

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE MATRIZ CUENCA**

CARRERA DE INGENIERÍA ELECTRÓNICA.

**“INTERFAZ DE CONTROL PARA UN BRAZO ROBOT
ARTICULADO BASADO EN SOFTWARE DE DESARROLLO
INTEGRAL”.**

Tesis previa a la obtención del
Título de Ingeniero Electrónico.

Autor: Rodrigo Andrés Escandón Cueva.

Director: Ing. Marco Alexander Carpio Alemán.

Cuenca - Ecuador

2013

**INTERFAZ DE CONTROL PARA UN BRAZO
ROBOT ARTICULADO BASADO EN
SOFTWARE DE DESARROLLO INTEGRAL**

DECLARACIÓN

Los conceptos desarrollados, el análisis realizado sobre el tema y las conclusiones de la presente monografía, son de exclusiva responsabilidad del autor. Y autorizo a la Universidad Politécnica Salesiana el uso de la misma con fines académicos.

Rodrigo Andrés Escandón Cueva

Cuenca, Junio del 2013



DECLARACIÓN

Yo Ing. Marco Carpio, certifico haber dirigido y revisado prolijamente cada uno de los capítulos de la tesis titulada “Interfaz de control para un brazo robot articulado basado en software de desarrollo integral”, realizada por el señor Rodrigo Andrés Escandón Cueva, y por lo tanto autorizo su presentación..

Ing. Marco Alexander Carpio Alemán

Cuenca, Junio del 2013

DEDICATORIA

Dedico este proyecto a mis amados padres que gracias a su apoyo he podido salir adelante y cumplir con mis metas, cualquier cosa que logre realizar en mi vida o llegue a ser, será gracias a ellos y su constante e incondicional soporte; Además debo agradecer a mis hermanas quienes han sido los mejores ejemplos a seguir para mí y espero que se vean reflejadas en mi esfuerzo y trabajo diario así como yo lo hago en el de ellas.

Rodrigo Andrés Escandón Cueva

**INTERFAZ DE CONTROL PARA UN BRAZO ROBOT
ARTICULADO BASADO EN SOFTWARE DE
DESARROLLO INTEGRAL.**

RODRIGO ANDRÉS ESCANDÓN CUEVA.

Egresado de la Carrera
Ingeniería Electrónica
Universidad Politécnica Salesiana sede Cuenca

Dirigido por:

MARCO ALEXANDER CARPIO ALEMÁN

Ingeniero Electrónico
Director de Carrera Universidad Politécnica Salesiana sede Cuenca
Carrera de Ingeniería Eléctrica



Cuenca –Ecuador

2013

RODRIGO ANDRÉS ESCANDÓN CUEVA

Interfaz de control para un brazo robot articulado basado en software de desarrollo integral.

Universidad Politécnica Salesiana, Cuenca – Ecuador 2013

Ingeniería Electrónica

Formato: 170x240 mm

Páginas: 167



Rodrigo Andrés Escandón Cueva

Egresado de la Carrera de Ingeniería Electrónica
Universidad Politécnica Salesiana sede Cuenca.

email: rescandon@ieee.org



Ing. Marco Alexander Carpio Alemán

Ingeniero Electrónico, Universidad Politécnica Salesiana
del Ecuador.
Director de carrera Universidad Politécnica Salesiana
Carrera de Ingeniería Eléctrica.

email : mcarpio@ups.edu.ec

Queda prohibida, salvo excepción prevista en la ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con la autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos o investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

© Universidad Politécnica Salesiana
CUENCA – ECUADOR – SUDAMERICA.

Rodrigo Andrés Escandón Cueva, *Interfaz de control para un brazo robot articulado basado en software de desarrollo integral.*

IMPRESO EN ECUADOR – PRINTED IN ECUADOR.

ÍNDICE GENERAL

CAPITULO 1. PROGRAMACIÓN DE ROBOTS INDUSTRIALES.	1
1.1. INTRODUCCIÓN:	1
1.1.1. Robótica Industrial.	1
1.1.2. Impacto de la robótica industrial en la actualidad.	2
1.2. PROGRAMACIÓN EN ROBOTS INDUSTRIALES.	2
1.2.1. Introducción:	2
1.2.2. Reseña Histórica.....	3
1.3. ARQUITECTURA DE PROGRAMACIÓN DE ROBOTS INDUSTRIALES.	6
1.4. MÉTODOS DE PROGRAMACIÓN DE ROBOTS INDUSTRIALES.....	8
1.4.1. Programación por guiado.	8
1.4.2. Programación textual.....	10
1.5. LENGUAJES DE PROGRAMACIÓN EN ROBOTS INDUSTRIALES.	10
1.5.1. Ambiente de programación.	12
1.5.2. Modelado del entorno.	12
1.5.3. Tipo de datos.....	13
1.5.4. Manejo de entradas / salidas (digital y analógicas)	14
1.5.5. Control de flujo de ejecución del programa.....	15
1.5.6. Control de movimiento.	16
CAPITULO 2. BRAZOS ROBOT INDUSTRIALES ANTROPOMÓRFICOS.	18
.....	
2.1 INTRODUCCION.....	18
2.2 ARQUITECTURA GENERAL DE UN SISTEMA ROBOTICO.....	18
2.3 BRAZOS ROBOT KUKA KR 16-2 Y KR 5-2 ARC HW.....	20
2.3.1 Estructura Mecánica.....	21
2.3.2 Sistema sensorial.	25
2.3.3 Sistema de control.	27
2.3.4 Elementos terminales.	30
2.4 BRAZO ROBOT FEEDBACK MENTOR 35-001	32
2.4.1 Estructura Mecánica.....	33
2.4.2 Sistema sensorial.	38
2.4.3 Sistema de control.	39
2.5 CINEMATICA DE ROBOTS ANTROPOMORFICOS.	40

2.6	PROGRAMACION EN KRL (KUKA ROBOT LENGUAJE).	43
2.6.1	Creación de programas secuenciales y de asignación de trayectorias.	44
2.7	INTRODUCCIÓN A LA COMUNICACIÓN INDUSTRIAL DE BRAZOS ROBOTS ARTICULADOS KUKA.	47
2.7.1	KUKA.Ethernet KRL XML.	48
2.7.2	KUKA.Robot Sensor Interface 3.1.	50
CAPITULO 3. DESARROLLO DE LA INTERFAZ.		52
3.1	INTRODUCCIÓN.	52
3.2	ENTORNOS DE DESARROLLO INTEGRADOS (IDE)	52
3.2.1	Matlab y Simulink.	53
3.2.2	Descripción del IDE.	54
3.2.1	Métodos de programación gráfica.	55
3.3	DESARROLLO DE LA INTERFAZ DE CONTROL.	57
3.3.1	Diseño de la interfaz gráfica de usuario.	58
3.3.2	Paneles de visualización y control de secuencias.	59
3.3.3	Área de simulación y monitoreo.	61
3.3.4	Controles de simulación y accionar real.	62
3.3.5	Menú de elección de robots y formas de programación.	62
3.3.6	Visualización de la posición y orientación del robot.	63
3.3.7	Paneles de programación.	64
3.3.8	Panel de generación de reportes.	68
3.3.9	Menú general del software.	69
3.3.10	Campo de control gráfico.	69
3.4	PROGRAMACION DE SISTEMAS DE CONTROL Y COMUNICACION.	71
3.4.1	Desarrollo de las librerías dedicadas para brazos robot.	72
3.4.2	Programación de sistemas de comunicación.	74
3.4.3	Programación de secuencias y envío de parámetros de control.	76
CAPITULO 4. PRUEBAS Y EVALUACIÓN DEL SISTEMA.		79
4.1	INTRODUCCIÓN.	79
4.2	PRUEBAS DEL SOFTWARE.	79
4.2.1	Pruebas de caja blanca.	80
4.2.1	Pruebas de caja negra.	90
4.3	PRUEBAS DE COMUNICACIÓN DEL SOFTWARE.	93
4.3.1	Proceso de adquisición de datos.	93
4.3.2	Análisis estadístico de los datos adquiridos.	97
4.4	ANÁLISIS DE RESULTADOS.	103
4.4.1	Calculo de precisión y exactitud.	103
4.4.2	Análisis de la respuesta física del sistema.	107

CAPITULO 5. RESUMEN, CONCLUSIONES Y RECOMENDACIONES.	112
5.1 CAPITULO 1.	112
5.2 CAPITULO 2.	113
5.3 CAPITULO 3.	114
5.4 CAPITULO 4.	118
APÉNDICES Y ANEXOS.	122
REFERENCIAS BIBLIOGRAFICAS	165
REFERENCIAS ELECTRÓNICAS	166

ÍNDICE DE FIGURAS

- Figura 1. a) Robot Unimate 1961 b) Robot Famulus 1973 [27, 28]. Página 4.
- Figura 2. Robot PUMA Unimation 1978 [26].
Página 5.
- Figura 3. Componentes estructurales Robot PUMA Unimation 1978 [13]. Página 5.
- Figura 4. Ejemplos de paneles de control. [33, 34].
Página 9.
- Figura 5. Relación entre diversos elementos de un entorno de programación. Página 13.
- Figura 6. Arquitectura básica de un sistema Robótico [15].
Página 19.
- Figura 7. Estructura mecánica Brazo robot antropomórfico KUKA KR 16-2 [16]. Página 22.
- Figura 8. Estructura mecánica Brazo robot KUKA KR5-2 ARC HW [17]. Página 22.
- Figura 9. Volumen de trabajo Brazo robot antropomórfico KUKA KR 16-2 [18]. Página 23.
- Figura 10. Volumen de trabajo Brazo robot antropomórfico KUKA KR 5-2 ARC HW [17]. Página 24.
- Figura 11. Tarjeta WAGO de acople de bus de campo DeviceNet [23]. Página 26.
- Figura 12. Esquema de los módulos I/O WAGO [22].
Página 26.
- Figura 13. KR C2 [19].
Página 28.
- Figura 14. Computadora embebida.
Página 29.
- Figura 15. Elementos adicionales del controlador.
Página 29.

- Figura 16. Controladores dedicados para grado de libertad.
Página 30.
- Figura 17. Ejemplos de Extremos efectores.
Página 31 [42].
- Figura 18. Garra de sujeción.
Página 32.
- Figura 19. Herramientas KR 5-2 ARC HW.
Figura 32.
- Figura 20. Estructura mecánica Brazo robot antropomórfico
Mentor 35-001 [24]. Página 33.
- Figura 21. Volumen de trabajo Brazo robot antropomórfico
Mentor 35-001. Página 34.
- Figura 22. Mecanismo reductor y de transmisión del movimiento
Mentor 35-001. Página 35.
- Figura 23. Mecanismo de transmisión del extremo efector del
Mentor 35-001. Página 37.
- Figura 24. Sistema diferencial mecánico del extremo efector del
Mentor 35-001. Página 37.
- Figura 25. Circuitos Equivalentes Driver de control [25]
Página 39.
- Figura 26. Algoritmo de Denavit y Hartenberg para un brazo robot
KUKA KR16-2 [4]. Página 41.
- Figura 27. Concepto de software Windows/ VxWorks [15].
Pagina 43.
- Figura 28. Programa base de una programación KRL [15].
Página 45.
- Figura 29. Estructura e interconexión del KRL XML [20].
Página 49.
- Figura 30. Estructura y componentes del IDE Matlab y Simulink
[36]. Página 54.
- Figura 31. Ejemplo de programación gráfica [38].
Página 56.
- Figura 32. Vista principal de la interfaz hombre máquina.
Página 58.

Figura 33.	Panel de control de secuencias. Página 59.
Figura 34.	Panel de visualización. Página 60.
Figura 35.	Imágenes de las variables del área de monitoreo. Página 61.
Figura 36.	Panel de simulación y accionar real. Página 62.
Figura 37.	Menú de elección de robots y formas de programación. Página 63.
Figura 38.	Paneles de visualización de Orientación y Posición. Página 64.
Figura 39.	Panel de programación asistida o manual. Página 65.
Figura 40.	Panel de programación para la generación de trayectorias. Página 67.
Figura 41.	Panel de programación real. Página 68.
Figura 42.	Panel de generación de reportes. Página 69.
Figura 43.	Menú del software. Página 69.
Figura 44.	Tolbox de Comunicación para el brazo robot Mentor. Página 70.
Figura 45.	Diagrama de implementación de las librerías de comunicación y control. Página 73.
Figura 46.	Tarjeta de adquisición de datos NI USB-6212 [12]. Página 75.
Figura 47.	Algoritmo básico para el control de movimiento. Página 77.
Figura 48.	Proceso de prueba para la evaluación de software [39]. Página 80.
Figura 49.	Algoritmo de control del diseño en realidad virtual. Página 82.

- Figura 50. Grafo de flujo del control del diseño en realidad virtual. Página 83.
- Figura 51. Algoritmo de control del manipulador robótico. Página 86.
- Figura 52. Grafo de flujo de control del manipulador robótico. Página 87.
- Figura 53. Ilustración gráfica del movimiento utilizado para la experimentación. Página 94.
- Figura 54. Muestra Adquirida de un sensor para ser aplicado en los procesos de prueba. Página 96.
- Figura 55. Histograma de las muestras del Primer grado de libertad del quinto proceso de prueba [Anexo B]. Página 97.
- Figura 56. Histograma de las muestras del Segundo grado de libertad del quinto proceso de prueba [Anexo B]. Página 98.
- Figura 57. Histograma de las muestras del Tercer grado de libertad del quinto proceso de prueba [Anexo B]. Página 98.
- Figura 58. Histograma de las muestras del Cuarto grado de libertad del quinto proceso de prueba [Anexo B]. Página 99.
- Figura 59. Histograma de las muestras del Quinto grado de libertad del quinto proceso de prueba [Anexo B]. Página 99.
- Figura 60. Parámetros relacionados a la distribución de probabilidad normal [6]. Página 100.
- Figura 61. Función de densidad de probabilidad del primer grado de libertad del quinto proceso de prueba [Anexo B]. Página 101.
- Figura 62. Función de densidad de probabilidad del segundo grado de libertad del quinto proceso de prueba [Anexo B]. Página 101.
- Figura 63. Función de densidad de probabilidad del tercer grado de libertad del quinto proceso de prueba [Anexo B]. Página 102.

- Figura 64. Función de densidad de probabilidad del cuarto grado de libertad del quinto proceso de prueba [Anexo B].
Página 102.
- Figura 65. Función de densidad de probabilidad del quinto grado de libertad del quinto proceso de prueba [Anexo B].
Página 102.
- Figura 66. Tendencia de cambio de las muestras de los diferentes experimentos. Página 106.
- Figura 67. Valores medidos directamente del sensor de posición.
Página 107.
- Figura 68. Respuesta física a un movimiento puntual del GDL1.
Página 108.
- Figura 69. Respuesta física a un movimiento puntual del GDL2.
Página 109.
- Figura 70. Respuesta física a un movimiento puntual del GDL3.
Página 109.
- Figura 71. Respuesta física a un movimiento puntual del GDL4.
Página 109.
- Figura 72. Respuesta física a un movimiento puntual del GDL5.
Página 110.

ÍNDICE DE TABLAS.

Tabla 1.	Lenguajes de programación y Sistemas operativos de las principales empresas de robótica industrial. Página 9.
Tabla 2.	Descripción y funcionamiento de los ejes en un brazo robot antropomórfico. Página 17.
Tabla 3.	Especificaciones mecánicas Brazo robot KUKA KR 16-2 [18]. Página 20.
Tabla 4.	Especificaciones mecánicas Brazo robot antropomórfico KUKA KR 5-2 ARC HW [17]. Página 21.
Tabla 5.	Descripción y funcionamiento de los ejes del brazo robot Mentor 35-001. Página 27.
Tabla 6.	Especificaciones mecánicas Brazo robot antropomórfico Mentor 35-001. Página 29.
Tabla 7.	Valores de operación típicos de los controladores [25]. Página 33.
Tabla 8.	Parámetros de Denavit y Hartenberg del brazo robot KUKA KR 16-2 [4]. Página 35.
Tabla 9.	Sentencias de programación para KRL [15]. Página 38.
Tabla 10.	Componentes KRL XML [20]. Página 42.
Tabla 11.	Pruebas de camino básico para el algoritmo de control del modelo en realidad virtual. Página 71.
Tabla 12.	Pruebas de bucles para el algoritmo de control del modelo en realidad virtual. Página 72.
Tabla 13.	Pruebas de camino básico para el algoritmo de control del manipulador robótico. Página 75.
Tabla 14.	Pruebas de bucles para el algoritmo de control del modelo en realidad virtual. Página 76.

Tabla 15.	Clases de equivalencia para la interfaz gráfica de usuario. Página 78.
Tabla 16.	Pruebas de caja negra para la interfaz gráfica de usuario. Página 79.
Tabla 17.	Muestras utilizadas para el 5to Proceso de prueba [Anexo A]. Página 82.
Tabla 18.	Compendio de valores de exactitud y precisión calculados. Página 90.
Tabla 19.	Promedio de valores de exactitud y precisión del sistema. Página 91.
Tabla 20.	Parámetros de respuesta del movimiento mecánico real. Página 95.

ACRÓNIMOS.

PDF.	Formato de documento portable.
HTML.	Lenguaje Marcado de Hipertexto.
IDE.	Entorno de Desarrollo Integrado.
GUI.	Interfaz Gráfica de Usuario.
GUIDE	Ambiente de Desarrollo de Interfaces Graficas de Usuario
PUMA	Brazo manipulador Programable Universal.
KR	Robot Kuka.
KRC	Controlador de Robot Kuka.
KCP	Panel de control Kuka.
VDT	Terminal de Visualización.
USB	Bus serial Universal.
TCP	Protocolo de Control de Transmisión.
VGA	Conjunto de Gráficos de Video
DAQ	Adquisición de datos.

CAPITULO 1. PROGRAMACIÓN DE ROBOTS INDUSTRIALES.

1.1. INTRODUCCIÓN:

En este capítulo se presenta una breve introducción sobre la robótica industrial, la evolución de la misma y sus conceptos más importantes, de manera de implantar ciertas nociones necesarias para el mejor entendimiento de este tipo de máquinas industriales y el realce e importancia que estos tienen a nivel social e industrial principalmente. Posteriormente se centrara el estudio introductorio en el tema puntual de la programación de sistemas robóticos industriales, tema central del capítulo que nos permitirá entender de mejor manera el alcance y naturaleza del proyecto descrito en la presente monografía.

1.1.1. Robótica Industrial.

La robótica como tal es considerada como una rama de aplicación tecnológica principalmente asociada a las ciencias de la electrónica y la mecánica, que estudia el diseño y construcción de máquinas capaces de desempeñar una tarea específica; etimológicamente el termino robot se deriva la palabra “robota”, que hace referencia a trabajo o labor, el termino obviamente abarca todos los tipos de robots, más específicamente y en base al concepto estándar de la Organización internacional de estándares, un robot industrial es: “Un manipulador multifuncional, reprogramable, con varios grados de libertad, capaz de mover piezas, accionar herramientas, o dispositivos especiales, según trayectorias programadas para realizar diversas tareas” [10].

La robótica industrial continua en desarrollo en base a la necesidad de los seres humanos de efectivizar el trabajo al tiempo que disminuye considerablemente el actuar humano en trabajos pesados o repetitivos con una velocidad y exactitud prácticamente imposible de igualar por personas, lo que ha permitido desarrollar de manera tal que se han creado diversos tipos de robots, dependiendo de la aplicación industrial necesaria.

1.1.2. Impacto de la robótica industrial en la actualidad.

En 1995 el mundo industrializado ya contaba con alrededor de 700.000 robots, 500.000 se encontraban en Japón, 120.000 en Europa y 60.000 en Estados Unidos. En la actualidad se estima que exista al menos un millón y medio de robots en aéreas industriales con una distribución porcentual parecida a la estimada en la década anterior; Además se plantea estadísticas sobre su evolución tales como que en Alemania la relación entre trabajadores humanos y robots industriales sea de 7 a 1 es decir que por cada 7 personas existe un robot, además se plantea que en Japón cada hora se instala cerca de 5 nuevos robots industriales. Un claro ejemplo de esto es la línea de ensamblaje automotriz de la empresa alemana BMW, en la cual cientos de robots trabajan en una línea de ensamblaje de 12 km de largo para ensamblar un auto completamente en aproximadamente 3 minutos una producción especializada que permite satisfacer la gran demanda que la empresa automotriz tiene en la actualidad.

El impacto social y económico que la robótica industrial tiene sobre el haber general de todo el planeta es muy grande ya que basta con considerar que de no contar con el 10% de robots industriales que se dedican a la fabricación de elementos electrónicos la vida cotidiana como se la conoce hoy en día no pudiese continuar y sería prácticamente imposible continuar procesos productivos actuales sin la ayuda de mecanismos robóticos que posean la precisión y velocidad que tienen en la actualidad [8].

1.2. PROGRAMACIÓN EN ROBOTS INDUSTRIALES.

1.2.1. Introducción:

La programación de robots industriales se centra principalmente en la estructuración de sentencias de control definidas para un sistema robótico específico cuyo objetivo es el de indicar la secuencia de acciones a realizar durante una tarea que el mencionado dispositivo se proponga ejecutar, considerando como principales los siguientes puntos de acción:

- Movimientos a puntos predefinidos.
- Manipulación de Objetos.

- Interacción y sincronización con el entorno.

Los lenguajes de programación son considerados como el planteamiento más general para resolver los problemas de comunicación hombre máquina, estos consisten en la utilización de programación a alto nivel por medio de un computador o un panel de control en el cual se realice la programación de las sentencias de control que definen el actuar del robot, este tipo de comunicación es necesaria para realizar tareas de montajes o de ensamblaje, labores que tienen un mayor grado de dificultad y difícilmente pudieran ser estructurados por medio de programación guiada, además de considerarse una programación estructurada en base a programas de alto nivel el robot adquiere una mayor versatilidad debido a que este puede ejecutar múltiples tareas solamente cambiando al programa apropiado.

1.2.2. Reseña Histórica.

La robótica con fines industriales propiamente inicio como es obvio con el primer robot industrial el cual fue creado en el año de 1948 por la empresa estadounidense Unimation; Unimate como fue nombrado es un brazo robot esférico que fue instalado en el año de 1961 en la empresa General Motors como parte de su línea de producción específicamente moviendo piezas de metal caliente, más tarde se utilizaría el mismo robot como parte fundamental de la línea de ensamblaje en el área automotriz, principalmente durante la década subsiguiente a su instalación debido a que en la década de 1970 la globalización de la robótica industrial hizo que Unimation se enfrentase a otras empresas compitiendo por el liderato del mercado mundial. El robot tenía la capacidad de almacenar hasta 4000 pasos de programa en un disco de almacenamiento magnético, este tipo de control le permitía realizar varias aplicaciones pero no se consideraba como programable, debido a que el actuar del robot estaban limitados a estos pasos y a las secuencias predeterminadas de los mismos con una mínima interacción con el usuario.

Al igual que Unimate existen otros dispositivos robóticos de otras empresas que en base principalmente a necesidades planteadas por compañías interesadas en automatización desarrollaron diseños y prototipos propios los cuales han marcado importantes avances en lo que se conoce como robótica industrial propiamente en la actualidad, Es claro el ejemplo de la empresa KUKA robotics, empresa alemana que hoy en día tiene gran prestigio en todo el mundo que en el año de 1973 desarrollo su

primer prototipo propio de robot industrial aplicado a la industria automotriz el dispositivo denominado Famulus es considerado el primer robot con seis ejes accionados electromagnéticamente, Al igual que el robot Unimate este dispositivo basaba su movimiento en pasos de control definidos por lo que no tenía un lenguaje de programación específico que le permitiese el desarrollo de secuencias y movimientos avanzados [8].



Figura 1. a) Robot Unimate 1961 b) Robot Famulus 1973 [27, 28].

El primer robot programable también creado por la empresa Unimation fue el robot PUMA (Programmable Universal Machine for Assembly) en 1978, este dispositivo era capaz de mover un objeto y colocarlo en cualquier orientación en un lugar deseado dentro de su alcance, El robot PUMA es considerado la base fundamental para los diseños de los robots actuales, debido a su diseño estructural en el cual se podían diferenciar varios grados de libertad basados en articulaciones rotacionales, muy parecidos a los robots antropomórficos utilizados en gran medida en la actualidad. La programación del PUMA además permitió hacer muy poco en el campo de la investigación la integración de sistemas de control más complejos como son algoritmos de inteligencia artificial, redes neuronales entre otros dotando al sistema robótico de mayores prestaciones y versatilidad con su entorno y programador.



Figura 2. Robot PUMA Unimation 1978 [26].

El brazo robot PUMA utiliza su propio sistema operativo (VAL II o VAL Plus), el cual fue diseñado específicamente para robots de la empresa Unimation, este sistema no permite programar nuevos algoritmos de control debido a que estos estuvieron previamente definidos en el controlador pero permitía la programación de secuencias determinadas por el usuario desde una terminal de computadora conocida también por sus siglas en ingles VDT (Visual Display Terminal), lo que permitía una amplia gama de aplicaciones en base a estos programas descritos.

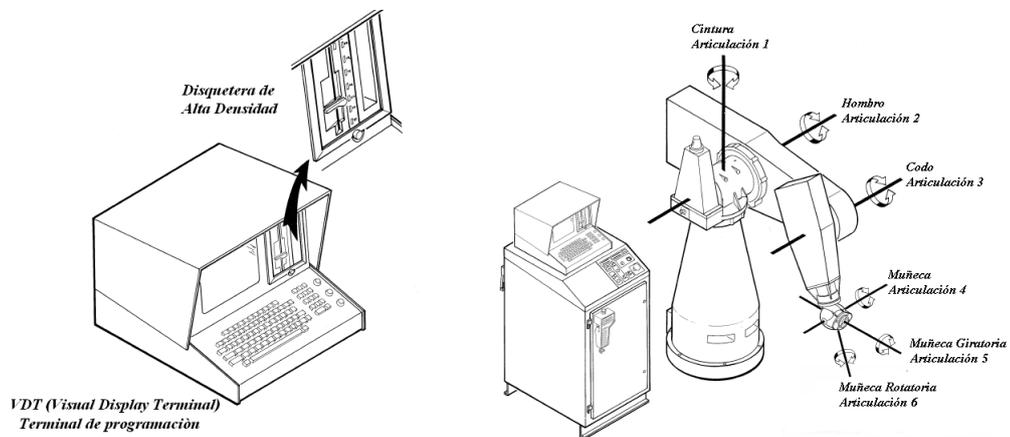


Figura 3. Componentes estructurales Robot PUMA Unimation 1978 [13].

El tipo de programación utilizado para describir estos programas estaban realizados de forma textual, esto significa que el usuario debe indicar la tarea al robot mediante el uso de un lenguaje de programación específico, es decir se describe las ordenes al robot y posteriormente son ejecutadas por el mismo previo una compilación o descarga de las mismas en el controlador del robot; Se realizara una

conceptualización más a profundidad de este y otros tipos de programación de robots más adelante el presente capítulo.

A manera de ejemplo se presenta un programa modelo en VAL II sistema operativo del robot PUMA desarrollado por la empresa Unimation que define una secuencia de movimientos específicos para un brazo robot PUMA.

PROG TOTO	SUBPROG TITI
1 MOVE A	1 TYPE "HE TERMINADO LA
2 CALL TITI	INSTRUCCIÓN DE MOVIMIENTO"
3 MOVE B	
4 CALL TITI	2 RETURN
5MOVE C	
6CALL TITI	
7MOVE D	
8CALL TITI	
9MOVE F	
10 IF (F>3) OR (G<=7.82)	
GOTO 9	
11 MOVE G	
RETURN	

A, B, C, etc. Son los nombres de los puntos a donde el robot deberá ir. Estos puntos están definidos por X, Y, Z, O, A y T que son las coordenadas cartesianas del punto a donde se quiere llegar (X, Y, Z) expresadas en milímetros y la orientación de la herramienta la cual está definida por las otras variables (O, A, T), expresadas en grados [14].

El ejemplo se presenta para escalear dos principales conceptos, primeramente se desea aclarar que los lenguajes de programación de robots industriales en su mayoría son propios de la compañía que los fabrica y además que a pesar de la gran evolución que los sistemas de programación de robots han tenido aun en la actualidad ciertas características predominan constantes en la programación de robots industriales.

1.3. ARQUITECTURA DE PROGRAMACIÓN DE ROBOTS INDUSTRIALES.

La programación textual puede ser clasificada en tres niveles, estos se definen dependiendo de que las órdenes programadas y enviadas al robot se refieran a los

movimientos propiamente dichos del robot, al estado en el que deben permanecer los objetos manipulados o al objetivo general a conseguir. Los niveles de programación se conocen como Nivel Robot, Nivel Objeto y Nivel Tarea respectivamente, estos se conceptualizan a continuación de manera de esclarecer las diferencias puntuales entre cada uno.

- **Nivel robot:** Para este nivel es necesario especificar cada movimiento que el robot deberá realizar para alcanzar su objetivo, así como velocidad, orientación, activación o desactivación de la herramienta etc. Este nivel de programación textual se caracteriza porque la tarea global debe ser descompuesta en subtareas, de manera que los algoritmos que el robot seguirá puedan ser interpretados fácilmente por los controladores que a su vez darán movimiento a los actuadores.
- **Nivel Objeto:** En este nivel de programación la complejidad se reduce notablemente ya que las órdenes descritas no están dirigidas al robot como tal sino que describen el accionar general del sistema basados únicamente en cómo deben situarse o colocarse los objetos inmiscuidos en el proceso global. Cabe destacar que las instrucciones programadas en este nivel requerirán generar instrucciones a nivel robot para ser descargadas en los controladores.
- **Nivel Tarea:** El nivel tarea comprendería únicamente una única sentencia que ya no especifique ni los objetos a moverse ni como el robot lo hará sino únicamente especificara cual es el objetivo que el robot deberá cumplir.

El estado actual de la programación de robots se mantiene en el primer nivel debido principalmente a la necesidad de generar códigos desde los niveles superiores a nivel robot, el mismo que es indispensable para los movimientos propiamente del robot, además el nivel robot permite especificar muchas características de los movimientos, normalmente necesarias en una aplicación industrial. Existen gran cantidad de lenguajes de programación desarrollados para el primer y segundo nivel de programación textual, como se mencionó anteriormente cada uno responde a robots específicos de las diferentes empresas mundiales de manufactura de robots industriales. (Véase Tabla.1.)

1.4. MÉTODOS DE PROGRAMACIÓN DE ROBOTS INDUSTRIALES.

Existen diversos criterios para realizar una clasificación de los métodos de programación de robots. Algunas atienden a la potencia del método, mientras que otras clasificaciones hacen referencia al sistema empleado para indicar la secuencia de acciones a realizar por el robot. Este segundo criterio ilustra de mejor manera los métodos implementados por las empresas que lideran en la actualidad el mercado de la robótica industrial. Además establece de forma muy general que un robot puede ser programado de dos formas diferentes el denominado por programación guiada o mediante un procedimiento textual; Existen robots además que conjugan ambos tipos de programación e incluso pueden estructurar secuencias de acción que basadas en ambos tipos de programación pueden definir un proceder más complejo para los actuadores robóticos. Debido a la complejidad que cada uno de los tipos de programación tiene es necesario el estudio de cada uno por separado y además los considerados subtipos de programación tanto para el método de guiado como para el textual.

1.4.1. Programación por guiado.

La programación por guiado o también conocida como de aprendizaje asistido consiste en una manipulación directa de los actuadores del robot por medio de una maqueta del mismo o llevándolo a las diferentes posiciones gracias a un controlador de posición embebido, de una u otra manera el posicionamiento de los diferentes puntos o trayectorias debe ser registrado por el robot para la posterior repetición de los mismos.

Considerando el estado de los actuadores en la programación guiada se pueden considerar dos subtipos de programación. Si suponemos que los actuadores están desconectados y la fuerza para establecer los diferentes puntos o trayectorias del robot provienen directamente del programador, es decir este toma al robot desde su extremo para guiarlo hasta los puntos requeridos al tiempo que el controlador registrara automáticamente la señal de los sensores de posición, este es considerado como una programación guiada pasiva directa.

Una variación de la programación guiada que resuelve el problema de manipular físicamente el robot es la cual se realiza por medio de un maniquí o doble del robot,

este doble posee la misma configuración del robot pero es mucho más ligero facilitando así la manipulación del mismo, al igual que en la programación pasiva directa el programador deberá llevar al maniquí a los diferentes puntos o trayectorias que desea el robot alcance al tiempo que el controlador almacena automáticamente los movimientos realizados, este tipo de programación es muy utilizada por robots de pintura integrados en una etapa final de una línea de fabricación industrial; Este tipo de programación es conocido como programación guiada pasiva por maniquí.

Otra variación de la programación guiada que quizás es la más utilizada en la actualidad es aquella que utiliza una botonera o mando analógico (joystick) normalmente diferenciado del sistema robótico y que por medio del cual controla los actuadores del robot para llevarlo a los puntos requeridos o por las trayectorias deseadas, estos puntos determinados no se registran automáticamente por el sistema de control del robot sino que se almacenan únicamente los expresados por el programador. La mayoría de las empresas de manufacturación de robots industriales de la actualidad poseen este tipo de dispositivo para la programación asistida de sus robots; A este tipo de programación se lo conoce como programación guiada activa.

En la actualidad los paneles de control o de programación han evolucionado acorde a la tecnología. Las principales empresas de manufactura de robots han desarrollado paneles táctiles en donde se conjuga todas las características de la programación guiada e incluso de la programación textual para el control óptimo y sintetizado de sus robots industriales [11].



(a)



(b)

Figura 4. Ejemplos de paneles de control.

a) KUKA SMARTPAD b) Reis Robotics ReisPAD [33, 34]

Considerando una segunda clasificación dependiendo de la potencia del sistema se definen dos tipos más específicos de programación guiada. El denominado guiado básico y guiado extendido.

El método de programación guiado básico el robot es consecutivamente guiado por los puntos por los que se quiere que este pase, durante una llamada fase de ejecución automática del programa. Durante esta etapa el controlador del sistema interpola estos puntos por medio de una trayectoria, es decir que los puntos siempre serán repetidos secuencialmente tal como se grabaron lo que dificulta la adicción de un controlador, el muestreo que el robot realiza sobre los puntos debe realizarse con una frecuencia muy alta, ejemplos de robots que utilicen este tipo de programación son los dedicados a pintar, pulir o soldar diferentes elementos que tengan un contorno específico y tengan que repetirse varias veces.

El método de programación de guiado extendido especifica los puntos por los que deberá pasar el robot, pero además define datos relativos de velocidad, tipos de trayectoria, precisión etc. Datos que son almacenados en conjunto con los puntos programados; normalmente este método de programación se realiza por medio de la botonera o mando analógico que el robot disponga. A pesar que el método de programación guiado es sumamente utilizado y es considerado además como indispensable por su facilidad de uso y mínimos requerimientos de memoria este método tiene varias limitaciones ya que la programación necesita el espacio propio del robot y en un entorno industrial sería necesario retirar al robot de la línea de producción para programarlo también no se cuenta con una documentación del programa lo que dificulta en gran medida realizar modificaciones en el mismo.

1.4.2. Programación textual.

La programación textual es una alternativa a los métodos de programación por guiado, el método consiste en indicar la tarea al robot por medio del uso de un lenguaje de programación determinado y normalmente específico para cada marca de robot. En este programa el robot recibe una serie de órdenes que son editadas en un programa textual y posteriormente son ejecutadas.

1.5. LENGUAJES DE PROGRAMACIÓN EN ROBOTS INDUSTRIALES.

En la actualidad no existe ninguna normalización en cuanto a los procedimientos de programación de robots. Por el contrario, cada fabricante ha desarrollado su método particular, el mismo que es válido solo para sus propios robots. Los intentos de estandarizar los procedimientos de programación de robots no han tenido el

reconocimiento ni la aceptación necesaria, ejemplos de este tipo de propuestas como los sistemas AL (Algol Like) de la universidad de Stanford en 1974, IRDATA de la universidad de Karlsruhe que fue aceptado como estándar por la VDI de Alemania pero al igual que otros intentos de unificación no han logrado trascender.

La falta de estandarización llevo a las empresas de manufacturación de sistemas robóticos a desarrollar lenguajes de programación propios para sus robots, a pesar que la mayoría tienen varias características similares, el no normalizar los sistemas de programación y comunicación dificulta la interacción de los robots no solo con otros dispositivos de igual o parecida naturaleza sino también con sistemas sensoriales y de control más complejos que permitirían una mayor interacción del robot con su entorno y en especial el desarrollo de proyectos de investigación que pretendan resolver estas u otras deficiencias.

La siguiente tabla es un compendio de las principales empresas dedicadas a la manufactura de robots, todas poseen los métodos de programación guiado y textual más se enfatiza en los sistemas operativos y lenguajes de programación propios de cada uno.

No.	Empresa.	Lenguajes de Programación.	Sistemas Operativos.
1	ABB	Rapid	Windows XP
2	Epson	SPEL+	Windows XP
3	FANUC	TPE	Propio
4	MITSUBISHI	MELFA	RT ToolBox
5	Kawasaki	Kawasaki AS	Propio
6	KUKA	KRL	Windows XP

Tabla 1. Lenguajes de programación y Sistemas operativos de las principales empresas de robótica industrial.

Debido a las necesidades comunes que todo sistema de programación de robots tiene se ha originado cierta afinidad entre la mayoría de empresas dedicadas a la programación de estos, poniendo en manifiesto una serie de características generales que se observan en la mayoría de sistemas. Tradicionalmente los requerimientos generales que se establecen para un sistema de programación de robots se contemplan en mayor o menor medida dependiendo de las empresas dedicadas a la robótica industrial en los siguientes puntos se conceptualiza las principales características [1].

1.5.1. Ambiente de programación.

Considerando la dificultad que supone programar las acciones de un manipulador industrial, la mayoría de los sistemas de programación procuran desarrollarse como interpretados de manera de poder realizar un seguimiento paso a paso en cada momento de la programación, evitando así el ciclo de edición compilación y ejecución en cada sentencia programada que implica entre otros aspectos disminución considerable de productividad y un alto precio en tiempo de programación.

Considerando aspectos técnicos todo sistema de programación de robots debe tener una buena capacidad de depuración y ejecución paso a paso, teniendo especialmente en cuenta la interacción en tiempo real que debe existir entre los distintos equipos de la célula con el controlador del robot, al tiempo que se aconseja una monitorización continua e ininterrumpida del programa. Algunos de los sistemas actuales incluso poseen sistemas operativos multitareas permitiendo el control simultáneo y sincronizado de varios robots o sistemas adjuntos al robot.

1.5.2. Modelado del entorno.

El modelo del entorno se define como la representación que el robot tiene de los objetos con los que interacciona, es decir este modelo especifica características geométricas como posición, orientación de los objetos y las dimensiones de los mismos.

Para definir las características denotadas de los objetos normalmente el sistema asigna a cada objeto una posición y orientación medida desde cierto sistema base, o referencial, frecuentemente conocido como sistema del mundo, de esta manera el sistema puede definir al objeto de manera única.

Modelos de entorno actuales facilitan la programación del robot en lo que refiere a los objetos involucrados en su actuar, este modelo conocido como relacional, permite establecer relaciones entre objetos. Estas relaciones se subdividen conceptualmente dependiendo de la unión física entre los mencionados objetos.

- Objetos independientes.
El movimiento de uno no afecta al otro.

- Dependencia de unión rígida.
El movimiento de uno implica al otro y viceversa.
- Dependencia de unión no rígida.
El movimiento de uno implica al otro pero al revés.

El modelo relacional, una vez programado se actualiza automáticamente durante la ejecución del programa, mediante una estructura que correlacione los diferentes objetos del sistema en general, simplificando así la tarea del programador.

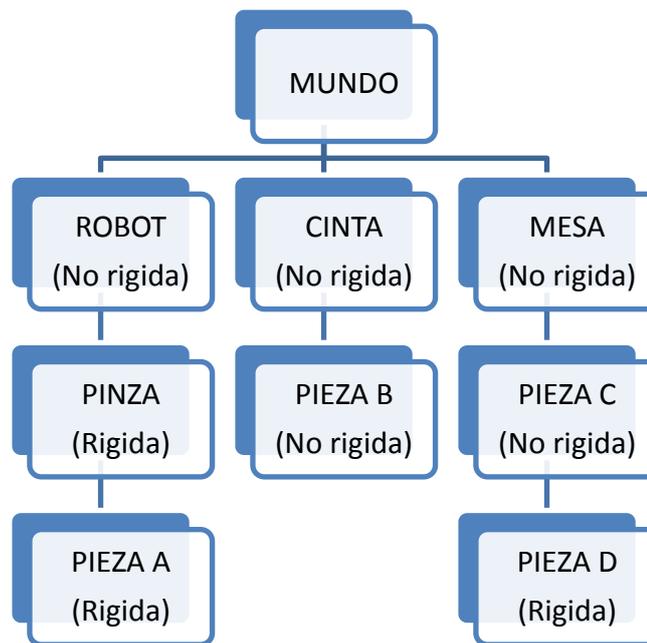


Figura 5. Relación entre diversos elementos de un entorno de programación.

1.5.3. Tipo de datos.

Como todo sistema de programación, aquellos destinados al control de robots deben contar con datos convencionales de programación, tipos de datos como enteros, reales, boléanos, etc. Son necesarios para una programación estructurada y normalmente utilizados como base de control para sentencias que involucren variables, las mismas que definirán funcionalidades del robot con su entorno o de manera específica una trayectoria o acción del robot.

Además de los típicos tipos de datos mencionados existen otros datos necesarios para la programación de robots, estos están destinados específicamente para operaciones de interacción con el entorno. Estos datos a los cuales el robot deberá acceder

definen entonces posición, orientación y los objetos que se encuentran involucrados en el proceso industrial.

El posicionamiento del robot se realiza por medio de coordenadas articuladas, las mismas que indican los valores de posición que cada articulación debe tener para llegar a un punto específico. Se consideran entonces datos referenciales que pueden ser manipulados por el programador, normalmente definidos como coordenadas cartesianas, las mismas que una vez asociadas a un sistema de referencia de coordenadas pueden ser estructuradas como las coordenadas del origen del robot y la orientación de la herramienta del mismo, la representación conjunta de posición y orientación se consigue agrupando las tres coordenadas cartesianas y las de orientación esta representación varía dependiendo del lenguaje de programación utilizado pero como norma todo sistema de programación sigue estos parámetros [11].

1.5.4. Manejo de entradas / salidas (digital y analógicas)

Como norma general al considerar un proceso industrializado completo, los robots deben tener la capacidad de interrelacionarse con su entorno, otras máquinas o elementos del sistema, por lo cual sistemas de comunicación adicionales son fundamentales para la integración y sincronización en los procesos industriales. Considerando el nivel más sencillo de comunicación mediante señales binarias de entrada y salida, el robot puede decidir comenzar una determinada acción o indicar a un dispositivo externo que comience con la suya.

El control de las señales binarias tanto de ingreso como de salida son manipuladas por el robot por medio de instrucciones de activación o desactivación de las mismas. En cuanto a las señales de ingreso el robot tiene la capacidad de leerlas y manipular el funcionamiento del programa en función de ellas, en la programación las señales pueden indicar interrupciones, saltos, condiciones, o incluso comenzar un proceso o ciclo de trabajo automáticamente cuando el controlador del robot recibe una señal determinada. Cabe recalcar que una interrupción debería poder monitorizar de manera automática la condición y aun en movimiento el robot deberá realizar lo citado por la interrupción automáticamente.

Si tomamos en cuenta un nivel superior de comunicación se pueden considerar protocolos estándar que permitan una mejor comunicación del robot con su entorno, tipos de comunicación como de red local, conexión punto a punto serial, entre otros permiten que por medio de un software instalado en un ordenador adicional al sistema robótico controle o supervise el funcionamiento del accionar del sistema robótico. A pesar que este tipo de comunicaciones actualmente no es predominante para la mayoría de empresas de robótica industrial, debido a que consideran suficiente la comunicación del robot con su entorno por medio de señales digitales o analógicas simples, existe una gran variedad de proyectos de investigación que involucran sistemas de comunicación industrial para actuadores robóticos en base a estos protocolos estándar de comunicación, abriendo un extenso campo de investigación y a nivel académico, ya que estos sistemas permitirían integrar todo tipo de sistemas sensoriales, de visión artificial o algoritmos de inteligencia artificial, de una manera más sencilla.

Cualquiera de los sistemas de comunicación mencionados debe cumplir el objetivo de permitir al robot relacionarse con su entorno, eléctricamente se considera a los sensores como la forma más común para lograr este enlace debido a que los sensores sin importar su naturaleza se conceptualizan como un dispositivo que genera una señal normalmente eléctrica en función de estímulos físicos naturales que se encuentren en su entorno o área de sensibilidad (calor, luz, temperatura, presión, etc.). Las señales eléctricas generadas pueden alterar la programación del robot para: modificar la trayectoria, Elegir entre diversas alternativas, Obtener la identidad y posición de objetos y sus características, cumplir con restricciones externas, entre otros.

1.5.5. Control de flujo de ejecución del programa.

Como todo sistema de programación, los lenguajes de programación para robots industriales deben permitir al programador especificar de alguna manera un flujo de ejecución de operaciones. Esto se logra por medio de estructuras de programación comunes como bucles, for, while, repeat, etc. También es de gran importancia la capacidad de procesamiento en paralelo, tanto para el control de varios robots con un solo programa, como para el control de un solo robot en una celda de trabajo cuyo

equipamiento adicional este bajo el control del programa del robot. Para ello se emplea señales de sincronismo y ejecución de tareas en paralelo.

Para la mayoría de aplicaciones es necesario tener además un control de interrupciones mediante las que distintos equipos en funcionamiento puedan interactuar con el robot. Debe tener la capacidad de asignar prioridades, enviar durante el desarrollo del programa señales de activación, desactivación, etc.

1.5.6. Control de movimiento.

Una característica necesaria de la programación de robots industriales es la capacidad de especificar el tipo de movimiento del robot, es decir que además de determinar el punto de destino y el inicial el sistema de programación debe poder determinar si el tipo de trayectoria es de punto a punto, coordenadas o de trayectoria continua. En este último se consideran entonces líneas rectas, interpolación circular etc., además en la programación se debe poder definir parámetros de, precisión, condicionamientos del sistema y velocidad. Este último normalmente tiene como factor común en la mayoría de sistemas de programación definirse como un valor porcentual de la velocidad total del robot para cada línea de acción facilitando así la alteración de la velocidad de todo el programa o de una o varias líneas específicas.

La principal características que define el control de movimiento de un robot industrial es el nivel de control y formas de programación que este tiene en relación a las trayectorias que se pueden programar. Todo sistema de programación de robots tiene características similares sobre este tema, si consideramos por ejemplo que el robot debe seguir una trayectoria recta para evitar obstáculos en su entorno, los sistemas de programación deberán tener un sistema de punto de paso para resolver este problema de manera más eficiente que con una simple trayectoria línea. El modo en que una determinada configuración (posición y orientación) se define como punto de paso, está relacionado con la especificación de la precisión con que se desea el robot alcance dicho punto. El controlador recibe la referencia del programa, y no permite una nueva referencia hasta que el robot no alcance la referencia deseada con la precisión programada. De este modo si la precisión con la que se desea alcanzar una determinada configuración se define como baja, el extremo del robot se encaminara hacia la configuración siguiente sin haber llegado a la anterior y sin tener

que disminuir apenas su velocidad. El resultado final es que el movimiento gana en continuidad y velocidad a costa de perder una precisión innecesaria en esos puntos.

A pesar que de una u otra forma todos los sistemas de programación de robots industriales tienen en común la programación de diversos tipos de trayectorias, cada empresa dedicada a este campo posee formas de programación específicas para sus robots, razón por la cual se profundizara la programación de trayectorias en capítulos siguientes, centrándonos en el lenguaje específico de programación que será utilizado para el proyecto [11].

