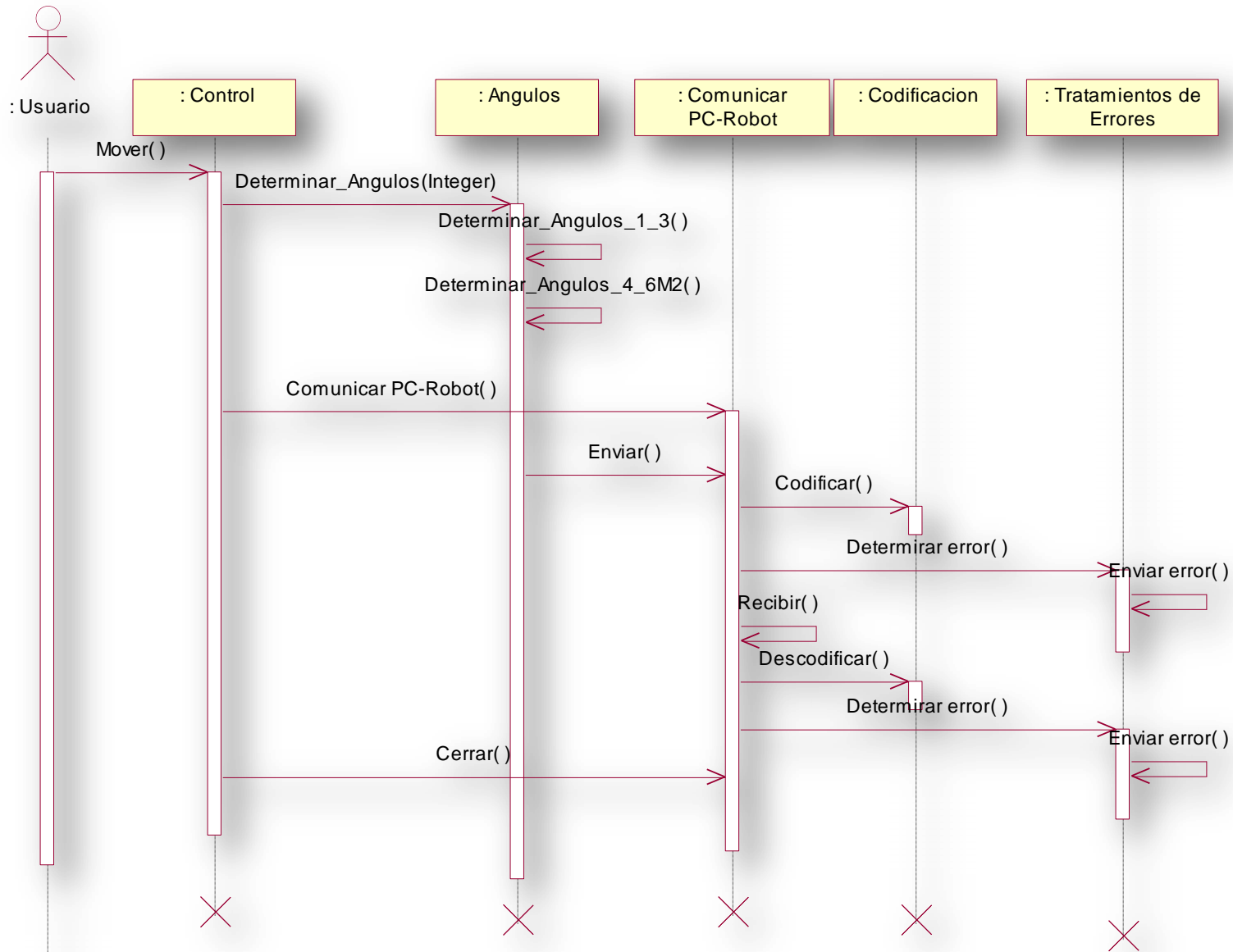


Fig.4 Anexo1 Diagrama de Interacción CU\_Mover Robot (Método Tradicional)