CAPITULO 2. BRAZOS ROBOT INDUSTRIALES ANTROPOMÓRFICOS.

2.1 INTRODUCCION.

Este capítulo se centra en el estudio específico y concreto de los brazos robot antropomórficos industriales que serán parte de nuestro idea, para los cuales se desarrollara la interfaz de control humano máquina, estos dispositivos concretamente son los robots industriales KUKA KR 16-2, KUKA KR 5-2 ARC HW y Feedback Mentor 35-001, el estudio se concentrara en sus aspectos físicos, características eléctricas, electrónicas, además de realizar un especial énfasis en el sistema de programación que estos robots utilizan o que serán desarrollados, todas las herramientas que poseen para estructurar sentencias de movimiento y funciones de trayectoria, de manera de conceptualizar los procesos necesarios para el correcto funcionamiento de los robots, dichas características serán utilizadas en gran medida debido que la interfaz procesara los datos programados por el usuario y en función de los parámetros ya estandarizados en los controladores permitirán realizar el movimiento del robot para el cual se desarrollaran las librerías de comunicación.

Considerando las características físicas, los robots tienen grandes diferencias, por lo cual el estudio deberá ser diferenciado en este ámbito, al igual que las características de programación y controladores lo que permitirá ampliar el uso de la interfaz para los tres robots en cuestión, aumentada la validez y la aplicabilidad del sistema como tal.

2.2 ARQUITECTURA GENERAL DE UN SISTEMA ROBOTICO.

Un sistema robótico estándar industrial, posee sin distinción de un modelo de manipulador específico los mismos parámetros constituyentes. Estos se consideran necesarios como mínimo para el correcto desarrollo de un proyecto completo o

aplicación con los robots, a pesar que pudieran existir subsistemas adicionales que complementen el accionar del robot o del sistema en general.

Es necesario acotar que existen ciertas variaciones de la arquitectura general, considerando las diferentes compañías que manufacturan los robots, además de las aplicaciones específicas para las cuales algunos robots son creados, por ejemplo aunque los elementos constituyentes mostrados a continuación son indispensables complementos de la arquitectura general, si consideramos los tipos de programación que existen el software o programa utilizado para la programación de secuencias puede ser presentado de diversas formas con la única restricción que implica el utilizarlo para programar las sentencias de control del robot en cuestión.

Los elementos que forman un sistema robótico están presentados a continuación:

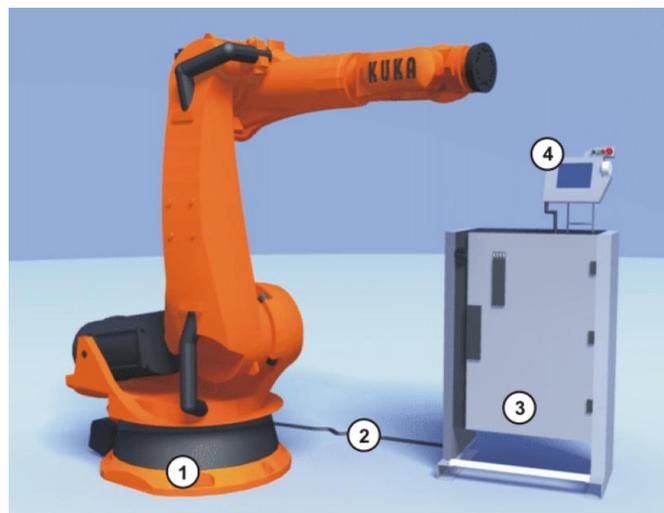


Figura 6. Arquitectura básica de un sistema Robótico [15].

1. Robot
2. Cables de unión.
3. Unidad de control del robot.
4. Unidad de programación.

Como se especificó, en un sistema general, el modelo o tipo del robot es independiente del sistema como tal, las características del robot como se verá más adelante varían en función de parámetros tales como velocidad, alcance, y carga nominal, es decir variables propias de la estructura mecánica como tal.

Los cables de unión son los encargados de transmitir las señales (normalmente eléctricas) de control desde el robot hacia el controlador y viceversa.

La unidad de control del robot, alberga los circuitos eléctricos y electrónicos de control, potencia, seguridad y comunicaciones, dependiendo del robot que se desea controlar la unidad de control varía en tamaño, y ciertas especificaciones técnicas, además de la versión en la que se encuentre, que normalmente actualiza ciertos parámetros de software para el control del robot.

El último parámetro constituyente de un sistema genérico, es la unidad de programación, este componente normalmente está ligado a la versión y características de software y conectividad de la unidad de control, debido a que la unidad de programación es la interfaz de interacción directa con el usuario, espacio en donde se realiza la programación y manejo del robot.

Los componentes constituyentes como se especificó anteriormente, varían de muchas formas dependiendo del robot a controlar y la versión de sus componentes especialmente de software, razón por la cual es necesario definir cada uno de forma separada y especificar características de cada uno, razón por la cual los siguientes apartados se centraron en conceptualizar todos los parámetros específicos de los sistemas robóticos que se utilizarán en el proyecto.

2.3 BRAZOS ROBOT KUKA KR 16-2 Y KR 5-2 ARC HW.

En este apartado se analizará de forma específica las características físicas y técnicas de los manipuladores robóticos de la empresa KUKA, por ser base fundamental del estudio preliminar del proyecto. Los robots al pertenecer a la misma empresa comparten muchos aspectos razón por la cual se abordará las características de su arquitectura en el mismo apartado, haciendo especial énfasis en la morfología de cada uno, considerando a esta como la estructura física, eléctrica y electrónica que todo robot industrial tiene como puntos en común. Estos elementos que poseen ciertas características que cumplen con objetivos específicos, en su conjunto permiten el correcto desenvolvimiento de los robots industriales:

- Estructura mecánica.
- Sistema sensorial.

- Sistema de control.
- Elementos terminales.

Cada uno de elementos antes mencionados serán examinados y conceptualizados para los brazos robot industriales antes mencionados, sobre los cuales se realizara el estudio preliminar, diferenciando cuando sea necesario los elementos que los robots entre si no tengan en común, además se hará énfasis en ciertos elementos constitutivos que conforman los elementos antes enlistados de manera de denotar la importancia que estos tienen en la estructura completa del robot y en su correcto funcionamiento.

2.3.1 Estructura Mecánica.

Mecánicamente, un brazo robot antropomórfico industrial está formado por eslabones unidos por medio de articulaciones rotacionales, las mismas que permiten un movimiento relativo entre cada par de eslabones consecutivos, Este tipo específico de robot industrial tiene sus características en base a la estructura de un brazo humano y al igual que la mayoría de manipuladores industriales actuales, los brazos robots antropomórficos KUKA tienen seis grados de libertad, permitiendo mover cada articulación con respecto a la anterior de forma independiente.

El número de articulaciones está basado en el hecho que es necesario un número mínimo de grados de libertad para posicionar y orientar un cuerpo de cualquier manera en el espacio, tres al menos que definen la posición y al menos dos para la orientación de la extremidad dependiendo de la resolución que se desee el robot tenga, en la cual normalmente el robot cuenta con una herramienta específica para realizar un trabajo determinado.

Como se mencionó anteriormente los brazos robot antropomórfico industriales KUKA poseen seis grados de libertad:

EJES	DESCRIPCIÓN.	FUNCIONALIDAD
1 – 3	Ejes principales	Posicionamiento
4 - 6	Ejes de la muñeca	Orientación.

Tabla 2. Descripción y funcionamiento de los ejes en un brazo robot antropomórfico.

Las siguientes imágenes (Fig.4, Fig.5) muestran la distribución física de los brazos robot y sus grados de libertad.

- Estructura mecánica KUKA KR 16-2

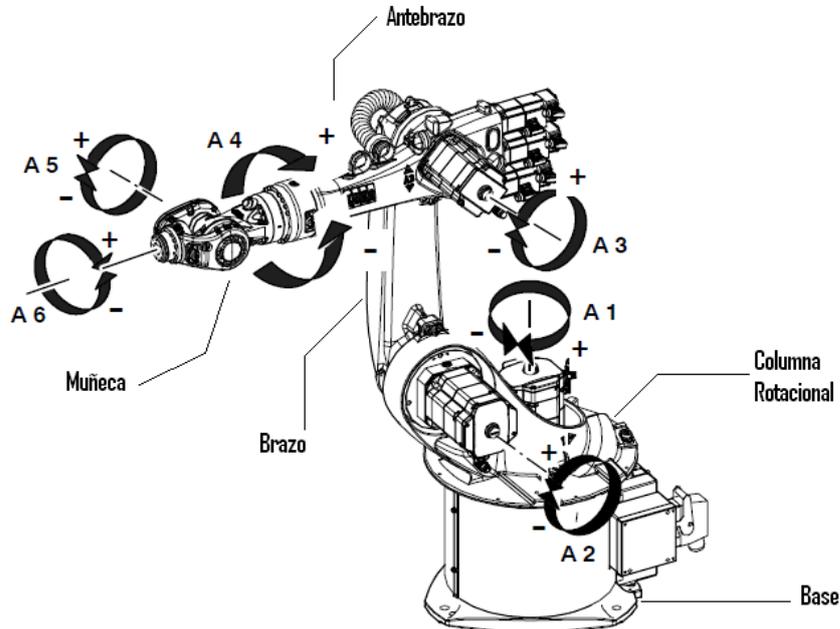


Figura 7. Estructura mecánica Brazo robot antropomórfico KUKA KR 16-2 [16].

Los grados de libertad están distribuidos de manera que el robot solvete el posicionamiento espacial en cualquiera punto de su área de trabajo.

- Estructura mecánica KUKA KR 5-2 ARC

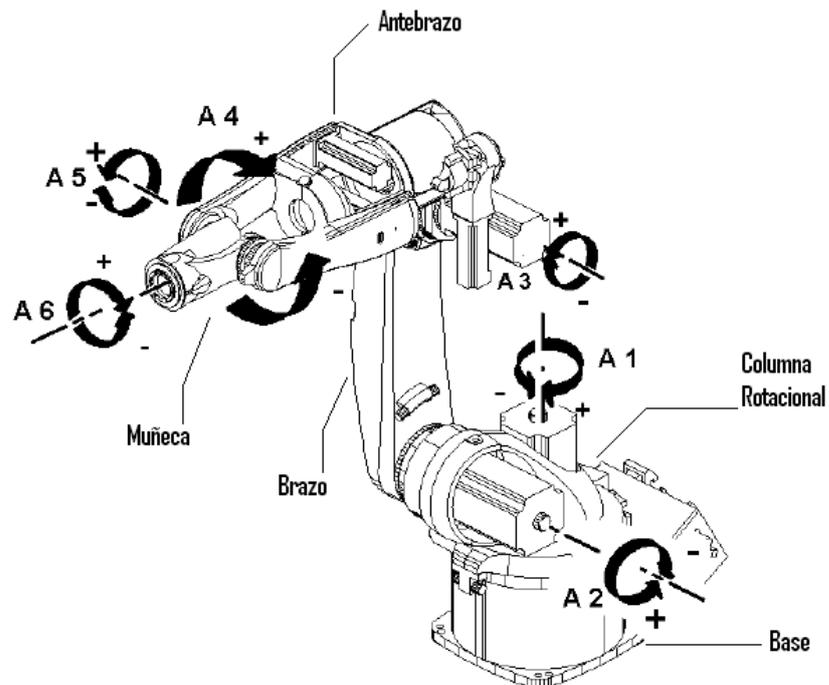


Figura 8. Estructura mecánica Brazo robot KUKA KR5-2 ARC HW [17].

Determinadas las características físicas de un robot antropomórfico, es importante además definir otro parámetro que está relacionado con su estructura física, su área de trabajo, este importante parámetro proporciona información sobre el alcance y movilidad que tiene el robot. Conceptualmente el área de trabajo o también conocido como volumen de trabajo es el espacio de puntos definidos a los cuales el robot puede llegar. Este espacio está definido en base al tamaño y disposición de los eslabones, por lo cual será diferente para cada robot.

Como un estándar informativo, el volumen de trabajo normalmente se expresa por medio de dos gráficos descriptivos de la vista superior y una vista lateral del robot dibujando el volumen al cual el robot puede acceder, ambas vistas están debidamente definidas por cotas estándar en milímetros (mm) mostrando el volumen completo, además de una tabla especificando el alcance máximo, carga nominal y carga extra dependiendo de la herramienta a utilizar, repetitibilidad, unidades de control, etc.

- Área de trabajo KUKA KR 16-2

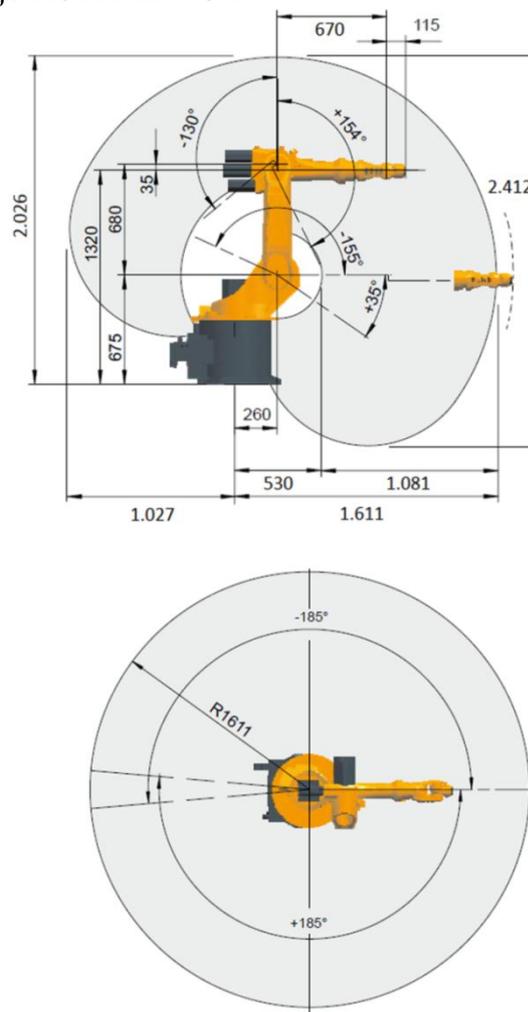


Figura 9. Volumen de trabajo Brazo robot antropomórfico KUKA KR 16-2 [18].

ITEM	DESCRIPCIÓN.	VALOR.
Cargas.	Carga	16 Kg
	Carga Adicional	10 Kg
Zona de trabajo.	Max. Alcance	1610 mm
Otros datos.	Numero de ejes	6
	Repetitibilidad	± 0.05 mm
	Peso	235 Kg.
	Posiciones de montaje	Suelo, techo
	Unidades de control	KR C2

Tabla 3. Especificaciones mecánicas Brazo robot KUKA KR 16-2 [18].

- Área de trabajo KUKA KR 5-2 ARC

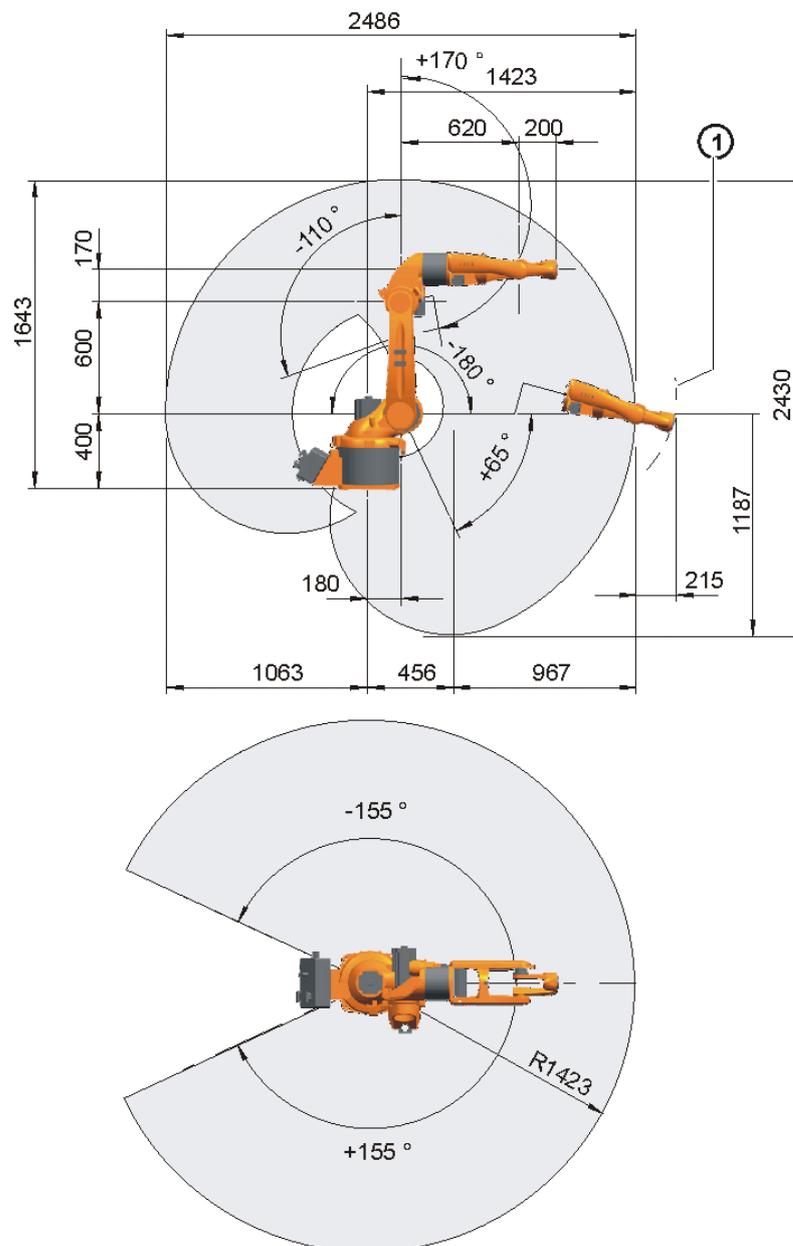


Figura 10. Volumen de trabajo Brazo robot antropomórfico KUKA KR 5-2 ARC HW [16].

ITEM	DESCRIPCIÓN.	VALOR.
Cargas.	Carga	5 Kg.
	Carga Adicional	12 Kg.
Zona de trabajo.	Max. Alcance	1423 mm
Otros datos.	Numero de ejes	6
	Repetitibilidad	±0.04 mm
	Peso	126 Kg.
	Posiciones de montaje	Suelo, Techo
	Unidades de control	KR C2

Tabla 4. Especificaciones mecánicas Brazo robot antropomórfico KUKA KR 5-2 ARC HW [16].

Otro de los parámetros de consideración en la estructura mecánica sobre robots industriales, es la utilización de transmisiones y reductores dentro de sí, estos dispositivos permiten el correcto movimiento del robot facilitando en muchas ocasiones el posicionamiento de los actuadores, así como los algoritmos de control que el robot utiliza para el posicionamiento y movimiento del robot.

Específicamente ambos robots KUKA, que son base del estudio preliminar poseen transmisiones que permiten colocar los actuadores fuera de las diferentes articulaciones lo que conlleva a reducir el peso de ciertas partes del robot, como se puede observar en las figuras anteriores (Véase Fig4, & Fig5), los actuadores rotacionales que definen el comportamiento de los grados de libertad finales de robot, es decir aquellos delimitan la orientación del extremo efector, están posicionados sobre el antebrazo de la estructura mecánica, el movimiento es llevado por medio de sistemas de transmisión hacia los eslabones que conforman la muñeca del robot en donde se posiciona el extremo efector o herramienta del robot.

Existe además sistemas reductores que normalmente están dispuestos en los sistemas electromecánicos de accionamiento, estos permiten no solo acrecentar el torque que los actuadores tienen y la fuerza que estos infligen sobre los eslabones sino que además reducen la velocidad de los motores eléctricos facilitando así el control de posición y velocidad que estos tienen.

2.3.2 Sistema sensorial.

Se puede considerar como un factor común de todo dispositivo robótico ciertos subsistemas de control que permiten la interacción de elementos ajenos al sistema

original del robot, sensores, transductores, otros dispositivos electromecánicos, etc. Estos sistemas basan su funcionamiento en dispositivos que reciben y generan señales que son legibles por el controlador del robot, algunos robots comerciales utilizan controladores lógicos programables o como son globalmente conocidos por sus siglas en inglés “PLC”, que permiten el paso de estas señales hacia el controlador del robot, refiriéndonos específicamente a los robots de la empresa KUKA, estos cuentan dentro de su unidad de control, con una tarjeta de la empresa WAGO que utiliza un sistema de retransmisión basado en el protocolo industrial DeviceNet, esta forma de comunicación permite realizar un control muy sencillo en base a señales digitales o analógicas que son interpretadas por el controlador tras ser estructuradas en un programa descrito en el lenguaje propio de programación KUKA.



Figura 11. Tarjeta WAGO de acople de bus de campo DeviceNet [23].

Los módulos utilizados para la adquisición y generación de señales de control digitales están dispuestos de forma separada entre sí pero direccionados por medio de una única unidad de acople (véase Fig8.), esta unidad central además de direccionar los diferentes puertos en la tarjeta, realiza la transmisión vía DeviceNet hacia el controlador del robot, esta abstracción permite ampliar la utilidad de la tarjeta debido a que se pueden adicionar módulos al sistema original ampliando el número de contactos de ingreso o salida o la naturaleza de las señales que estos utilicen.

Dentro de los controladores de los robots que son causa de nuestro estudio se encuentran dispuestos para ambos manipuladores una tarjeta I/O (ingreso/salida) de señales digitales de las características antes mencionadas. Estas tarjetas poseen 16

contactos eléctricos para adquirir señales digitales y 16 para generación de señales de igual naturaleza.

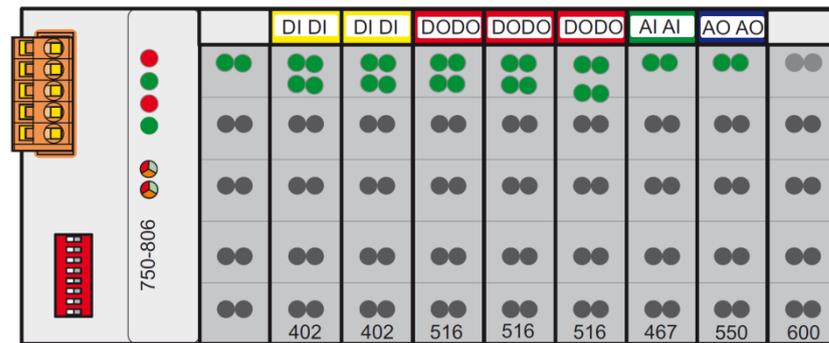


Figura 12. Esquema de los módulos I/O WAGO [22].

La figura anterior esquematiza de manera muy simple la disposición normal de un módulo completo de adquisición y generación WAGO, en la imagen se puede diferenciar la tarjeta de acople, los módulos de ingresos digitales (DI) y salidas digitales (DO), módulos que están dispuestos en los controladores de los robots industriales como se verá posteriormente. Debido a que esta forma de comunicación tiene varios inconvenientes en lo que respecta al nivel de abstracción para con los sistemas que se pudiesen utilizar ajenos al sistema robótico original y por ende no contempla un elemento del todo necesario para la estructuración del proyecto no será analizado a profundidad.

2.3.3 Sistema de control.

Un sistema de control como su nombre indica, es el encargado de controlar los procesos que rigen el funcionamiento de todo el robot, desde adquirir las señales de ingreso de la tarjeta de adquisición de datos o sensores internos, hasta interpretar los programas descritos por el usuario de forma textual o guiada; el controlador procesa la información y en base a algoritmos previamente establecidos envía las señales necesarias hacia los actuadores para que el sistema robótico en su totalidad realice una acción determinada como alcanzar un punto, o seguir una trayectoria dictaminada por el usuario, para una aplicación cualquiera. Al igual que los robots estudiados las unidades de control desarrolladas por la empresa KUKA poseen ciertas características que los determinan, como nomenclaturas específicas y además

tienen todos los subsistemas constituyentes de un sistema de control estándar, siendo estos:

- Dispositivos de almacenamiento.
- Interfaz de usuario.
- Unidad central de procesamiento.
- Unidades de memoria.
- Sistemas de comunicación.
- Controladores de grados de libertad.

Cada uno de estos subsistemas tienen un objetivo para el sistema de control, y todos están dispuestos en un solo módulo, que será definido en base a los robots estudiados. El controlador KR C2 (KUKA Robot Control Versión 2), es un módulo diseñado en base a necesidades individuales, que utiliza componentes estándar de procesamiento garantizando fiabilidad y fácil integración de nuevos sistemas.

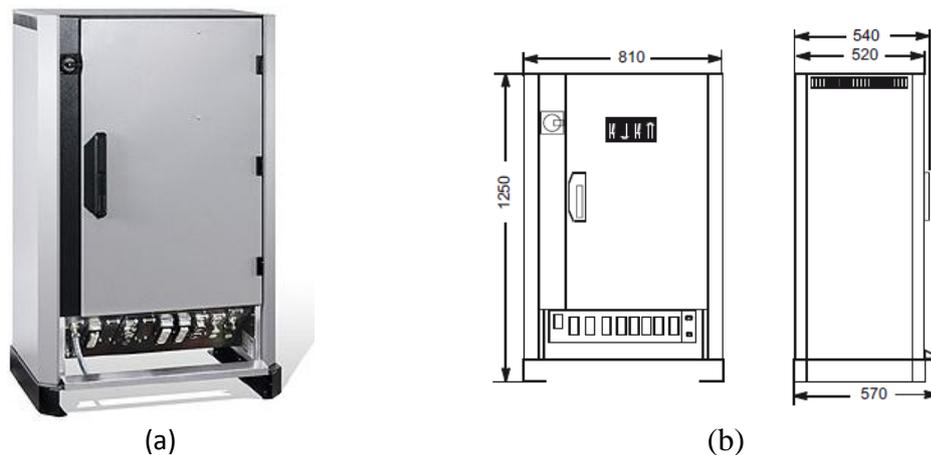


Figura 13. KR C2.

a) Vista Normal. b) Dimensiones y cotas [19].

El control que será utilizado por los robots se destaca por su fácil mantenimiento y nivel de estandarización, debido a que este controlador puede trabajar con robots desde 5Kg hasta 500Kg, es decir de baja, media y alta carga. Físicamente el controlador es un armario de dimensiones estándar (Véase Fig10) dentro del cual se pueden observar los elementos constituidos enlistados anteriormente y de los cuales se realizara una breve explicación.

Los dispositivos de almacenamiento o memoria, procesamiento y sistemas de comunicación se encuentran englobados todos sobre una única computadora

embebida de control, la misma que como ordenador principal del sistema solventa todas las características de estos subsistemas.



(a)



(b)

Figura 14. Computadora embebida.
a) Vista Lateral. b) Vista Superior.

Los sistemas de comunicación estándar que el ordenador posee, están basados en protocolos estándar, tales como RS232 para comunicación serial, USB, TCP para comunicación Ethernet, puertos VGA para la visualización del ambiente de programación e interacción con el usuario y protocolos industriales para la comunicación con las unidades de control para cada grado de libertad.

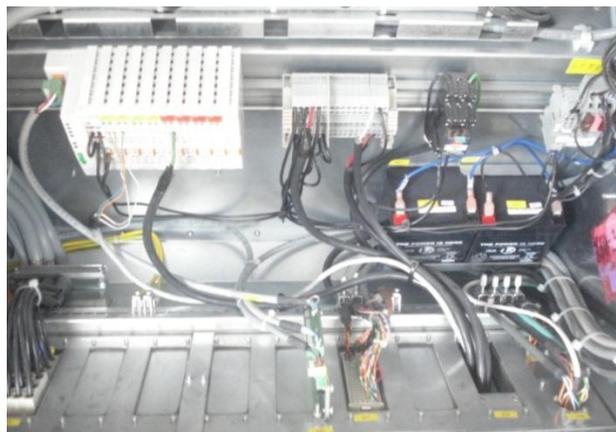


Figura 15. Elementos adicionales del controlador.

Como se puede ver en la figura anterior dentro del armario se encuentran ciertos elementos de seguridad y de comunicación adicionales a los antes definidos, tales

como baterías de respaldo y la tarjeta de adquisición de datos que se conceptualizo en el apartado anterior.

La memoria interna al igual que la unidad central de procesamiento (CPU), están concebidas dentro de la computadora como un sistema integrado completo, cumpliendo así con todas las características que una unidad de control para manipuladores robóticos debe tener.

El último parámetro constituyente del controlador y el de mayor importancia son los controladores de grados de libertad, estos controladores en el caso de los robots KUKA son dedicados para cada uno de los grados de libertad y específicamente rigen sus algoritmos de control al posicionamiento y control de velocidad de los actuadores electromecánicos para cada articulación, como se mencionó anteriormente los manipuladores robóticos que son base de nuestro estudio poseen seis grados de libertad, por lo que el controlador posee seis dispositivos dedicados para cada uno de estos grados de libertad, como se muestra en la siguiente figura.



Figura 16. Controladores dedicados para grado de libertad.

2.3.4 Elementos terminales.

Una de las principales características de los robots industriales de la empresa KUKA, es su gran versatilidad y sencillas formas de cambio de sus herramientas, también conocidas como extremos efectores o elementos terminales. Estos elementos que

varían según la aplicación para la cual se va a utilizar el robot, normalmente poseen ciertos protocolos de comunicación y software dedicado para su uso sobre la plataforma de programación, debido a esto se los considera como sistemas completos e independientes que pueden ser adquiridos e instalados por separado y en función de la necesidad del usuario.

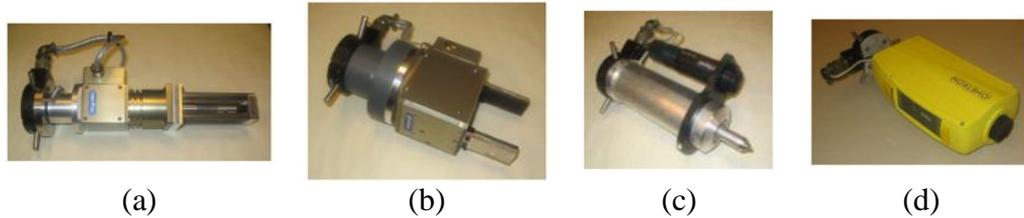


Figura 17. Ejemplos de Extremos efectores.

a) Valve operation tool. b) Gripper tool. c) Vibration sensor. d) Laser vibrometer. [42].

La imagen muestra algunos ejemplos de extremos efectores que la empresa KUKA, dispone para ser adaptados a los dispositivos robóticos, la mayoría de estos sistemas cómo se mencionó anteriormente basan su control y accionar en programas dedicados cuya programación de alguna manera se fusiona con la programación regular, debido a este precepto es muy importante denotar en base al presente proyecto que las sentencias de control que manipulen el actuar de estos dispositivos son diferentes a aquellas que controlan la posición y orientación del robot, por lo que el proyecto no se considera para una aplicación puntual en base a un tipo de herramienta específico.

Los robots sobre los que se realiza el estudio, están instalados con diferentes herramientas cada uno. Para el manipulador KUKA KR 16-2 se cuenta con una garra de sujeción controlada por una electro-válvula neumática, la misma que es accionada por medio de pulsos de control enviados desde la tarjeta de adquisición de datos y controlada por un programa de KUKA dedicado a esta, denominado, “KUKA GRIP-TECH”, la fuerza y el nivel de control que esta herramienta posee, la convierte en una de las herramientas más versátiles y muy útiles para aplicaciones de sujeción y traslado.



Figura 18. Garra de sujeción.

El robot KUKA KR 5-2 ARC HW es un manipulador diseñado con ciertas características especiales que lo hace idóneo para aplicaciones de soldadura, más el dispositivo específico sobre el cual se desarrollara el proyecto refiere aplicaciones de sujeción y control de trayectorias por lo que sus extremos efectores han sido diseñados únicamente para sujeción de elementos, la primera herramienta es un electroimán comandado directamente por una fuente de corriente continua, cuyo control se limita al accionamiento por medio de la tarjeta de adquisición de datos, y un segundo extremo efector es una herramienta para sujetar un elemento de escritura.



(a)



(b)

Figura 19. Herramientas KR 5-2 ARC HW.

a) Electroimán de sujeción. b) Herramienta de escritura.

2.4 BRAZO ROBOT FEEDBACK MENTOR 35-001

Este apartado analizara las características del manipulador Mentor 35-001 de la empresa Feddback al igual que se hizo con sus pares de la empresa KUKA, es decir se analizara características físicas y técnicas del robot, realizando énfasis en la morfología, de manera de poder conceptualizar parámetros que será utilizados en gran medida para el desarrollo de la interfaz, debido a que varios de los parámetros físicos tendrán injerencia en la programación del software final.

La estructura física, eléctrica y electrónica del robot será conceptualizada por medio de la definición los siguientes parámetros.

- Estructura mecánica.
- Sistema sensorial.
- Sistema de control.

2.4.1 Estructura Mecánica.

El brazo robot está compuesto por eslabones unidos por medio de articulaciones rotacionales, es decir un brazo robot antropomórfico, a diferencia de los manipuladores anteriores, el robot Mentor 35-001 posee únicamente cinco grados de libertad que pueden realizar movimientos de forma independiente. Como se mencionó anteriormente se necesita un número mínimo de grados de libertad para posicionar y orientar un cuerpo de cualquier manera en el espacio, tres que definen la posición y para este robot en particular dos que definirán la orientación del extremo efector.

EJES	DESCRIPCIÓN.	FUNCIONALIDAD
1 – 3	Ejes principales	Posicionamiento
4 – 5	Ejes de la muñeca	Orientación.
6	Herramienta	Sujeción de elementos

Tabla 5. Descripción y funcionamiento de los ejes del brazo robot Mentor 35-001.

La siguiente imagen (Fig.18) muestra la distribución física de los brazos robot y sus grados de libertad.

- Estructura mecánica MENTOR 35-001

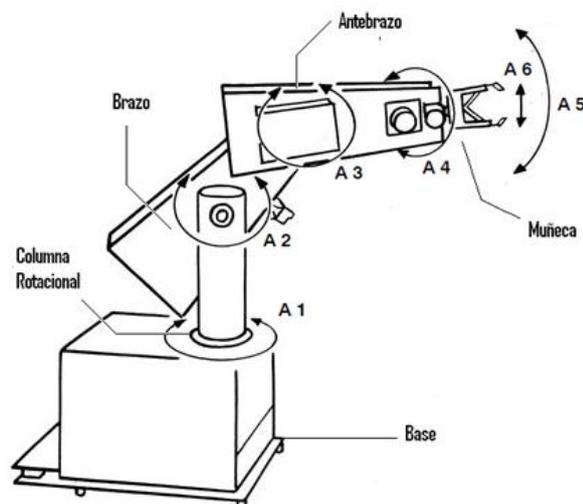


Figura 20. Estructura mecánica Brazo robot antropomórfico Mentor 35-001 [24].

Los grados de libertad están distribuidos de manera que el robot solvente el posicionamiento espacial en cualquiera punto de su área de trabajo.

El volumen de trabajo del robot define el alcance que este tiene, considerando el estándar para la presentación de este criterio se utilizara las imágenes representativas y tabla de especificaciones del robot tal como se hizo anteriormente.

- Área de trabajo Mentor 35-001

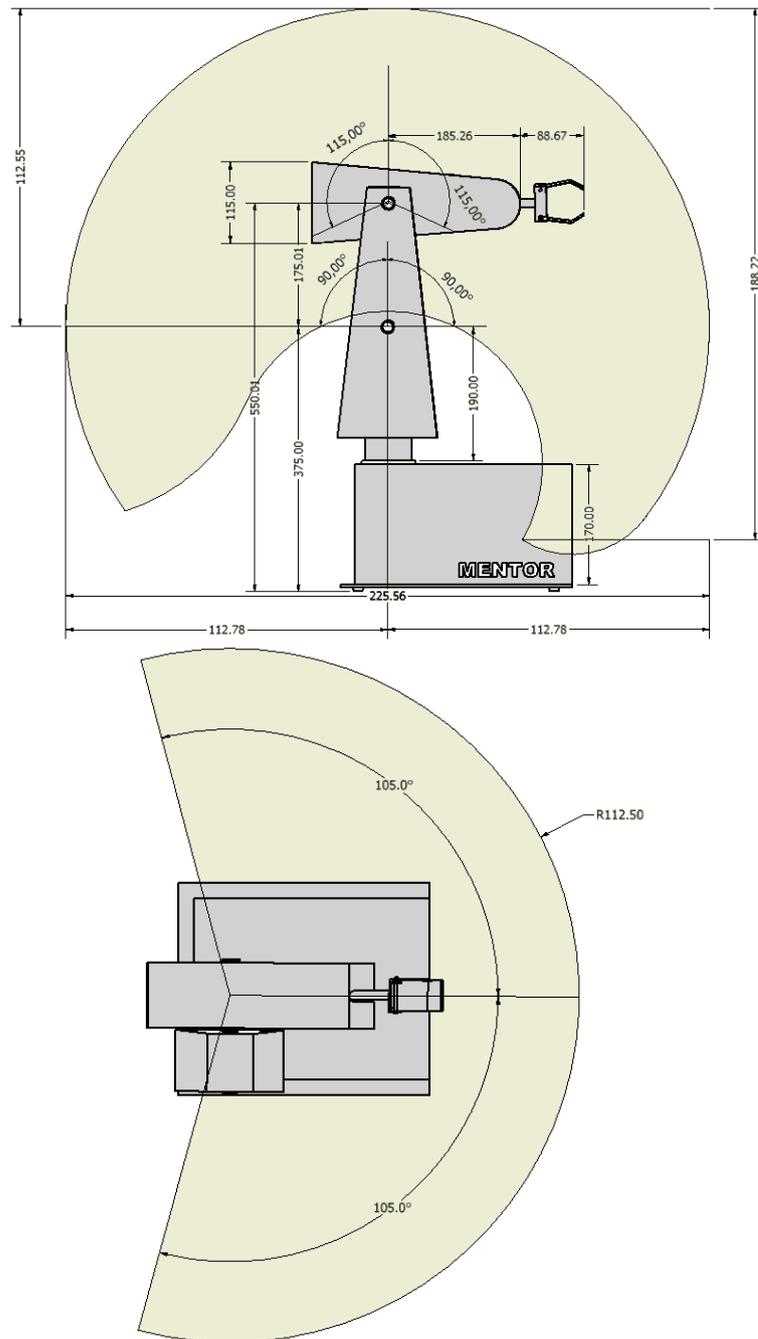


Figura 21. Volumen de trabajo Brazo robot antropomórfico Mentor 35-001.

ITEM	DESCRIPCIÓN.	VALOR.
Cargas.	Carga estándar.	1 Kg
Zona de trabajo.	Max. Alcance	428 mm
Otros datos.	Numero de ejes	6
	Repetitibilidad	±2 mm
	Peso	18 Kg.
	Posiciones de montaje	Suelo.

Tabla 6. Especificaciones mecánicas Brazo robot antropomórfico Mentor 35-001.

Las características físicas del robot, proporcionan a diferencia de los estudiados anteriormente, un acercamiento más práctico a los mecanismos transmisores, reductores y acoplamientos que el robot posee, académicamente y en base a los objetivos del proyecto esta característica permite dotar de mayores prestaciones al robot, razón por la cual los principales mecanismos serán brevemente citados a continuación.

- Sistemas reductores y cajas de engranajes.

Un sistema reductor aplicado específicamente a la robótica cuyo objetivo sea el de mover un eslabón no solo disminuye la velocidad del actuador, en este caso rotacional para el mejor control del posicionamiento del eslabón; sino que además transmite el movimiento de un punto físico a otro aumentando a su vez la fuerza del actuador permitiendo así movilizar un eslabón más robusto o una serie de eslabones conectados entre sí.

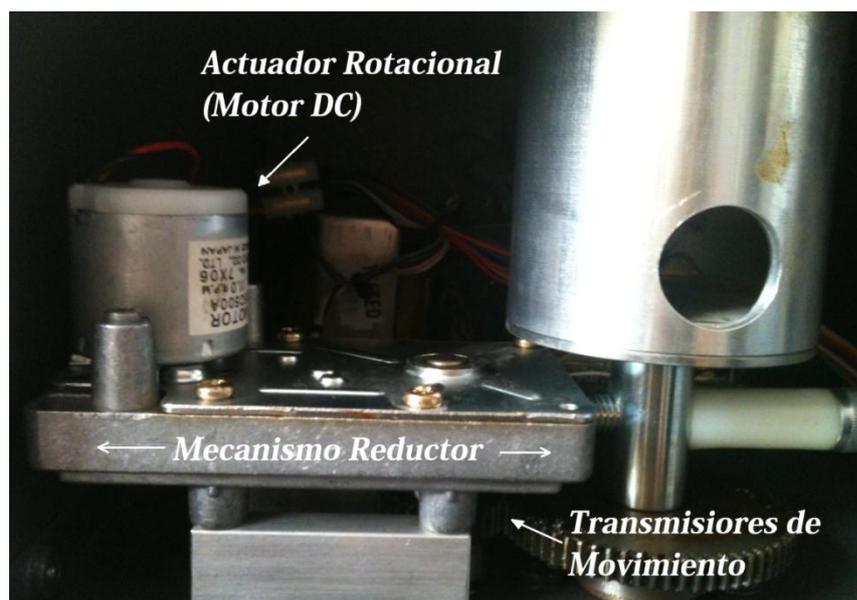


Figura 22. Mecanismo reductor y de transmisión del movimiento Mentor 35-001.

La imagen muestra el sistema completo del actuador para la columna rotacional del robot estudiado, en donde se puede apreciar los elementos que lo constituyen, el actuador rotacional unido a un primer mecanismo reductor o caja de engranes el mismo que transmite el movimiento hacia un segundo complemento de engranes que proporcionan el movimiento más lento y fuerte hacia el eslabón como tal. Cabe recalcar que la distancia entre el eje del actuador y el eje de rotación del eslabón como tal físicamente es considerable más los cálculos de cinemática mostrados en apartados posteriores consideran el actuador como todo el sistema completo facilitando la identificación del eje de rotación y el punto de injerencia del actuador como tal.

- Sistemas de transmisión del movimiento.

Un sistema de transmisión de forma general transfiere el movimiento mecánico producido por un actuador desde un punto a otro, en el caso de sistemas reductores como se mostró anteriormente el sistema puede alterar las características del movimiento como el torque o la velocidad, dependiendo de la aplicación. Más existen otros tipos de transmisores que únicamente utilizan mecanismos como engranes, piñones cables o cadenas para transmitir el movimiento a distancias considerables, el robot Mentor utiliza un sistema de transmisión especializado para accionar la pinza de sujeción que posee como extremo efecto, el actuador de este sistema es un motor de corriente continua que enrolla un cable metálico halándolo a lo largo de todo el antebrazo del robot, este cable a su vez genera un movimiento lineal en la base de la muñeca el cual cierra o abre la pinza, si el cable esta retraído o no respectivamente.

La siguiente imagen muestra los elementos definidos anteriormente del sistema de transmisión, el cable que permite la transmisión del movimiento se encuentra sujetado fuera del eslabón mostrado (antebrazo), al actuador rotacional que está ubicado en el eslabón adyacente (brazo).

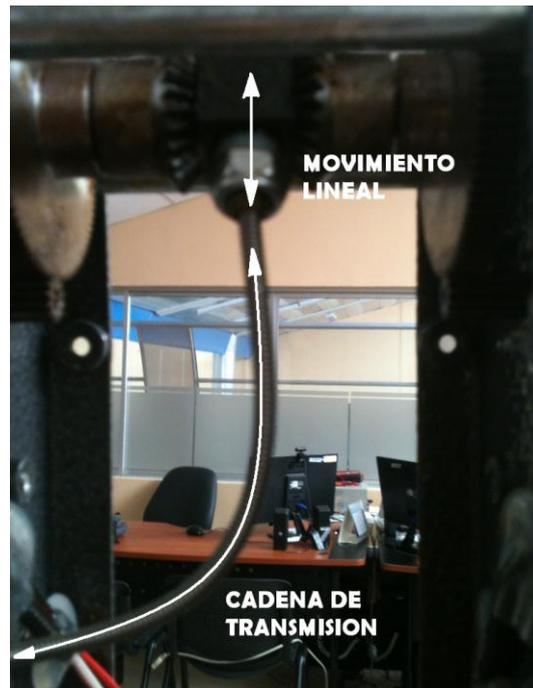


Figura 23. Mecanismo de transmisión del extremo efector del Mentor 35-001.

La utilidad principal del sistema es la de aprovechar todo el torque generado por el actuador rotacional y dirigirlo a la pinza aumentando así la carga máxima que el robot puede tener, además el sistema aliviana el eslabón por donde se realiza el movimiento (antebrazo) permitiendo realizar los movimientos de este con mayor velocidad.

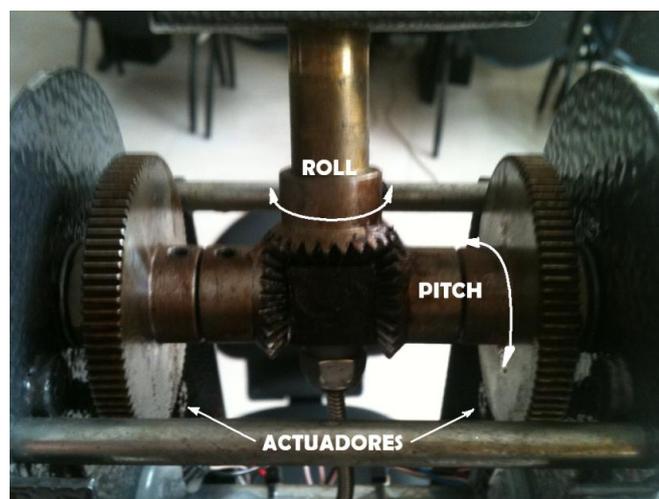


Figura 24. Sistema diferencial mecánico del extremo efector del Mentor 35-001.

Un último tipo de mecanismo, es aquel formado por varios elementos transmisores y actuadores, que estructurados de una forma específica generan un trabajo en conjunto que afectara a uno o más eslabones, en el caso del robot estudiado, este posee un

sistema denominado “diferencial”, el cual está formado por tres piñones engranados entre si los cuales son movidos por dos engranes de acople que a su vez están unidos directamente al eje de los actuadores rotacionales, el sistema diferencial permite el movimiento de dos grados de libertad en un solo eslabón dependiendo de la dirección que los actuadores tomen en conjunto, lo cual es muy útil para realizar los movimientos de orientación del extremo efector del robot. Como se especificó anteriormente el robot Mentor posee dos grados de libertad para definir su orientación, estos grados son conocidos por el movimiento que realizan: rotación e inclinación, más han sido generalizados por su traducción en inglés: Roll y Pitch, La imagen expuesta (Fig.21), muestra el sistema diferencial y sus componentes, cabe recalcar que la utilidad de este tipo de sistemas también se denota en la modelación cinemática debido a la utilidad que implica considerar un solo punto para dos grados de libertad diferentes.

2.4.2 Sistema sensorial.

Los manipuladores robóticos, poseen sistemas internos que proporcionan información útil al controlador para realizar los lazos retroalimentados de control para el posicionamiento o control de velocidad del robot, en el caso del robot estudiado, este posee transductores electromecánicos rotacionales que emiten una señal eléctrica dependiendo de la posición física en la que se coloquen, al estar sobre las articulaciones del robot permite conocer el valor de las variables articulares en base a la señal eléctrica recibida, esta señal deberá ser acondicionada en base a las restricciones de cada grado de libertad de manera de poder utilizar directamente este valor, este acondicionamiento será parte de la interfaz y el usuario podrá manipularla de manera de alterar la relación entre la señal eléctrica y la variable articular.