## Trabajos citados

1. **Wikipedia.** [En línea] [Citado el: 20 de 3 de 2008.] <http://es.wikipedia.org/wiki/Campusbot>.
2. **RoboticSpot.** Robot Scara. [En línea] [Citado el: 7 de Enero de 2008.]  
<http://www.roboticspot.com/spot/artic.shtml?todo=&block=4&newspage=robotsindustriales>.
3. **Instituto Tecnológico de Nuevo Laredo.** Instituto Tecnológico de Nuevo Laredo. [En línea] [Citado el: 8 de enero de 2008.]  
[http://64.233.169.104/search?q=cache:\\_R22RqUzHR0J:www.itnuevolaredo.edu.mx/maestros/sis\\_com/takeyas/Apuntes/Inteligencia%2520Artificial/Apuntes/tareas\\_alumnos/Robotica/Robotica2\(2005-II\).pdf+%22Manipulador+de+Stanford%22+%2B+robot&hl=es&ct=clnk&cd=17](http://64.233.169.104/search?q=cache:_R22RqUzHR0J:www.itnuevolaredo.edu.mx/maestros/sis_com/takeyas/Apuntes/Inteligencia%2520Artificial/Apuntes/tareas_alumnos/Robotica/Robotica2(2005-II).pdf+%22Manipulador+de+Stanford%22+%2B+robot&hl=es&ct=clnk&cd=17).
4. **baarriio.** Sueños De Un Robot. [En línea] 15 de abril de 2008. [Citado el: 20 de mayo de 2008.]  
<http://baarriio.blogspot.es/>.
5. **INFORMÁTICA DE GESTIÓN.** Salamanca, España : s.n., diciembre de 1999.
6. **Universidad de Sevilla.** Departamento de Arquitectura y Tecnología de Computadores. [En línea] [Citado el: 6 de enero de 2008.]  
[http://icaro.eii.us.es/descargas/Tema%20\\_4\\_%20parte\\_4\\_y\\_ultima.pdf](http://icaro.eii.us.es/descargas/Tema%20_4_%20parte_4_y_ultima.pdf);  
[http://www.esi.us.es/php/infgen/aulav/aeronaves/tema4\\_parte3.pdf](http://www.esi.us.es/php/infgen/aulav/aeronaves/tema4_parte3.pdf).
7. **Microsoft.com.** MSDN. *MSDN-ES*. [En línea] [Citado el: 8 de enero de 2008.]  
[http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cskon/html/vdrfAQuickSurveyOfCSharpFeatures\\_PG.asp](http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cskon/html/vdrfAQuickSurveyOfCSharpFeatures_PG.asp).
8. **urriellu.net.** [En línea] [Citado el: 11 de junio de 2008.] <http://urriellu.net/es/articles-software/csharp-advantages.html>.
9. **desarrolloweb.** [En línea] [Citado el: 11 de junio de 2008.]  
<http://www.desarrolloweb.com/articulos/1329.php>.
10. **wikipedia.** wikipedia. [En línea] [Citado el: 11 de junio de 2008.]  
[http://es.wikipedia.org/wiki/Proceso\\_Unificado](http://es.wikipedia.org/wiki/Proceso_Unificado).
11. **disca.** [En línea] [Citado el: 11 de junio de 2008.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
12. **Gómez Batista, Yalice y Rodríguez Montoya, Roberto.** *Diseño de un sistema de control con microcontrolador para el robot ARMDROID 1*. Santiago de Cuba, Santiago de Cuba, Cuba : Universidad de Oriente, 2004. Trabajo de diploma Tesis en opción al grado de Ingeniero en Automática.
13. **Universidad de Sevilla.** [En línea] [http://www.esi.us.es/php/infgen/aulav/aeronaves/tema4\\_parte3.pdf](http://www.esi.us.es/php/infgen/aulav/aeronaves/tema4_parte3.pdf).

14. **Pozo Quintero, Abelardo del y Sebastian Arbide, Ricardo.** *La Síntesis Cinemática y Dinámica de dos Robots Industriales*. [ed.] Ernesto Morfa. 112. Ciudad de la Habana : Academia, 1989. Vol. 1.
15. **BibliotecaInformatica.Net.** BibliotecaInformatica.Net. [En línea] [Citado el: 17 de mayo de 2008.] <http://bibliotecainformatica.iespana.es/hardware/drivers.htm>.
16. **ALEGSA.** DICCIONARIO DE INFOMÁTICA INTERNET Y TECNOLOGÍAS. [En línea] [Citado el: 17 de Mayo de 2008.] <http://www.alegsa.com.ar/Dic/controlador.php>.
17. **Alvarez, Sofia y Hernandez, Anaisa.** *Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML*. Ciudad de la Habana : s.n., 2000.
18. **Larman, Craig.** *UML y Patrones*. Ciudad de la Habana : Félix Varela, 2004.
19. **Saavedra Gutierrez, Jorge A.** El mundo informático. [En línea] [Citado el: 28 de mayo de 2008.] <http://jorgesaavedra.wordpress.com/2006/08/17/patrones-grasp-craig-larman/>.

## GLOSARIO

---

- i** Instituto de Cibernética, Matemática y Física
- ii** Sistema de coordenadas de la mano o agarre con respecto a la base, está dada por la matriz  $U_1$ , o sea:  $T_6 = U_1 = A_1 * A_2 * A_3 * A_4 * A_5 * A_6$
- iii** Teoría de Mecanismos y maquinas
- iv** Hace referencia a los Robots que poseen múltiples patas o pies (imitan a los artrópodos en funciones y formas).
- v** ECMA International es una organización internacional basada en memberships de estándares para la comunicación y la información. Adquirió el nombre ECMA International en 1994, cuando la European Computer Manufacturers Association (ECMA), cambió su nombre para expresar su alcance internacional. Como consecuencia de esta decisión, el nombre ya no se considera un acrónimo y no se escribe completamente en mayúsculas. La organización fue fundada en 1961 para estandarizar los sistemas computarizados en Europa. La membership está abierta a las empresas que producen, comercializan o desarrollan sistemas computarizados o de comunicación en Europa.
- vi** La Organización Internacional para la Estandarización o International Organization for Standardization (ISO), que nace después de la segunda guerra mundial (fue creada en 1946), es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.
- vii** Es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Desde el punto de vista técnico, se debería escribir "Patrones GRAS" en vez de "Patrones GRASP" pero la segunda expresión suena mejor en inglés, idioma en que fue acuñado el término.