Otro tipo de sistemas sensoriales son aquellos que permite al robot comunicarse o realizar movimientos en base a sistemas externos, como los mencionados anteriormente para los robots de la empresa KUKA, el robot Mentor considerando sus prestaciones académicas se utilizó para el desarrollo integro de la interfaz razón por la cual el control de los actuadores del robot y todo el monitoreo del mismo se realizara por medio de la interfaz, dotando al robot de gran accesibilidad en base a los parámetros de programación desarrollados en la misma y todas las prestaciones del ordenador, esto concluye que el robot no posee sistemas sensoriales externos

propios, más los atributos dotados por la interfaz permitirán desarrollar sistemas de control en base a cualquier señal o sistema ajena al sistema robótico original.

2.4.3 Sistema de control.

Como se explicó en el apartado anterior el robot estudiado fue utilizado íntegramente para el desarrollo de la interfaz, razón por la cual el sistema de control del robot Mentor se limita a los sistemas de comunicación y a los Controladores para cada grado de libertad desarrollados para el mismo, a diferencia de los parámetros de los robots antes descritos que además poseen dispositivos de almacenamiento, interfaces de usuario, unidades de procesamiento y memoria, parámetros que se encuentran embebidos en la interfaz y serán estudiados posteriormente.

Los driver del robot realizan el control del movimiento de cada actuador que describe un grado de libertad, este control se lo realiza por medio de señales lógicas permitiendo girar los actuadores rotacionales en diferentes direcciones, además alternado los niveles de tensión el controlador también puede realizar control de velocidad para los diferentes grados de libertad. El circuito electrónico equivalente mostrado a continuación permite observar los elementos constitutivos de cada driver y la forma en la que este acondiciona la señal de ingreso para ser emitida con la potencia y tensiones necesarias para mover el actuador.

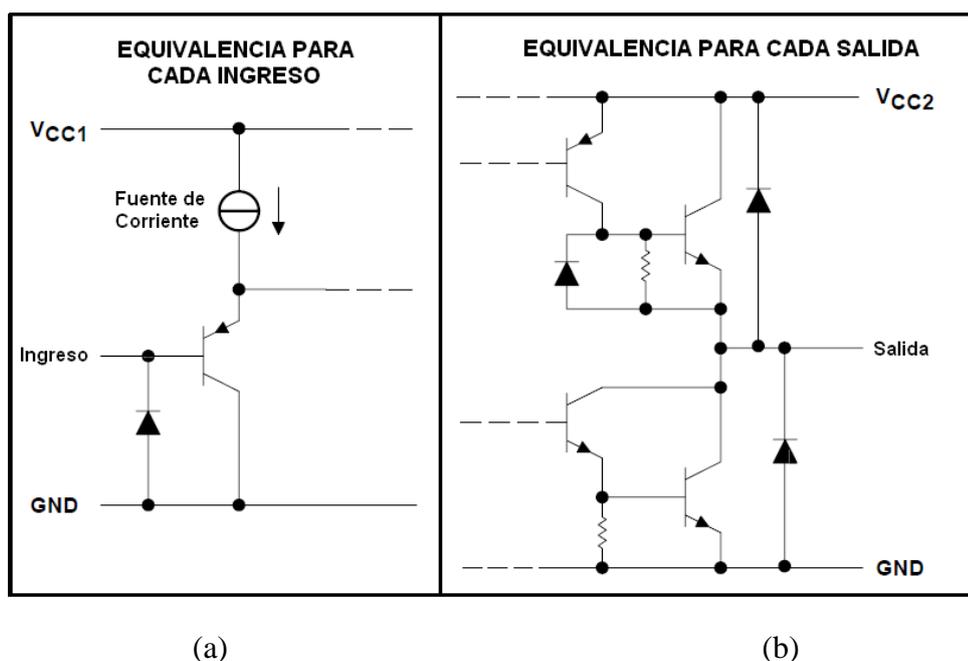


Figura 25. Circuitos Equivalentes Driver de control
a) Circuito de ingresos, b) Circuito de salida [25].

Las características técnicas del encapsulado final del driver dan relación de la utilidad del mismo, considerando los elementos constitutivos del robot estudiado:

PARÁMETRO	VALOR MÍNIMO	VALOR MÁXIMO	UNIDAD.
Tensión de alimentación	4.5	36	V
Tensión lógica niveles altos	2.3	7	V
Tensión lógica niveles bajos	-0.3	1.5	V
Corriente de salida por canal	600	1000	mA.
Corriente pico de salida	1.2	2	A
Temperatura de operación	0	70	°C
Tiempo de transición	300		ns

Tabla 7. Valores de operación típicos de los controladores [25].

Cabe recalcar que los driver se encuentran dispuestos en una sola tarjeta electrónica que dispone los valores de tensión necesarios, además de permitir el paso de las señales de control y censado hacia y desde el robot respectivamente, también posee canales que permiten la conexión física de la tarjeta de adquisición de datos, equipo que realiza la comunicación con el ordenador por medio de un protocolo de comunicación USB, parámetros que serán descritos más adelante.

2.5 CINEMATICA DE ROBOTS ANTROPOMORFICOS.

La cinemática de un robot industrial de manera general se concentra en la descripción analítica del movimiento espacial de un manipulador robótico como una función del tiempo, y al considerar como primordial el posicionamiento y orientación del extremo efector la principal preocupación de la cinemática es establecer una relación entre la base fija o inmóvil del robot y su extremo efector, en función de los valores que sus articulaciones toman. Este apartado analizara brevemente los problemas de cinemática de robots antropomórficos, parámetros que pueden ser utilizados para todos los robot estudiados, considerando las respectivas variaciones físicas, más serán aplicados únicamente a uno de ellos.

Existen dos consideraciones o también denominados problemas fundamentales sobre la cinemática, el primero radica en encontrar la posición y orientación del extremo efector respecto a un sistema de coordenadas que se toma como referencia, normalmente posicionado sobre la base fija del robot. Para determinar las ecuaciones de cinemática directa y con ello el modelo cinemático del robot se necesita conocer

Articulación i	θ_i	d_i	a_i	α_i
1	q_1	d_1	a_1	+90
2	q_2	0	a_2	0
3	q_3	0	a_3	-90
4	q_4	d_4	0	-90
5	q_5	0	0	+90
6	q_6	d_6	0	0

Tabla 8. Parámetros de Denavit y Hartenberg del brazo robot KUKA KR 16-2 [4].

$$A_3^0 = \begin{bmatrix} C1C2C3 - C1S2S3 & -S1 & -C1C2S3 - C1S2C3 & C1C2C3a3 - C1S2S3a3 + C1C2a2 + C1a1 \\ S1C2C3 - S1S2S3 & C1 & -S1C2S3 - S1S2C3 & S1C2C3a3 - S1S2S3a3 + S1C2a2 + S1a1 \\ S2C3 + C2S3 & 0 & -S2S3 + C2C3 & S2C3a3 + C2S3a3 + S2a2 + d1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 1. Matriz de transformación homogénea que relaciona los tres primeros grados de libertad de un brazo robot antropomórfico KUKA KR 16-2.

$$A_6^3 = \begin{bmatrix} C4C5C6 - S4S6 & -C4C5S6 - S4C6 & C4S5 & C4S5d6 \\ S4C5C6 + C4S6 & -S4C5S6 + C4C6 & S4S5 & S4S5d6 \\ -S5C6 & S5S6 & C5 & C5D6 + d4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 2. Matriz de transformación homogénea que relaciona los tres grados finales de libertad de un brazo robot antropomórfico KUKA KR 16-2.

$$T = A_3^0 A_6^3$$

Ecuación 3. Matriz de transformación homogénea completa para un brazo robot antropomórfico KUKA KR 16-2.

La siguiente consideración del modelo cinemático es el problema de la cinemática inversa, tal como su nombre lo indica realiza lo contrario a la cinemática directa ya que en lugar de definir la posición del extremo efector en función de los valores articulares de cada uno de sus grados de libertad, la cinemática inversa considera como parámetros conocidos al punto y orientación del extremo efector y en función a esto determina que valores articulares deberían tener los diferentes grados de libertad, en el caso de los robots que son caso de nuestro estudio, estos valores articulares serían ángulos que los actuadores electromecánicos deberían tener.

Debido a que se considera el actuar de un usuario, que determine la trayectoria que el robot debe seguir, es necesario denotar que las ecuaciones de cinemática inversa son más utilizadas, ya que el usuario únicamente tiene injerencia sobre los puntos de inicio y fin de una trayectoria y el tipo de trayectoria a utilizarse, esto será la base para realizar el control cinemático en posteriores capítulos y las funciones de trayectoria que estarán programadas en la interfaz y sobre la cual se podrán realizar las diferentes pruebas de todos los tipos de trayectorias posibles.

2.6 PROGRAMACION EN KRL (KUKA ROBOT LENGUAJE).

Este apartado analizara las características específicas del lenguaje de programación de KUKA, acotando el estudio preliminar que se realizó para la futura implementación del software realizado en ambientes académicos utilizando los robots de la empresa KUKA, este lenguaje de programación permite de manera general controlar los movimientos de los robots y todas las características de estos movimientos, las particularidades del software se compartirán con las programadas en la interfaz, el software además permitirá la comunicación con el ordenador por medio de un protocolo de comunicación definido.

De forma general, los componentes que conforman el software de programación KUKA, son:

- KUKA System Software (KSS) 5.x.
- Windows XP embedded 2.x incl. Service Pack 2.
- Windows 95 hasta KUKA System Software 4.x.

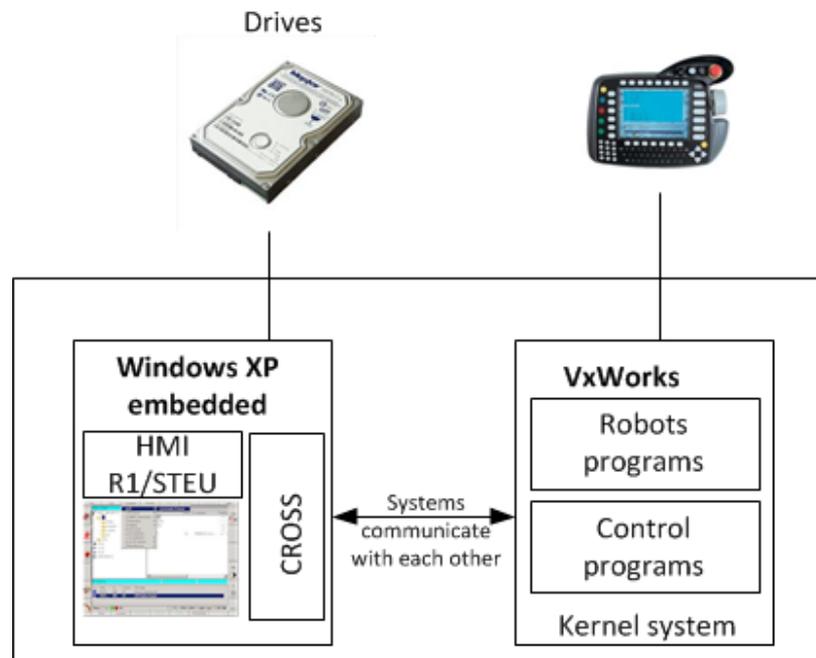


Figura 27. Concepto de software Windows/ VxWorks [15].

Sobre estos componentes se encuentran definidos programas que facilitan la interacción con el usuario, normalmente definidos como una interfaz, humano maquina estructurada sobre el KCP. El entorno de programación está estructurado al igual que un software en el sistema operativo Windows cualquiera con una serie de

menús desplegables y ventanas, a las cuales se puede acceder de forma sencilla por medio de botones de acceso rápido.

Se denota entonces sobre el KUKA System Software, 3 grupos de usuarios, diferenciados por las tareas del personal del robot, estos tres grupos de usuarios son:

- Usuario:
 - ✓ El usuario puede realizar tareas de puesta en servicio (Ajuste, medición de herramientas, medición de base.)
 - ✓ Creación y corrección de programas de trabajo simples mediante formularios de línea.
- Experto:
 - ✓ Ampliación de la programación por el lenguaje de programación superior KRL.
 - ✓ Creación de programas de trabajo para operaciones complejas mediante técnicas de subprogramas o aplicación de diversos bloques.
 - ✓ Utilización de programación de movimiento numérica.
 - ✓ Proyección de sistemas de bus de campo (Profibus o Devicenet)
- Administrador:
 - ✓ Configuración adicional de Plug-Ins.

El control que se realiza sobre el robot, está basado en los movimientos que la estructura mecánica propiamente realizara, de forma muy general se pueden englobar todos estos movimientos bajo en concepto de trayectorias, es decir formas definidas en las que un robot se movilizara entre dos puntos o más puntos; Existen múltiples formas de programar un robot como se describió en capítulos anteriores, mas todas las formas de programación o las herramientas existentes realizan o definen una llamada “función de trayectoria”, relación matemática que expresa como se movilizara el robot, cumpliendo una trayectoria específica para moverse de un punto a otro.

2.6.1 Creación de programas secuenciales y de asignación de trayectorias.

Utilizando el KCP, el programador puede estructurar diferentes sentencias de control y dependiendo de sobre qué grupo de usuario se encuentra, el usuario tendrá la oportunidad de realizar movimientos del robot o en relación al ambiente del mismo.

Un programa secuencial se realiza posicionando al robot, en diferentes puntos dentro de su volumen de trabajo, al grabar estos puntos el usuario puede definir ciertas características en el programa de manera de que el robot se mueva sobre estos puntos con una trayectoria específica y con una velocidad determinada, luego de estructurar estas sentencias de acción el robot puede repetir estos movimientos de forma automática o guiada por el usuario, cumpliendo así con una secuencia de acción completa para cualquier aplicación específica.

Todos los programas descritos en KRL, están estructurados como un lenguaje textual, la programación asistida implica el movimiento del robot por medio del KCP, estos movimientos implicaran en el programa textual puntos a los cuales el robot deberá acceder y se visualizaran como líneas de comando descritas de forma textual. De forma general todo programa descrito en KRL está estructurado de igual forma, con una línea de iniciación INI, donde efectúa la comprobación de los datos para iniciar el movimiento del robot, además se denotan líneas en donde el robot toma una posición inicial, llamada HOME, entre estas dos líneas se desarrollara todo el programa.

```

1  INI
2  PTP HOME  Vel= 100 % DEFAULT
3
4
5  PTP HOME  Vel= 100 % DEFAULT

```

Figura 28. Programa base de una programación KRL [15].

Como se mencionó anteriormente los movimientos que el robot realizara estarán en base a funciones de trayectoria definidas por el usuario, estas trayectorias se programan en KRL por medio de líneas de formulario para cada movimiento. Estas instrucciones para la programación de movimientos, lógica de entrada-salida, entre otros, dependen además de software dedicado que el robot puede tener, como es el caso del programa para el control de una herramienta específica. Las principales sentencias de control que se pueden programar están enlistadas en la siguiente tabla:

Movimiento	
PTP	Instrucciones de movimiento “Punto a punto”, “Lineal” y “Circular”
LIN	
CIRC	
Lógica	
WAIT	Funciones de espera dependientes de tiempo.

WAIT FOR	Funciones de espera dependientes de una señal.
OUT: OUT	Funciones de conmutación simples.
OUT: PULSE	Funciones de impulso simples.
OUT: SYN OUT	Funciones de conmutación dependientes de trayectoria.
OUT: SYN PULSE	Funciones de impulso dependientes de trayectoria.
IBUS-Seg. On/off	Acoplar o desacoplar un segmento Interbus.
Comentario	
NORMAL	Inserción de una línea de comentario en un programa.
SELLO	Línea de comentario con fecha y hora en un programa.
Asistente KRL	
PTP	Movimiento absoluto “Punto a punto”
PTP_REL	Movimiento relativo “Punto a punto”
LIN	Movimiento lineal absoluto.
LIN_REL	Movimiento lineal relativo.
CIRC	Movimiento circular absoluto.
CIRC_REL	Movimiento circular relativo.

Tabla 9. Sentencias de programación para KRL [15].

De las sentencias enlistadas, las funciones de trayectoria se estructuran utilizando de forma común las sentencias de movimiento, cada una refiere una trayectoria diferente por lo que es necesario especificar las características de cada uno.

- PTP (Punto a punto): El robot se mueve sobre los puntos guardados, a lo largo de la trayectoria más rápida.
- LIN (Lineal): El robot se mueve a lo largo de una trayectoria lineal con una velocidad definida.
- CIRC (Circular): El robot se mueve a lo largo de una trayectoria circular.

Estos movimientos pueden realizarse de dos formas diferentes, considerando la forma en la que el robot se aproximara a los puntos de la trayectoria, las formas son por medio de un movimiento aproximado o exacto. Las sentencias condicionales además de estructuras de programación como bucles y condiciones se las realiza de igual forma, con la diferencia que el acceso a estas, está restringido a los usuarios experto y administrador. Una vez que el programa secuencial este estructurado con las expresiones o sentencias de control para las diferentes trayectorias que los robots realizaran, se puede establecer el funcionamiento manual de toda la secuencia programada o de igual forma el funcionamiento automático con una velocidad determinada.

2.7 INTRODUCCIÓN A LA COMUNICACIÓN INDUSTRIAL DE BRAZOS ROBOTS ARTICULADOS KUKA.

Los manipuladores robóticos, casi en su totalidad y sin considerar la empresa a la que pertenezcan tienen ciertos sistemas o parámetros que permiten una comunicación con elementos externos a su sistema original, esto significa que los movimientos del robot están en función de señales o sentencias de control descritas por un sistema externo, que normalmente es un sensor, ordenador, programador lógico u otro robot. En cualquiera de estos casos el controlador del robot utiliza un protocolo de comunicación definido para el envío y recepción de datos desde el sistema externo hacia el robot y viceversa. Las diferentes empresas de manufactura de robots diseñan sus prototipos con protocolos de comunicación que son estructurados por medio de sentencias de software dispuestas en el mismo sistema operativo o lenguaje de programación. En este apartado se finaliza el estudio introductorio realizado para los actuadores robots de la empresa KUKA, haciendo énfasis en la conceptualización y uso del software necesarios para la futura implementación del software de control en los mencionados robots.

La empresa KUKA ha desarrollado software dedicados para todos los tipos de comunicación industrial que tiene a su disposición. De forma individual y por medio de sentencias básicas condicionales un robot KUKA puede realizar la comunicación con otros sistemas, por medio de la tarjeta de adquisición de datos de su subsistema sensorial, si se desea considerar al robot en un conjunto industrial con otros dispositivos de igual naturaleza por ejemplo en una línea de ensamblaje, varios robots KUKA, se pueden conectar simultáneamente por medio de un ordenador lógico programable (PLC) y hacia un ordenador, esta estructura jerárquica permitirá que los robots trabajen en conjunto.

Adicionalmente a los protocolos antes mencionados los manipuladores de la empresa KUKA, tienen la característica de ampliar su gama de sistemas de comunicación por medio de software dedicados específicos para cada uno de los protocolos, estos además varían dependiendo de la versión del controlador que gobierna al robot.

De forma general se puede especificar dos protocolos de transmisión de datos que permiten la interacción con robots de la empresa KUKA, el protocolo RS232 de transmisión serial y el protocolo TCP/IP para la transmisión vía Ethernet.

Los robots KUKA se comunican, con otros dispositivos sobre puertos seriales por medio de RS-232, utilizando un protocolo muy complejo pero efectivo para asegurar la transmisión con un error mínimo, este proceso implica entonces que la programación tanto en KRL, como en el ordenador o sistema que se comunicara con el controlador del robot, deberá regirse a todos los parámetros que contempla el algoritmo para minimizar el error al mínimo, a pesar que la comunicación serial bidireccional asíncrona es un protocolo de comunicación muy fiable que continua siendo utilizada en innumerables aplicaciones, tiene ciertas desventajas sobre todo en lo respectivo a velocidad de transmisión, razón por la cual para aplicaciones en las que se pretende controlar el robot en tiempo real es común utilizar el protocolo TCP/IP vía Ethernet.

Como se comentó anteriormente la empresa KUKA, desarrolla software dedicados para aplicaciones específicas, en el caso de ampliar el control de los manipuladores robóticos por medio de un sistema externo al original utilizando un protocolo de comunicación vía Ethernet, KUKA utiliza los software KUKA.Ethernet KRL XML y KUKA.Robot Sensor Interface 3.1, ambos son paquetes adicionales que de manera general utilizan el protocolo de comunicación antes mencionado para comunicarse con sistemas externos, los siguientes puntos muestran las características más importantes de cada uno de los software mencionados.

2.7.1 KUKA.Ethernet KRL XML.

El software es un paquete de tecnología adicional, con las siguientes funciones y características:

- Transmisión de datos del controlador del robot hacia un sistema externo y viceversa.
- Envío y recepción de datos por medio de un programa en KRL.
- Hasta nueve sistemas externos pueden conectarse al controlador.
- El controlador del robot se considera siempre como un cliente que se conecta a un servidor que será el sistema externo.
- Los datos recibidos se almacenan de manera de que los datos no se pierdan.

- Una etiqueta XML entrante puede establecer directamente una salida del controlador.

Los datos se transmiten mediante Ethernet, por el protocolo TCP/IP y por medio de sentencias XML, bajo las siguientes características:

- El programa descrito KRL puede enviar sentencias como un solo elementos siempre y cuando no tenga más de 80 caracteres.
- El algoritmo de Nagel incorporado en el protocolo TCP / IP optimiza el tamaño de paquete para la transmisión de datos en el bus. Así, no es posible predecir el tamaño de los paquetes enviados en el bus.

La aplicación en el sistemas externo debe determinar por si misma si una estructura XML está completa, basándose en la formación del mismo. Estos pueden ser archivados mientras se espera por ejemplo el fin de una etiqueta en el elemento raíz.

Si se pretende que el sistema externo únicamente transmita datos hacia el robot la implementación es muy sencilla, solo es necesario enviar sentencias XML bien formadas hacia el controlador del robot.

Si el sistema externo también recibe datos del controlador, es necesario en la mayoría de casos, integrar un analizador XML en el servidor, simplificando así la lectura de los datos y evitando sentencias complejas.

Los componentes del software, los componentes del software y su interconexión, están descritos a continuación:

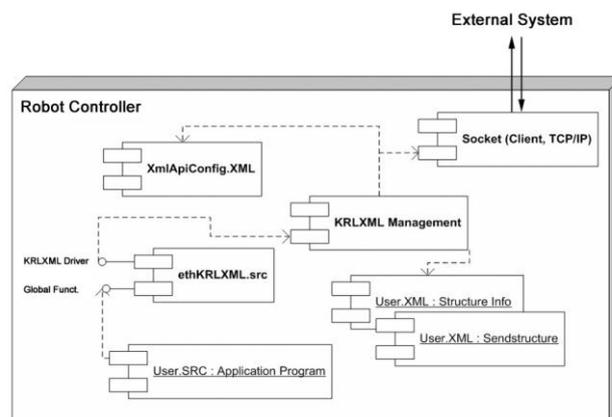


Figura 29. Estructura e interconexión del KRL XML [20].

COMPONENTE.	DESCRIPCIÓN.
Socket. (Cliente TCP/IP)	Interface para el intercambio de datos con un sistema externo.
Manejo KRLXML.	Núcleo del sistema.
ethKRLXML.src	Archivo fuente que contiene todas las funciones del usuario.
XmlApiConfig.xml	Archivo donde están definidos los parámetros de comunicación. <ul style="list-style-type: none"> • Nombre del dispositivo. • Dirección IP. • Número del puerto.
User. XML	Archivo donde la estructura XML de los datos a ser transmitida está definida: <ul style="list-style-type: none"> • Información de estructura: estructura XML para recepción de datos. • Estructura de envío: estructura XML para transmisión de datos.
User. SRC	Programa de aplicación que se utiliza para el intercambio de datos.

Tabla 10. Componentes KRL XML [20].

2.7.2 KUKA.Robot Sensor Interface 3.1.

El software es un paquete de tecnología adicional, con las siguientes funciones y características:

- Transmisión de datos del controlador del robot hacia un sistema de censado externo y viceversa.
- Envío y recepción de datos vía Ethernet o utilizando el sistema de ingresos y salidas del controlador del robot.
- Procesamiento cíclico de la señal y evaluación de la frecuencia de ciclo.
- Influencia en los movimientos del robot o la ejecución del programa procesando las señales de sensores.
- Para el intercambio de datos por medio del sistema de ingresos y salidas I/O, los datos y señales del sensor son leídas y escritas por medio del software y procesadas para el control de un programa en KRL.

Para el intercambio de datos por medio de Ethernet, el controlador del robot se comunica con el sistema sensorial en tiempo real, utilizando el protocolo UDP/IP, en donde el usuario debe configurar los datos en un archivo XML.

- La transmisión cíclica de datos desde el controlador al sistema de censado y viceversa se realiza de forma paralela, a la ejecución del programa, es decir se puede realizar diferentes operaciones para con el robot al mismo tiempo.

La aplicabilidad de estos métodos de transmisión de datos hacia un sistema externo, específico, siendo este un ordenador en donde se realizara la interfaz de control para los movimientos y funciones de trayectorias de los robots, es la base fundamental del diseño e implementación de la interfaz como tal, la cual se desarrollara en capítulos posteriores.

CAPITULO 3. DESARROLLO DE LA INTERFAZ.

3.1 INTRODUCCIÓN.

El presente capítulo describe la estructura de diseño, programación y componentes de la interfaz de control como tal, cada uno de los parámetros utilizados y principales algoritmos utilizados para solventar los problemas de comunicación, control y monitoreo del robot en cuestión a utilizarse para el proyecto. El diseño de la interfaz pretende satisfacer todos los parámetros asociados con los objetivos del proyecto, razón por la cual las herramientas de control y monitoreo están realizadas bajo parámetros académicos acordes al uso que se pretende implementar, esta razón también afectó la elección del software en el cual se desarrolló la programación de toda la interfaz.

El capítulo además expondrá los conceptos necesarios sobre entornos de programación integrados, los diferentes ambientes y técnicas de programación y la intercomunicación entre ellos, de manera de dotar a la interfaz de mayores prestaciones académicas aumentando así su utilización.

3.2 ENTORNOS DE DESARROLLO INTEGRADOS (IDE)

Un entorno de desarrollo integrado, conocido por sus siglas en inglés IDE, es un programa informático que puede ser dedicado bajo un único lenguaje de programación o puede utilizar varios; De forma general está compuesto por un conjunto de herramientas de programación, normalmente utilizados como herramientas de aplicación, en conjunto todo IDE posee los siguientes componentes básicos para su accionar:

- Editor de código.
- Compilador.
- Interprete
- Depurador
- Interfaces gráfica de usuario.

Estos componentes permiten al usuario estructurar programas que serán utilizados para un aplicación definida y dependiendo de la aplicación permitirán incluso la comunicación con sistemas externos, para su comunicación, control o monitoreo.

3.2.1 Matlab y Simulink.

Como se mencionó anteriormente, un IDE puede trabajar bajo distintos ambientes de programación manteniendo cierto nivel de intercomunicación entre ellos de manera de utilizar los mismos componentes básicos de acción como son compilador, interprete, depurador además de aumentar las prestación que el software de manera general puede ofrecer al usuario para realizar o solventar un aplicación definida.

Debido a la naturaleza del proyecto es necesario utilizar un programa que ofrezca no solo la posibilidad de programación textual o secuencial sino además un entorno de programación que permita simular y comunicar en tiempo real los programas o algoritmos descritos para el control o monitoreo de sistemas reales como es en el caso del proyecto un sistema electromecánico complejo, debido a esto se optó por utilizar un programa que no solo cumple con estos parámetros sino que además ofrece un ambiente de programación muy conocido en el ámbito académico; MATLAB y su plataforma de programación y simulación grafica SIMULINK, serán los programas sobre las cuales se desarrollara la interfaz y algoritmos de comunicación tanto de control como de monitoreo para los brazos robot industriales.

MathWorks es la empresa que realiza el software de desarrollo; Mundialmente conocida como líder en el desarrollo de software de cálculo matemático, y sus programas son acreditados como la principal herramienta para procesos de investigación, innovación y desarrollo. Matlab es un entorno de programación para desarrollo de algoritmos, análisis de datos, visualización y cálculo numérico. Por su parte Simulink es un entorno gráfico para la simulación y el diseño basado en modelos de sistemas dinámicos multidominio y embebidos. La compañía produce cerca de 100 productos adicionales para tareas especializadas, como el análisis de datos y procesamiento de imágenes [37].

Los componentes o herramientas que los sistemas utilizan para el desarrollo de las diferentes aplicaciones se definen como son mundialmente conocidos, por su traducción en inglés “ToolBox” estas herramientas son variadas y engloban

prácticamente todos los procesos de diseño, simulación e implementación de todo sistema eléctrico, electrónico o mecánico, permitiendo así la integración de diversos sistemas de control para ser implementados en una simulación de un sistema electro mecánico o estructurado para ser utilizado en uno de forma real.

Los IDE pueden generar aplicaciones por si solas o pueden ser parte de aplicaciones existentes; Esta característica presente también en el software elegido permite la creación de interfaces de usuario adaptativas y flexibles, programables de manera amigable al usuario y que pueden internamente utilizar todas las herramientas, mencionadas anteriormente que el software posee; esto permitirá la manipulación de la interfaz como tal por parte del usuario de manera que este pueda interactuar no solo con los ejemplos propuestos si no que permitirá también evaluar diseños propios estructurados del usuario.

3.2.2 Descripción del IDE.

A pesar de las marcadas diferencias entre los entornos de programación del software de desarrollo Matlab y Simulink, estos productos comparten una gran estructura de aplicaciones, procesos en paralelo y generación de código, que permiten al usuario realizar procesos en conjunto, permitiendo analizar o procesar de manera más completa la información.

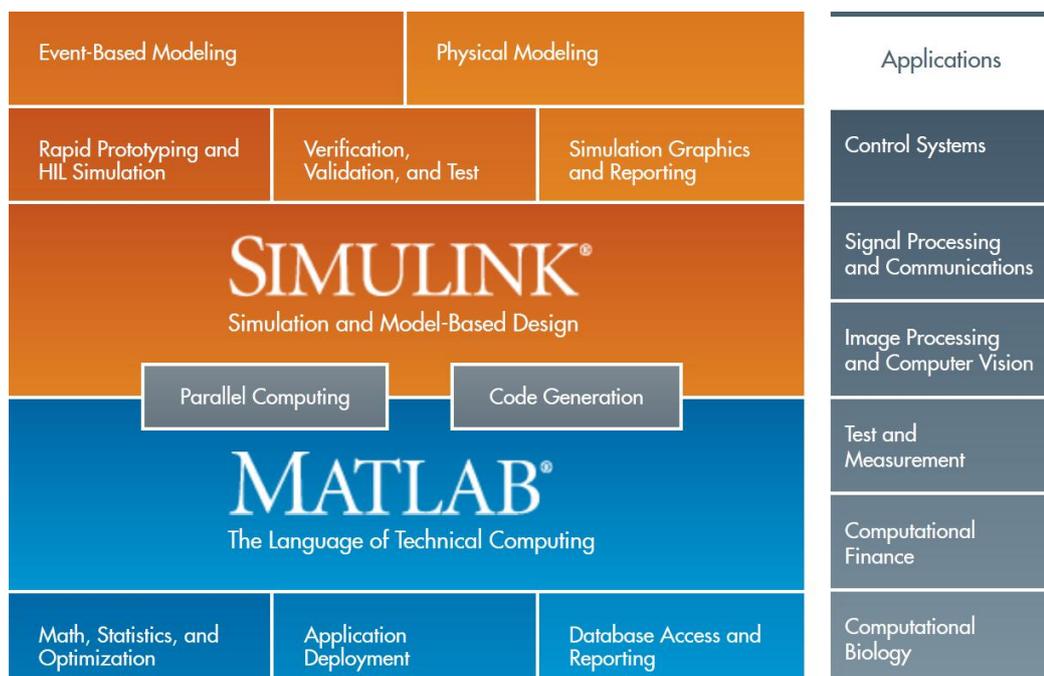


Figura 30. Estructura y componentes del IDE Matlab y Simulink [36].

La imagen muestra el mapa completo de la estructura del IDE, especificando las principales funciones que cada uno tiene por separado además de resaltar las aplicaciones más importantes o herramientas en los campos de implementación más conocidos.

Sobre la estructura es importante resaltar dos aspectos principales que afectan directamente el desarrollo de la interfaz, al poder desarrollar una aplicación en base a una interfaz gráfica el código desarrollado podrá ser concebido de forma individual dando como resultado parámetros de control que no afectaran el actuar de otros algoritmos desarrollados en la plataforma de simulación de modelos reales, y además de ser necesario la interconectividad entre los dos entornos permiten utilizar algoritmos o códigos iguales en ambos entornos permitiendo incluso la creación de nuevas librerías, que en el caso del proyecto serán en su mayoría sobre comunicación hacia sistemas reales.

3.2.1 Métodos de programación gráfica.

Considerando la gran cantidad de prestaciones que el ambiente de programación grafica representa para el proyecto, es necesario denotar ciertos métodos y conceptos básicos sobre este tipo de programación, su funcionamiento y técnicas de desarrollo.

De forma general, un entorno de programación gráfico, permite realizar operaciones o desarrollar algoritmos por medio de la utilización de bloques, al igual que en una programación clásica de forma descrita, la secuencia o la forma en la que se utilicen ciertas herramientas o líneas de código dan como resultado una acción diferente en una función o estructura normalmente descrita por medio de variables, de forma parecida la secuencia de bloques que estarán intercomunicados entre sí en el ambiente de programación grafico darán como resultado el cambio de una variable de un estado a otro para una aplicación específica.

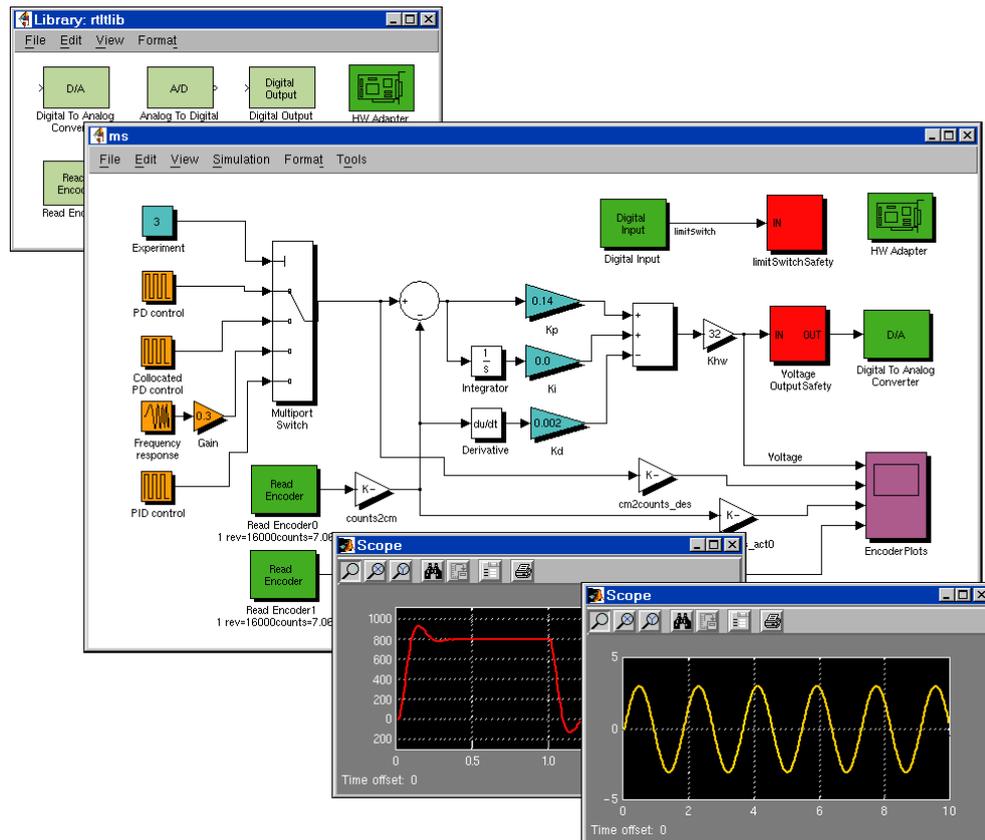


Figura 31. Ejemplo de programación gráfica [38].

A pesar que la programación gráfica permitiría realizar cualquier función, matemática o algoritmo de decisión los bloques están desarrollados en su mayoría para aplicaciones de simulación y para el desarrollo de procesos de control hacia sistemas reales, permitiendo realizar diversos métodos de comunicación y de documentación.

Los principales métodos de programación gráfica, dependen de la plataforma sobre la cual se realiza la misma, considerando el ambiente escogido para la realización del proyecto los métodos de programación, se limitan al uso de librerías que permiten desarrollar modelos de sistemas reales ya sean eléctricos, electrónicos, mecánicos o biológicos. Estos sistemas a su vez pueden sufrir cambios en relación al modelo original con la ayuda de transductores de señal, este precepto permite cambiar una señal considerada de un sistema modelado real a una señal matemática que está expuesta a cambios para el procesamiento de la misma. Por medio de un segundo grupo de librerías la señal se procesa dependiendo de la aplicación para la cual se desarrollara el proyecto, permitiendo en una última instancia el uso de estas variables

para la documentación de las mismas o el control de un sistema externo en base a estas y a parámetros de comunicación predefinidos.

Otros métodos de programación que no refieren el control de variables básicas de programación, constantes para una función matemática, ni señales o parámetros propios de sistemas reales, están definidas en un tercer grupo de librerías que adquieren datos directamente de tarjetas embebidas del ordenador o aparte del sistema general, librerías o herramientas de procesamiento de imágenes, adquisición de datos para aplicaciones de audio y voz, permiten la manipulación directa de estas variables para ser utilizadas en cualquier aplicación netamente de software o para ser empleadas como referencias de control para sistemas reales externos.

Debido a la versatilidad y veracidad de los resultados que los sistemas simulados o modelos matemáticos diseñados en Simulink tienen, existen varios programas que permiten interconexión con otros software, de manera de dotar de mayor realismo o permitiendo una mejor percepción del sistema real modelado que se pretende controlar. Es así que varios complementos como la visualización de sistemas mecánicos o patrones de realidad virtual son muy utilizados en la actualidad para dar realce a los modelos realizados en programación gráfica, complementos que serán utilizados en la interfaz y serán descritos posteriormente [37].

3.3 DESARROLLO DE LA INTERFAZ DE CONTROL.

La interfaz de usuario desarrollada pretende como objetivo principal el controlar los movimientos del robot industrial, en base a funciones de trayectorias, o parámetros de movimiento definidos por el usuario, considerando la aplicación de esta en ámbitos académicos se pretende que la interfaz no solo facilite la identificación de las plantas robóticas como tal, sino también permita realizar reportes sobre el uso de la interfaz, cierto nivel de dinamismo para la estructuración aleatoria de trayectorias, control offline de los robots en base a un modelo simulado, además de control online para el caso de movimientos reales, esto compuesto con los diferentes métodos y sentencias de monitoreo permitirán asimilar y poner en práctica los conceptos sobre robótica en lo que respecta a la estructuración y cálculo de funciones de trayectoria. Una segunda etapa que permitirá mayores prestaciones en niveles académicos inferiores se referirá a la realizada en el ambiente de programación grafica en donde

se permitirá no solo realizar comunicación directa con los actuadores sino alterar también parámetros propios de control del robot.

3.3.1 Diseño de la interfaz gráfica de usuario.

El diseño preliminar de la primera etapa de la interfaz de control consiste en una interfaz humana máquina que pretende ser fácil de utilizar para cualquier estudiante, quien podrá utilizarla como herramienta para el entendimiento de criterios y aplicación de conceptos de robótica, razón por la cual será necesario de conocer nociones básicas sobre robótica industrial para la correcta utilización del software mas no conocimientos avanzados en programación textual propiamente, ya que toda la programación de la interfaz será reducida al máximo.

Las herramientas utilizadas para el desarrollo de la interfaz gráfica de usuario, deberán tener acceso a todos los paquetes de herramientas de Matlab además de tener interconectividad con Simulink, razón por la cual se optó por desarrollar esta primera etapa íntegramente utilizando el entorno de desarrollo de interfaces graficas de usuario propia de Matlab, conocido por sus siglas en ingles GUIDE (GraphicalUserInterfaceDevelopmentEnvironment).

En base a los parámetros mecánicos eléctricos y electrónicos mencionados en capítulos anteriores, que el brazo robot que es base de nuestro estudio posee se desarrollaron los diferentes elementos de control de la interfaz, los cuales serán descritos individualmente a continuación.

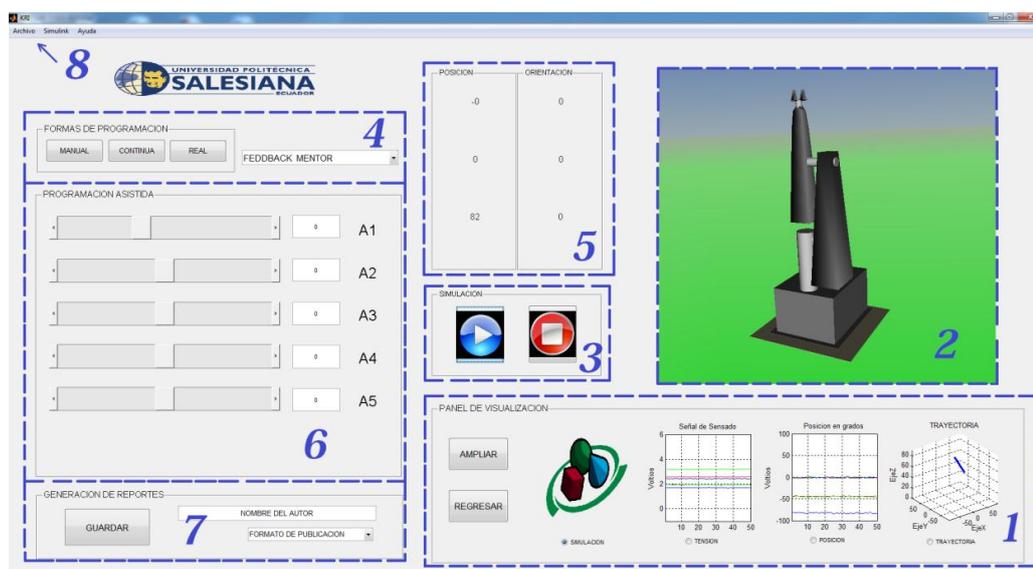


Figura 32. Vista principal de la interfaz hombre máquina.

Como se puede observar en la imagen, la primera etapa de la interfaz está constituida de varios campos, cada uno de ellos está desarrollado para solventar un proceso o una aplicación de la interfaz, cada uno de los campos son dinámicos es decir que cambian en función de los parámetros que el usuario elija, mas su posición y función general es la misma a pesar que los parámetros sean específicos dependiendo de cada aplicación.

En base a la numeración descrita en la imagen anterior, (Fig23) se procederá a enumerar cada uno de los campos para su posterior explicación y función en la interfaz.

1. Paneles de visualización y control de secuencias.
2. Área de simulación y monitoreo.
3. Controles de simulación y accionar real.
4. Menú de elección de robots y formas de programación.
5. Visualización de la posición y orientación del robot.
6. Paneles de programación.
7. Panel de generación de reportes.
8. Menú general del programa.

Cada uno de los campos dinámicos puede variar entre diferentes paneles que realizaran diversas aplicaciones o funciones en la interfaz, pero poseen cierta correlación entre ellos, razón por la cual comparten la misma área en el diseño de la interfaz gráfica.

3.3.2 Paneles de visualización y control de secuencias.

El primer campo de la interfaz refiere al área en donde se realizara la adquisición de datos que formara la secuencia final de movimientos del brazo robot para grado de libertad del mismo, este panel posee herramientas que adquieren la posición del robot de las diversas formas de programación y las almacena en una base de datos que posteriormente dará la pauta de movimientos al brazo robot en cuestión.

	COLUMNA (A1)	HOMBRO (A2)	CODO (A3)	ROLL (A4)	PITCH (A5)	YAW (A6)	PINZA
1	0	0	0	0	0	0	1
2	93	45	-81	0	0	0	1
3	93	45	-81	112	-35	0	1
4	-18	67	-104	112	-35	0	1
5	-18	67	-104	0	0	0	1
6	0	0	0	0	0	0	1

Figura 33. Panel de control de secuencias.

La imagen permite apreciar, las herramientas que se posee el primer panel, entre las anteriormente mencionadas se tiene la tabla que muestra la base de datos adquirida por los diversos métodos de programación y los botones para guardar la posición del robot, además se cuenta con botones para borrar la base de datos, controlar el actuar de la pinza y llevar al robot a su posición inicial o denominada “home”; Posición que suele ser muy importante para el cálculo cinemático y correcto posicionamiento del robot.

Sobre el mismo campo se encuentra el segundo panel de la interfaz, el actuar de este panel denominado como: “panel de visualización”, está subordinado al actuar del panel de control de secuencias ya que como su nombre lo indica este panel permite visualizar todos los parámetros adquiridos del robot y mostrarlos en el área de simulación y monitoreo después de realizada y ejecutada la secuencia de movimientos.

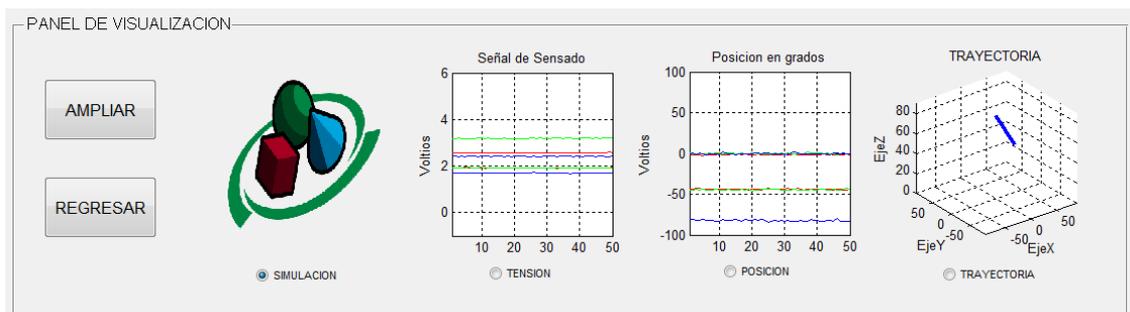


Figura 34. Panel de visualización.

La imagen permite observar todas las variables que el panel de visualización permite monitorear, estas variables denominadas como: simulación, tensión, posición y trayectoria, permiten visualizar en el área de simulación y monitoreo, el modelo en realidad virtual del robot elegido, los niveles de tensión adquiridos de los sensores de posición, la posición en grados de los principales grados de libertad y una imagen en tres dimensiones de la trayectoria realizada por el robot durante la secuencia estructurada, respectivamente. Debido a la dependencia que este panel tiene con respecto al panel de secuencias las herramientas del usuario se limitan a escoger cuál de las variables se desea ampliar en el área de visualización y si se desea regresar al panel de control de secuencias.

3.3.3 Área de simulación y monitoreo.

El segundo campo que constituye la interfaz gráfica de usuario es la denominada: “Área de simulación y monitoreo”, este campo se encuentra constituido por un campo grafico dinámico que en función de los parámetros elegidos en el panel de visualización permite observar las diferentes variables denominadas en el apartado anterior. La principal característica de este campo es la respuesta en tiempo real del modelo en realidad virtual, durante la fase y en función de los parámetros de los paneles de programación, además de la respuesta de cada una de las variables posterior al movimiento del robot.

El tema sobre el modelo de realidad virtual, no solo se considera una herramienta de visualización sino que es el principal método de aproximación offline para la programación de los movimientos del robot, el modelo realizado a su vez no solo posee características físicas y visualmente iguales al robot que es parte de nuestro estudio, sino que además realiza una gran aproximación al modelo real permitiendo utilizarlo para la estructuración de ecuaciones de trayectorias y cinemática del robot, proporcionando acceso al estudiante a programar y probar los movimientos del robot sin necesidad de moverlo físicamente, lo cual resulta muy útil al no tener seguridad de la respuesta física resultante de la programación se esté realizando.

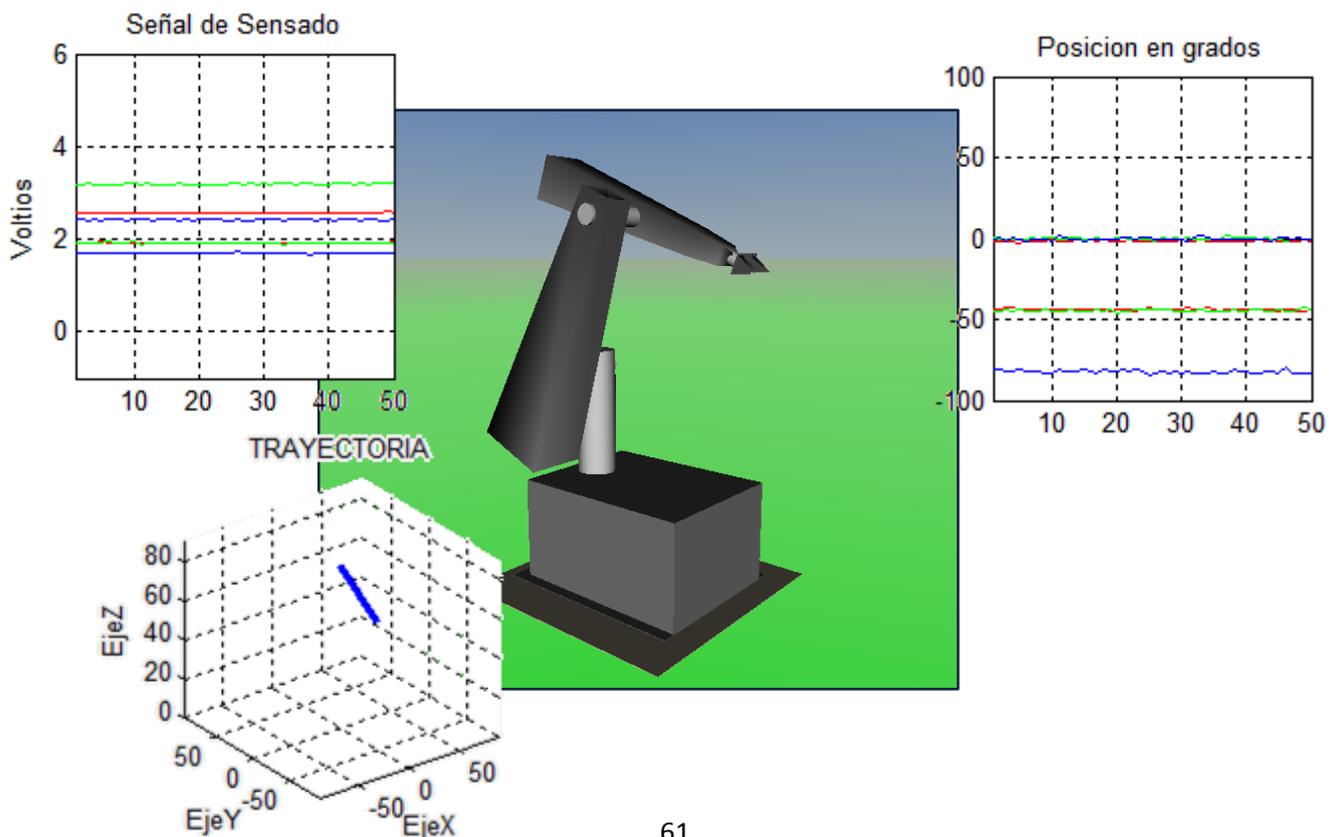


Figura 35. Imágenes de las variables del área de monitoreo.

3.3.4 Controles de simulación y accionar real.

El tercer campo refiere al panel de simulación, este panel posee las herramientas básicas de simulación para comenzar y finalizar o parar la misma, más estas herramientas pudieran considerarse como las más importantes de la interfaz debido a que estas no solo controlan la simulación sino que además accionan los movimientos reales del robot y permiten el cambio entre los paneles de control de secuencias y visualización, permitiendo así el monitoreo completo del robot tras realizar el control de los movimientos del mismo.



Figura 36. Panel de simulación y accionar real.

Cabe recalcar que el algoritmo principal descrito en la interfaz, que relaciona el actuar conjunto de los diferentes paneles así como la comunicación con los robots y cuyo principal ejecutor se describe en este panel será profundizado en apartados posteriores de forma global.

3.3.5 Menú de elección de robots y formas de programación.

El cuarto campo de la interfaz, permite al usuario escoger que robot se pretende controlar; La interfaz cambiara ciertos parámetros de programación dependiendo del robot que se escoja, sobre todo en lo que refiere a los grados de libertad que cada robot tiene, las variables que forman parte del informe de sesión, además inicializara los parámetros de comunicación de ser escogido el robot sobre el cual se realizaran los movimientos reales.

El otro grupo de herramientas que posee este campo son las formas de programación que permiten al usuario escoger que tipo de programación desea realizar y en base a esto permite visualizar los diferentes paneles de programación los mismos que agregaran parámetros de movimiento a la secuencia descrita en el panel de control de secuencias.

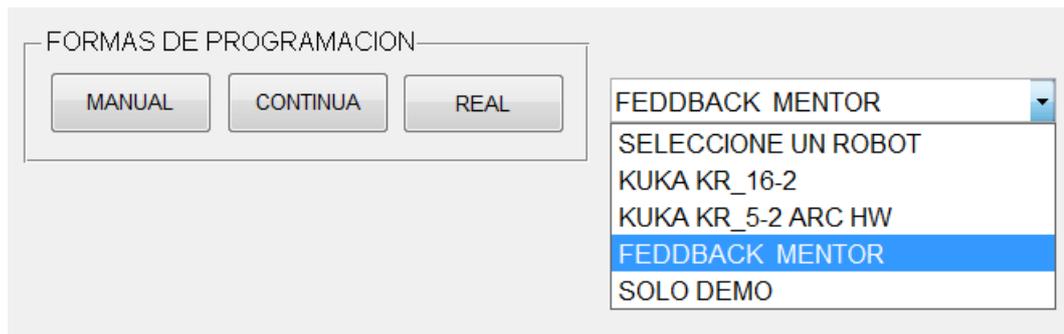


Figura 37. Menú de elección de robots y formas de programación.

El panel de formas de programación varía en función del robot escogido en el menú, debido a que no todos los robots tienen la posibilidad de programación real, por limitaciones propias del método de comunicación utilizado.

3.3.6 Visualización de la posición y orientación del robot.

El campo de visualización de posición y orientación del robot permanece inmóvil durante las etapas de programación, debido a que la interfaz adquiere los valores de posición y orientación del extremo efector, en base a las ecuaciones de cinemática directa y los valores reales adquiridos por los sensores de posición del robot, por esta razón los paneles tanto de posición como de orientación muestran los valores únicamente después de que el robot realice sus movimientos reales.

La imagen siguiente muestra en primera instancia (Fig28. a) los paneles en su estado estático es decir antes de realizarse los movimientos reales del robot, permitiendo observar la posición donde se mostrarán los valores en centímetros de la posición de extremo efector, y de igual forma la orientación en grados del mismo. En la segunda imagen (Fig28. b), se muestra la respuesta después del movimiento del robot, los valores de posición y orientación ocupan los mismos espacios de coordenadas X, Y, Z. y los valores de orientación A, B, C, mostrados en la figura anterior.

POSICION		ORIENTACION	
"X"	"A"	17	87
"Y"	"B"	42	68
"Z"	"C"	44	81

(a)

(b)

Figura 38. Paneles de visualización de Orientación y Posición.

a) Panel previo al movimiento del robot. b) Panel posterior al movimiento del robot.

Al considerar que la respuesta mostrada en los paneles está en base al movimiento real y adquisición de los sensores del robot, el monitoreo de este parámetro se realiza después de cada movimiento completo, la imagen mostrada en la variable de trayectoria del panel de monitoreo es una aproximación de los diferentes valores de posición calculados después de cada secuencia de movimientos completa.

3.3.7 Paneles de programación.

Como se mencionó en el campo del menú para escoger las formas de programación, este campo involucra varios tipos de programación, cada uno tiene como resultado el agregar valores a la base de datos del control de secuencias, utilizando diferentes métodos de programación, considerando de manera general, cada método se asemeja con los estudiados en el primer capítulo, dando un mayor acercamiento a los conceptos manejados sobre programación de robótica industrial.

El primer panel del campo de tipos de programación es el denominado: "Programación asistida", las herramientas dinámicas de este panel permiten al estudiante programar cada uno de los grados de libertad por separado utilizando una aproximación analógica al utilizar los slider que refieren cada uno de las articulaciones del robot, además de tener un campo numérico dinámico en donde no

solo se observa el valor del slider sino también se puede ingresar el valor exacto del ángulo al cual se quiere llegar con una articulación determinada. El panel de control de secuencias adquiere el valor de ángulo de estas herramientas y directamente las almacena en la base de datos para formar la secuencia de movimientos que el robot realizara.

Es importante aclarar también que este método de programación se realiza offline es decir que físicamente el robot no sigue los movimientos del slider.



Figura 39. Panel de programación asistida o manual.

Los movimientos del robot se observan gracias al modelo realizado en realidad virtual, permitiendo al usuario manipular los actuadores como se desee hasta la obtener la posición preciosa antes de grabar el mismo en la secuencia de movimientos.

El segundo panel se refiere a las funciones matemáticas de trayectoria que describen el movimiento del robot, estas funciones dependen de los algoritmos de control que el robot físicamente tenga implementados. razón por la cual se optó por desarrollar trayectorias referidas en espacio articular utilizando una interpolación cubica, esta función matemática calcula los valores articulares que el robot deberá adquirir para desplazar sus grados de libertad a una posición específica, el objetivo de este tipo de

trayectoria es mover el robot lo más rápido posible de un punto a otro, esto se consigue moviendo todos los actuadores de forma paralela haciendo que el robot alcance su posición y orientación final en la trayectoria más veloz.

El cálculo de la función de trayectoria, está en base a constantes o parámetros, los mismos que indican ciertas características del movimiento considerando principalmente los valores de las variables articulares y el espacio de tiempo en el cual se realizara los movimientos, académicamente se puede realizar el cálculo de estas variables por parte del usuario de manera de solventar un movimiento específico completo, es decir dos valores de cada variable articular, uno inicial y otro final. Las constantes se calculan de la siguiente manera:

$$\begin{aligned}
 a &= q_{inicial} \\
 b &= 0 \\
 c &= \frac{3[q_{final} - q_{inicial}]}{t_{final}^2} \\
 d &= -\frac{2[q_{final} - q_{inicial}]}{t_{final}^3}
 \end{aligned}$$

Ecuación 4. Parámetros de trayectoria articular utilizando interpolación cubica.

Estos valores fueron obtenidos en base a consideraciones iniciales predeterminadas para el robot, las mismas en donde se especifica que los valores de las variables articuladas, definidos en las ecuaciones por la letra “q”, tanto iniciales como finales deben coincidir con los valores deseados del movimiento, además se considera que la velocidad inicial y final del robot deben ser nulas es decir antes de realizar el movimiento y después de haberlo terminado. Estas condiciones se solventan utilizando un polinomio de tercer orden.

$$q(t) = a + bt + ct^2 + dt^3$$

Ecuación 5. Polinomio de tercer orden utilizado para la función de trayectoria.

La imagen muestra el panel y la distribución de campos que se utiliza para la creación de los valores que formaran parte de la secuencia de movimientos.

GENERACION DE TRAYECTORIAS

TRAYECTORIA ARTICULAR
INTERPOLACION CUBICA

$$q(t) = a + bt + ct^2 + dt^3$$

Ingresar el parametro "a"

Ingresar el parametro "b"

Ingresar el parametro "c"

Ingresar el parametro "d"

Figura 40. Panel de programación para la generación de trayectorias.

Al resolver el polinomio de tercer orden en base a las condiciones iniciales, se obtiene las ecuaciones de cada parámetro (Ecuación 4), de esta manera el panel permite al estudiante colocar únicamente el valor de los parámetros y calcula el valor articular que formara parte de la secuencia de acción descrita en el panel de secuencias.

El tercer panel al que se puede acceder en el campo de tipos de programación es el denominado como: “programación real”, este panel a diferencia de los otros paneles permite realizar una programación online con el robot real, es decir las herramientas dispuestas en este panel se comunican directamente con el robot de manera que el usuario tenga control directo y en tiempo real sobre los actuadores del robot colocándolo en una posición específica, tras alcanzar la posición deseada se puede grabar dicha posición en la secuencia de acción, tal como se lo realiza con los otros paneles.

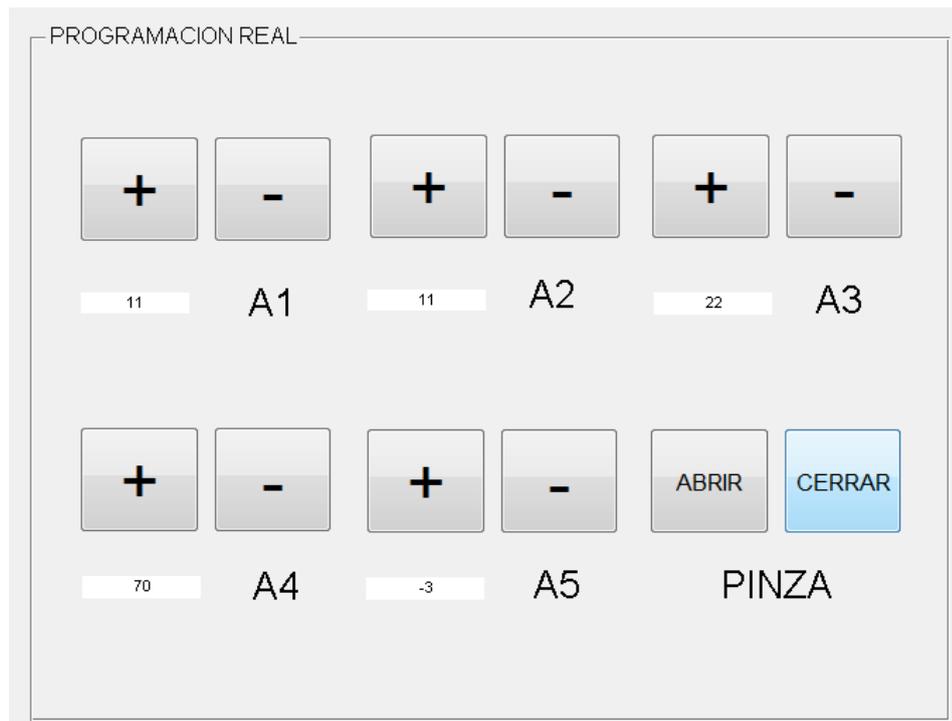


Figura 41. Panel de programación real.

El panel está constituido de herramientas para cada grado de libertad y un espacio de texto que permite visualizar el grado que se alcanzó con cada uno, de estos campos de texto estático se adquieren los valores para la base de datos.

3.3.8 Panel de generación de reportes.

El último panel propiamente dicho es el encargado de generar un archivo de reporte del uso de la interfaz durante la sesión. Académicamente será necesario documentar la utilización del software razón por la cual el panel adquiere los datos de programación de la interfaz y genera un reporte en formato "PDF" o "HTML", en el cual se coloca el nombre del usuario y la siguiente información técnica:

- Tipo de Robot Utilizado.
- Fecha y hora de la sesión.
- Datos generados en la secuencia de acción.
- Parámetros utilizados en el panel de generación de trayectorias.
- Ultimo valor de posición y orientación del robot.

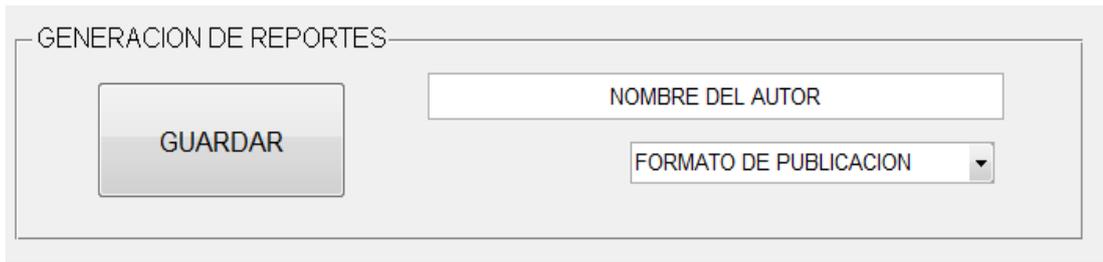


Figura 42. Panel de generación de reportes.

Las herramientas del panel incluyen un menú para elegir el tipo de formato para la generación del reporte, un campo para escribir el nombre del autor o usuario del software y un botón que permite guardar el archivo generado en el ordenador.

3.3.9 Menú general del software.

El último campo es parte del software como tal, conformado por varios menús que realizan diferentes acciones para la interfaz, el primero es el menú de herramientas básicas, que permiten abrir o guardar un reporte de sesión de manera de utilizar datos de sesiones anteriores en una nueva sesión; Además de esto el segundo menú denominado como: “Simulink” es aquel que realiza la comunicación con la segunda parte de la interfaz abriendo un modelo determinado en el campo de programación gráfica, que contiene las librerías de comunicación necesaria para realizar el control de los brazos robot, el cual se analizara en el siguiente apartado.

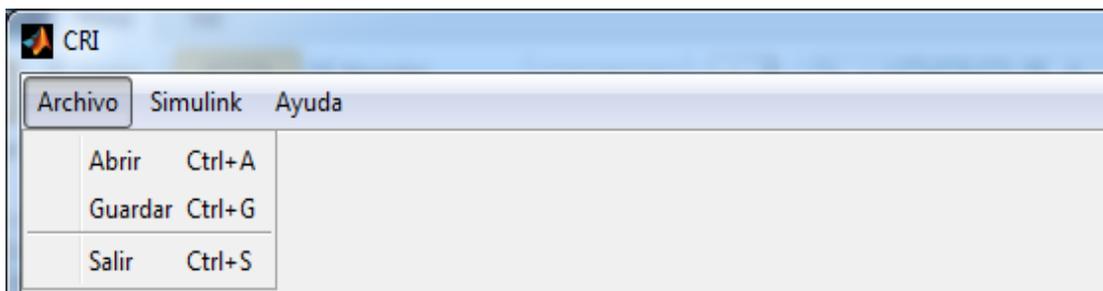


Figura 43. Menu del software.

El último menú de “Ayuda”, permitirá acceder a documentación sobre el software, sus desarrolladores y el manual de operación completo de la interfaz en un archivo completo ajunto.

3.3.10 Campo de control gráfico.

El campo de control grafico refiere al modelo programado en el entorno de Simulink, el objetivo general de esta segunda parte de la interfaz es la de permitir al estudiante

un mayor control sobre el robot, accediendo incluso y dependiendo del robot que se utilice al lazo de control del mismo, dependiendo así de los algoritmos de control que el estudiante realice y no los programados en la interfaz, razón por la cual las librerías creadas que conforman el “toolbox”, afín a la interfaz refieren únicamente a la comunicación con el robot y el movimiento como tal dependerá de las funciones de trayectoria y parámetros de control que el estudiante diseñe.

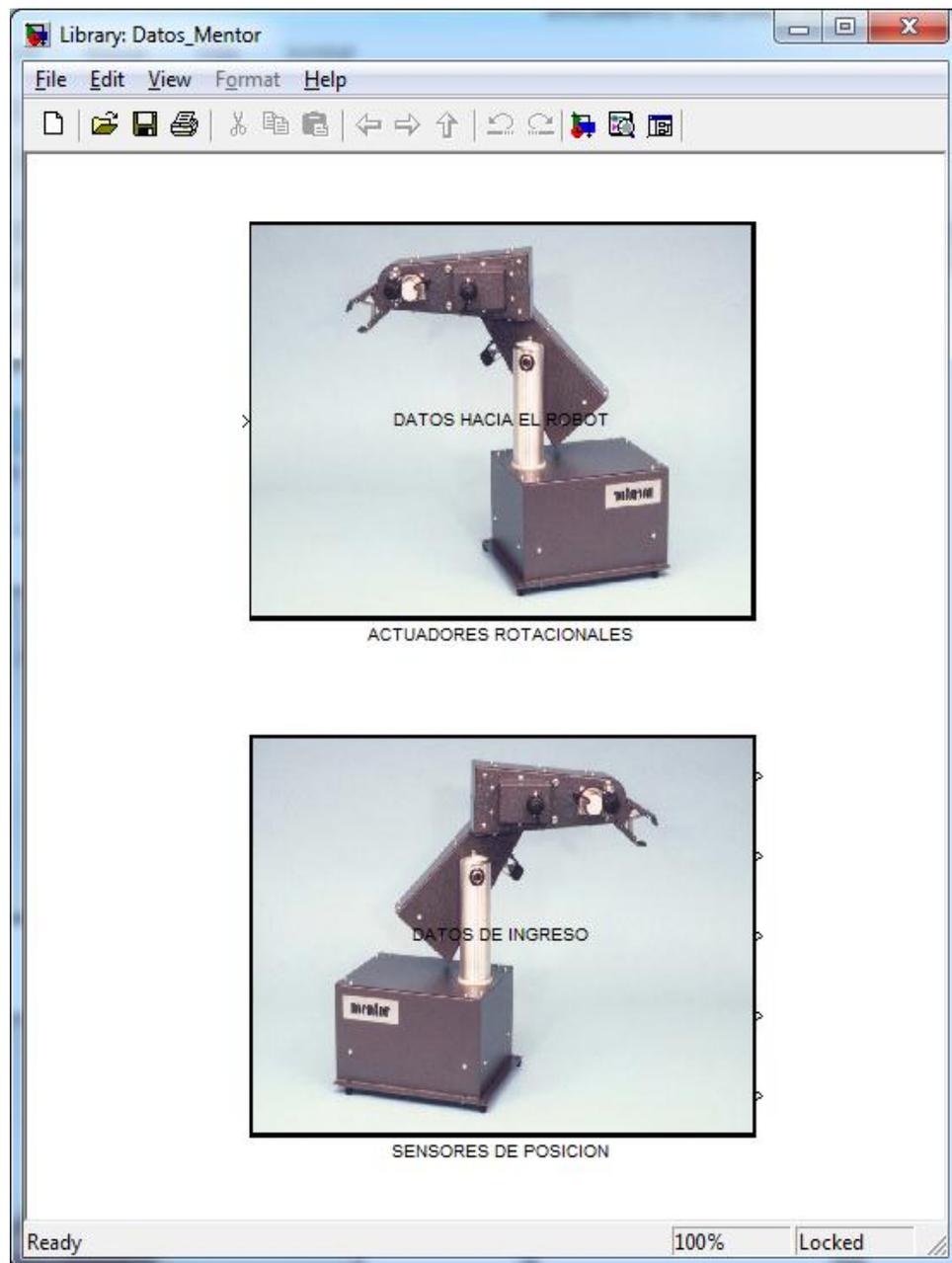


Figura 44. Tolbox de Comunicación para el brazo robot Mentor.

La librería de comunicación cuenta con dos bloques que realizan la adquisición y envío de las señales desde y hacia el robot, la razón por la cual se concibieron por

separado estos parámetros es para facilitar al estudiante un diseño completo del sistema de control o funciones de trayectoria para el robot.

El bloque de ingreso de datos hacia el robot denominado como “Actuadores rotacionales”, recibe una única señal conjunta que se pretende establecer como un vector que contenga los diferentes niveles lógicos que implican el movimiento de los actuadores que conforman el robot, este nivel de abstracción permitirá al estudiante el cambio a voluntad de las características de estos niveles lógicos lo que supone un mayor control sobre el movimiento del robot en base a los algoritmos diseñados en el campo de programación gráfica.

El bloque de salida o datos de ingreso denominado como “Sensores de posición” cuenta con 5 salidas diferentes, una por cada grado de libertad, el bloque cambia los valores de tensión adquiridos por los sensores a variables directamente en grados, es decir el valor de las diferentes variables articulares del robot, las señales de censado permitirán al estudiante realizar sistemas de control retroalimentados para el posicionamiento específico del robot.

Los sistemas de comunicación que rigen la programación de ambos bloques así como los utilizados en la primera parte de la interfaz, están basados en estándares propios para el robot, que se desarrollaron en base a sistemas de comunicación estándar propios del software IDE y a los cuales se tiene acceso por medio de la interfaz y que fueron acondicionados en los bloques del toolbox para realizar el correcto control del robot. Los parámetros técnicos completos utilizados tanto en la comunicación como en el control de los movimientos del robot serán expuestos a detalle en el siguiente apartado.

3.4 PROGRAMACION DE SISTEMAS DE CONTROL Y COMUNICACION.

Definidos todos los parámetros que constituyen el software, tanto en la interfaz de usuario como las principales características de las librerías de comunicación creadas para la programación gráfica, se procederá a explicar cada uno de los parámetros técnicos desarrollados para el correcto funcionamiento de todas las herramientas antes descritas. Los parámetros detallados tanto de control como de comunicación son compartidos para ambas partes constituyentes de la interfaz razón por la cual se utilizara criterios generales que abarquen ambos tipos de programación.

De forma general la programación de los sistemas de comunicación se refiere a la integración y acondicionamiento en la interfaz de los protocolos estándar de comunicación hacia sistemas reales a los cuales el sistema IDE tiene acceso y en base a los cuales se estructuró un sistema de comunicación propio de acuerdo a los parámetros eléctricos, electrónicos y mecánicos del manipulador robótico.

La programación de los sistemas de control se refiere a la utilización del sistema de comunicación para enviar parámetros de control que realizaran movimientos en el robot, es decir que dependiendo de las diferentes variables que el usuario programe o disponga durante el uso de la interfaz, el sistema de control acondiciona estas variables para que se puedan enviar bajo los parámetros de comunicación y permitan el correcto movimiento del robot ya sea en un control manual o por medio de funciones de trayectoria.

3.4.1 Desarrollo de las librerías dedicadas para brazos robot.

Una librería se considera como un grupo de subprogramas utilizados para el desarrollo de otros programas autónomos, esto permite esclarecer el uso de las librerías creadas para su uso en la interfaz, si se considera que la interfaz gráfica de usuario contiene las librerías creadas o descritas dentro de los algoritmos programados en el software como tal, la creación de las librerías su uso, se limitaría a aquellas que se desarrollaron para ser utilizadas independientemente en el campo de programación gráfica, razón por la cual se describirá el desarrollo de las librerías únicamente en base a este tipo de programación, mas es preciso acotar que todos los parámetros utilizados en la programación grafica son los mismos que se utilizaron en la programación descrita de la interfaz de usuario inicial ya que los sistemas de comunicación que son la base tanto para el monitoreo como el control del manipulador robótico están basados en las características propias del software IDE que son las mismas para ambos ámbitos de programación.

El esquema de implementación de las librerías se presenta a continuación.

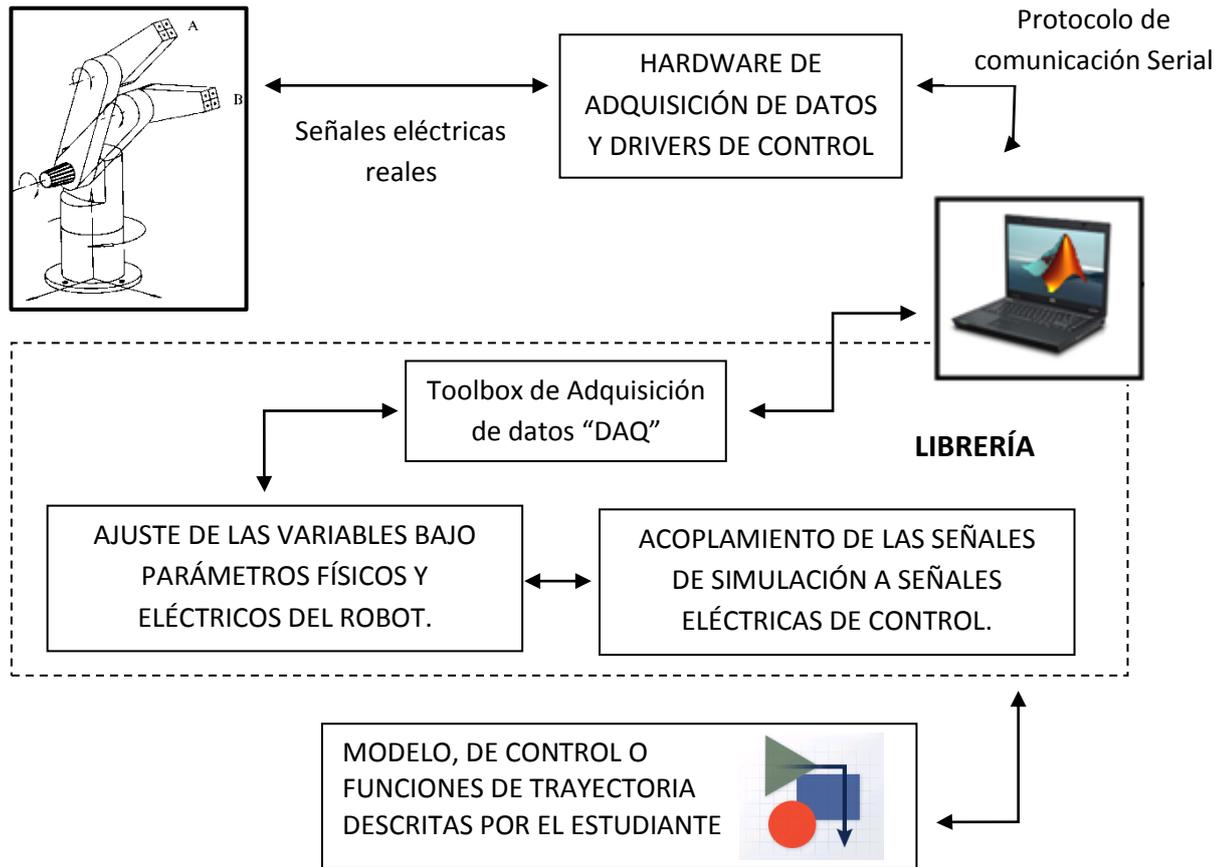


Figura 45. Diagrama de implementación de las librerías de comunicación y control.

La imagen permite observar los criterios que se tomaron en cuenta para la creación de los procesos que formaron la librería, esto además permite discernir el esquema real de comunicación del sistema completo.

Para la creación de librerías se utilizaron los parámetros propios del sistema IDE que permiten la creación de librerías en base a una simulación o diseño propio, además de guardar el diseño y permitir restringir el acceso a los bloques que lo constituyen la librería permite utilizarlos directamente desde el menú principal de herramientas de Simulink.

Utilizando parámetros del diseño mecánico, el modelo cinemático del robot y herramientas matemáticas propias de Simulink se acoplaron las señales entregadas por el toolbox de adquisición de datos, que realiza la comunicación utilizando un protocolo serial bajo el estándar USB de manera de convertirlas en señales que no solo son manipulables directamente por el usuario sino que además refieren información útil directamente al sistema, el segundo bloque que se tuvo en consideración es el de acoplamiento de señales de simulación a señales eléctricas, este bloque directamente

permite controlar el movimiento de los actuadores del robot utilizando señales lógicas simples.

3.4.2 Programación de sistemas de comunicación.

Como se mencionó anteriormente el sistema de comunicación que es la base de la librería de control desde y hacia el robot; este está estructurado utilizando un protocolo de comunicación serial bajo el estándar de bus universal o USB, este protocolo permite enviar de forma bidireccional tanto los datos medidos del robot como los datos de control enviados desde el ordenador, al hardware de adquisición y control de datos acondiciona las señales en una etapa de potencia eléctrica para realizar los movimientos físicos del robot como tal, la programación del sistema de comunicación se complementa con los parámetros propios que el sistema IDE posee para la comunicación USB, mismos que están constituidos en el Toolbox de adquisición de datos, por esta razón es primordial entender el funcionamiento técnico del toolbox, el hardware de adquisición utilizado y las principales características del protocolo de comunicación utilizado.

El bus universal en serie conocido por sus siglas en inglés USB es como se mencionó anteriormente un estándar industrial que utiliza el protocolo de comunicación serial para la transmisión de datos, creado en 1990, es en la actualidad el método de comunicación de periféricos más utilizado, a pesar que comparte las características básicas de su protocolo de comunicación enviando la información bit a bit de forma individual el USB puede llegar a alcanzar velocidades de transmisión de hasta 600 Mb/s dependiendo de la versión que utilice el periférico o hardware con el que se pretenda realizar la comunicación, esto se consigue utilizando una señalización diferencial que permite una comunicación bidireccional full dúplex. Esta velocidad entonces asegura que la comunicación con el robot y la respuesta física del mismo se puedan considerar en tiempo real considerando la respuesta de la interfaz como tal.

El toolbox de adquisición de datos utilizado como vínculo para las señales es un software desarrollado para el sistema IDE que tiene como objetivo adquirir señales de sistemas o procesos físicos reales, por medio de hardware especializado, por ejemplo el sonido producido por una persona hablando que es adquirido por un micrófono, o en el caso del proyecto la señal eléctrica emitida por los sensores de posición del robot que es adquirida por una tarjeta electrónica de adquisición; el

toolbox simplemente acondiciona el valor físico medido en variables finitas y direcciona como señales de simulación o variables para el entorno de programación textual para su posterior procesamiento. De igual forma envía la señal o variable estructurada desde los ambientes de programación acondicionando una señal eléctrica que se pueda utilizar en el proceso físico real.



Figura 46. Tarjeta de adquisición de datos NI USB-6212 [12].

La imagen muestra la tarjeta de adquisición de datos que es utilizada para el acondicionamiento de las variables tanto de monitoreo como de control recibidas y enviadas del robot, el equipo fue construido y comercializado por la empresa National Instruments, como parte de una línea de instrumentos industriales para la adquisición y control de plantas o elementos que constituyan un proceso industrial. Mas estos dispositivos son mundialmente usados en el ámbito académico para prácticas de laboratorio en las asignaturas que implique la interacción con elementos físicos normalmente electromecánicos o eléctricos; razón por la cual se optó por este equipo dando aun mayor acercamiento académico al proyecto como tal.

Las características técnicas de la tarjeta incluyen:

- 8 entradas analógicas (16 bits, 400 kS/s)
- 2 salidas analógicas (16 bits a 250 kS/s),
- 32 E/S digitales, dos contadores de 32 bits.

- Energizado por bus USB para mayor movilidad, conectividad de señal integrada
- NI signalstreaming para transferencia de datos bidireccional a alta velocidad en USB.
- Compatible con LabVIEW, ANSI C/C++, C#, Visual Basic .NET, Visual Basic 6.0 y Matlab. [12]

Es importante recalcar que la programación del toolbox de adquisición de datos, así como el uso de la tarjeta de adquisición no son requisitos académicos para el uso del software final, debido a que los algoritmos descritos en la interfaz, así como el diseño de las librerías poseen los parámetros técnicos necesarios para su utilización, permitiendo al estudiante concentrar su uso en parámetros específicos de robótica y control.

3.4.3 Programación de secuencias y envío de parámetros de control.

Las secuencias que el usuario del software programe ya sea en la interfaz de usuario, utilizando las herramientas predeterminadas de programación o en el espacio de programación gráfica, refieren ciertos criterios que deben ser considerados para los algoritmos de control, dichos criterios dependen del controlador electrónico que el robot posee, razón por la cual es necesario exponer las consideraciones que se siguieron para el desarrollo de los controladores y las secuencias de movimiento.

Si consideramos que una secuencia de movimiento como la expuesta en los paneles de la interfaz de usuario, es una sucesión de movimientos del robot; se puede esclarecer el algoritmo de control realizado para un movimiento específico, debido a que una sucesión de movimientos compartiría los mismos criterios.

La siguiente imagen (Fig.36) muestra un diagrama de flujo que define el algoritmo básico de control utilizado para realizar los movimientos físicos del robot, las variables se conciben como vectores sobre los cuales están definidos los diferentes valores articulares de cada grado de libertad, es decir los valores de posición censados y los valores lógicos de control para el movimiento; De esta forma se asegura un movimiento continuo y en paralelo de todos los actuadores, realizando el movimiento de forma veloz y cumpliendo con las características propias del movimiento de una trayectoria en espacio articular. Cabe recalcar que el algoritmo de control puede variar si el estudiante realiza un cambio en el espacio de programación gráfica, basta con respetar los parámetros del controlador físico que utiliza señales

lógicas como ingresos de la etapa de potencia de los actuadores, para realizar un algoritmo diferente para el control de los grados de libertad del robot.

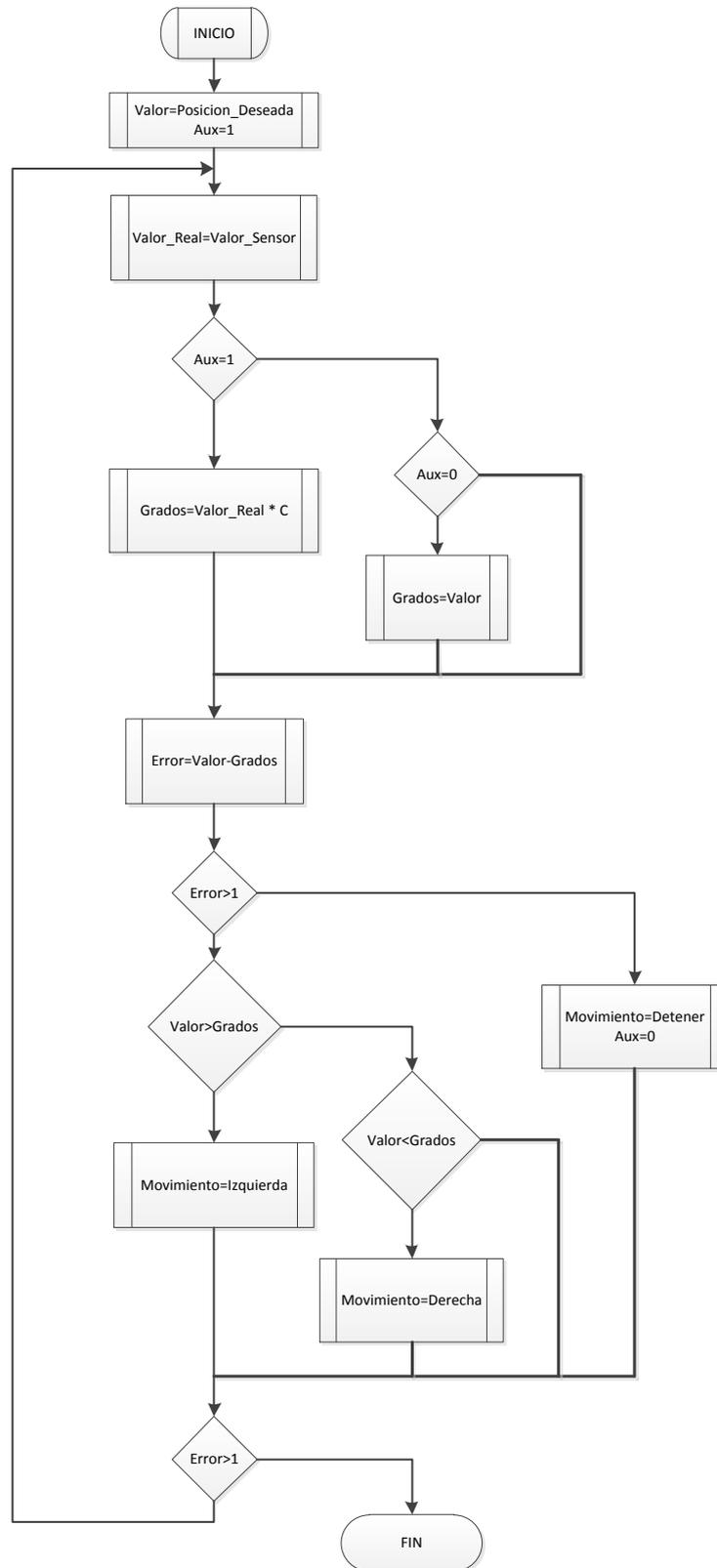


Figura 47. Algoritmo básico para el control de movimiento.

Las funciones de trayectoria como se expuso anteriormente están basadas en una aproximación matemática que calcula el valor de las variables articulares que el robot deberá poseer para moverse de un punto a otro, considerando lo antes expuesto sobre los parámetros de control para un movimiento o para una secuencia de movimientos, las funciones de trayectoria compartirían los mismos criterios ya que la respuesta de estas funciones derivarían en valores articulares que formarían parte de la secuencia como tal.

CAPITULO 4. PRUEBAS Y EVALUACIÓN DEL SISTEMA.

4.1 INTRODUCCIÓN.

Este capítulo abordara las pruebas de funcionamiento realizadas al sistema así como las técnicas de validación que determinan la validez del software desarrollado para el control del brazo robot articulado. Al ser aplicado en un sistema electromecánico, además de las pruebas preliminares de validación del programa como tal será necesario el desarrollo de pruebas físicas para definir parámetros de control y respuestas por parte del manipulador robótico a las señales de control emitidas por la interfaz.

En primer lugar se denotan las pruebas específicas para el software desarrollado, estas pruebas se desarrollaron en base al diseño de casos de prueba, los mismos que sirvieron para mejorar las diferentes funciones constituyentes de la interfaz, así como la reducción de errores de programación, previo a las pruebas de laboratorio, es decir aquellas realizadas sobre un ambiente controlado, permitiendo verificar que los resultados sean correctos; estas pruebas no solo permitirán establecer criterios de valoración del software de control como tal, sino que además permitirán un mejor acercamiento a las prestaciones físicas del robot determinando valores tales como precisión, exactitud, y repetibilidad.

4.2 PRUEBAS DEL SOFTWARE.

El programa desarrollado, pretende marcar una pauta para la utilización de un manipulador robótico a ser utilizado en el ámbito académico, dotando de mejores herramientas a los estudiantes para aplicar los conceptos generales sobre control y robótica industrial, para lo cual es necesario que la interfaz de usuario y todas las herramientas que esta posea, tengan la menor cantidad posible de errores. Este apartado incorpora las pruebas de software realizadas para la identificación y reducción de estos errores garantizando así la calidad del software.

Las pruebas realizadas se desarrollaron en base a parámetros estándar para la creación de los llamados: “casos de prueba”, estos casos de prueba que en conjunto tienen la intención de descubrir un error, razón por la cual una prueba se considera exitosa si esta permite descubrir un error en el programa; cabe recalcar que la prueba no asegura la ausencia de errores, solo puede demostrar que existen defectos de software.

La siguiente imagen muestra el esquema general para el proceso de prueba, en donde se analizara las especificaciones del software y su código fuente, además incluye el procedimiento de prueba formalmente dicha. El resultado del proceso de prueba concluirá si el software es aceptable además de determinar los posibles errores del programa.

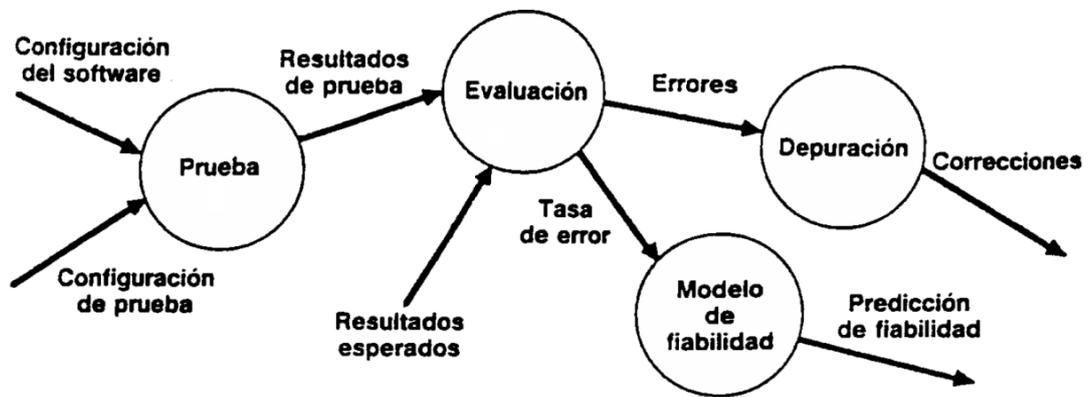


Figura 48. Proceso de prueba para la evaluación de software [39].

Como se mencionó anteriormente el proyecto deberá pasar por una serie de pruebas diseñadas con el objetivo de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y tiempo, estos casos de prueba se diseñan en base a parámetros específicos que permitan comprobar si cada función es operativa, además de permitir probar cada componentes de forma adecuada, estos parámetros son dictados por medio de dos formas diferentes que suponen probar cualquier producto de ingeniería. Las pruebas de caja negra y caja blanca como se denominan a estos procesos de evaluación, serán desarrolladas íntegramente a continuación en función de los datos específicos obtenidos de la interfaz de control.

4.2.1 Pruebas de caja blanca.

Los casos de prueba diseñados por medio del método de caja blanca realizan un exhaustivo examen de los detalles procedimentales, comprobando la secuencia de los

algoritmos y examinando el estado del programa en varios puntos, por esta razón las pruebas realizadas bajo los preceptos de caja blanca suponen tener programas perfectos, debido a su nivel de prueba, definiendo todos los caminos lógicos, desarrollando pruebas para todos estos caminos y evaluando los resultados obtenidos.

El principal inconveniente de las pruebas de caja blanca es el tiempo que prolonga excesivamente los planes de desarrollo del software, razón por la cual normalmente se utiliza este criterio para los caminos lógicos o algoritmos más significativos. Este precepto lo consideraremos directamente, al aplicar esta forma de prueba únicamente en las secuencias de control del robot las cuales se consideran como principales para el proyecto. Para cada una de las funciones los casos de prueba diseñados garantizarán lo siguiente:

- Se ejecutan al menos una vez los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internas. [39]

Estas consideraciones se solventan por medio de dos pruebas globales diferentes, las cuales serán desarrolladas para los algoritmos del proyecto mencionados.

- Pruebas de camino básico.

El método obtiene una medida de complejidad basada en los algoritmos implementados en el software, lo que permite realizar varias pruebas garantizando que cada camino se ejecute al menos una vez, esta medida se obtiene construyendo en primera instancia un grafo de flujo del algoritmo en cuestión con el cual se determinarán las pruebas respectivas, como se muestra a continuación.

- a. Notación del grafo de flujo del programa.

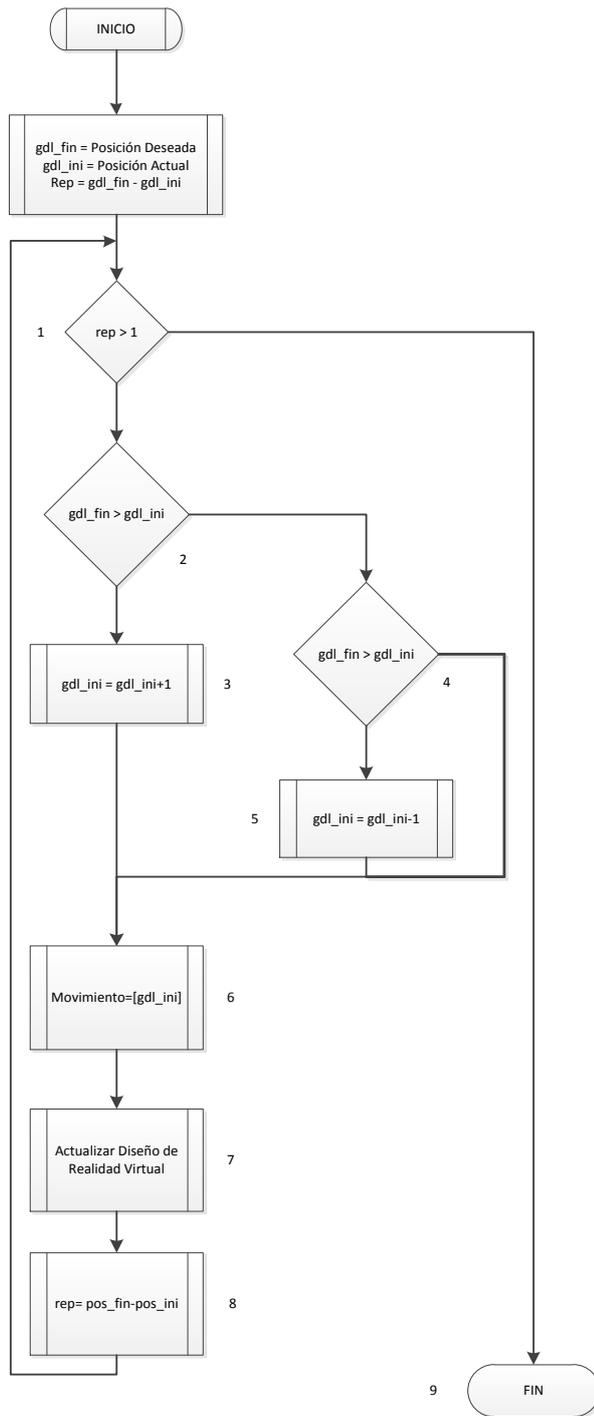


Figura 49. Algoritmo de control del diseño en realidad virtual.

El diseño mostrado a continuación, cambia la notación gráfica del algoritmo y utiliza únicamente nodos referidos con los números correspondientes a las funciones o procesos del algoritmo, unidos por medio de flechas o aristas que indican la secuencia o flujo del programa.

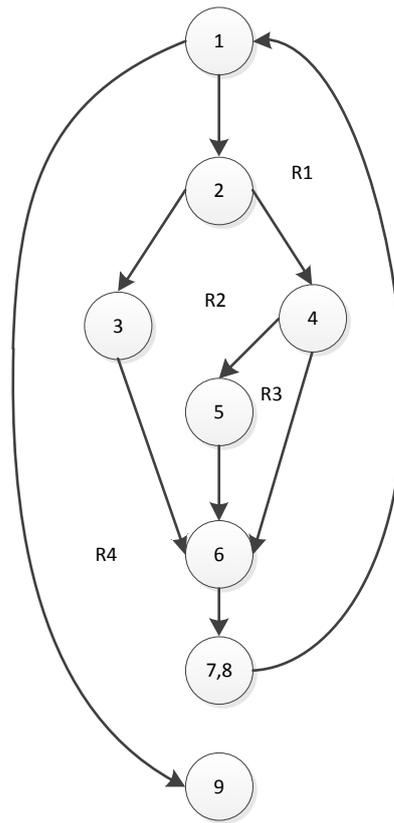


Figura 50. Grafo de flujo del control del diseño en realidad virtual.

El grafo de flujo, presentado en la imagen anterior permite observar varios elementos importantes para el método de camino básico:

- Nodos: Representados por una o más sentencias procedimentales.
- Aristas: Representan el flujo de control (flechas).
- Regiones: Representan los bucles del programa o conjunto de procesos.

Estos elementos son de gran importancia para el siguiente movimiento a realizarse en las pruebas de camino básico como se mostrara a continuación.

b. Complejidad Ciclomática.

La complejidad ciclomática determinada por $V(G)$ es la medida que se calcula para determinar el número de pruebas necesarias a realizar.

$$V(G) = \text{No. de Regiones} = 4$$

$$V(G) = \text{Aristas} - \text{Nodos} + 2 = 10 - 8 + 2 = 4$$

$$V(G) = \text{Nodos de predicado} + 1 = 3 + 1 = 4$$

La complejidad ciclomática nos indica cuantos caminos independientes tiene el algoritmo, en este caso 4.

- 1-9
- 1-2-3-6-7-8-1-9
- 1-2-4-5-6-7-8-1-9
- 1-2-4-6-7-8-1-9

c. Pruebas de camino básico.

Se debe realizar una prueba para cada camino propuesto en el apartado anterior.

Prueba	Camino a comprobar	Parámetros de ingreso	Respuesta valida	Respuesta invalida	Resultado
1	1-9	$gdl_ini=gdl_fin$	El diseño en realidad virtual no realiza ningún movimiento	El diseño en realidad virtual realiza cualquier tipo de movimiento	Exitoso
2	1-2-3-6-7-8-1-9	$gdl_ini>gdl_fin$	El diseño en realidad virtual realiza el movimiento articular hacia la derecha	El diseño en realidad virtual realiza cualquier movimiento diferente a uno articular hacia la derecha.	Exitoso
3	1-2-4-5-6-7-8-1-9	$gdl_ini<gdl_fin$	El diseño en realidad virtual realiza el movimiento articular hacia la izquierda	El diseño en realidad virtual realiza cualquier movimiento diferente a uno articular hacia la izquierda	Exitoso
4	1-2-4-6-7-8-1-9	$gdl_ini=gdl_fin$ Indeterminado	El diseño en realidad virtual no realiza ningún movimiento	El diseño en realidad virtual realiza cualquier tipo de movimiento	Indeterminado

Tabla 11: Pruebas de camino básico para el algoritmo de control del modelo en realidad virtual.

- Prueba de bucles.

El siguiente grupo de pruebas de caja blanca a realizarse son las determinadas a validar la construcción de los bucles, debido a la gran importancia que estos tienen en casi todos los algoritmos implementados en software. Para nuestro caso se realizaran el siguiente análisis:

El bucle simple presentado en el algoritmo, realiza “n” iteraciones donde “n” varía en función de un proceso, las pruebas a realizarse contemplaran lo siguiente:

- Saltar el bucle.
- Pasar una vez por el bucle.
- Pasar dos veces por el bucle.

- Pasar “m” veces por el bucle con $m > n$.
- Hacer $n-1$, n y $n+1$ pasos por el bucle.
 - a. Pruebas de bucles.

Las iteraciones se consideraran constantes para asegurar la ejecución de todas las pruebas propuestas.

Prueba	Acción a realizar	Parámetros de ingreso	Respuesta valida	Respuesta invalida	Resultado
1	Saltar el bucle.	$gdl_ini=gdl_fin$	El diseño en realidad virtual no realiza ningún movimiento	El diseño en realidad virtual realiza cualquier tipo de movimiento	Exitoso
2	Pasar una vez por el bucle.	$gdl_ini=gdl_fin \pm 1$	El diseño en realidad virtual alcanza su posición final realizando un único movimiento	El diseño en realidad virtual realiza más de un único movimiento	Exitoso
3	Pasar dos veces por el bucle.	$gdl_ini=gdl_fin \pm 2$	El diseño en realidad virtual alcanza su posición final realizando dos movimientos	El diseño en realidad virtual realiza más de dos movimientos	Exitoso
4	Pasar “m” veces por el bucle con $m > n$.	$rep=rep+m$	El diseño en realidad virtual realiza los movimientos sin llegar a la posición deseada.	El diseño en realidad virtual llega a la posición deseada.	Exitoso
5	Hacer $n-1$, n y $n+1$ pasos por el bucle.	$rep=rep+1$	El diseño en realidad virtual realiza los movimientos sin llegar a la posición deseada.	El diseño en realidad virtual llega a la posición deseada.	Exitoso
		$rep=rep$	El diseño en realidad virtual alcanza la posición deseada.	El diseño en realidad virtual no llega a la posición deseada.	Exitoso
		$rep=rep-1$	El diseño en realidad virtual realiza los movimientos sin llegar a la posición deseada.	El diseño en realidad virtual llega a la posición deseada.	Fallido

Tabla12: Pruebas de bucles para el algoritmo de control del modelo en realidad virtual.

Cabe recalcar que tanto el algoritmo, como el grafo de flujo presentados fueron diseñados a partir del movimiento de un solo grado de libertad del modelo en

realidad virtual del robot, debido a que todos siguen el mismo precepto de forma paralela, analizar un solo proceso se considera suficiente para la realización de pruebas de software; al igual que el siguiente análisis que refiere a la comunicación y control del robot real.

a. Notación del grafo de flujo del programa.

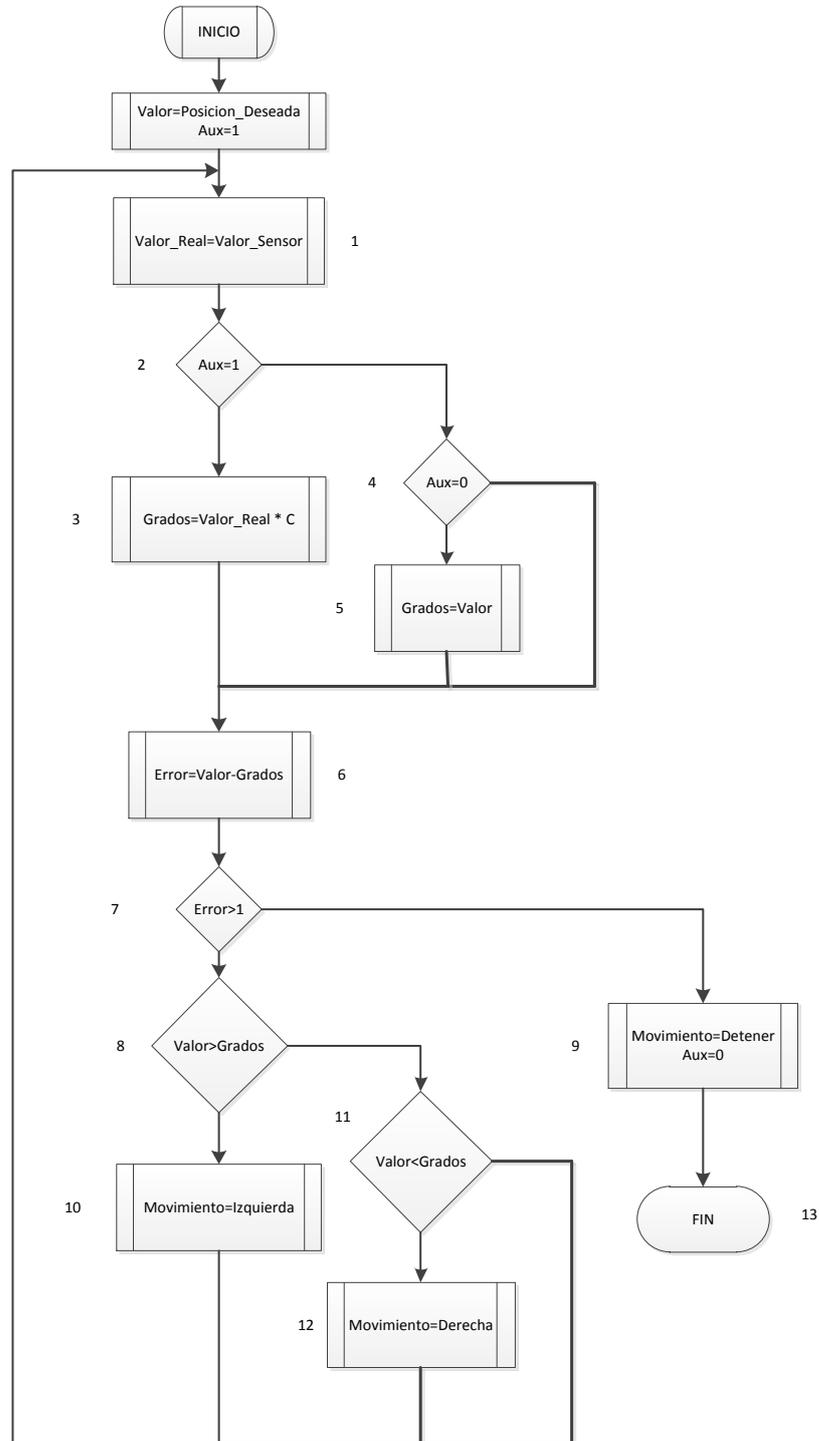


Figura 51. Algoritmo de control del manipulador robótico.

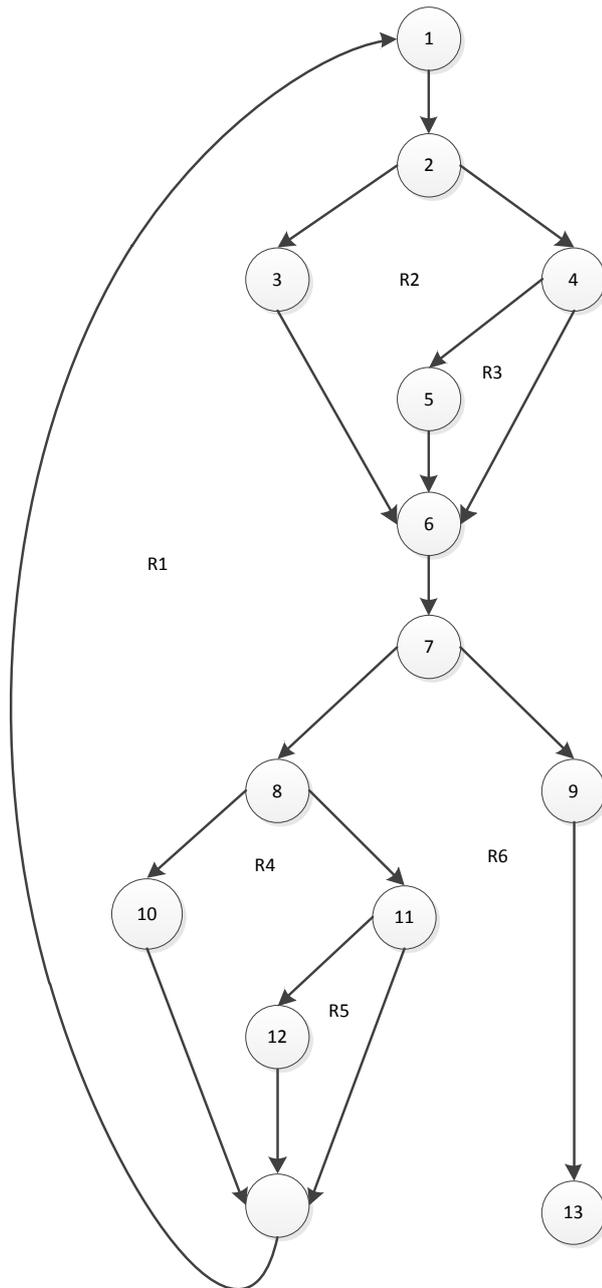


Figura 52. Grafo de flujo de control del manipulador robótico.

Al igual que se realizó anteriormente es necesario determinar el valor de la complejidad ciclomática para el posterior diseño de los casos de prueba.

b. Complejidad Ciclométrica

$$V(G) = \text{No. de Regiones} = \mathbf{6}$$

$$V(G) = \text{Aristas} - \text{Nodos} + 2 = 17 - 13 + 2 = \mathbf{6}$$

$$V(G) = \text{Nodos de predicado} + 1 = 5 + 1 = \mathbf{6}$$

La complejidad ciclométrica nos indica los seis caminos independientes que tiene el algoritmo.

- 1-2-3-6-7-9-13
 - 1-2-4-5-6-7-9-13
 - 1-2-4-6-7-9-13
 - 1-2-3-6-7-8-10-1-2-4-6-7-9-13
 - 1-2-3-6-7-8-11-12-1-2-4-6-7-9-13
 - 1-2-3-6-7-8-11-1-2-4-6-7-9-13
- Pruebas de camino básico.

Se debe realizar una prueba para cada camino propuesto en el apartado anterior, al igual que se hizo anteriormente las variables que son parte de parámetros de ingreso serán forzadas para adecuarse a la prueba respectiva.

Prueba	Camino a comprobar	Parámetros de ingreso	Respuesta valida	Respuesta invalida	Resultado
1	1-2-3-6-7-9-13	Grados=Valor	El brazo robot no realiza ningún movimiento	El brazo robot realiza cualquier tipo de movimiento	Exitoso
2	1-2-4-5-6-7-9-13	Aux=0	El brazo robot no realiza ningún movimiento	El brazo robot realiza cualquier tipo de movimiento	Exitoso
3	1-2-4-6-7-9-13	Aux=Indeterminado	El brazo robot no realiza ningún movimiento	El brazo robot realiza cualquier tipo de movimiento	Indeterminado
4	1-2-3-6-7-8-10-1-2-4-6-7-9-13	Valor>Grados	La articulación referida realiza un movimiento hacia la derecha	La articulación referida realiza cualquier movimiento diferente a uno articular hacia la derecha	Exitoso
5	1-2-3-6-7-8-11-12-1-2-4-6-7-9-13	Valor<Grados	La articulación referida realiza un movimiento hacia la izquierda	La articulación referida realiza cualquier movimiento diferente a uno articular hacia la izquierda	Exitoso
6	1-2-3-6-7-8-11-1-2-4-6-7-9-13	Valor=Grados Indeterminados	El brazo robot no realiza ningún movimiento.	El brazo robot realiza cualquier tipo de movimiento	Indeterminado

Tabla13: Pruebas de camino básico para el algoritmo de control del manipulador robótico.

- Prueba de bucles.

Las pruebas realizadas sobre el bucle principal del algoritmo contemplaran las cinco pruebas generales, propuestas por el método, descrito y realizado anteriormente.

a. Pruebas de bucles.

Al igual que en las pruebas anteriores las iteraciones se consideraran constantes para asegurar la ejecución de todas las pruebas propuestas.

Prueba	Acción a realizar	Parámetros de ingreso	Respuesta valida	Respuesta invalida	Resultado
1	Saltar el bucle.	Grados=Valor	El brazo robot no realiza ningún movimiento	El brazo robot realiza cualquier tipo de movimiento	Exitoso
2	Pasar una vez por el bucle.	Grados=Valor±1	El brazo robot alcanza su posición final realizando un único movimiento, en cada articulación.	El brazo robot realiza más de un único movimiento para alcanzar la posición deseada o no llega a dicha posición.	Exitoso
3	Pasar dos veces por el bucle.	Grados=Valor±2	El brazo robot alcanza su posición final realizando dos movimientos en cada articulación.	El brazo robot realiza más de dos movimientos para alcanzar la posición deseada o no llega a dicha posición.	Exitoso
4	Pasar “m” veces por el bucle con $m > n$.	Error=Error+m	El brazo robot realiza los movimientos sin llegar a la posición deseada.	El brazo robot llega a la posición deseada.	Exitoso
5	Hacer n-1, n y n+1 pasos por el bucle.	Error=Error+1	El brazo robot realiza los movimientos sin llegar a la posición deseada.	El brazo robot llega a la posición deseada.	Exitoso
		Error=Error	El brazo robot realiza los movimientos llegando a la posición deseada.	El brazo robot no llega a la posición deseada.	Exitoso
		Error=Error-1	El brazo robot realiza los movimientos sin llegar a la posición deseada.	El brazo robot llega a la posición deseada.	Fallido

Tabla14: Pruebas de bucles para el algoritmo de control del modelo en realidad virtual.

El análisis de los datos entregados por las diferentes pruebas realizadas serán, estudiados a profundidad en apartados posteriores, realizando comparaciones y complementos en relación a las pruebas de comunicación y movimientos del robot.

4.2.1 Pruebas de caja negra.

Los casos de prueba diseñados por medio del método de caja negra, a diferencia de los analizados en el apartado anterior obvia el comportamiento interno y la estructura del programa, concentrando el análisis en la interfaz del software, razón por la cual el estudio de los casos de prueba aportaran en gran medida al proyecto. Los casos de prueba de la caja negra pretenden demostrar los siguientes parámetros del programa:

- Las funciones del software son operativas.
- Las entradas se aceptan de forma correcta.
- Se produce salidas correctas.
- La integridad de la información externa se mantiene.

Los errores que se pueden descubrir utilizando el análisis de caja negra comprenden: funciones incorrectas o ausentes, errores de la interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores en rendimiento y errores de inicialización o terminación. Estas consideraciones se solventan por medio de una prueba denominada “Prueba de partición equivalente”, la cual será ampliada a continuación al será aplicada para la interfaz gráfica de usuario del proyecto [39].

- Pruebas de partición equivalente.

El método de partición equivalente divide el dominio de entrada de un programa en clases de datos, originando así casos de prueba que examinan todas las acciones posibles para cada operación o condición de entrada, el proceso de forma general consiste en identificar los diferentes complementos de la interfaz del software y analizar los campos de interacción con el usuario, es decir aquellas herramientas sobre las cuales el usuario puede ingresar o cambiar un dato o variable, estas se consideran condiciones externas o entradas, que como se verá a continuación pueden ser de diferente naturaleza dependiendo normalmente del tipo de herramienta que se utilice para la interacción o el tipo de variable sobre la cual se tiene control; a su vez la forma en la que el programa responde se consideran clases de equivalencia, las

mismas que pueden ser concebidas como válidas o inválidas, dependiendo del diseño de la interfaz y los objetivos planteados para la programación de la misma.

La siguiente tabla muestra la división de las diferentes entradas de interacción con el usuario que se consideran principales para el desenvolvimiento completo del actuar de la interfaz de control, comenzando con los parámetros necesarios para el desarrollo del informe de sesión hasta terminar con las herramientas descritas para el control de movimiento del robot como tal, posteriormente a la identificación de las clases se procederá a estructurar los casos de prueba, procediendo según los parámetros estándar, que dictan realizar un caso de prueba para cada clase de equivalencia válida e inválida, procurando según sea las condiciones externas cubrir la mayor cantidad de clases de equivalencia válidas y cada clase de equivalencia inválida de forma individual.

a. Identificación de las clases de equivalencia.

Condiciones Externas (Entradas)	Tipo	Clases de equivalencia válidas (CEV)	Clases de equivalencia inválidas (CEI)
Nombre del usuario	Valor (cadena)	1: Cualquier cadena, palabra o palabras	2: No se ingresa una palabra o palabras
Formato de publicación	Valores específicos	3: "FORMATO DE PUBLICACION"	6: Valor diferente a los valores específicos.
		4: "PDF"	
		5: "HTML"	
Seleccione un robot	Valores específicos	7: "SELECCIONE UN ROBOT"	12: Valor diferente a los valores específicos.
		8: "KUKA KR 16-2"	
		9: "KUKA KR 5-2"	
		10: "FEEDBACK MENTOR"	
Movimiento manual A1	Rango	13: (-95<= valor <=155)	14: Valor<-95
			15: Valor>155
Movimiento manual A2	Rango	16: (-90<= valor <=90)	17: Valor<-90
			18: Valor>90
Movimiento manual A3	Rango	19: (-115<= valor <=115)	20: Valor<-115
			21: Valor>115
Movimiento manual A4	Rango	22: (-180<= valor <=180)	23: Valor<-180
			24: Valor>180
Movimiento manual A5	Rango	25: (-50<= valor <=50)	26: Valor<-50
			27: Valor>50
Ingresar el parámetro "a"	Valor (Vector)	28: Se ingresa un vector numérico de 5 posiciones	29: Cualquier cadena alfanumérica de menos de 5 posiciones.
			30: Cualquier cadena alfanumérica de más de 5 posiciones.

Ingresar el parámetro "b"	Valor (Vector)	31: Se ingresa un vector numérico de 5 posiciones	32: Cualquier cadena alfanumérica de menos de 5 posiciones.
			33: Cualquier cadena alfanumérica de más de 5 posiciones.
Ingresar el parámetro "c"	Valor (Vector)	34: Se ingresa un vector numérico de 5 posiciones	35: Cualquier cadena alfanumérica de menos de 5 posiciones.
			36: Cualquier cadena alfanumérica de más de 5 posiciones.
Ingresar el parámetro "d"	Valor (Vector)	37: Se ingresa un vector numérico de 5 posiciones	38: Cualquier cadena alfanumérica de menos de 5 posiciones.
			39: Cualquier cadena alfanumérica de más de 5 posiciones.

Tabla 15: Clases de equivalencia para la interfaz gráfica de usuario.

b. Casos de prueba.

Prueba	Clases de equivalencia	Prueba a realizar					Resultado
1	1era	"Rodrigo Escandón"					Generación de reporte normal.
2	2da	"45A18CD70F"					Nombre de usuario erróneo
3	3ra	"FORMATO DE PUBLICACION"					"Escoja un tipo de formato para realizar su reporte"
4	4ta	PDF					Reporte generado en PDF
5	5ta	HTML					Reporte generado en HTML
6	6ta	"" (Opción indeterminada)					Orden Errónea
7	7ma	SELECCIONE UN ROBOT					"Elija por favor una opción"
8	8va	KUKA KR 16-2					Diseño virtual del robot KUKA
9	9na	KUKA KR 5-2					Diseño virtual del robot KUKA
10	10ma	FEEDBACK MENTOR					Diseño control y comunicación con el robot Mentor
11	11va	SOLO DEMO					Panel de demostraciones
12	12va	"" (Opción indeterminada)					Orden Errónea
13	13a, 16a, 19a, 22a, 25a	A1	A2	A3	A4	A5	Adquisición de datos correcta para el movimiento del robot.
		-90	-80	-100	-180	-50	
14	13b, 16b, 19b, 22b, 25b	A1	A2	A3	A4	A5	Adquisición de datos correcta para el movimiento del robot.
		100	80	100	180	50	
15	14, 17, 20, 23, 26	A1	A2	A3	A4	A5	Adquisición de datos errónea y error en el movimiento debido al espacio de trabajo del robot
		-100	-100	-150	-200	-60	
16	15, 18, 21, 24, 27	A1	A2	A3	A4	A5	Adquisición de datos errónea y error en el movimiento debido al espacio de trabajo del robot
		200	100	150	200	60	
17	28, 31, 34, 37	a	b	c	d	Calculo correcto de variables articulares y movimiento del robot en la trayectoria esperada	
		[90, 45, 60,	[0, 0, 0,	[10.8, 20.2, 15.5,	[5.2, 3.4, 7.8		

		45, 30]	0, 0]	8.8, 2.2]	4.1, 3.3]	
18	29, 32, 35, 38	a	b	c	d	Calculo incorrecto de variables que se derivaron en un error dirigido al usuario
		[90, 45, 45]	[10, 5]	[10.8, 20.2]	[5.2, 3.4, 7.8]	
19	30, 33, 36, 39	a	b	c	d	Calculo incorrecto de variables que se derivaron en movimientos erróneos del robot.
		[90, 45, 60,	[0, 0, 0,	[10.8, 20.2, 15.5,	[5.2, 3.4, 7.8	
		45, 30,	0, 0,	8.8, 2.2,	4.1, 3.3,	
		10]	0]	5.7]	4.2]	

Tabla16: Pruebas de caja negra para la interfaz gráfica de usuario.

Los casos de pruebas a los cuales se sometió al software resultaron en cambios internos en la interfaz, lo que mejoró las prestaciones del programa, más los resultados al igual que los obtenidos anteriormente serán analizados más adelante.

4.3 PRUEBAS DE COMUNICACIÓN DEL SOFTWARE.

Este apartado contemplara las pruebas de laboratorio realizadas sobre el brazo robot articulado real. Al considerar estas pruebas en su totalidad, respuestas físicas del robot a las señales emitidas por el ordenador en base a los parámetros de control de la interfaz; estos parámetros obtenidos también permitirán esclarecer criterios sobre los parámetros de comunicación al tiempo que se definirán los valores de precisión y exactitud del sistema.

4.3.1 Proceso de adquisición de datos.

El proceso por el cual se realizaron las pruebas consistió en el movimiento del actuador robótico en una secuencia específica, es decir desde un punto inicial hasta un segundo punto final. De manera de asegurar que la experimentación produzca resultados concluyentes tanto en medidas de control como de comunicación del software, se optó por realizar varios procesos, los cuales se constituyeron de dos experimentos cada uno, permitiendo realizar un análisis exhaustivo de la respuesta física del robot.

Los experimentos concebidos como tal se desarrollaron en base a parámetros específicos, los cuales debían repetirse continuamente, estos consistieron en el movimiento puntual del brazo robot desde una posición específica hacia otra, cada

experimento se realizó de dos formas diferentes, por medio de un movimiento secuencial de los grados de libertad, es decir uno por uno y por medio de un movimiento continuo bajo los parámetros de una trayectoria en espacio articular, de manera de abarcar todas las posibles formas de programación que la interfaz posee.



Figura 53. Ilustración gráfica del movimiento utilizado para la experimentación.

Cada uno de los experimentos mencionados que forman parte de un proceso se repitieron continuamente bajo los parámetros de técnicas de investigación denominadas como de observación, las mismas que contemplaron los siguientes pasos:

- A. Determinar el objeto, situación o caso.
- B. Determinar los objetivos de la observación.
- C. Determinar la forma con que se van a registrar los datos
- D. Observar cuidadosa y críticamente.
- E. Registrar los datos observados.
- F. Analizar e interpretar los datos.
- G. Elaborar conclusiones.
- H. Elaborar el informe de observación.

La tabla completa que contempla todos los experimentos y procesos realizados se encuentra adjunto a la presente monografía (Anexo A), razón por la cual no se

mostrara toda la experimentación sino únicamente el caso de un proceso específico, de manera de realizar el análisis de resultados general posteriormente.

Experimento 9					
punto inicial: GD1=20, GD2=100, GD3=-80, GD4=-60, GD5=70					
punto final: GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30					
Repetición.	A1	A2	A3	A4	A5
1	-39,5299	79,5348	58,3499	9,53755	29,105
2	-39,5433	79,5466	56,231	10,455	30,32465
3	-37,5781	80,2806	66,4942	9,03446	29,0531
4	-42,6768	80,2348	58,0776	9,55674	29,9184
5	-43,3548	78,6833	65,8315	9,62634	29,6415
6	-41,3173	79,7077	56,2456	10,5733	29,1396
7	-41,7411	82,9738	62,977	9,86309	30,0049
8	-44,2022	79,5132	56,5005	9,09364	30,08875
9	-41,4688	79,8052	67,4725	10,7509	30,3164
10	-40,3128	79,9152	56,6915	9,06405	29,5203
11	-42,1559	78,7066	60,5786	9,74471	28,8145
12	-43,1086	78,6823	60,5826	9,56715	30,59333
13	-44,3587	76,8797	59,1001	10,5142	29,8665
14	-41,4058	80,4973	60,932	10,8397	30,2991
15	-37,9927	80,9345	65,8129	9,77431	29,2261
16	-41,1712	80,8671	58,7838	10,8693	29,2434
17	-37,9752	79,4258	60,9815	9,12324	29,9876
18	-41,9081	79,3338	60,9042	9,24161	30,8702
19	-38,1541	79,0992	59,7921	10,4254	29,1396
20	-39,1156	80,0248	60,9977	10,9581	29,7799
Experimento 10					
Punto inicial: GD1=-75, GD2=0, GD3=90, GD4=120, GD5=-25					
Punto final: GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30					
Repetición.	A1	A2	A3	A4	A5
1	-43,1455	77,5089	56,7902	9,44877	29,34688
2	-41,3667	80,1042	59,1765	9,33039	30,351
3	-42,4839	83,2536	59,3538	9,06405	30,135
4	-39,7595	79,1819	57,9713	10,0111	30,8273
5	-39,5176	81,3794	59,1605	9,59674	29,9357
6	-40,3983	80,4408	59,4076	9,15283	29,1396
7	-42,0286	79,3352	59,6582	9,09364	30,1607
8	-39,7528	80,1477	57,2841	9,47837	30,4203
9	-38,255	82,2587	59,4545	9,47837	30,7644
10	-42,8339	78,1565	56,9672	10,4846	30,5241
11	-39,6203	77,6124	63,766	9,86309	29,4165
12	-40,7435	79,2497	59,5739	10,0702	30,8356
13	-38,7248	77,7714	54,1685	9,3008	29,2607
14	-39,6344	79,6134	60,9187	10,3366	29,9537

15	-39,1918	79,1335	62,1449	10,8693	30,5587
16	-41,6211	79,208	57,4062	10,0702	29,6588
17	-37,4107	81,4106	60,8233	10,3662	29,7972
18	-42,6379	79,3558	58,0605	9,77431	30,9568
19	-42,9515	82,335	60,8372	9,3008	29,6242
20	-41,3126	77,6096	58,5783	10,1886	29,5549

Tabla 17: Muestras utilizadas para el 5to Proceso de prueba [Anexo A].

Los datos mostrados en la tabla anterior, contemplan aquellos adquiridos durante un proceso completo. Tal como se explicó anteriormente los procesos constan de dos experimentos que permiten analizar el resultado en conjunto, tanto de los movimientos en secuencia como de trayectorias referidas a un mismo punto, en tanto que el segundo experimento repite el destino del movimiento pero desde un punto de partida distinto verificando así el desenvolvimiento del robot de desde diferentes puntos pero referidos al mismo punto final.

Cada uno de los datos mostrados son el resultado de aplicar una media estadística de los datos entregados por cada uno de los sensores de posición del robot, después que este alcanza su punto final, es decir se toma la referencia estadística de una serie de datos entregados por el sensor de posición, de manera de acrecentar la veracidad de la muestra y el proceso de prueba como tal.

Como se observa en la siguiente figura, a pesar que la respuesta del sensor en relación al grado alcanzado es prácticamente lineal, existe una leve fluctuación la misma que pudiera ampliar el error en el posterior análisis de datos, razón por la cual se optó por tomar la media como muestra censada de un total de 100 muestras adquiridas por el sensor, como se explicó anteriormente.

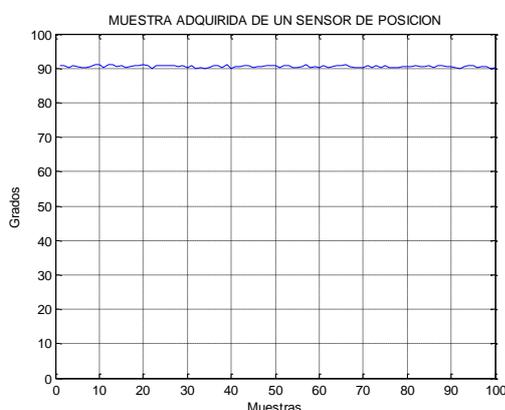


Figura 54. Muestra Adquirida de un sensor a ser aplicado en los procesos de prueba.

4.3.2 Análisis estadístico de los datos adquiridos.

Cada uno de los procesos realizados que son parte de las pruebas del sistema, como el mostrado anteriormente (Tabla 16), serán procesados por medio de herramientas estadísticas que permitirán determinar los parámetros de validación de las pruebas físicas realizadas. En primer lugar se utilizara el histograma como herramienta estadística para mostrar de forma gráfica los valores representados de cada variable, en este caso el movimiento y posicionamiento del robot por medio de cada una de sus articulaciones.

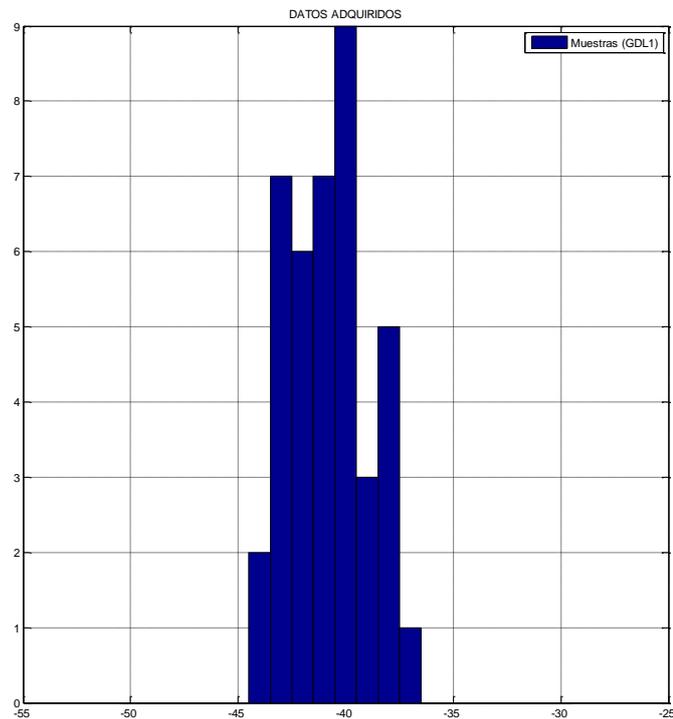


Figura 55. Histograma de las muestras del Primer grado de libertad del quinto proceso de prueba [Anexo B].

La herramienta permite observar una preferencia, tendencia o frecuencia por parte de la muestra para ubicarse en una determinada región o valor del rango completo, esto nos permitirá determinar referencias del comportamiento de un grado de libertad en particular frente a un movimiento dictado por la interfaz.

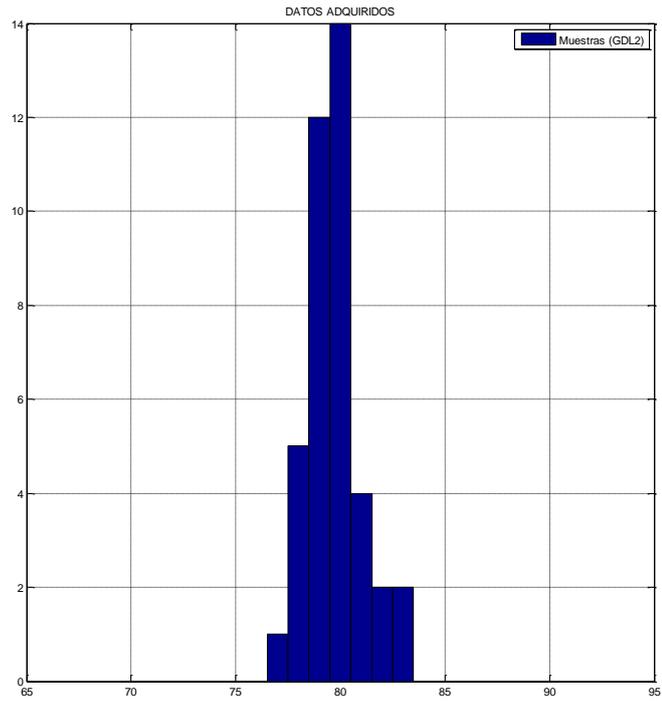


Figura 56. Histograma de las muestras del Segundo grado de libertad del quinto proceso de prueba [Anexo B].

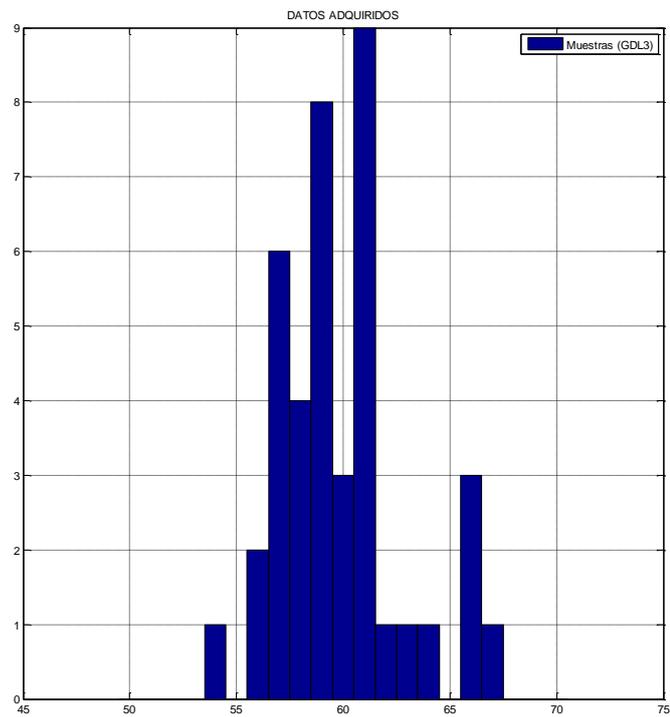


Figura 57. Histograma de las muestras del Tercer grado de libertad del quinto proceso de prueba [Anexo B].

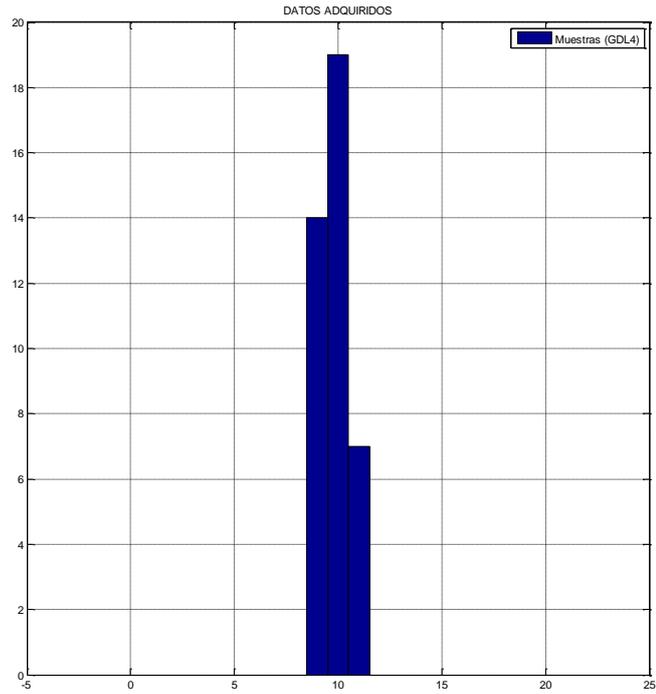


Figura 58. Histograma de las muestras del Cuarto grado de libertad del quinto proceso de prueba [Anexo B].

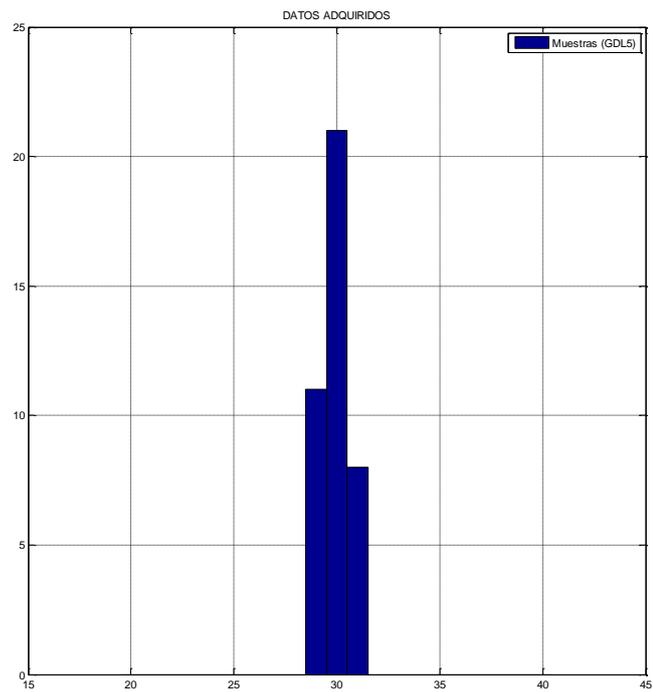


Figura 59. Histograma de las muestras del Quinto grado de libertad del quinto proceso de prueba [Anexo B].

Completando el análisis para todos los grados de libertad del proceso mostrado se establece así la frecuencia y error absoluto con el cual el robot llega a las posiciones

dictadas. Esta herramienta permitirá además calcular los parámetros de precisión y exactitud, procesos que se realizarán posteriormente.

La siguiente herramienta estadística que se utilizara es conocida como “distribución de probabilidad”, esta herramienta permite modelar fenómenos de diferente naturaleza de forma general sin necesidad de conocer incluso el fenómeno como tal. Al considerar varios tipos de distribuciones que son utilizadas dependiendo de su aplicación, será necesario para nuestro caso utilizar la distribución gaussiana o denominada distribución normal, esta es utilizada con mayor frecuencia al considerarse óptima para fenómenos reales; Su función de densidad tiene una forma acampana como se verá a continuación y expresa el resultado en valores comprendidos entre 0 y 1 permitiendo establecer criterios estadísticos en base al comportamiento de las muestras analizadas sin importar su naturaleza, la distribución satisface la siguiente ecuación.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-u}{\sigma}\right)^2}$$

Ecuación 6. Función de densidad de probabilidad normal [6].

Donde u (mu), es la media, σ es la desviación estándar y σ^2 en la varianza.

Debido a que los parámetros estadísticos sobre los cuales está en función la distribución serán utilizados para el análisis de datos, su conceptualización se la realizara posteriormente, mas es importante denotar ciertos parámetros que se obtendrán como respuesta del modelo obtenido de la distribución de probabilidad.

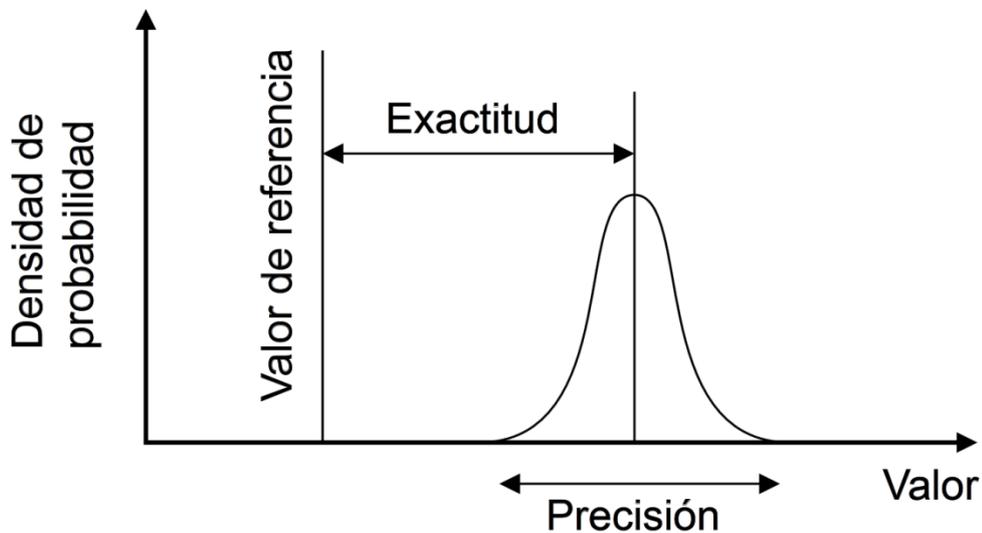


Figura 60. Parámetros relacionados a la distribución de probabilidad normal [6].

La imagen ilustra las medidas que nos interesan para medir la precisión y exactitud del sistema. Utilizando la función de densidad de probabilidad aplicada al ejemplo del proceso de prueba mostrado se presentan a continuación la respuesta del modelo estadístico para cada grado de libertad, de manera de utilizarlas posteriormente de forma general y en conjunto con los demás procesos para relacionar conceptualmente los parámetros mencionados y obtener las medidas requeridas del sistema completo.

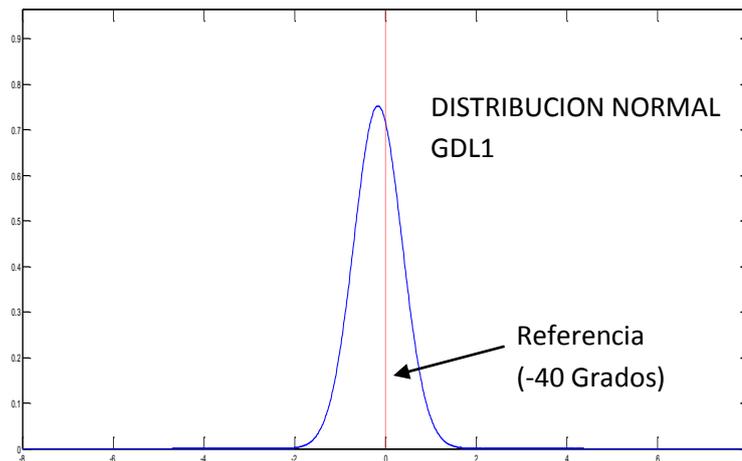


Figura 61. Función de densidad de probabilidad del primer grado de libertad del quinto proceso de prueba [Anexo B].

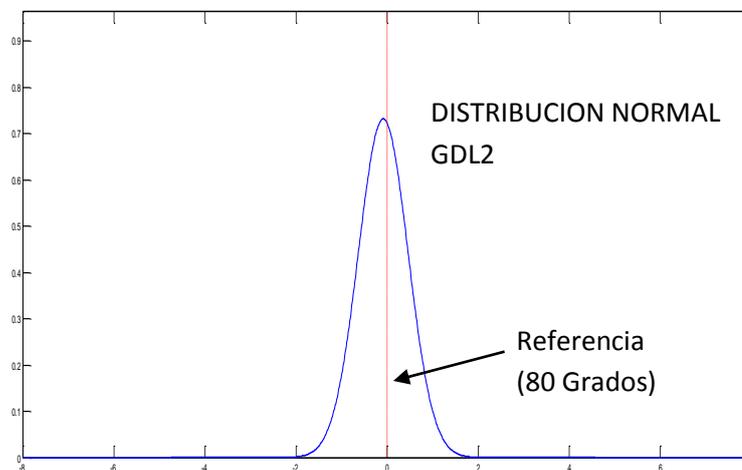


Figura 62. Función de densidad de probabilidad del segundo grado de libertad del quinto proceso de prueba [Anexo B].

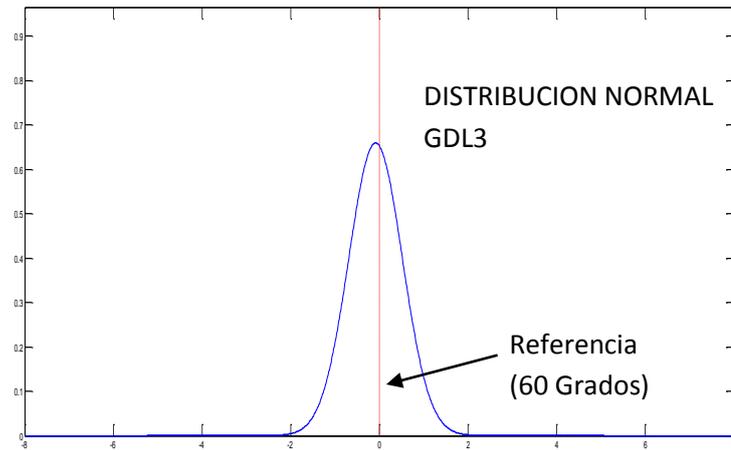


Figura 63. Función de densidad de probabilidad del tercer grado de libertad del quinto proceso de prueba [Anexo B].

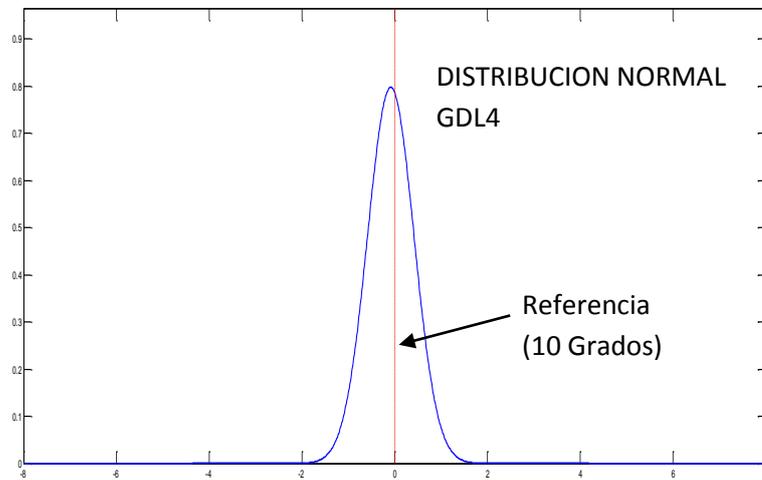


Figura 64. Función de densidad de probabilidad del cuarto grado de libertad del quinto proceso de prueba [Anexo B].

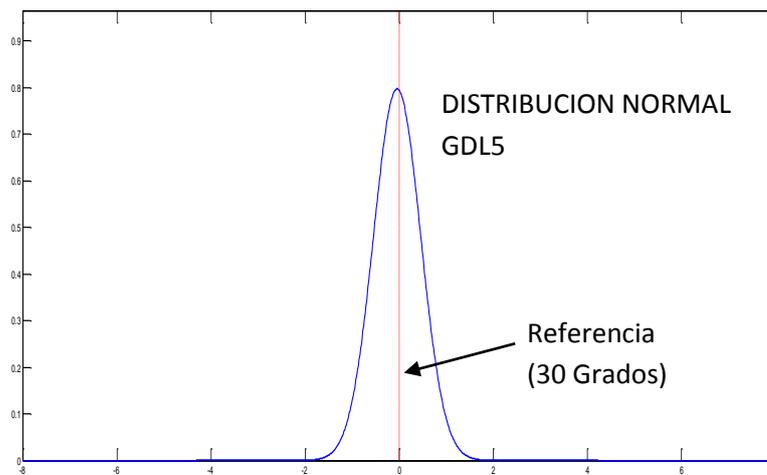


Figura 65. Función de densidad de probabilidad del quinto grado de libertad del quinto proceso de prueba [Anexo B].

Las gráficas muestran cierta relación con los histogramas presentados, debido a que la forma de la distribución también cambia en relación a la frecuencia donde se dan las muestras, mas no se permite apreciar las muestras de forma específica ni la localización y frecuencia real de muestras incorrectas, mas como se mencionó anteriormente la importancia de la distribución de probabilidad radica en el cálculo de los parámetros de precisión y exactitud que avalaran las pruebas y el sistema completo en base no solo del ejemplo presentado sino de todos los procesos de medición realizados.

Tal como se referencio en los histogramas y las funciones de probabilidad la información completa del análisis estadístico de todos los procesos y experimentos se encuentra adjunto a la presente (Anexo B), razón por la cual no se mostraran individualmente los resultados de todos los procesos ni experimentos, más se procederá al análisis general de los resultados obtenidos de manera de calcular los valores que avalaran los procesos de prueba realizados en el siguiente apartado.

4.4 ANÁLISIS DE RESULTADOS.

En este apartado se realizara el análisis de los datos adquiridos y procesados por medio de las herramientas estadísticas presentadas en el apartado anterior, este proceso permitirá esclarecer atributos del sistema de control y definir los parámetros relativos al software de comunicación desarrollado.

En una primera instancia se definirán y calcularan los parámetros estadísticos de precisión y exactitud en base al muestreo y análisis de todos los procesos de prueba realizados, posteriormente se realizara un breve análisis de la respuesta física por medio de la utilización de la respuesta a un cambio brusco de cada grado de libertad.

4.4.1 Calculo de precisión y exactitud.

En procesos de ingeniería, la precisión es la capacidad de un instrumento o sistema de dar el mismo resultado en mediciones diferentes realizadas en iguales condiciones [6]. Considerando los experimentos que fueron parte de los procesos de prueba se puede expresar que el procedimiento para la evaluación de este parámetro es correcto; mas es importante denotar que para el cálculo de precisión, que será considerado como general para el sistema se utilizara un cálculo específico en base a

reglas estadísticas, además de utilizar todos los experimentos realizados asegurando una relación más significativa al análisis de resultados.

Referidos a parámetros estadísticos, la precisión es la dispersión que tiene un conjunto de valores obtenidos de una medición específica normalmente referidos a una magnitud, la precisión es mayor mientras la dispersión de las muestras sea menor, análogamente existe una relación estadística que estima la dispersión de una muestra, y que es componente de una función de probabilidad normal, la desviación estándar, valor que será utilizado para el cálculo posterior del parámetro de precisión. La distribución normal también permite definir el parámetro de exactitud, al igual que la precisión, la exactitud puede ser referida en base a un parámetro estadístico, en este caso la media, el cual calcula la tendencia central de un conjunto de muestras.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Ecuación 7. Ecuación utilizada para el cálculo de la media o promedio [6].

Donde n es el número de datos de la muestra x .

La fórmula utilizada para el cálculo de la media, permitirá a su vez calcular el valor de exactitud que por concepto en un grupo de muestras es la diferencia entre el valor esperado y la media, la misma que será considerada como el valor medido.

$$\text{Exactitud} = |\text{Valor Medido} - \text{Valor Esperado}|$$

Ecuación 8. Ecuación utilizada para el cálculo de la exactitud [21].

Tomando como referencia el cálculo y parámetros de la media (Ecuación 7), se procederá a describir la ecuación que permite calcular la desviación estándar, valor que será utilizado como parámetro de precisión tal como se mencionó anteriormente.

$$\sigma = \sqrt{\left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)}$$

Ecuación 9. Ecuación utilizada para el cálculo de la desviación estándar [6].

El compendio del análisis estadístico (Anexo B), permite observar todos los parámetros estadísticos calculados para cada proceso de prueba en base a las muestras obtenidas de los experimentos, utilizando las ecuaciones anteriormente

definidas; más el cálculo general de todo el sistema se realizara en base a una aproximación utilizando en compendio los valores de exactitud y precisión de todos los datos obtenidos.

GRADO DE LIBERTAD	EXACTITUD	PRECISIÓN	PROCESO
GDL1	0.2376	1.5591	1
GDL2	2.0486	1.4419	
GDL3	2.2543	4.3419	
GDL4	0.2166	1.2152	
GDL5	0.4989	1.1625	
GDL1	1.7844	1.4864	2
GDL2	0.2369	2.1620	
GDL3	0.8721	2.8679	
GDL4	2.1482	3.0241	
GDL5	0.6643	2.2574	
GDL1	0.4833	1.1715	3
GDL2	1.0763	1.6882	
GDL3	3.3440	3.246	
GDL4	0.2944	0.9546	
GDL5	0.0279	0.5456	
GDL1	0.2301	1.4065	4
GDL2	0.1881	1.5372	
GDL3	0.2803	2.5286	
GDL4	0.0467	0.6465	
GDL5	0.1130	0.5714	
GDL1	0.8115	1.8888	5
GDL2	0.2572	1.4351	
GDL3	0.129	2.9519	
GDL4	0.1527	0.6	
GDL5	0.0961	0.5905	

Tabla18: Compendio de valores de exactitud y precisión calculados.

La tabla mostrada nos permitirá realizar una aproximación de la exactitud y precisión del sistema utilizando una media estadística; cabe recalcar que estos parámetros fueron calculados de forma específica para cada grado de libertad intencionalmente debido a los objetivos académicos de la interfaz creada, es necesario conocer las limitaciones eléctricas, electrónicas, mecánicas que el sistema al tiempo que se valida el proceso propio del software realizado, más un análisis de propio de control de posición, velocidad o control cinemático dependerá del uso de la interfaz por parte del usuario final, sobre todo en el espacio de programación gráfica, donde se podrá describir de mayor y mejor forma estos procesos.

GRADO DE LIBERTAD	EXACTITUD	PRECISIÓN
GDL1	0.70938	1.50246
GDL2	0.76142	1.65288
GDL3	1.37594	2.75858
GDL4	0.57172	1.28808
GDL5	0.27804	1.02548

Tabla19: Promedio de valores de exactitud y precisión del sistema.

El promedio de valores precisa la exactitud y precisión del sistema, como se mencionó anteriormente era importante el realizar pruebas aleatorias de movimientos que abarquen el mayor espacio del volumen de trabajo del robot de manera de realizar pruebas que sean relevantes al proyecto, este proceso nos permite observar que los diferentes experimentos acarrear ciertas características al análisis final, específicamente se puede denotar que dependiendo del experimento realizado los valores tanto de precisión como exactitud varían en todos los grados de libertad sin embargo la relación proporcional utilizada se considera correcta debido a una tendencia de cambio de las variables articuladas en un solo experimento.

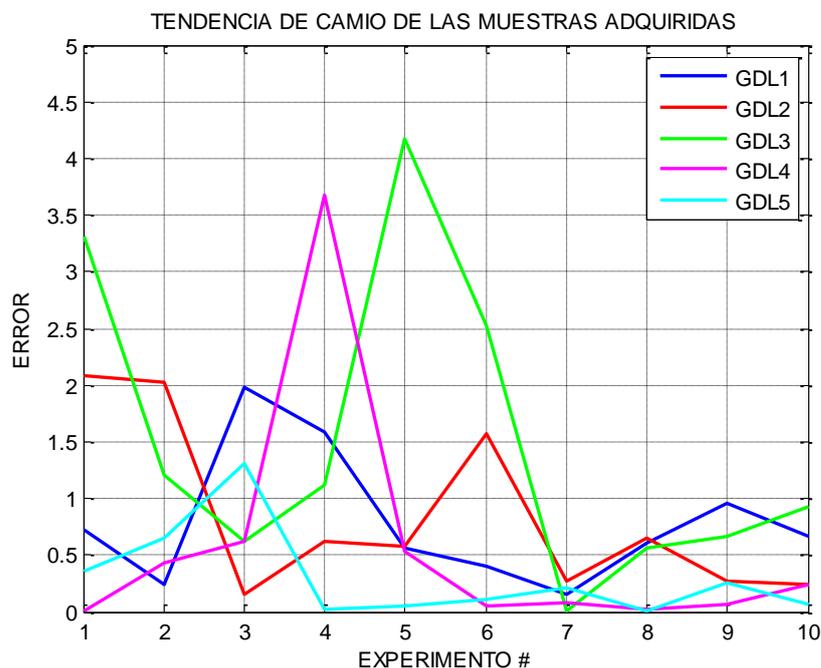


Figura 66. Tendencia de cambio de las muestras de los diferentes experimentos.

La imagen muestra que el tercer grado de libertad posee un error mayor a los demás, debido al error acumulado de inercia, es decir que al considerar como una cadena cinemática el manipulador robótico acarrea un error físico que se denota de mayor manera al final de esta, también cabe recalcar que los experimentos realizados

afectan la respuesta del sistema dependiendo si estos implican mayor esfuerzo mecánico para el robot, esta relación aunque no considerada para el cálculo de los parámetros de validación es importante para la relación física y del trabajo del manipulador robótico, razón por la cual se agregó un análisis estadístico grafico basado en cuartiles que proporciona una visión general de la simetría de las muestras para cada experimento (Anexo C).

4.4.2 Análisis de la respuesta física del sistema.

Este apartado se enfocara en el análisis del resultado físico que el manipulador robótico presenta a un movimiento de trayectoria articular o movimiento puntual. Partiremos de considerar el proceso de medición que se utilizó para la medición de posición de un grado de libertad en particular de manera generalizar su utilización, a todos los grados de libertad del manipulador robótico, después se expondrá un esquema grafico que relaciona la respuesta física del robot de manera de poder analizar ciertos criterios propios de control aplicados al proyecto.

El sistema de censado para cada grado de libertad está compuesto por un transductor activo de posición rotacional que cambia un movimiento rotacional a un valor equivalente de resistencia, adicionando un circuito eléctrico se adquirió el valor de la tensión suministrada por el sistema de censado.

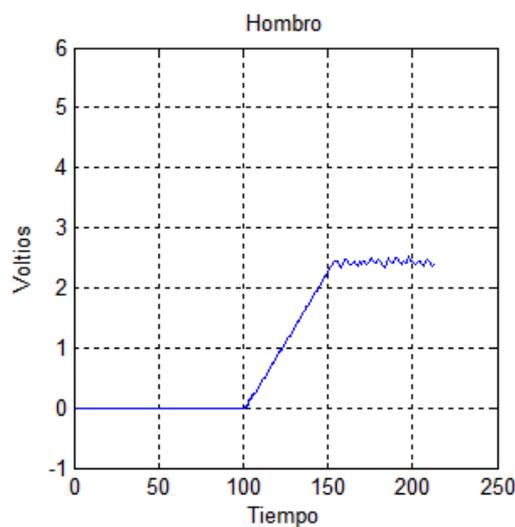


Figura 67. Valores medidos directamente del sensor de posición.

La imagen muestra la señal medida de tensión en voltios que el sensor suministra al sistema durante el movimiento de uno de los grados de libertad, a esta se aplicó una linealización en base al volumen de trabajo del robot y a cada uno de los sensores

asociados a cada grado de libertad. La ecuación mostrada a continuación define las variables a considerar para la linealización de cada sensor.

$$\theta = \left(\frac{\phi_{max} - \phi_{min}}{V_{max} - V_{min}} \right) (V - V_{min}) + \phi_{min}$$

Ecuación 10. Ecuación utilizada para la linealización del sensor de posición.

Donde θ es el valor de la variable articular en grados, ϕ_{max} & ϕ_{min} son los límites que definen el volumen de trabajo de cada grado de libertad, en grados, $V_{max} - V_{min}$ son los límites de tensión que el sensor puede entregar y V es el valor que el sensor tiene en cada momento o cambio de estado, medido en voltios.

El proceso de censado permitió formar un diagrama con los valores obtenidos de cada variable articular durante un movimiento específico, permitiendo perfilar el comportamiento físico real que cada eslabón asociado a estas variables presenta.



Figura 68. Respuesta física a un movimiento puntual del GDL1.

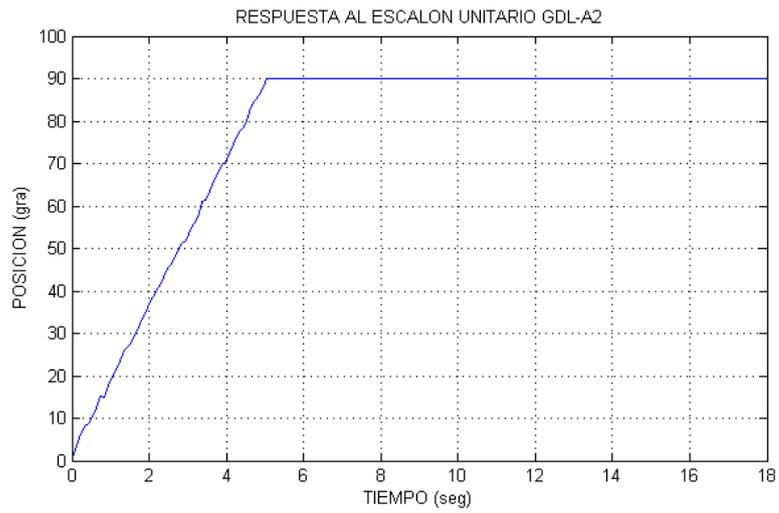


Figura 69. Respuesta física a un movimiento puntual del GDL2.

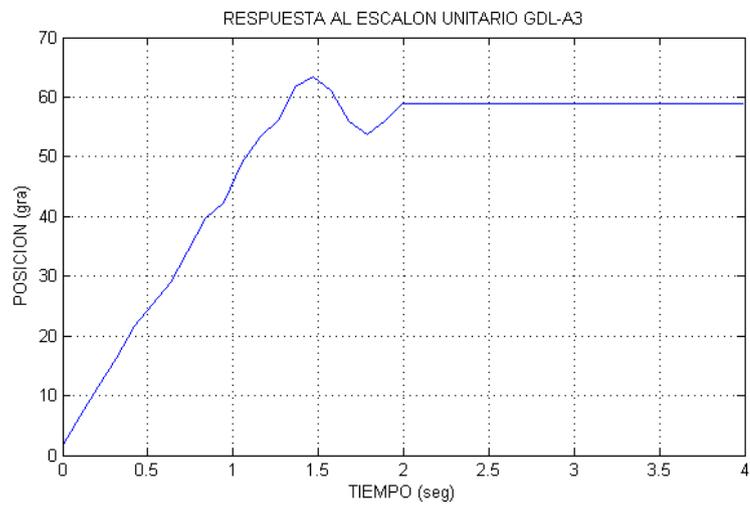


Figura 70. Respuesta física a un movimiento puntual del GDL3.

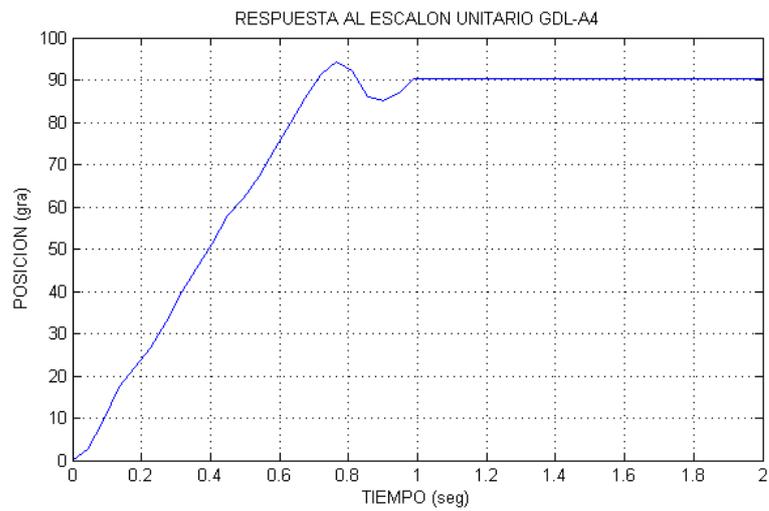


Figura 71. Respuesta física a un movimiento puntual del GDL4.

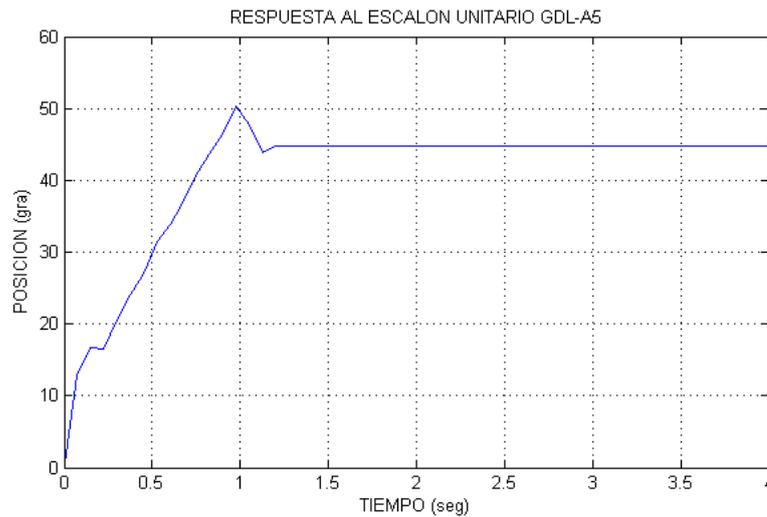


Figura 72. Respuesta física a un movimiento puntual del GDL5.

De las imágenes mostradas, se puede obtener ciertos parámetros de control que permiten establecer criterios sobre la resolución física que tiene el robot y especificaciones durante un movimiento, es decir desde que comienza una trayectoria y hasta llegar al nivel de referencia deseado.

Grado de libertad	Tiempo de establecimiento.	Valor real alcanzado.
GDL1	5.670 seg.	89.29062°
GDL2	5.024 seg.	90.76142°
GDL3	2.042 seg.	58.62406°
GDL4	1.030 seg	90.57172°
GDL5	1.213 seg.	44.72196°

Tabla 20: Parámetros de respuesta del movimiento mecánico real.

Las pruebas realizadas, tenían como objetivo el evaluar el comportamiento del robot tras ser expuesto a un movimiento brusco, todos fueron concebidos desde un punto inicial donde le valor de la variable articular era cero grados y movidos hacia un punto específico. Todos los valores de precisión y exactitud que fueron desarrollados anteriormente exponen un error de cierta forma en estado estable, es decir una vez que las articulaciones del robot alcanzan sus valores articulados propuestos de referencia, mas es importante definir los parámetros antes enlistados para tener una referencia del comportamiento del robot durante la trayectoria.

Del análisis previo se pudiera denotar que la respuesta de aquellos grados de libertad previo el tiempo de establecimiento que poseen un error mayor es muy rápida, es decir denota una respuesta subamortiguada, dependiendo de la secuencia que se

deseo realizar pueden llegar a tener un máximo paso de cuatro grados más a la referencia, el mismo que físicamente es imperceptible; en cuanto a las respuestas de aquellos grados de libertad que poseen un error mínimo y cuyo error en estado estable es muy cercano a cero son más lentos razón por la cual se pudieran considerar críticamente amortiguados, mas al tratarse de los dos primeros grados de libertad y considerando el peso que estos deben mover la respuesta se considera satisfactoria.

CAPITULO 5. RESUMEN, CONCLUSIONES Y RECOMENDACIONES.

En este capítulo se expondrán las principales ideas desarrolladas durante la monografía a manera de resumen permitiendo una adecuada elaboración de conclusiones del proyecto y recomendaciones sobre su uso y futura implementación. Este compendio enfocara los diversos capítulos tratados, de forma específica dando una mejor percepción de los temas tratados en cada uno

5.1 CAPITULO 1.

Durante el desarrollo del primer capítulo se recopiló información conceptual sobre robótica en general, haciendo énfasis en las nociones necesarias para el proyecto, como el efecto que este campo de estudio tiene en la sociedad e industria, razón por la cual es tan importante su enseñanza tanto teórica como práctica. Posteriormente se realizó un estudio preliminar sobre la programación de robots industriales teniendo como principal objetivo describir el estado del arte actual en este tema; estos preceptos permitieron dimensionar el proyecto y aumentar su validez sobre todo en el ámbito académico. Después de realizado el análisis respectivo sobre la arquitectura de programación se pudo especificar de mejor manera el objetivo alusivo a la apertura que la interfaz pretende dar a los estudiantes, que la utilicen para realizar innovación referida a los niveles actuales sobre los cuales se define la programación industrial de robots. Se realizó a continuación una descripción más específica de los métodos de programación utilizados para manipuladores robóticos, este estudio a su vez se reflejó en gran medida durante el diseño del software de control desarrollado para el proyecto. Las conclusiones de este capítulo son:

- El estudio preliminar sobre la actividad actual tanto en el campo de aplicación específica orientada a la academia, así como en el campo de innovación fue muy importante debido a que permitió dimensionar el proyecto correctamente además de establecer importantes parámetros de diseño que fueron utilizados en la programación de la interfaz en ambas

etapas, permitiendo concentrar los esfuerzos en la correcta aplicación de las herramientas de visualización, comunicación y control que la interfaz posee.

- El análisis correspondiente a los tipos de programación y en mayor medida a los prototipos y aplicaciones utilizados actualmente, permitió programar de forma dinámica la interfaz asegurando un mejor uso de la misma y diseño de prestaciones que no solo permitirán al usuario final utilizar el prototipo para avalar los conocimientos teóricos desarrollados de forma conceptual sino que tendrá la posibilidad de innovar en campos de programación aun no utilizados.

5.2 CAPITULO 2.

El desarrollo del segundo capítulo se centró en la descripción de todos los componentes que constituyen un brazo robot antropomórfico, tipo de robot sobre el cual se desarrolló el proyecto y eran necesarios conocer para la elaboración de la interfaz, centrando el estudio en los componentes físicos del robot, además de los principales parámetros de su actuar. Posteriormente se analizó los cálculos utilizados para el posicionamiento físico del robot, estos cálculos además serían utilizados como base para la implementación de las funciones de trayectoria que posteriormente se analizarían durante la programación de la interfaz. En este punto del proyecto además se realizó énfasis en la importancia académica que tiene un acercamiento práctico profundo para mejorar el aprendizaje de conceptos sobre robótica industrial, a más que solo un acercamiento práctico sobre dispositivos industriales, razón por la cual después de definir los elementos constitutivos de los manipuladores se realizó un estudio de implementación basado en las características necesarias para realizar la instalación de software y hardware sobre un robot industrial real de manera de en el futuro ampliar el alcance del proyecto y objetivarlo a la innovación basada completamente sobre aplicaciones industriales, este estudio complementario permitió definir los parámetros a los cuales se desea llegar en un ámbito académico acoplando el diseño de la interfaz a las necesidades propias del proyecto y en base al manipulador robótico sobre el cual se realizó el proyecto de implementación práctica. Las conclusiones obtenidas de este capítulo son:

- A pesar de contar con atributos especiales propios de un sistema robótico industrial la aplicación directa de programación de alto nivel no permite denotar ciertos aspectos que son necesarios para la enseñanza integral de la robótica, considerando la aplicabilidad académica, utilizar un robot cuyo lazo de control sea manipulable y sobre el cual se pueda observar todos los componentes constituyentes de un manipulador permitió no solo abarcar todos los temas tratados en las asignaturas referentes a robótica sino que permite trascender a asignaturas de control basando la práctica de los estudiantes en una planta de múltiples variables.
- La validez del proyecto fue probada durante el desarrollo de los temas tratados en las asignaturas referidas a la robótica industrial, a pesar que el prototipo de la interfaz desarrollado no pudo implementarse debido a que este necesariamente debe ser tratado sobre un manual de prácticas que especifique relación con el esquema capitular de las materias, se pudo asegurar la validez del proyecto planteando la necesidad de un mayor acercamiento a plataformas robóticas que permitan una mayor interacción con el usuario.
- El estudio complementario realizado en este capítulo, permitió establecer las principales recomendaciones del proyecto posterior a su implementación, debido principalmente a que la siguiente etapa en el estudio de la asignatura debería concentrarse en su totalidad en la programación de los manipuladores robóticos considerando estándares industriales específicos, el estudio facilitara la ejecución del proyecto al presentar en detalle los principales parámetros de control uso e instalación de los programas que utilizados para esta etapa del proyecto, permitiendo además al usuario proyectar aplicaciones netamente de innovación en el área.

5.3 CAPITULO 3.

El tercer capítulo comprendió, en primer lugar la conceptualización de un sistema de desarrollo integrado, de manera de dar realce y denotar la importancia que un sistema de estas características aportaría al proyecto, las razones por la cual se optó por las plataformas de Matlab y Simulink, y una breve descripción de los parámetros que estos programa poseen y aportarían para el desarrollo de la interfaz, posteriormente

se realizó la descripción del desarrollo de la interfaz comenzando con el diseño de ambas etapas, la estructuración de las herramientas y la programación de los algoritmos de comunicación y control con el manipulador robótico, cada proceso implicó un proceso de diseño basado en las especificaciones necesarias para solventar los objetivos del proyecto. Fue importante denotar en primera instancia el proceso por el cual se concibieron los atributos gráficos y la disposición de las herramientas para la interfaz gráfica de usuario, cada una de las herramientas de control o movimiento, se escogieron para asegurar la correcta función de un tipo de programación como los descritos en capítulos anteriores, o aseguramiento de los movimientos y parámetros necesarios para la estructuración de una secuencia de movimientos; para después ser programados en funciones que podrán ser utilizadas en un solo programa. Los objetos descritos en el programa que no formaban parte de un proceso cuya finalidad era el control del brazo robot, es decir aquellos que cumplían las funciones de guardado, apertura, información o publicación de reportes se diseñaron por el contrario bajo parámetros básicos de interacción académica, es decir herramientas cuyo objetivo es netamente académico y complementa la funcionalidad del proyecto para los objetivos bajo los cuales fue programada. La siguiente etapa de la interfaz de control contemplo un estudio preliminar sobre la programación gráfica y sus fundamentos, en donde se pudo observar la estrecha relación que tiene con la programación textual, y la apertura para la creación de nuevas librerías basadas en diseños específicos. Esto permitió diseñar y guardar librerías propias del robot que fueran utilizadas desde el programa principal y que internamente poseen los parámetros de comunicación necesarios para él envío de sentencias de control hacia el robot, permitiendo al estudiante elaborar controladores diferentes a los utilizados en la primera etapa de la interfaz, al tiempo que podrán hacer uso de otros Toolbox para proyectos o prácticas de innovación referida a la robótica industrial o la formas de programación avanzada. Finalmente se explicó el proceso por el cual se programó los sistemas de comunicación y el algoritmo de control implantado en la interfaz de usuario, haciendo énfasis en la tecnología utilizada y las características de los elementos utilizados para la comunicación y control. De este capítulo se obtuvieron las siguientes conclusiones.

- La investigación previa a la elaboración del prototipo realizado para el proyecto, en especial al comparar interfaces de usuario cuyo objetivo es el de

controlar elementos físicos, permitió escoger el software de desarrollo integrado Matlab & Simulink cuya principal aplicabilidad es en el campo académico y de investigación, más la razón principal para optar por este software es debido a las prestaciones del mismo, Matlab a diferencia de sus pares cuya programación se realiza en forma descrita o secuencial, y que además permite el desarrollo de interfaces gráficas de usuario como C++, Plataformas de Java, Optab, etc. Posee una característica propia y sumamente importante para el proyecto que es la interacción directa con el ambiente de programación gráfica, herramienta que solventa por sí sola el objetivo de permitir innovación por parte del usuario en el ámbito de la robótica industrial. Cabe recalcar que varios proyectos similares actualmente comerciales y utilizados en ámbitos académicos utilizan una plataforma de programación gráfica para el control de diversas plantas electromecánicas; En su mayoría estos dispositivos utilizan “Simulink” al igual que el presente proyecto o su similar de la empresa National Instruments “Labview”, mas ambos tipos de control presentan una deficiencia en lo que respecta a la conceptualización de varios temas que se abordarían en una asignatura académica estructurada, sino que únicamente se limitan a brindar herramientas para realizar prácticas de laboratorio. Esta deficiencia se pretende solventar con la etapa de la interfaz desarrollada como una interfaz gráfica de usuario. Considerando lo expuesto es necesario entonces acotar las principales recomendaciones sobre el tema, las cuales implican la estructuración de un esquema que indique el alcance de la interfaz y las asignaturas a las cuales se pudiese acoplar, es decir que es necesario en base al temario académico realizar un manual de prácticas en función de los contenidos de las mencionadas asignaturas utilizando las herramientas que proporciona la interfaz.

- El desarrollo de la interfaz de usuario permitió definir individualmente las herramientas que fueron programadas para solventar los distintos objetivos de la interfaz de usuario, la principal característica es la interacción que cada herramienta tiene con las demás, en base al tipo de programación escogida, permitiendo al usuario acceder no solo a un control completo del robot sino que a su vez podrá tener una retroalimentación y un compendio descrito de la

utilización de la interfaz de forma general. La herramienta que relaciona el diseño en realidad virtual permitió acrecentar las prestaciones de la interfaz, debido a que el usuario tiene la oportunidad de programar el robot sin necesidad de realizar movimientos físicamente, el movimiento “offline” como es conocido además de asegurar el correcto movimiento secuencial del robot, asegura la integridad física del manipulador así como de los elementos a su alcance y el programador, el dinamismo de la herramienta permite también interacción con el usuario durante la ejecución de las trayectorias, esto permite aseverar el movimiento real del manipulador al tiempo que asegura que los parámetros mostrados en el panel de visualización sean correctos. La siguiente herramienta que interactúa con los diferentes métodos de programación así como con el usuario permitiendo asegurar un correcto desempeño de las secuencias programadas así como la respuesta física del manipulador son las dispuestas en el panel de trayectorias, el dinamismo de la base de datos que alberga las trayectorias supone una forma de interacción con el usuario a un nivel más avanzado, debido a que dependiendo de la forma de programación, esta base de datos no solo muestra los datos adquiridos por el modelo en realidad virtual o las herramientas de programación “offline”, sino que además guarda los datos procesados de las trayectorias descritas por las ecuaciones en la programación continua, no solo mostrando el resultado del cálculo preliminar sino permitiendo al estudiante asegurar el resultado del movimiento que pretende dar al robot. Las últimas herramientas que aportan significativamente al desarrollando del usuario en la interfaz son aquellas que se programaron objetivamente para fines académicos, estas pretenden complementar lo realizado por el usuario para documentar lo realizado y dar continuidad al proyecto además de ser un apoyo académico tras realizar una práctica o proyecto acorde a un tema específico de la asignatura.

- Los parámetros de comunicación dispuestos en la programación global de la interfaz se observan de primera mano en las librerías creadas para ser utilizadas en la etapa de programación gráfica, como se mencionó los atributos de esta permite ampliar las posibilidades que el usuario tiene, no solo para innovar en procesos afines a la robótica sino de programar

controladores más robustos o que cumplan con un fin académico específico, por ejemplo el comparar la respuesta física de los diferentes tipos de controladores o las características de estos, mas es importante denotar ciertas características y limitaciones que esta etapa tiene, por ejemplo es necesario conocer de antemano el protocolo de comunicación creado, la forma de transmisión de la señal binaria de control y la forma como se presenta las señales de retroalimentación, además el usuario deberá necesariamente tener conocimientos previos sobre el uso de Simulink, de manera de aumentar las posibilidades de interacción con el robot y el uso de otros “toolbox” que ayuden al usuario a resolver un problema específico, mas al considerarse una herramienta utilizada usualmente, en el campo académico especialmente en aplicaciones que involucran diseño de prototipos y control de dispositivos se considerara un prerrequisito ya solventado.

- Todas las herramientas programadas y dispuestas en la interfaz se realizaron con fines académicos, y tomando en consideración el objetivo que pretende dar un acercamiento más practico a los estudiantes, el hardware utilizado para la comunicación con el manipulador robótico debía ser utilizado bajo iguales parámetros es decir que los dispositivos utilizados deben ser familiares para el estudiante, permitiendo incluso la manipulación del hardware de control por parte del estudiante de ser necesario. La tarjeta de adquisición de datos utilizada para el proyecto permite una interacción directa con el software de control proporcionando no solo una mejor comunicación bajo parámetros estándar específicos sino que además permite el desarrollo de aplicaciones en tiempo real, al conocer de forma específica los parámetros y protocolo del sistema de comunicación las aplicaciones se amplían a los diseños que el estudiante pueda realizar solventando así los objetivos relacionados a la innovación en temas de robótica industrial.

5.4 CAPITULO 4.

El cuarto capítulo expone, las pruebas de software y funcionamiento electromecánico realizadas para el proyecto, terminada la estructuración de la interfaz de programación así como el acondicionamiento del hardware de control, de forma general las pruebas se dividieron en aquellas que analizaron la interfaz como tal, es decir las denominadas pruebas de software, y por otro lado las pruebas de

comunicación las mismas que involucraron un análisis más exhaustivo basado en pruebas de laboratorio en cuanto al comportamiento y respuesta física del robot tras recibir las señales de control desde la interfaz. Las primeras pruebas consistieron en un análisis comprobatorio que permitió encontrar varias fallas en lo que respecta al software, solventando los parámetros de cada función, algoritmo y herramienta dispuesta en la interfaz de manera de asegurar su correcto funcionamiento, las pruebas se orientaron obviamente a la aplicación, haciendo énfasis en la correcta movilidad del manipulador y la interacción de los diversos servicios con otras herramientas y sobre todo entre las etapas que conforman la interfaz; Cabe recalcar que estas pruebas fueron diseñadas en base a parámetros normados específicos para la comprobación de software, estas reglas no solo permitieron un análisis secuencial y correcto sino que suponen una respuesta que valide la programación de la interfaz. El segundo grupo de pruebas se concentró en realizar diversos procesos, que a su vez estaban constituidos de experimentos los cuales, básicamente se referían al movimiento del robot en secuencias de movimientos y trayectorias, de los cuales se adquiría datos de los sensores electromecánicos de posición del robot, mismos que eran sometidos a una adecuación de manera de realizar un procesamiento estadístico y posterior análisis, estos datos permitieron asegurar no solo la validez del proyecto sino también algunos parámetros de interés académico especialmente en el ámbito mecánico que afectaron en cierta medida el comportamiento del robot. Las conclusiones de este capítulo son las siguientes:

- Previo la aplicación de las pruebas orientadas al software estas fueron estructuradas en base a normativas propias para un programa de computador específico, aunque el proyecto de forma general está orientado al control de un proceso físico por medio de un circuito electrónico, el prototipo desarrollado en realidad es netamente un software, razón por la cual era necesario asegurar que las funciones programadas y todas las herramientas dispuestas en la interfaz cumplan con sus objetivos, por esta razón se vio importante no solo concentrarse en realizar pruebas físicas del proyecto sino también pruebas propias de un programa informático, en este caso ambas etapas que conforman la interfaz de usuario. Los casos de pruebas expusieron todos los posibles errores de diseño, programación o ejecución del programa permitiendo así una depuración completa, este proceso se logró debido a la

correcta utilización de los lineamientos para el diseño de los casos de prueba, al ser estándares internacionalmente aceptados se puede concluir con un resultado satisfactorio que cumpla con los objetivos y expectativas del proyecto. Además de las pruebas realizadas, es decir caja negra y blanca, existen otro tipo de pruebas específicas para el desarrollo de software, entre las más conocidas están las pruebas por pares, pruebas alfa y beta, sin embargo estas pruebas no se pueden ser aplicadas para el proyecto por tratarse de un prototipo cuya aplicación debe ser evaluada no solo por usuarios finales, sino también por desarrolladores con experiencia, es decir estudiantes y docentes respectivamente, además estas pruebas como se mencionó anteriormente es necesario realizarlas en base a una planificación curricular, permitiendo así un análisis completo y más exhaustivo previo a la implementación final del proyecto.

- El análisis estadístico evidenció una respuesta física del robot a los parámetros de control de la interfaz muy aceptable evidenciando no solo la correcta programación de la misma sino la factibilidad que el robot puede tener para una aplicación específica como se vio en el desarrollo del capítulo, estos datos fueron procesados a partir de una adquisición de datos reales entregados directamente por el robot en función de su posición, lo que también permite concluir que el nivel de aceptabilidad de dichos sensores es bueno; el proceso por el cual las señales eléctricas se acondicionaron a parámetros físicos admito no solo manipularlos sino que además permitió un análisis más exhaustivo y que otorgue una mayor percepción al resultado real de los movimientos secuenciales del robot, así como la efectividad que el sistema de retroalimentación tiene. Mas es importante denotar que los parámetros medidos de precisión y exactitud varían en base al experimento que se realice, y de forma general a la trayectoria que se desee realizar, por esta razón es importante señalar que los algoritmos de control que se utilicen pueden presentar variaciones dependiendo obviamente del tipo de control utilizado y si se realizan variaciones de en el protocolo de comunicación, esto implica entonces que al utilizar la interfaz los proyectos a realizarse sobre todo en el ámbito de la innovación deben en la medida de lo posible atenerse

a los procesos de control definidos en la monografía asegurando así una mínima respuesta física a lo mostrado en el análisis de resultados.

- Las respuestas físicas del manipulador robótico, al igual que los parámetros de precisión y exactitud se realizaron y definieron por separado para cada grado de libertad del robot debido a que las especificaciones técnicas del robot en lo que respecta a la estructura mecánica implica un error de inercia, esto es debido a que el robot como tal puede ser concebido como una cadena cinemática, es decir una sucesión de eslabones que a pesar de tener independencia de movimiento entre si también tienen cierta relación en especial en lo que refiere a atributos mecánicos, al momento de realizar un movimiento relativo a una función de trayectoria, estos atributos afectan el desenvolvimiento de los eslabones que se encuentran más lejanos en la cadena cinemática base, esta es la razón por la que el tercer grado de libertad presenta un error superior a sus predecesores, es decir que no solo tiene su error propio en estado estable sino que además suma los errores de inercia de la estructura mecánica que se acrecienta cuando los grados de libertad anteriores se mueven. Si consideramos entonces este precepto es por añadidura suponer las razones por las cuales los experimentos que implicaban un movimiento que debía superar no solo el peso del robot como tal sino además la gravedad propiamente dicha, tenían un error mayor en especial en el segundo y tercer grado de libertad, debido a que el error por sí mismo aumentaba, además de verse afectado por una mayor inercia del robot, todo este análisis implica un resultado obvio en cuanto a lo que respecta a la precisión y exactitud del sistema completo como tal. Aunque de forma casi imperceptible tanto en la relación física como los atributos de control de la interfaz existe también un error de redondeo y de apreciación, dispuestos por el método de adquisición de datos y linealización de los sensores, este error que no se puede medir al considerarlo como intrínseco de los atributos del instrumento de medición se descartó y por lo mismo no se considera como significativo para el desenvolvimiento del robot durante las pruebas físicas ni los algoritmos de control dispuestos en la interfaz.

APÉNDICES Y ANEXOS.

ANEXO A

**DATOS ADQUIRIDOS DE LOS
PROCESOS DE PRUEBA**

Experimento General: Movimiento Secuencial de todos los actuadores a la vez (Secuencia) y uno por uno (Cada uno)											
Repetición/GD L	A1		A2		A3		A4		A5		
	Medido	Referencia									
Experimento 1 punto inicial: GD1=GD2=GD3=GD4=GD5=0 a un punto GD1=-70, GD2=-90, GD3=-45 GD4=GD5=0											
1	Cada Uno	-71,1103	-70	-92,8194	-90	-44,3335	-45	-1,9426	0	-1,1736	0
	Secuencia	-72,1164	-70	-91,2542	-90	-50,9352	-45	-0,641	0	-1,0315	0
2	Cada Uno	-70,4769	-70	-93,1896	-90	-43,4769	-45	1,0806	0	-0,898	0
	Secuencia	-71,1769	-70	-92,5463	-90	-49,7503	-45	-0,9892	0	-1,1462	0
3	Cada Uno	-69,0319	-70	-92,599	-90	-47,5598	-45	0,6051	0	0,7015	0
	Secuencia	-72,2113	-70	-92,3414	-90	-52,0418	-45	-0,8825	0	-1,0941	0
4	Cada Uno	-69,8555	-70	-93,3931	-90	-48,1822	-45	-1,801	0	0,9401	0
	Secuencia	-72,241	-70	-91,6001	-90	-49,0705	-45	0,6416	0	0,6786	0
5	Cada Uno	-69,8948	-70	-92,2692	-90	-45,646	-45	3,3068	0	1,7012	0
	Secuencia	-70,7858	-70	-91,1301	-90	-50,1627	-45	-0,6897	0	-0,6131	0
6	Cada Uno	-68,7026	-70	-89,3494	-90	-49,0954	-45	0,9755	0	0,3001	0
	Secuencia	-73,0995	-70	-93,0272	-90	-50,4453	-45	-0,8141	0	-1,7015	0
7	Cada Uno	-70,6557	-70	-91,1138	-90	-43,6091	-45	1,6513	0	1,0886	0
	Secuencia	-70,7014	-70	-91,8114	-90	-50,4291	-45	-0,0864	0	-1,6825	0
8	Cada Uno	-68,8638	-70	-92,0302	-90	-45,1388	-45	1,0409	0	0,6525	0
	Secuencia	-71,906	-70	-91,8934	-90	-51,2696	-45	0,761	0	-1,6337	0
9	Cada Uno	-70,3083	-70	-93,8364	-90	-39,732	-45	1,124	0	1,9412	0
	Secuencia	-70,39	-70	-91,91	-90	-51,2323	-45	-0,9133	0	-1,5864	0
10	Cada Uno	-69,6207	-70	-91,3447	-90	-54,0246	-45	-1,3281	0	-1,6515	0
	Secuencia	-71,2839	-70	-92,0757	-90	-50,0006	-45	-1,1539	0	-0,8301	0

PROCESO 1

Experimento 2 punto inicial: GD1=-87, GD2=-75, GD3=-75, GD4=0, GD5=-17 a un punto GD1=-70, GD2=-90, GD3=-45 GD4=GD5=0

11	Cada Uno	-70,0836	-70	-91,3852	-90	-53,269	-45	1,5023	0	0,6425	0
	Secuencia	-70,7175	-70	-93,0217	-90	-46,0994	-45	-0,8132	0	-1,1115	0
12	Cada Uno	-70,0456	-70	-91,5505	-90	-41,0982	-45	0,9944	0	1,924	0
	Secuencia	-70,868	-70	-93,9442	-90	-52,2425	-45	-0,3261	0	-0,9751	0
13	Cada Uno	-68,0133	-70	-91,9118	-90	-38,7836	-45	0,4901	0	0,9761	0
	Secuencia	-71,4066	-70	-93,1305	-90	-51,2698	-45	-1,0117	0	-0,6515	0
14	Cada Uno	-68,6281	-70	-93,1256	-90	-46,2515	-45	1,4418	0	0,9218	0
	Secuencia	-71,3255	-70	-93,0635	-90	-52,5332	-45	-1,2498	0	-1,6897	0
15	Cada Uno	-68,9132	-70	-91,6014	-90	-46,4513	-45	-1,2245	0	-1,9817	0
	Secuencia	-71,4768	-70	-93,9268	-90	-51,561	-45	-0,3963	0	-1,1261	0
16	Cada Uno	-66,9033	-70	-90,776	-90	-38,5825	-45	-0,9732	0	-1,2492	0
	Secuencia	-73,0255	-70	-93,0615	-90	-47,6504	-45	-0,3581	0	-1,1951	0
17	Cada Uno	-69,4129	-70	-90,8766	-90	-47,2239	-45	-1,3447	0	-1,4417	0
	Secuencia	-66,6125	-70	-88,095	-90	-49,3166	-45	-1,7114	0	-1,1244	0
18	Cada Uno	-68,7285	-70	-92,3922	-90	-39,6546	-45	-1,4034	0	-1,3189	0
	Secuencia	-71,314	-70	-90,636	-90	-46,0973	-45	1,525	0	0,7481	0
19	Cada Uno	-69,4319	-70	-93,248	-90	-46,1152	-45	-1,1115	0	-1,1515	0
	Secuencia	-66,8768	-70	-87,4106	-90	-39,8313	-45	0,0963	0	0,1513	0
20	Cada Uno	-70,4934	-70	-93,4137	-90	-48,851	-45	-1,5389	0	-1,3992	0
	Secuencia	-70,7955	-70	-93,8379	-90	-41,1535	-45	-1,1966	0	-1,8673	0

PROCESO 1

Experimento 3 punto inicial: GD1=80, GD2=-75, GD3= -50, GD4=GD5=0 a un punto GD1=40, GD2=-40, GD3=35 GD4=80, GD5=0											
21	Cada Uno	38,0782	40	-38,5778	-40	39,9904	35	78,9904	80	1,1251	0
	Secuencia	37,4349	40	-38,2372	-40	30,4134	35	74,7512	80	0,5681	0
22	Cada Uno	40,6618	40	-33,7442	-40	31,7377	35	82,2056	80	-3,5994	0
	Secuencia	37,3786	40	-42,3381	-40	32,3021	35	75,8201	80	-0,9482	0
23	Cada Uno	36,3111	40	-38,3331	-40	33,8353	35	82,8257	80	-3,3057	0
	Secuencia	36,1609	40	-41,8064	-40	30,1415	35	80,3554	80	2,3957	0
24	Cada Uno	37,2996	40	-37,6302	-40	36,8647	35	80,2982	80	0,8145	0
	Secuencia	37,2109	40	-42,1532	-40	36,39	35	80,2826	80	-3,1407	0
25	Cada Uno	40,0617	40	-41,6425	-40	33,841	35	77,0129	80	-0,5761	0
	Secuencia	39,9782	40	-43,6376	-40	32,114	35	76,6491	80	-3,0446	0
26	Cada Uno	37,7888	40	-38,485	-40	38,3902	35	78,5992	80	2,9829	0
	Secuencia	35,6115	40	-40,6416	-40	31,1065	35	80,5231	80	-2,1742	0
27	Cada Uno	37,4923	40	-37,3594	-40	37,9486	35	82,228	80	-3,1461	0
	Secuencia	37,2504	40	-41,9932	-40	32,8158	35	76,447	80	-3,5152	0
28	Cada Uno	38,5276	40	-38,5539	-40	39,0687	35	81,6478	80	1,5551	0
	Secuencia	38,5835	40	-38,5403	-40	30,2748	35	75,9293	80	-2,948	0
29	Cada Uno	38,1159	40	-40,4222	-40	37,4929	35	84,4933	80	-3,7013	0
	Secuencia	38,1837	40	-40,1036	-40	30,3914	35	83,16	80	-3,2971	0
30	Cada Uno	37,9597	40	-38,8838	-40	37,9635	35	75,4901	80	-3,0981	0
	Secuencia	40,2891	40	-43,9237	-40	34,3614	35	79,9706	80	0,9861	0

PROCESO 2

Experimento 4 punto inicial: GD1=-15, GD2=35-75, GD3=100, GD4=90, GD5=0 a un punto GD1=40, GD2=-40, GD3=35 GD4=80, GD5=0											
31	Cada Uno	37,3082	40	-40,9764	-40	37,9611	35	83,4992	80	-3,1071	0
	Secuencia	37,2722	40	-41,0651	-40	31,7036	35	77,1729	80	-0,6109	0
32	Cada Uno	36,4477	40	-39,254	-40	32,884	35	77,3157	80	-1,1059	0
	Secuencia	36,217	40	-39,1803	-40	31,0998	35	74,0981	80	0,6651	0
33	Cada Uno	42,5131	40	-41,4991	-40	33,3691	35	75,5541	80	-2,8512	0
	Secuencia	40,1057	40	-41,4764	-40	32,3289	35	75,1787	80	-2,6915	0
34	Cada Uno	37,6785	40	-40,4418	-40	37,6875	35	74,5866	80	0,5772	0
	Secuencia	37,0864	40	-40,4335	-40	31,9913	35	74,014	80	0,618	0
35	Cada Uno	39,3321	40	-42,8502	-40	34,8126	35	75,6927	80	-2,551	0
	Secuencia	37,8071	40	-42,5718	-40	31,4729	35	81,0955	80	-2,3314	0
36	Cada Uno	40,2143	40	-38,5694	-40	34,9501	35	75,5031	80	2,9781	0
	Secuencia	40,2089	40	-42,4516	-40	35,0612	35	75,4882	80	2,8851	0
37	Cada Uno	39,5231	40	-39,3195	-40	36,8106	35	77,0065	80	-0,6109	0
	Secuencia	39,6001	40	-39,3675	-40	32,0086	35	74,8787	80	0,5617	0
38	Cada Uno	38,1925	40	-35,6898	-40	38,0801	35	74,1966	80	0,6912	0
	Secuencia	38,0553	40	-41,2335	-40	36,397	35	75,6703	80	2,7861	0
39	Cada Uno	37,1878	40	-40,2526	-40	32,1043	35	75,9318	80	2,562	0
	Secuencia	37,1002	40	-40,2202	-40	32,4332	35	75,9077	80	2,8253	0
40	Cada Uno	39,1573	40	-43,0332	-40	32,4931	35	77,0771	80	-0,7818	0
	Secuencia	37,2381	40	-42,5843	-40	32,0249	35	76,5241	80	-1,01236	0

PROCESO 2

Experimento 5 punto inicial: GD1=-60, GD2=-65, GD3=-60, GD4=-80 GD5=40 a un punto GD1=10, GD2=-30, GD3=-100 GD4=-28 GD5=4											
41	Cada Uno	11,0514	10	-29,8047	-30	-102,0348	-100	-30,1239	-28	3,1239	4
	Secuencia	10,6312	10	-30,1337	-30	-104,6381	-100	-29,9018	-28	4,5141	4
42	Cada Uno	11,1789	10	-29,2179	-30	-108,4635	-100	-29,9647	-28	3,4789	4
	Secuencia	10,7846	10	-30,4035	-30	-106,5422	-100	-30,6033	-28	4,4525	4
43	Cada Uno	10,661	10	-30,6513	-30	-108,7944	-100	-28,8459	-28	4,7201	4
	Secuencia	9,0527	10	-30,6066	-30	-107,9575	-100	-27,5142	-28	3,4222	4
44	Cada Uno	11,9128	10	-33,6219	-30	-102,1086	-100	-27,7509	-28	4,7375	4
	Secuencia	9,5254	10	-32,4553	-30	-101,3528	-100	-26,5555	-28	3,6718	4
45	Cada Uno	11,2814	10	-28,9102	-30	-109,0813	-100	-27,8101	-28	4,6682	4
	Secuencia	10,089	10	-29,1474	-30	-105,3538	-100	-26,8485	-28	3,7102	4
46	Cada Uno	9,255	10	-30,444	-30	-98,8451	-100	-28,5204	-28	4,2702	4
	Secuencia	10,351	10	-30,5056	-30	-98,3926	-100	-30,15	-28	3,1482	4
47	Cada Uno	10,9571	10	-32,229	-30	-103,389	-100	-28,7571	-28	3,17992	4
	Secuencia	9,3879	10	-29,9674	-30	-106,0804	-100	-27,7509	-28	4,14908	4
48	Cada Uno	11,9044	10	-27,5311	-30	-101,9334	-100	-27,307	-28	4,7548	4
	Secuencia	9,7127	10	-32,1531	-30	-101,9844	-100	-28,1357	-28	4,2356	4
49	Cada Uno	11,309	10	-28,122	-30	-105,9333	-100	-28,5204	-28	4,27023	4
	Secuencia	10,3799	10	-30,464	-30	-99,5946	-100	-28,6388	-28	4,4606	4
50	Cada Uno	10,8755	10	-32,2876	-30	-102,2839	-100	-28,402	-28	3,9067	4
	Secuencia	11,0177	10	-33,0352	-30	-108,5867	-100	-28,5204	-28	4,2529	4

PROCESO 3

Experimento 5 punto inicial: GD1=60, GD2=65, GD3=60, GD4=-80 GD5=-60 a un punto GD1=10, GD2=-30, GD3=-100 GD4=-28 GD5=4

51	Cada Uno	12,2231	10	-31,833	-30	-100,3716	-100	-27,0999	-28	4,599	4
	Secuencia	8,8756	10	-34,351	-30	-100,2354	-100	-28,8459	-28	3,57797	4
52	Cada Uno	11,6511	10	-31,2885	-30	-104,7826	-100	-28,2244	-28	3,8029	4
	Secuencia	9,291	10	-31,2405	-30	-107,3899	-100	-28,7867	-28	3,83756	4
53	Cada Uno	11,6012	10	-30,9752	-30	-100,5765	-100	-28,7867	-28	4,58174	4
	Secuencia	11,5004	10	-32,2271	-30	-100,6384	-100	-28,1948	-28	4,14908	4
54	Cada Uno	12,2266	10	-30,5672	-30	-100,6406	-100	-27,307	-28	3,4221	4
	Secuencia	8,7955	10	-34,4481	-30	-106,7133	-100	-27,7509	-28	4,52982	4
55	Cada Uno	11,8064	10	-28,426	-30	-99,6101	-100	-28,9939	-28	3,97602	4
	Secuencia	8,187	10	-33,0672	-30	-107,6052	-100	-28,1948	-28	3,35298	4
56	Cada Uno	12,3114	10	-29,2259	-30	-101,5842	-100	-28,5796	-28	3,28376	4
	Secuencia	8,8682	10	-32,8029	-30	-107,8245	-100	-27,8693	-28	3,4049	4
57	Cada Uno	11,0423	10	-28,8808	-30	-101,9121	-100	-27,8101	-28	3,5779	4
	Secuencia	8,8243	10	-32,6845	-30	-101,9474	-100	-27,5438	-28	3,62989	4
58	Cada Uno	9,4362	10	-30,3658	-30	-99,3515	-100	-27,8693	-28	4,23561	4
	Secuencia	9,2382	10	-30,6058	-30	-107,056	-100	-27,0703	-28	3,128	4
59	Cada Uno	11,2335	10	-31,0285	-30	-99,7879	-100	-28,88459	-28	3,17992	4
	Secuencia	10,9949	10	-33,0872	-30	-100,1604	-100	-28,8459	-28	4,84134	4
60	Cada Uno	11,2626	10	-32,1716	-30	-101,1805	-100	-27,4846	-28	4,4606	4
	Secuencia	8,6427	10	-32,0853	-30	-101,0396	-100	-27,0111	-28	4,18369	4

PROCESO 3

Experimento 7 punto inicial: GD1=95, GD2=60, GD3=120, GD4=55, GD5=65 a un punto GD1=50, GD2=30, GD3=80, GD4=-100, GD5=-40

61	Cada Uno	51,3645	50	31,2234	30	81,4283	80	-99,1022	-100	-40,4324	-40
	Secuencia	49,4315	50	31,2777	30	76,9943	80	-99,3094	-100	-39,6536	-40
62	Cada Uno	50,333	50	29,0263	30	79,3775	80	-99,339	-100	-39,2382	-40
	Secuencia	50,1962	50	30,8928	30	79,6162	80	-100,967	-100	-40,7612	-40
63	Cada Uno	50,2526	50	31,3601	30	81,7345	80	-100,996	-100	-40,9343	-40
	Secuencia	49,9673	50	28,3058	30	81,609	80	-100,168	-100	-40,9516	-40
64	Cada Uno	51,7806	50	29,6149	30	77,796	80	-99,1416	-100	-40,5189	-40
	Secuencia	48,3356	50	28,8863	30	77,3349	80	-99,4869	-100	-40,692	-40
65	Cada Uno	50,4427	50	30,6362	30	80,5876	80	-99,1022	-100	-40,1555	-40
	Secuencia	50,4433	50	28,9277	30	80,9414	80	-100,641	-100	-39,3075	-40
66	Cada Uno	50,9688	50	31,3952	30	77,2256	80	-100,671	-100	-39,6709	-40
	Secuencia	50,8307	50	28,1205	30	81,5215	80	-99,99	-100	-40,6574	-40
67	Cada Uno	49,7483	50	30,3061	30	84,3061	80	-100,523	-100	-39,7055	-40
	Secuencia	49,643	50	30,9154	30	79,6517	80	-99,6349	-100	-40,9516	-40
68	Cada Uno	51,2676	50	30,1417	30	79,7489	80	-99,6349	-100	-40,6228	-40
	Secuencia	48,671	50	30,1809	30	79,6986	80	-100,434	-100	-40,5708	-40
69	Cada Uno	51,3425	50	31,1087	30	82,5848	80	-100,197	-100	-39,1517	-40
	Secuencia	48,9831	50	30,9883	30	77,1785	80	-99,5165	-100	-40,8824	-40
70	Cada Uno	50,7519	50	31,057	30	82,2015	80	-99,5461	-100	-39,3594	-40
	Secuencia	48,3012	50	31,0177	30	78,3764	80	-100,049	-100	-40,0343	-40

PROCESO 4

Experimento 8 punto inicial: GD1=0, GD2=-80, GD3=-20, GD4=10, GD5=-20 a un punto GD1=50, GD2=30, GD3=80, GD4=-100, GD5=-40

71	Cada Uno	46,887	50	28,1909	30	83,066	80	-100,61495	-100	-40,3459	-40
	Secuencia	46,675	50	27,2202	30	79,5556	80	-100,848	-100	-40,2766	-40
72	Cada Uno	47,6745	50	30,2496	30	87,4239	80	-100,108	-100	-39,64844	-40
	Secuencia	47,4374	50	30,2688	30	78,3024	80	-100,848	-100	-39,4286	-40
73	Cada Uno	50,7377	50	31,6789	30	78,9563	80	-100,641	-100	-40,2939	-40
	Secuencia	48,0673	50	28,7955	30	78,802	80	-99,1318	-100	-39,7055	-40
74	Cada Uno	51,5963	50	31,8069	30	77,9628	80	-99,3094	-100	-39,2902	-40
	Secuencia	48,8974	50	28,4432	30	80,4272	80	-99,9013	-100	-39,6363	-40
75	Cada Uno	48,2177	50	26,8649	30	82,1587	80	-99,0134	-100	-40,3805	-40
	Secuencia	47,6107	50	27,2145	30	78,986	80	-99,9309	-100	-40,3805	-40
76	Cada Uno	50,6285	50	33,0329	30	77,8541	80	-99,0726	-100	-40,7266	-40
	Secuencia	50,5924	50	27,9207	30	82,2873	80	-99,8421	-100	-40,1036	-40
77	Cada Uno	51,6701	50	31,8781	30	81,4362	80	-100,108	-100	-39,4632	-40
	Secuencia	47,9863	50	28,4967	30	77,2225	80	-99,1614	-100	-40,2247	-40
78	Cada Uno	50,62	50	28,8767	30	77,0978	80	-100,878	-100	-40,4151	-40
	Secuencia	50,625	50	27,4221	30	80,0237	80	-100,789	-100	-39,3767	-40
79	Cada Uno	50,6811	50	30,028	30	79,1842	80	-100,848	-100	-40,7266	-40
	Secuencia	50,6859	50	30,0684	30	79,1933	80	-99,7237	-100	-40,3459	-40
80	Cada Uno	50,2582	50	30,617	30	73,9905	80	-99,4869	-100	-39,1863	-40
	Secuencia	50,1936	50	28,0181	30	74,9458	80	-99,4278	-100	-40,3112	-40

PROCESO 4

Experimento 9 punto inicial: GD1=20, GD2=100, GD3=-80, GD4=-60, GD5=70 a un punto GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30

81	Cada Uno	-39,5299	-40	79,5348	80	58,3499	60	9,53755	10	29,105	30
	Secuencia	-39,5433	-40	79,5466	80	56,231	60	10,455	10	30,32465	30
82	Cada Uno	-37,5781	-40	80,2806	80	66,4942	60	9,03446	10	29,0531	30
	Secuencia	-42,6768	-40	80,2348	80	58,0776	60	9,55674	10	29,9184	30
83	Cada Uno	-43,3548	-40	78,6833	80	65,8315	60	9,62634	10	29,6415	30
	Secuencia	-41,3173	-40	79,7077	80	56,2456	60	10,5733	10	29,1396	30
84	Cada Uno	-41,7411	-40	82,9738	80	62,977	60	9,86309	10	30,0049	30
	Secuencia	-44,2022	-40	79,5132	80	56,5005	60	9,09364	10	30,08875	30
85	Cada Uno	-41,4688	-40	79,8052	80	67,4725	60	10,7509	10	30,3164	30
	Secuencia	-40,3128	-40	79,9152	80	56,6915	60	9,06405	10	29,5203	30
86	Cada Uno	-42,1559	-40	78,7066	80	60,5786	60	9,74471	10	28,8145	30
	Secuencia	-43,1086	-40	78,6823	80	60,5826	60	9,56715	10	30,59333	30
87	Cada Uno	-44,3587	-40	76,8797	80	59,1001	60	10,5142	10	29,8665	30
	Secuencia	-41,4058	-40	80,4973	80	60,932	60	10,8397	10	30,2991	30
88	Cada Uno	-37,9927	-40	80,9345	80	65,8129	60	9,77431	10	29,2261	30
	Secuencia	-41,1712	-40	80,8671	80	58,7838	60	10,8693	10	29,2434	30
89	Cada Uno	-37,9752	-40	79,4258	80	60,9815	60	9,12324	10	29,9876	30
	Secuencia	-41,9081	-40	79,3338	80	60,9042	60	9,24161	10	30,8702	30
90	Cada Uno	-38,1541	-40	79,0992	80	59,7921	60	10,4254	10	29,1396	30
	Secuencia	-39,1156	-40	80,0248	80	60,9977	60	10,9581	10	29,7799	30

PROCESO 5

Experimento 10 punto inicial: GD1=-75, GD2=0, GD3=90, GD4=120, GD5=-25 a un punto GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30											
91	Cada Uno	-43,1455	-40	77,5089	80	56,7902	60	9,44877	10	29,34688	30
	Secuencia	-41,3667	-40	80,1042	80	59,1765	60	9,33039	10	30,351	30
92	Cada Uno	-42,4839	-40	83,2536	80	59,3538	60	9,06405	10	30,135	30
	Secuencia	-39,7595	-40	79,1819	80	57,9713	60	10,0111	10	30,8273	30
93	Cada Uno	-39,5176	-40	81,3794	80	59,1605	60	9,59674	10	29,9357	30
	Secuencia	-40,3983	-40	80,4408	80	59,4076	60	9,15283	10	29,1396	30
94	Cada Uno	-42,0286	-40	79,3352	80	59,6582	60	9,09364	10	30,1607	30
	Secuencia	-39,7528	-40	80,1477	80	57,2841	60	9,47837	10	30,4203	30
95	Cada Uno	-38,255	-40	82,2587	80	59,4545	60	9,47837	10	30,7644	30
	Secuencia	-42,8339	-40	78,1565	80	56,9672	60	10,4846	10	30,5241	30
96	Cada Uno	-39,6203	-40	77,6124	80	63,766	60	9,86309	10	29,4165	30
	Secuencia	-40,7435	-40	79,2497	80	59,5739	60	10,0702	10	30,8356	30
97	Cada Uno	-38,7248	-40	77,7714	80	54,1685	60	9,3008	10	29,2607	30
	Secuencia	-39,6344	-40	79,6134	80	60,9187	60	10,3366	10	29,9537	30
98	Cada Uno	-39,1918	-40	79,1335	80	62,1449	60	10,8693	10	30,5587	30
	Secuencia	-41,6211	-40	79,208	80	57,4062	60	10,0702	10	29,6588	30
99	Cada Uno	-37,4107	-40	81,4106	80	60,8233	60	10,3662	10	29,7972	30
	Secuencia	-42,6379	-40	79,3558	80	58,0605	60	9,77431	10	30,9568	30
100	Cada Uno	-42,9515	-40	82,335	80	60,8372	60	9,3008	10	29,6242	30
	Secuencia	-41,3126	-40	77,6096	80	58,5783	60	10,1886	10	29,5549	30

PROCESO 5

ANEXO B

ANÁLISIS

ESTADÍSTICO DE

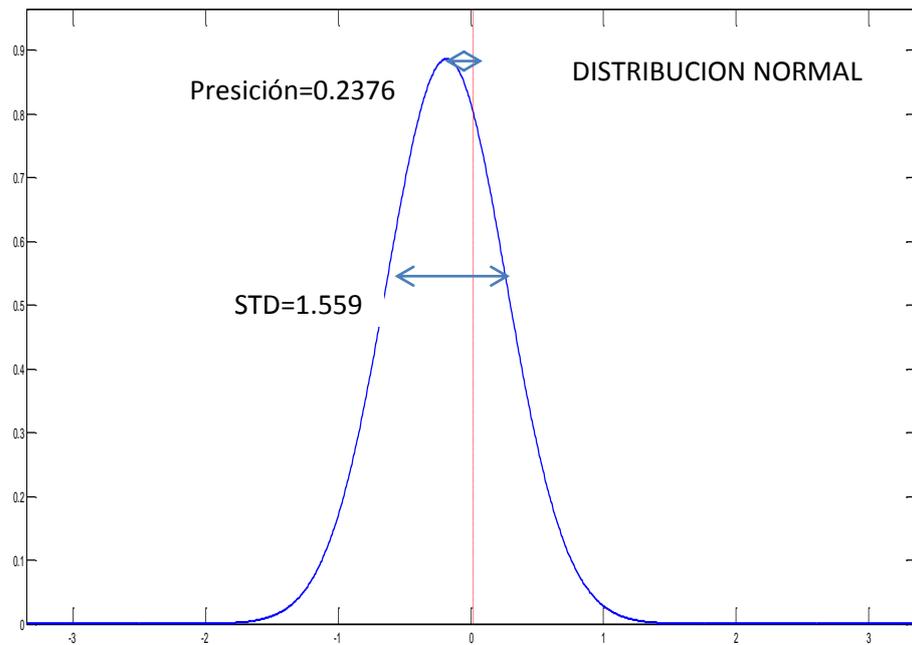
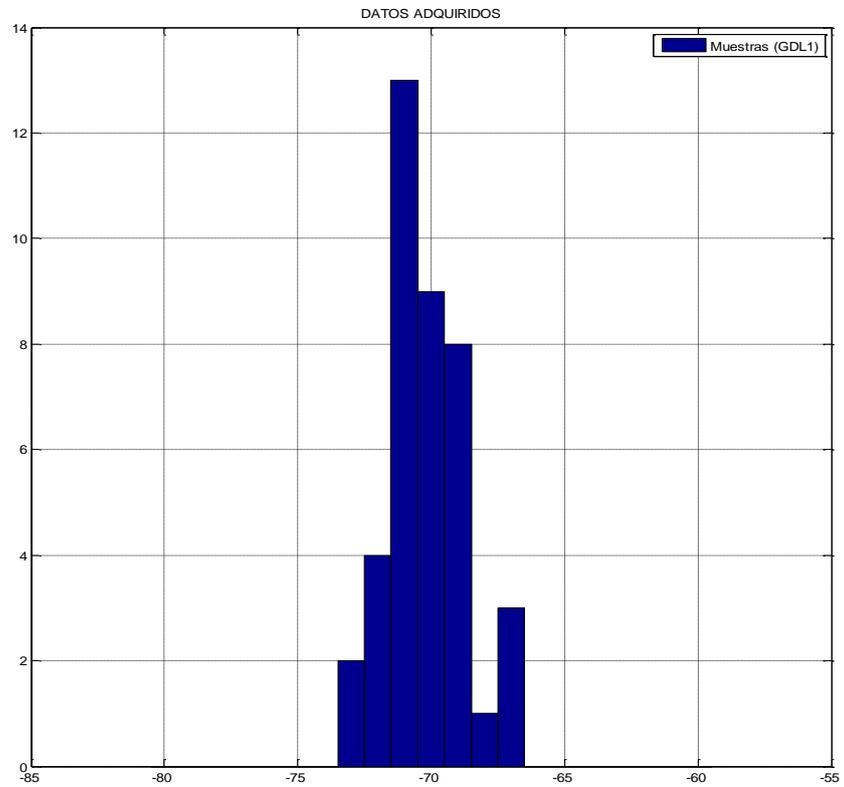
DATOS ADQUIRIDOS

Proceso 1 GRADO-1

Punto final GD1=-70, GD2=-90, GD3=-45, GD4=0, GD5=0.

Media	-70.2376	Exactitud.	0.2376
Desviación Estándar	1.5591	Precisión.	1.5591

GDL1

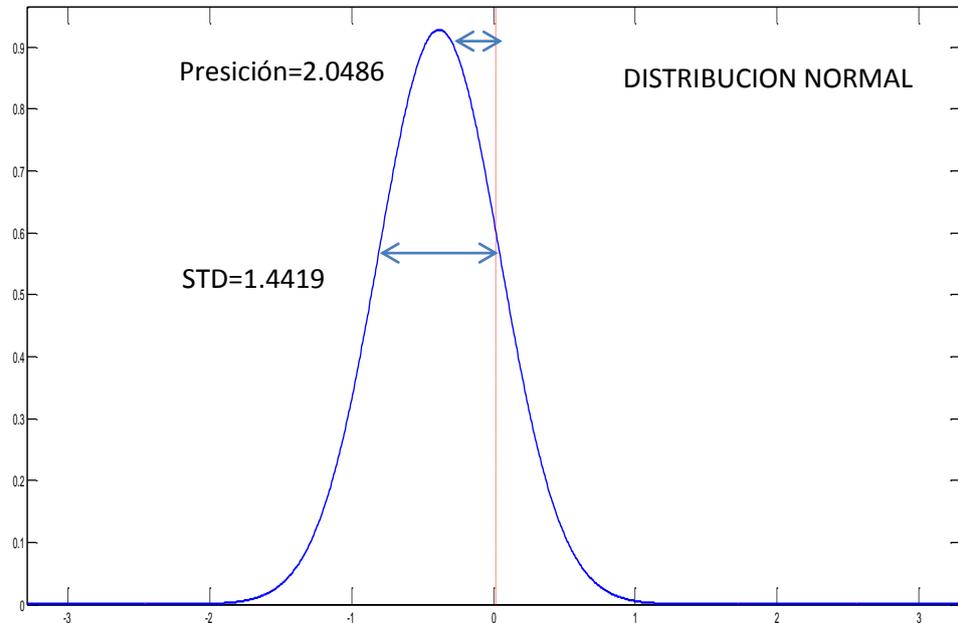
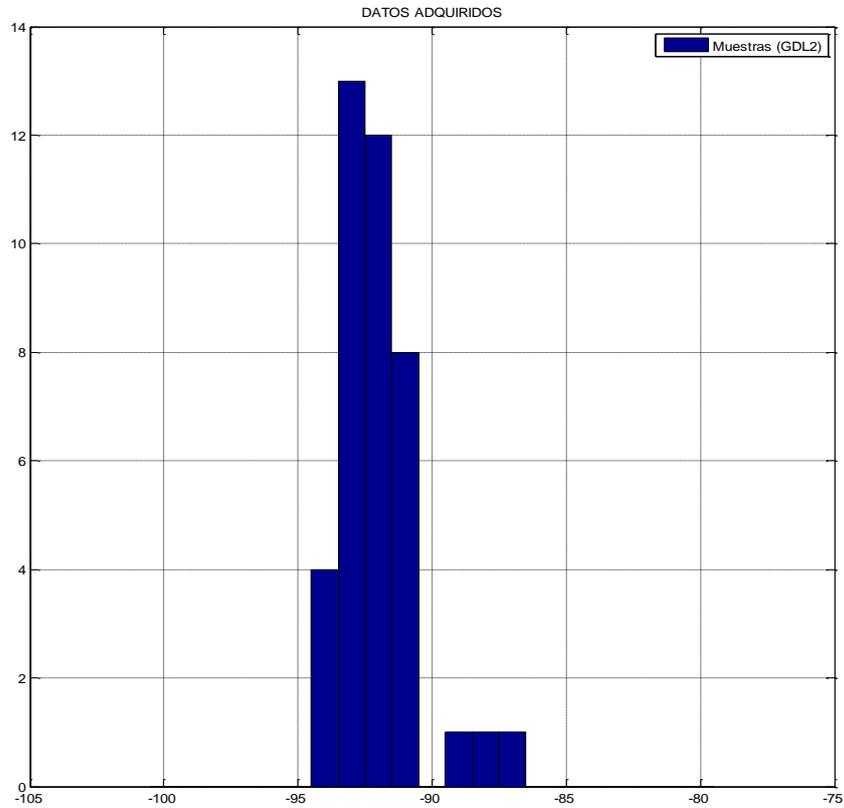


Proceso 1 GRADO-2

Punto final GD1=-70, GD2=-90, GD3=-45, GD4=0, GD5=0.

Media	-92.0486	Exactitud.	2.0486
Desviación Estándar	1.4419	Precisión.	1.4419

GDL2

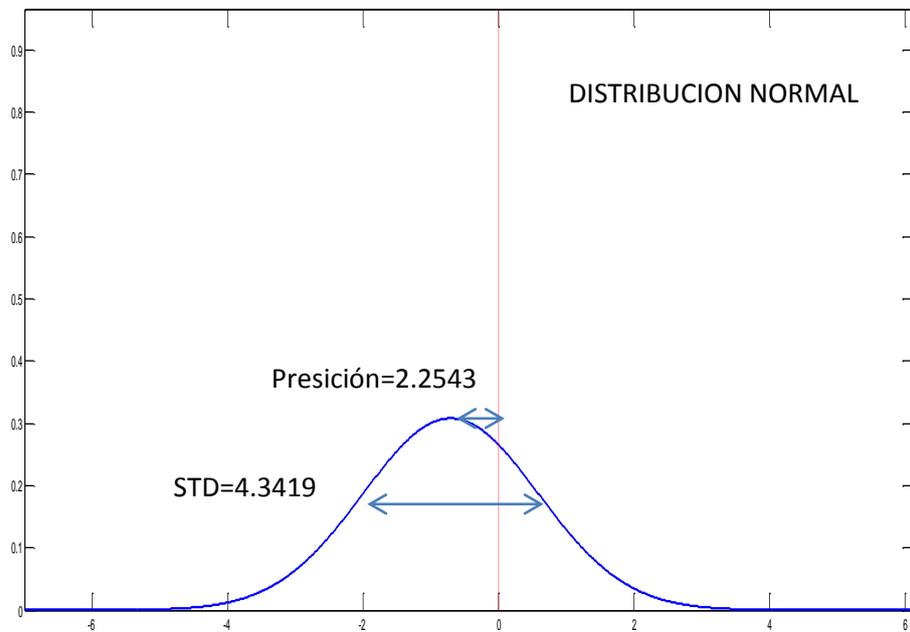
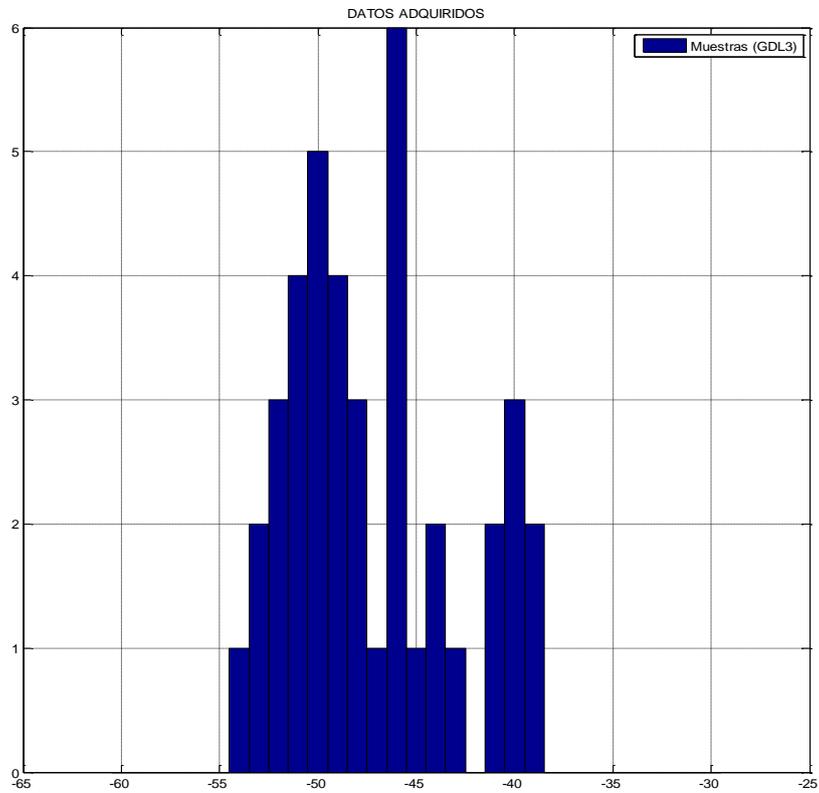


Proceso 1 GRADO-3

Punto final GD1=-70, GD2=-90, GD3=-45, GD4=0, GD5=0.

Media	-47.2543	Exactitud.	2.2543
Desviación Estándar	4.3419	Precisión.	4.3419

GDL3

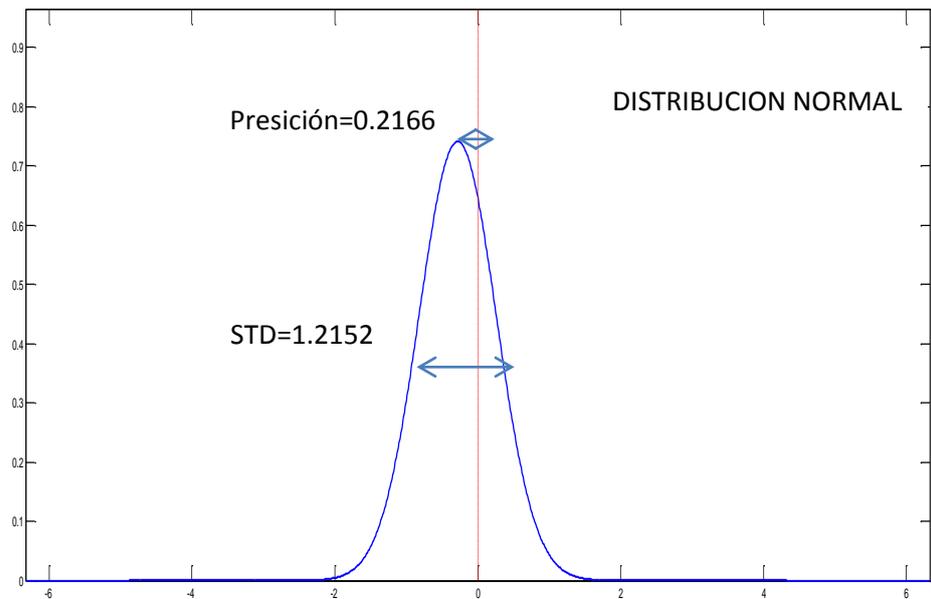
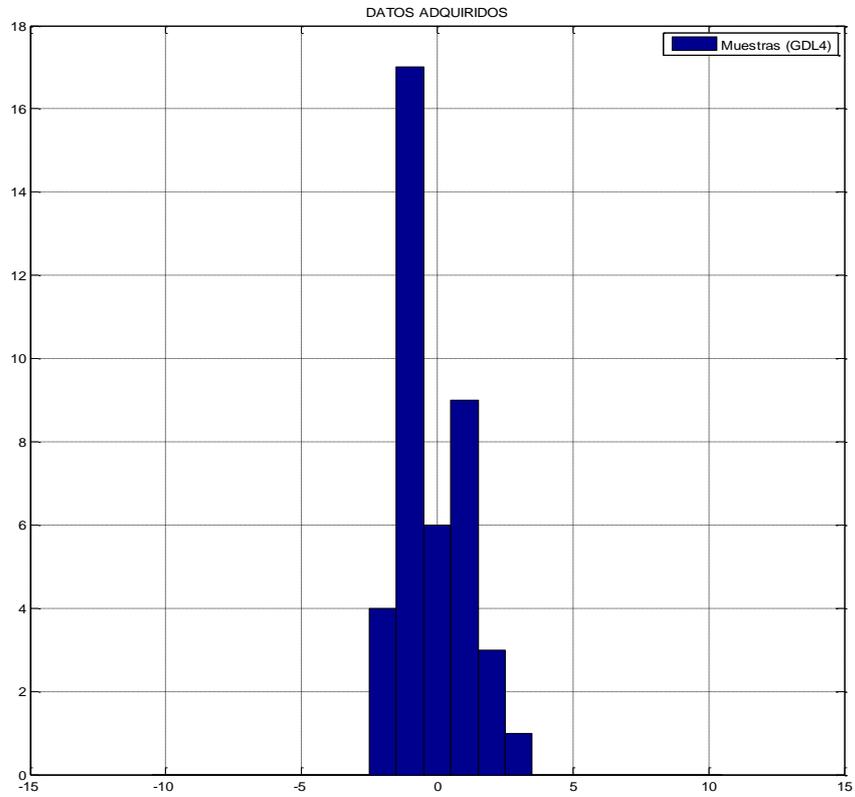


Proceso 1 GRADO-4

Punto final GD1=-70, GD2=-90, GD3=-45, GD4=0, GD5=0.

Media	-0.2166	Exactitud.	0.2166
Desviación Estándar	1.2152	Precisión.	1.2152

GDL4

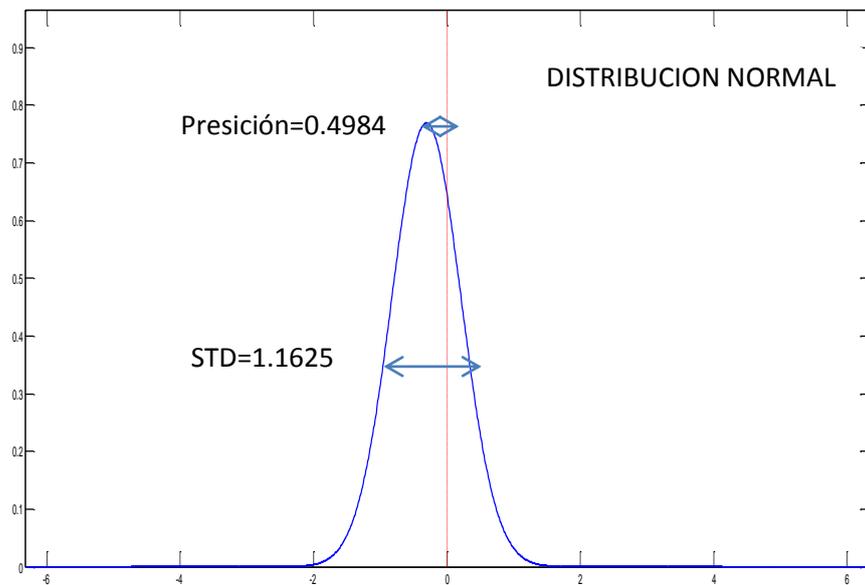
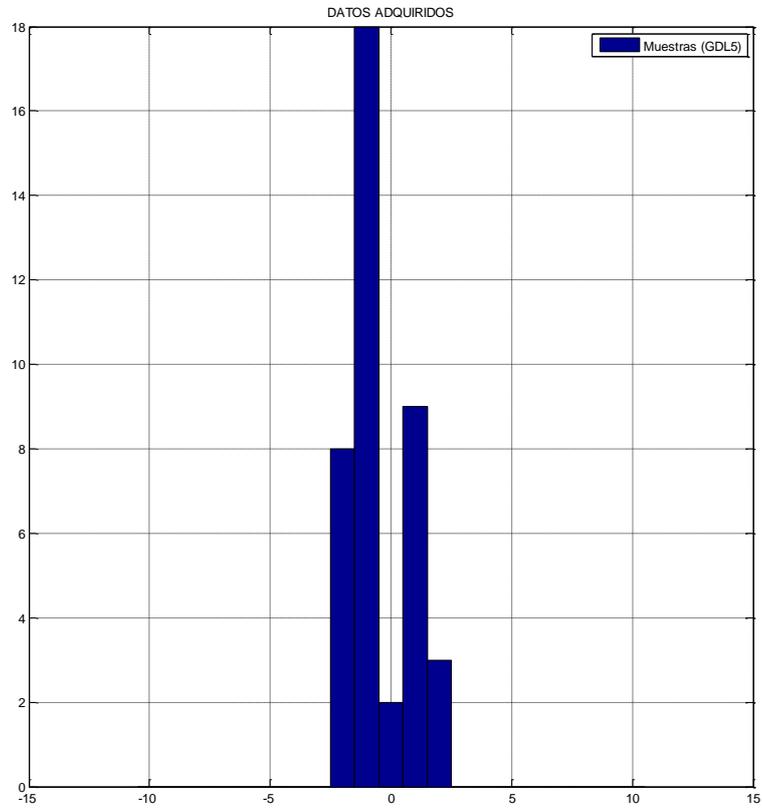


Proceso 1 GRADO-5

Punto final GD1=-70, GD2=-90, GD3=-45, GD4=0, GD5=0.

Media	-0.4989	Exactitud.	0.4989
Desviación Estándar	1.1625	Precisión.	1.1625

GDL5

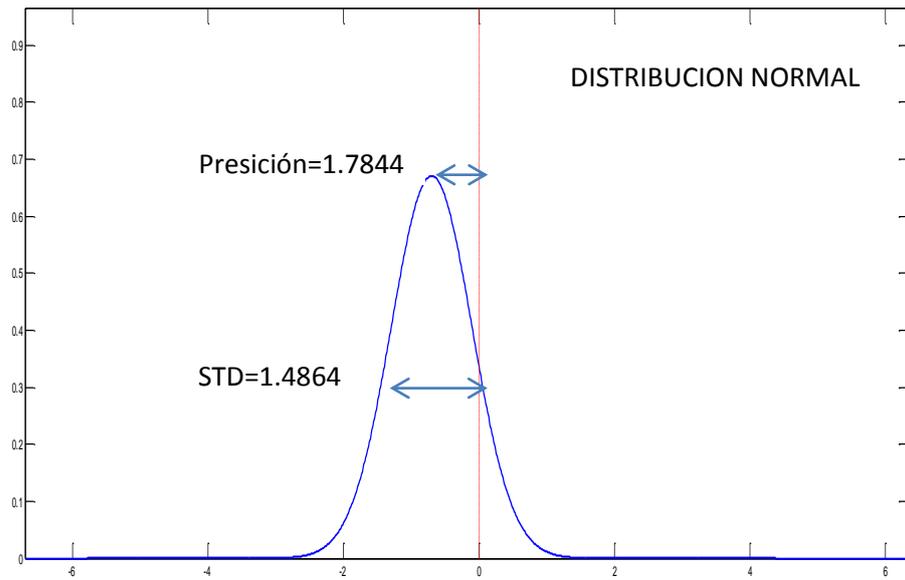
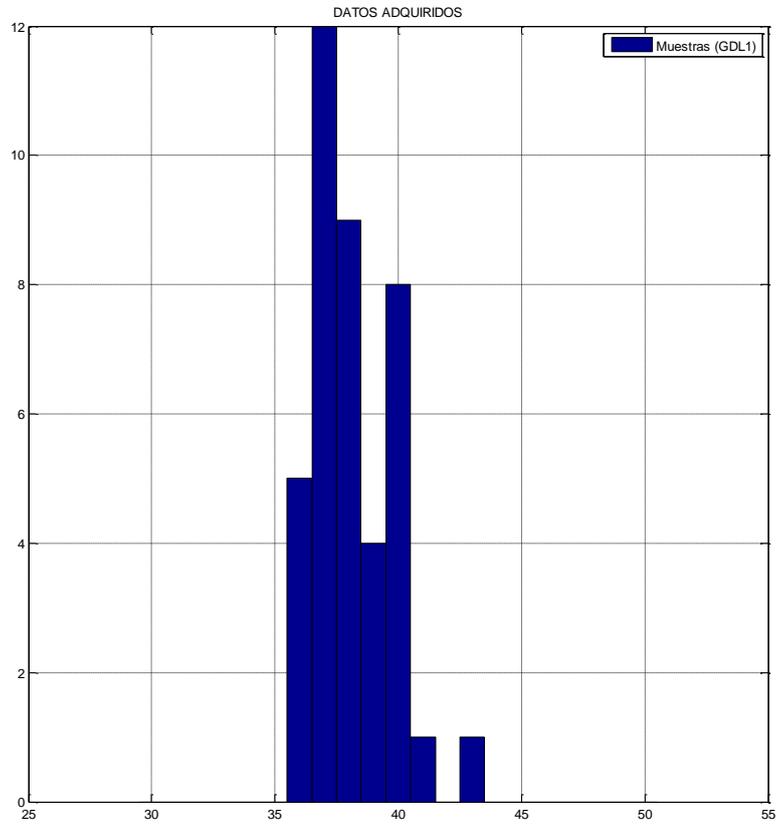


Proceso 2 GRADO-1

Punto Final GD1=40, GD2=-40, GD3=35 GD4=80, GD5=0

Media	38.2156	Exactitud.	1.7844
Desviación Estándar	1.4864	Precisión.	1.4864

GDL1

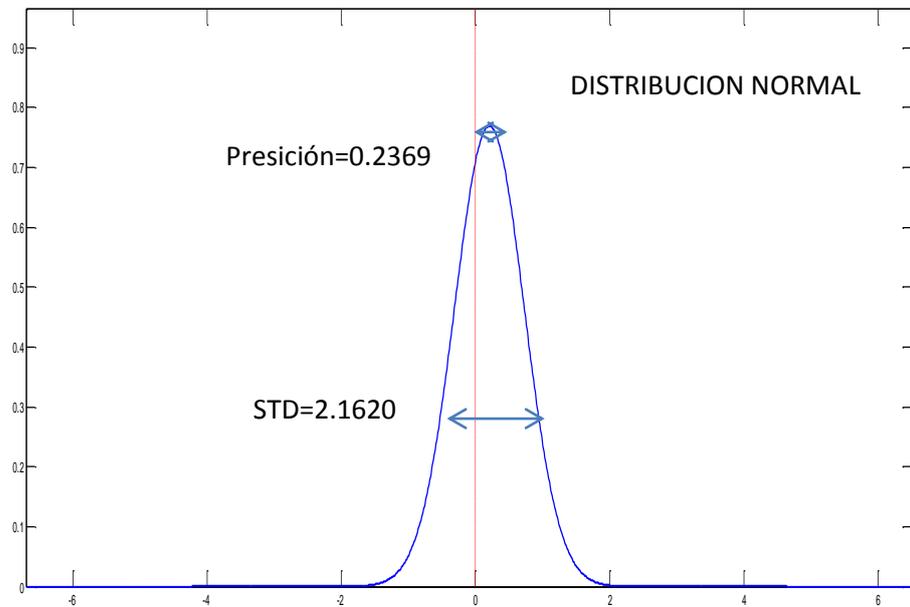
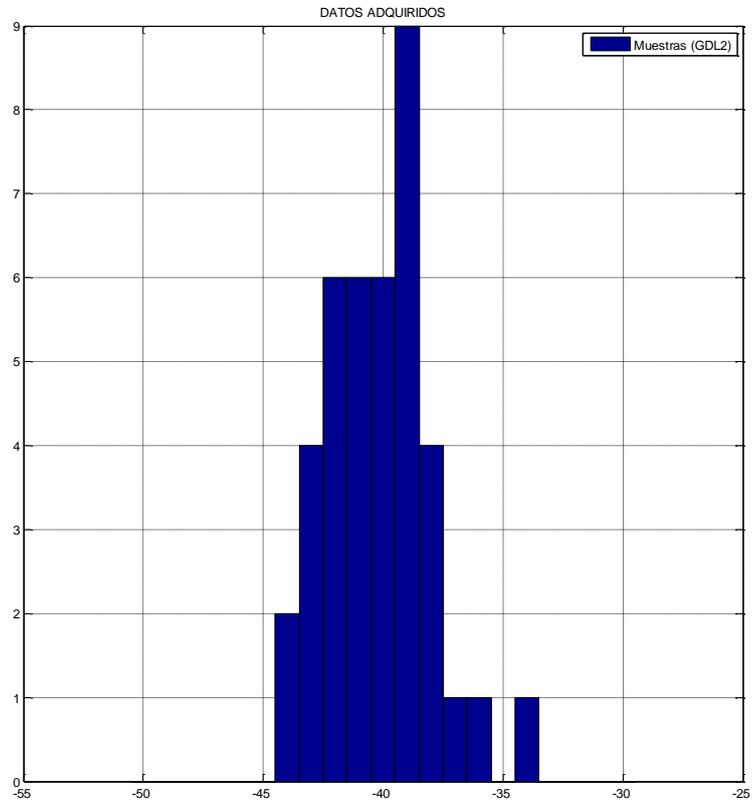


Proceso 2 GRADO-2

Punto Final GD1=40, GD2=-40, GD3=35 GD4=80, GD5=0

Media	-40.2369	Exactitud.	0.2369
Desviación Estándar	2.1620	Precisión.	2.1620

GDL2

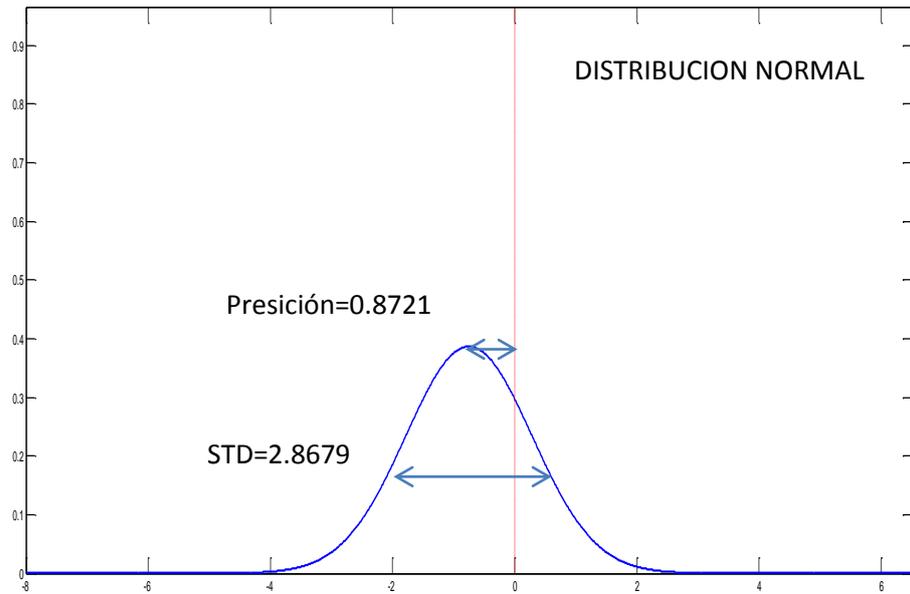
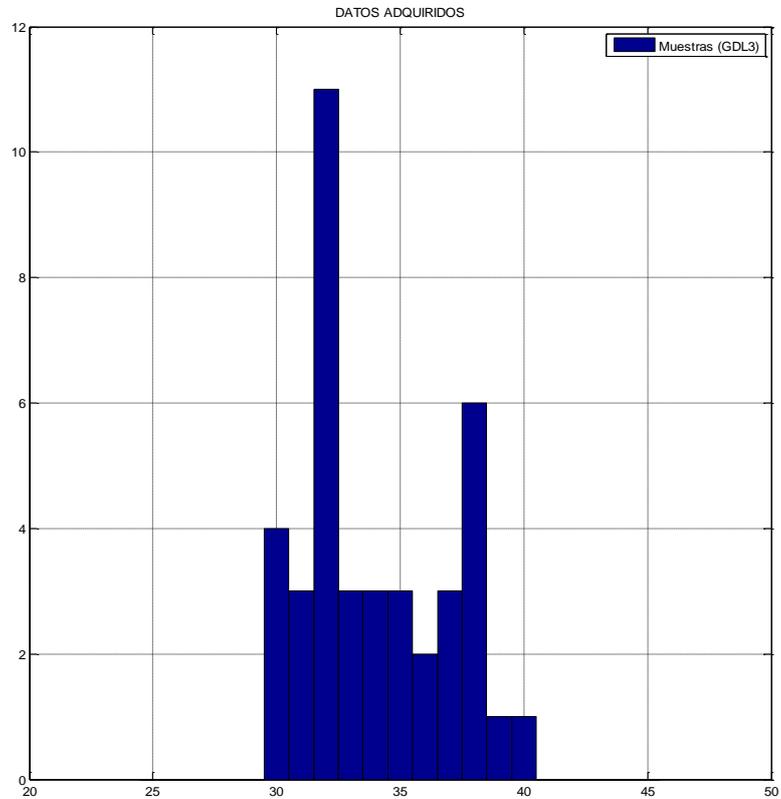


Proceso 2 GRADO-3

Punto Final GD1=40, GD2=-40, GD3=35 GD4=80, GD5=0

Media	34.1279	Exactitud.	0.8721
Desviación Estándar	2.8679	Precisión.	2.8679

GDL3

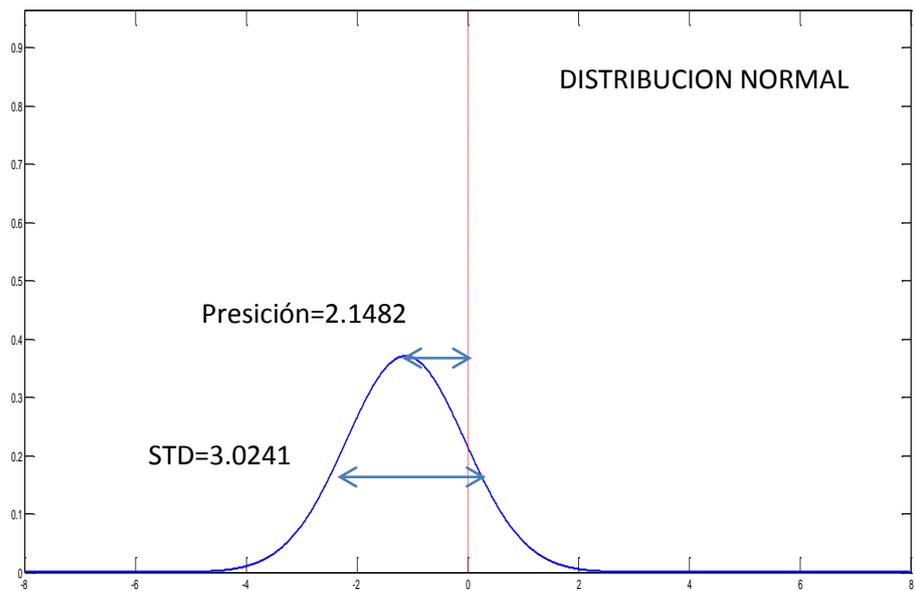
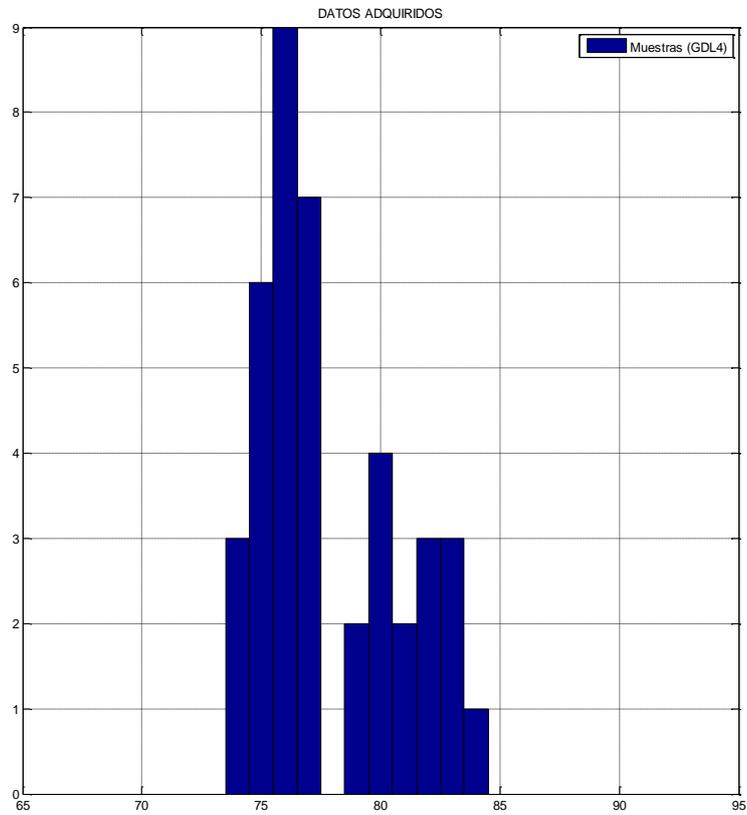


Proceso 2 GRADO-4

Punto Final GD1=40, GD2=-40, GD3=35 GD4=80, GD5=0

Media	77.8518	Exactitud.	2.1482
Desviación Estándar	3.0241	Precisión.	3.0241

GDL4

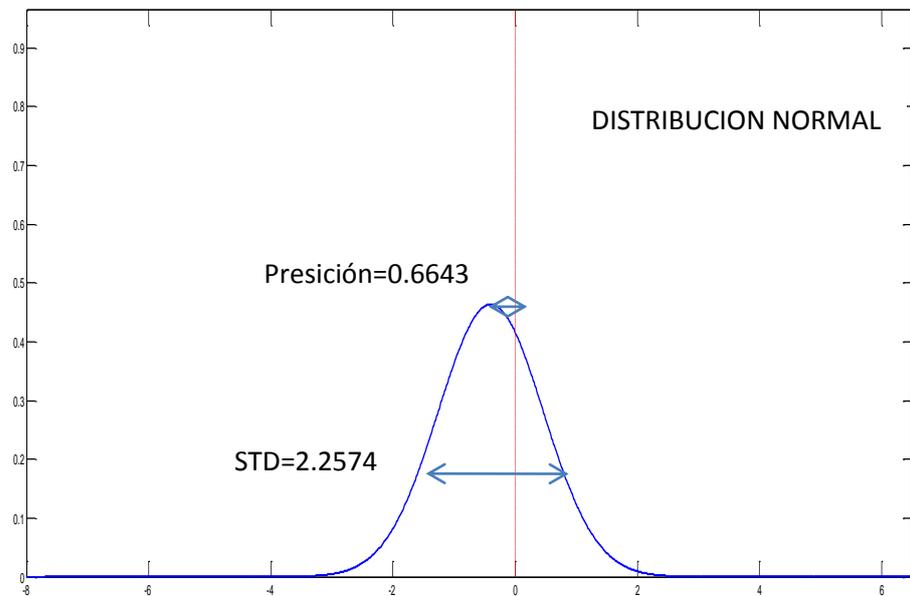
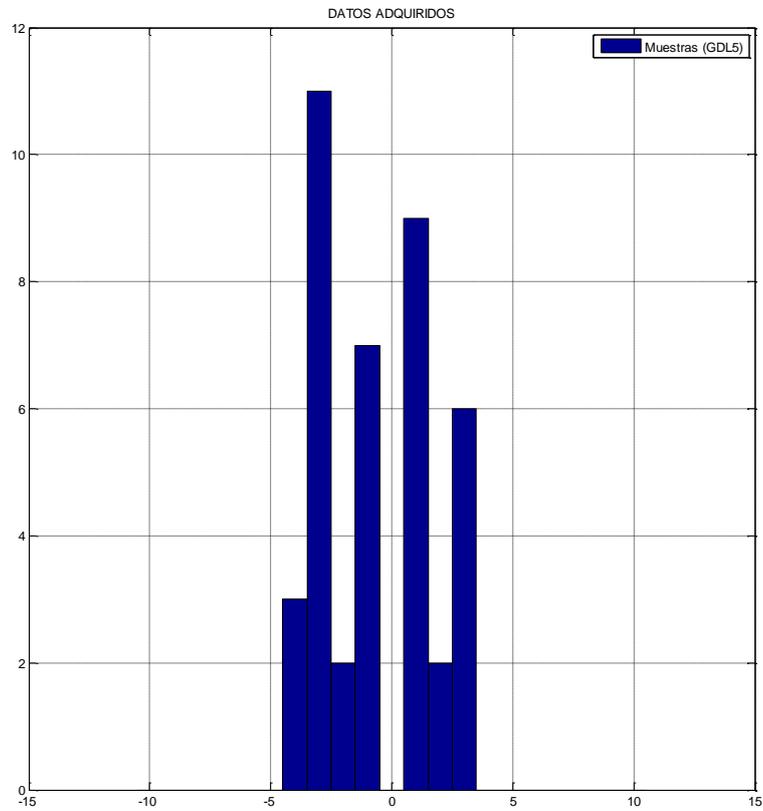


Proceso 2 GRADO-5

Punto Final GD1=40, GD2=-40, GD3=35 GD4=80, GD5=0

Media	-0.6643	Exactitud.	0.6643
Desviación Estándar	2.2574	Precisión.	2.2574

GDL5

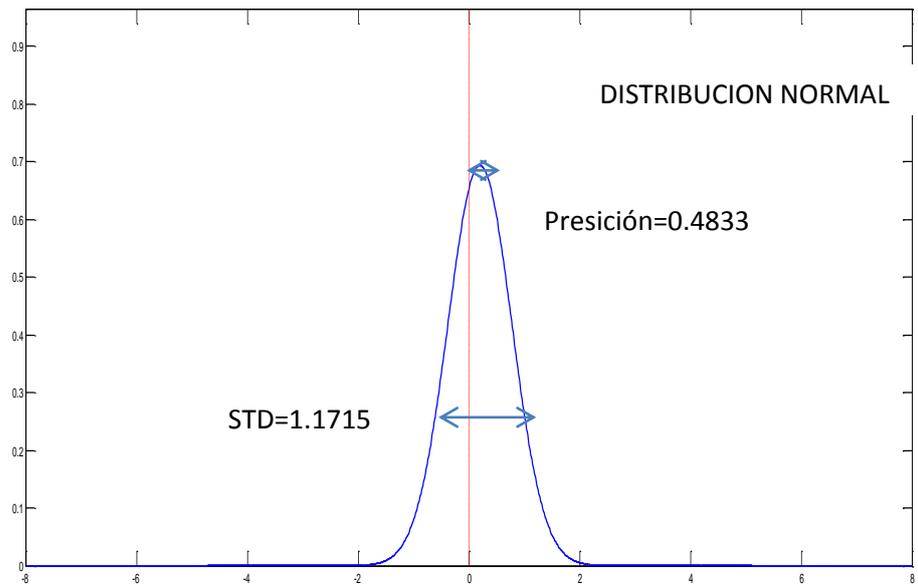
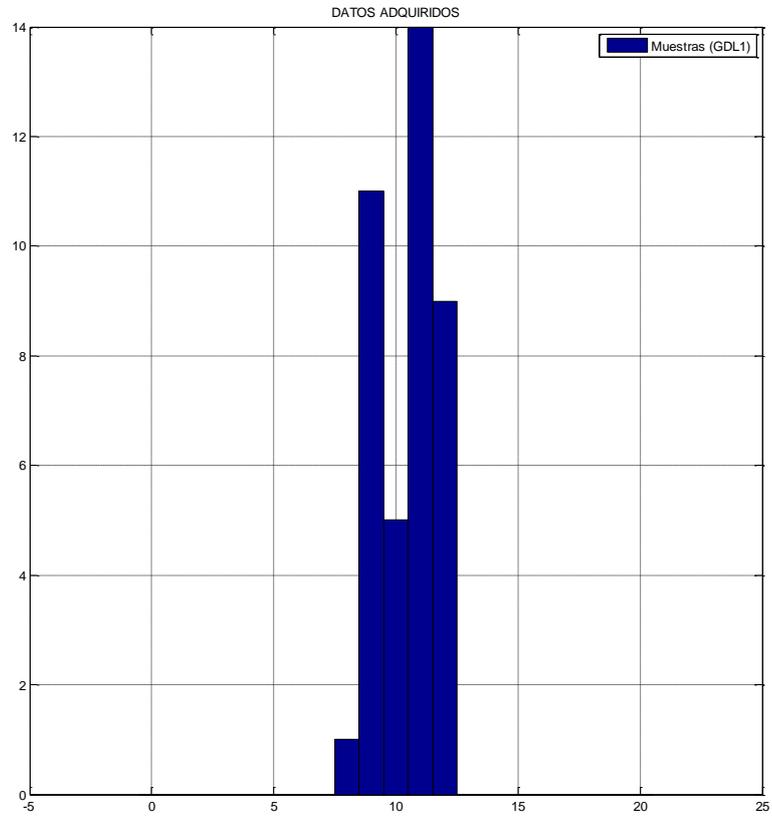


Proceso 3 GRADO-1

Punto Final GD1=10, GD2=-30, GD3=-100 GD4=-28 GD5=4

Media	10.4833	Exactitud.	0.4833
Desviación Estándar	1.1715	Precisión.	1.1715

GDL1

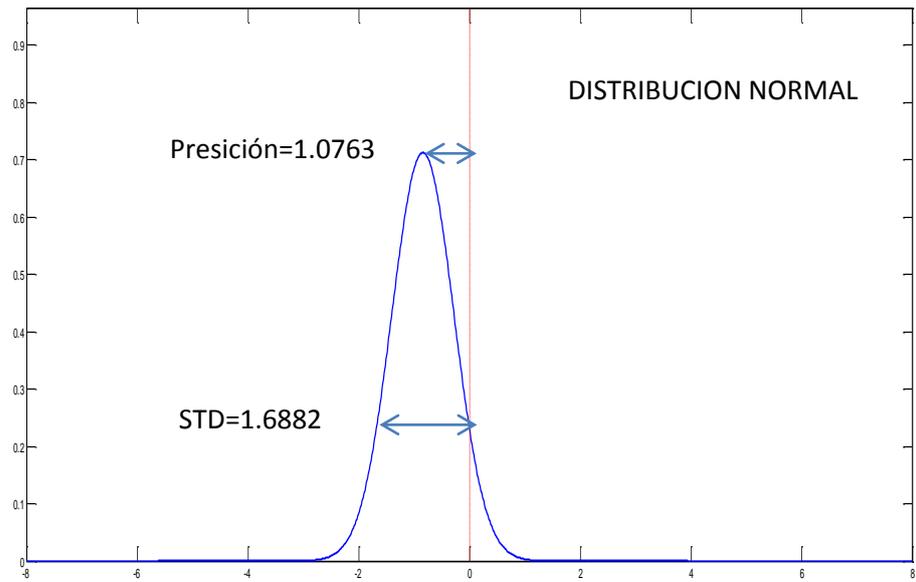
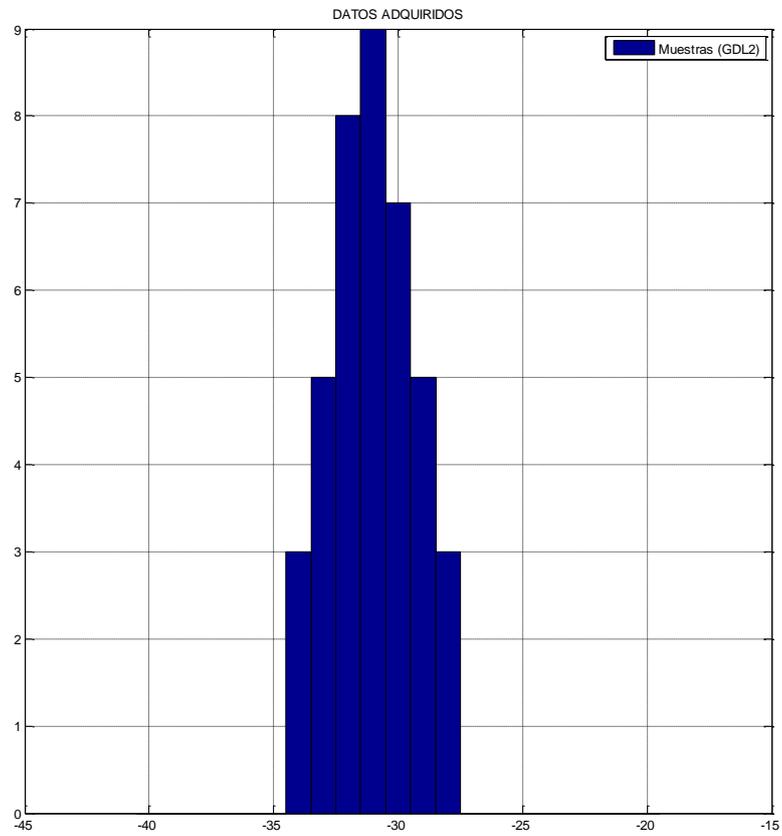


Proceso 3 GRADO-2

Punto Final GD1=10, GD2=-30, GD3=-100 GD4=-28 GD5=4

Media	-31.0763	Exactitud.	1.0763
Desviación Estándar	1.6882	Precisión.	1.6882

GDL2

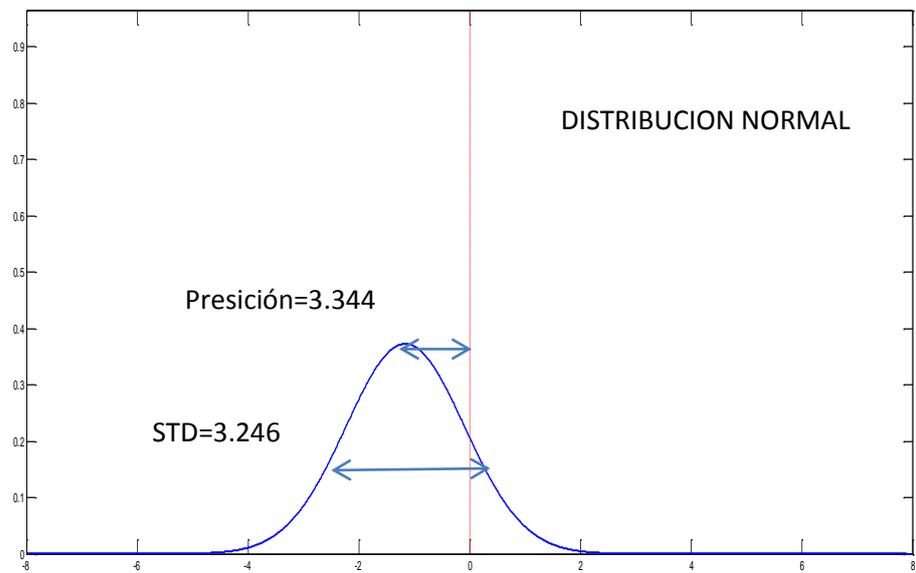
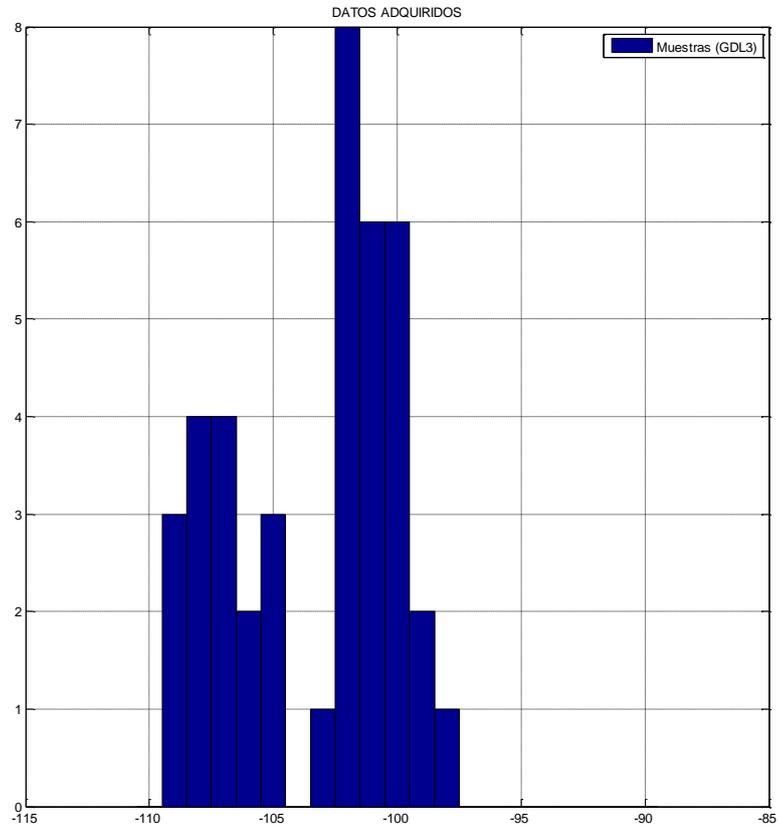


Proceso 3 GRADO-3

Punto Final GD1=10, GD2=-30, GD3=-100 GD4=-28 GD5=4

Media	-103.3440	Exactitud.	3.3440
Desviación Estándar	3.246	Precisión.	3.246

GDL3

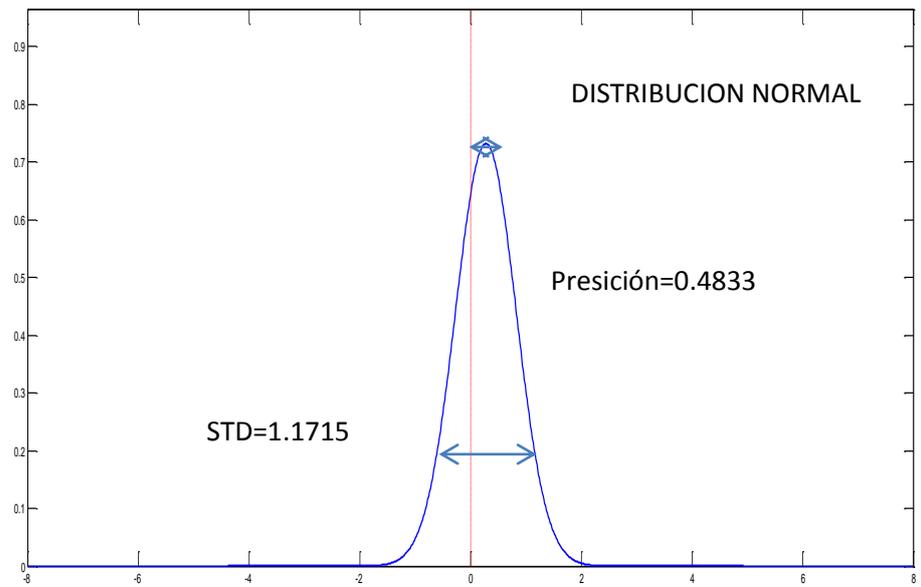
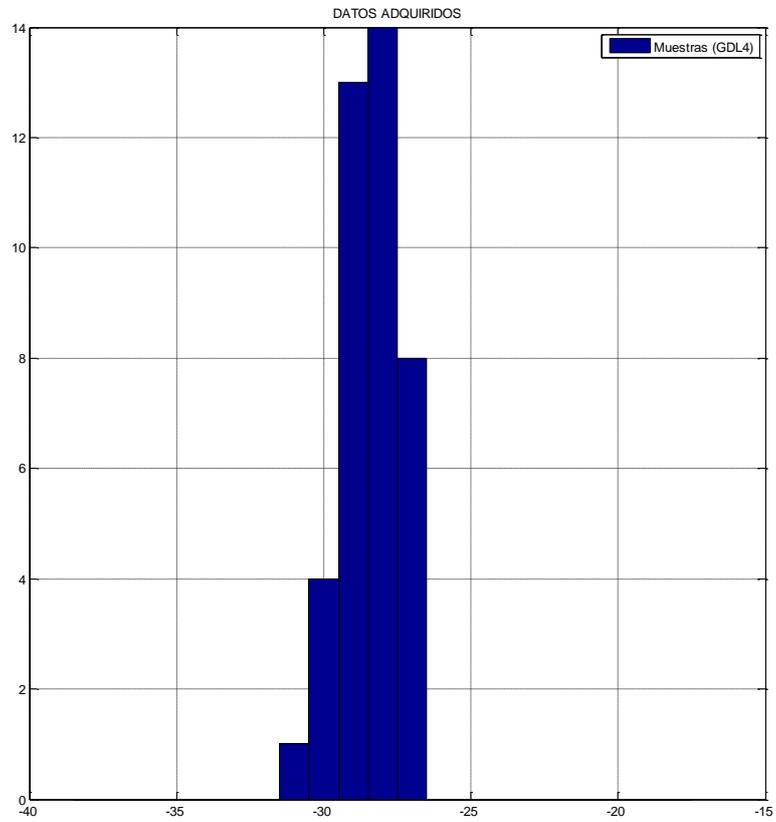


Proceso 3 GRADO-4

Punto Final GD1=10, GD2=-30, GD3=-100 GD4=-28 GD5=4

Media	-28.2944	Exactitud.	0.2944
Desviación Estándar	0.9546	Precisión.	0.9546

GDL4

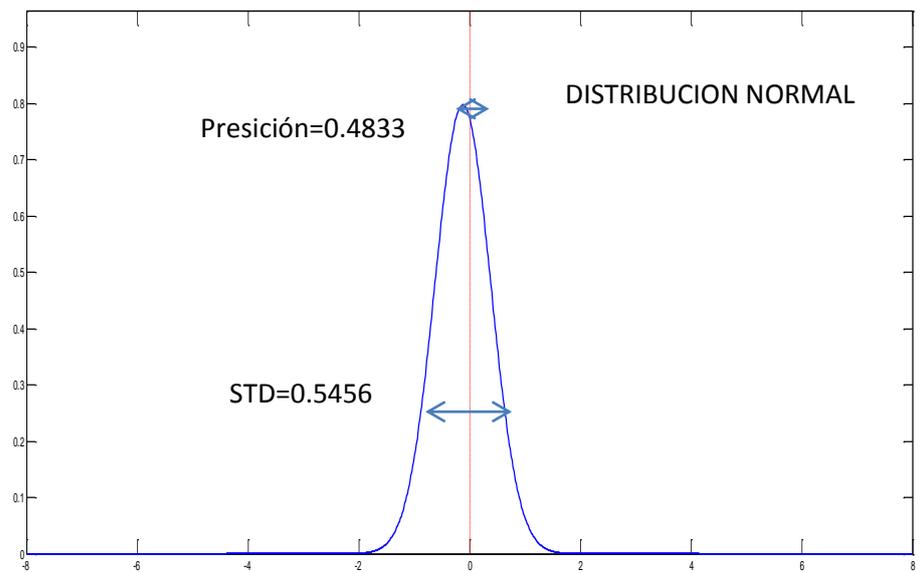
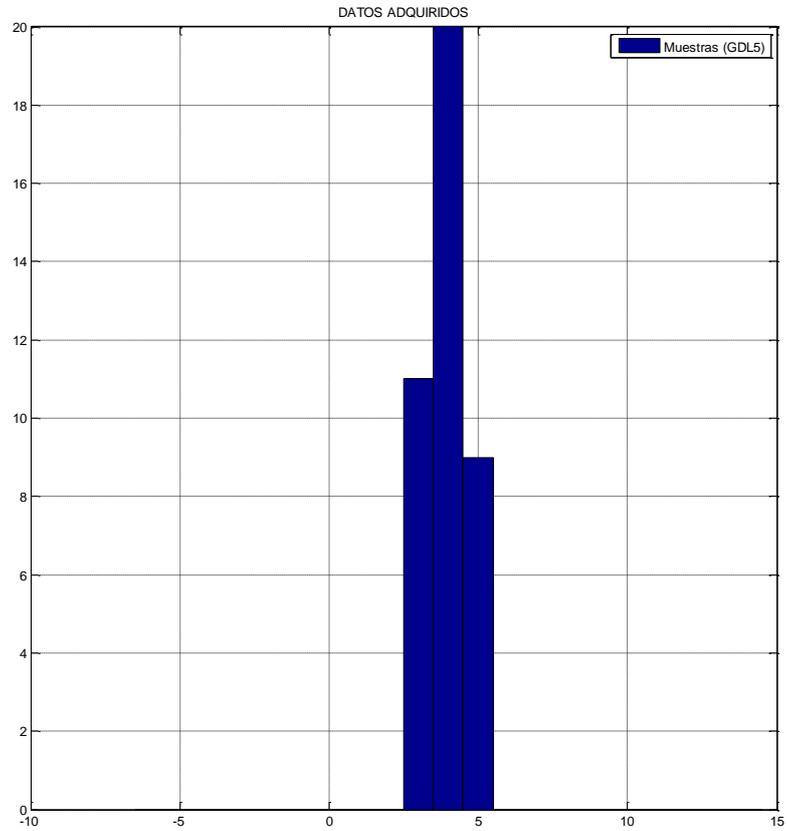


Proceso 3 GRADO-5

Punto Final GD1=10, GD2=-30, GD3=-100 GD4=-28 GD5=4

Media	3.9721	Exactitud.	0.0279
Desviación Estándar	0.5456	Precisión.	0.5456

GDL5

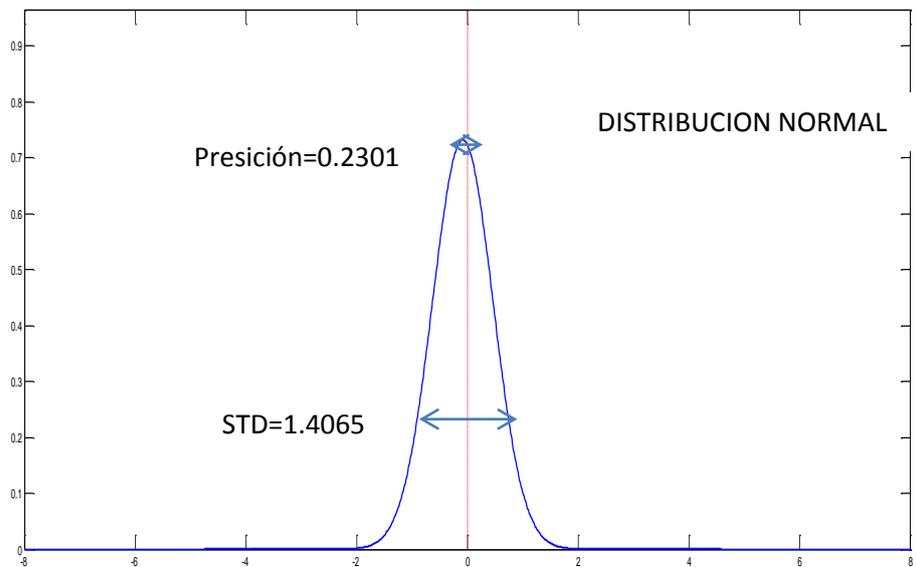
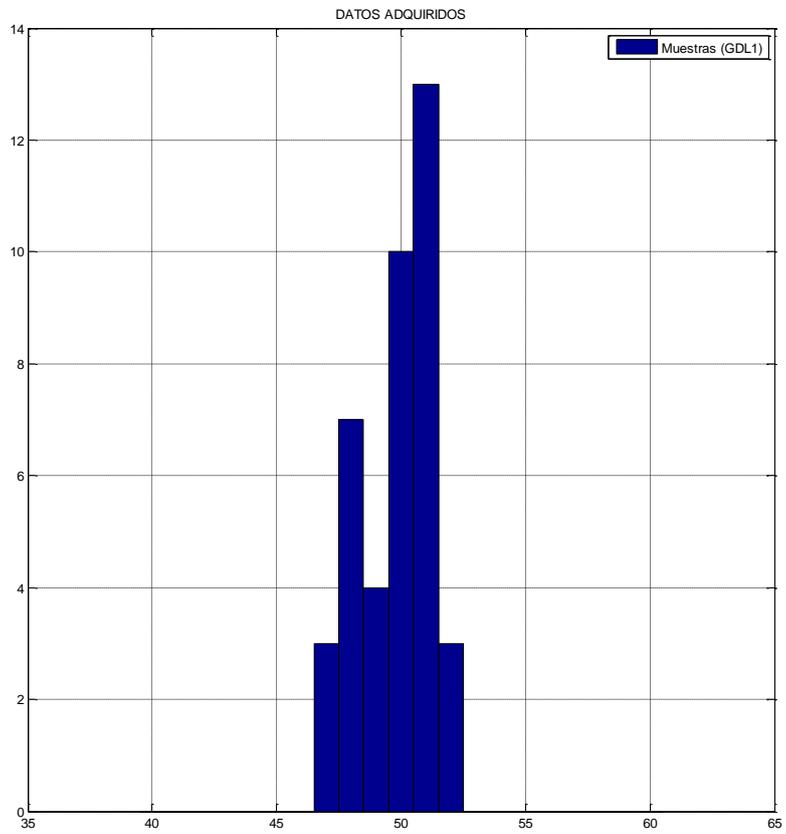


Proceso 4 GRADO-1

Punto Final GD1=50, GD2=30, GD3=80, GD4=-100, GD5=-40

Media	49.7699	Exactitud.	0.2301
Desviación Estándar	1.4065	Precisión.	1.4065

GDL1

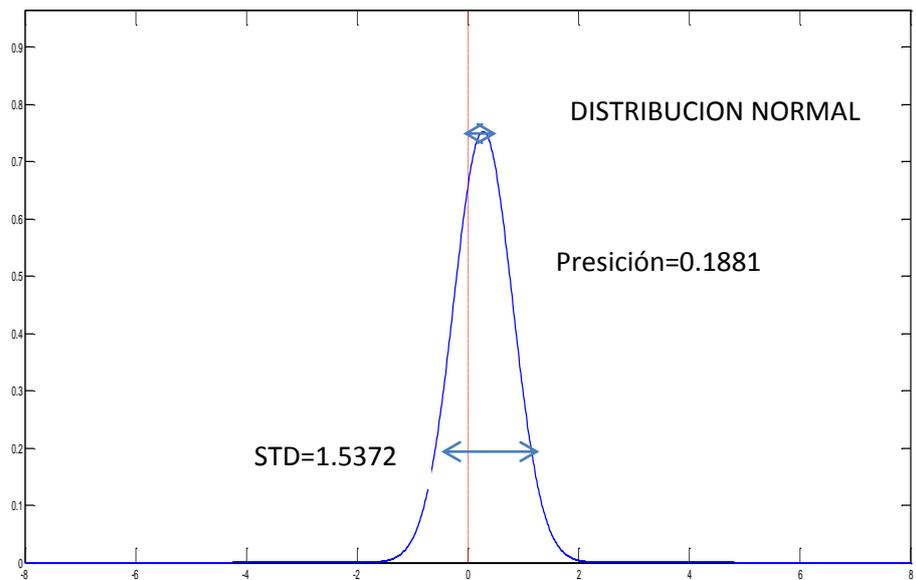
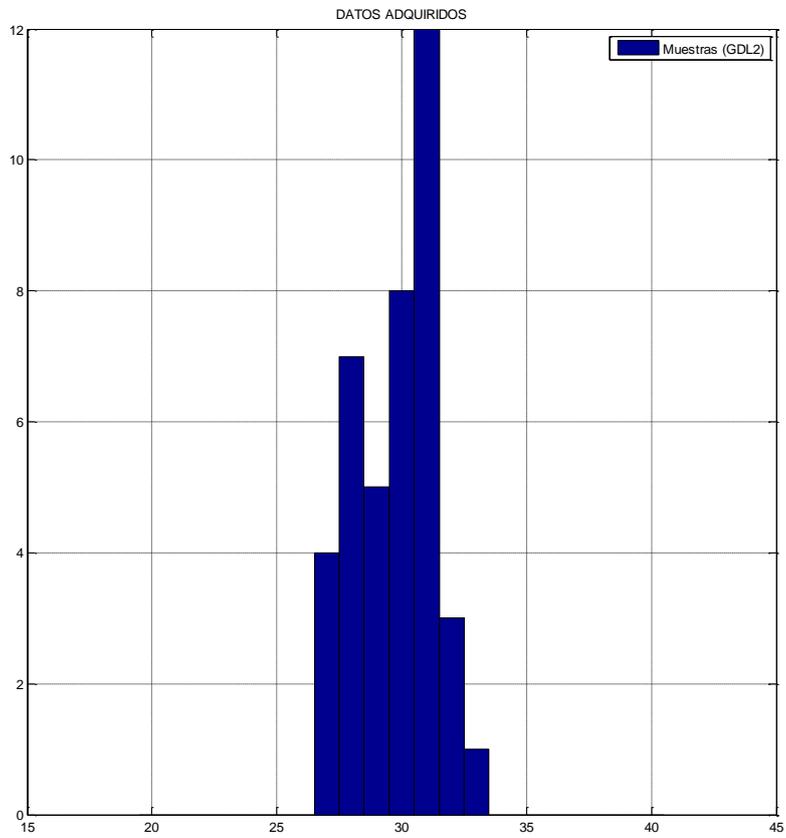


Proceso 4 GRADO-2

Punto Final GD1=50, GD2=30, GD3=80, GD4=-100, GD5=-40

Media	29.8119	Exactitud.	0.1881
Desviación Estándar	1.5372	Precisión.	1.5372

GDL2

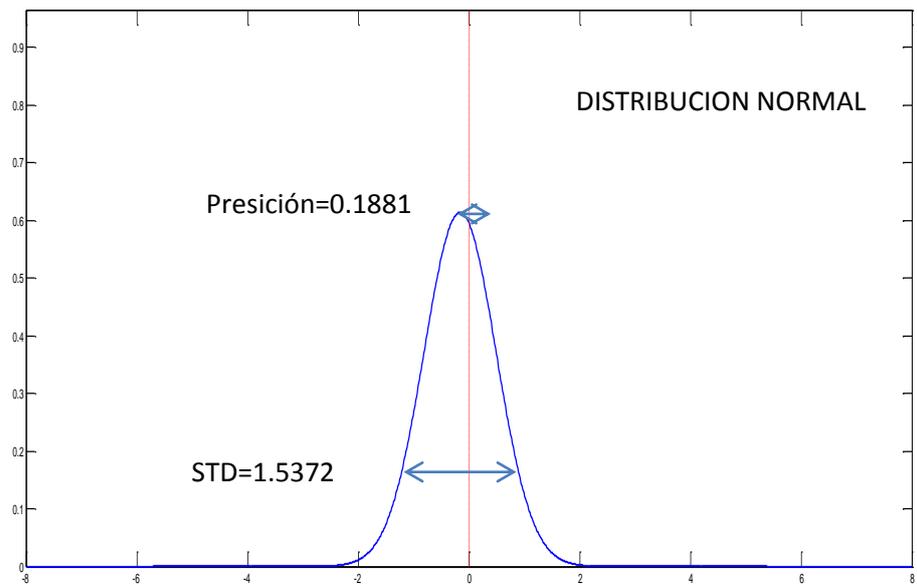
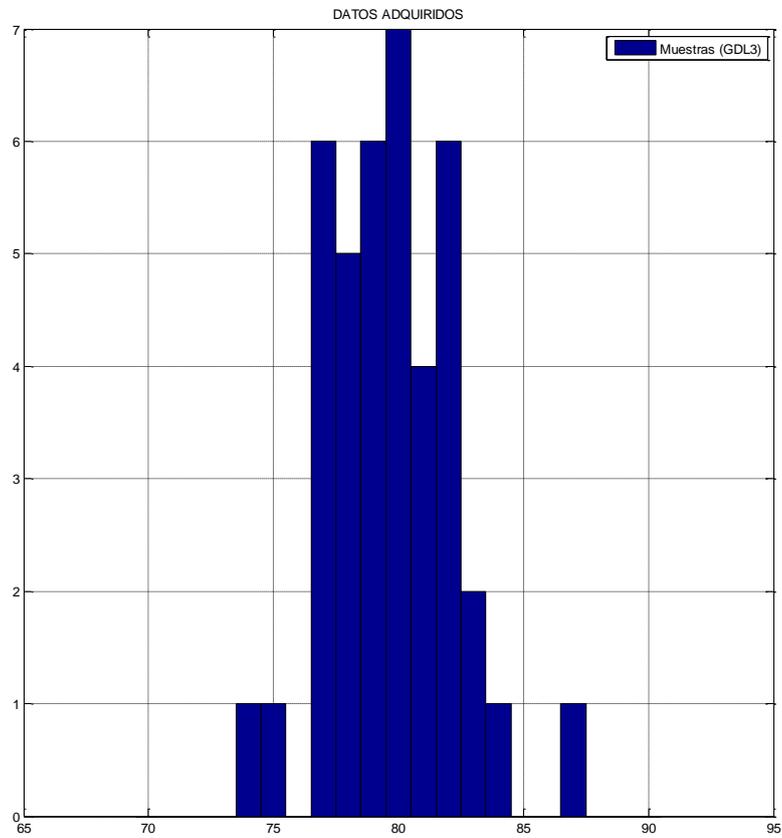


Proceso 4 GRADO-3

Punto Final GD1=50, GD2=30, GD3=80, GD4=-100, GD5=-40

Media	79.7197	Exactitud.	0.2803
Desviación Estándar	2.5286	Precisión.	2.5286

GDL3

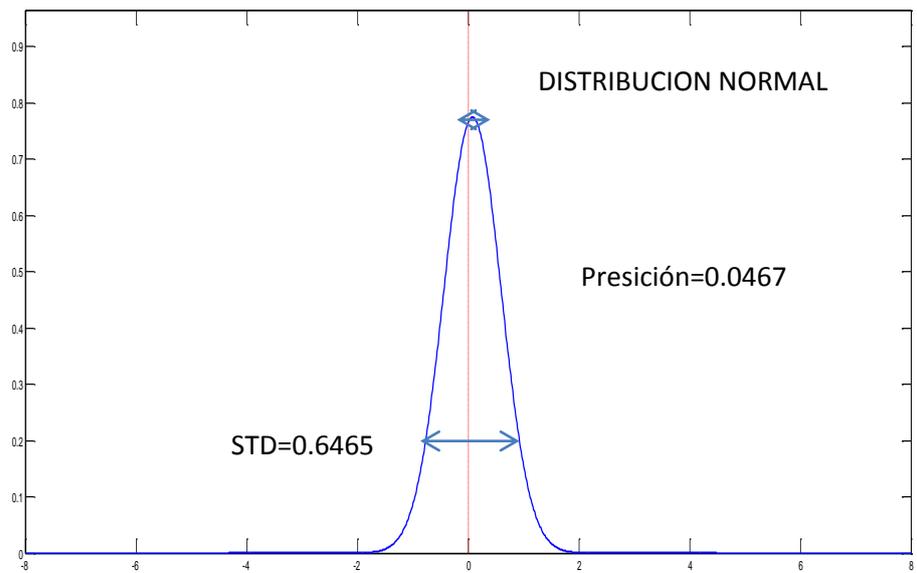
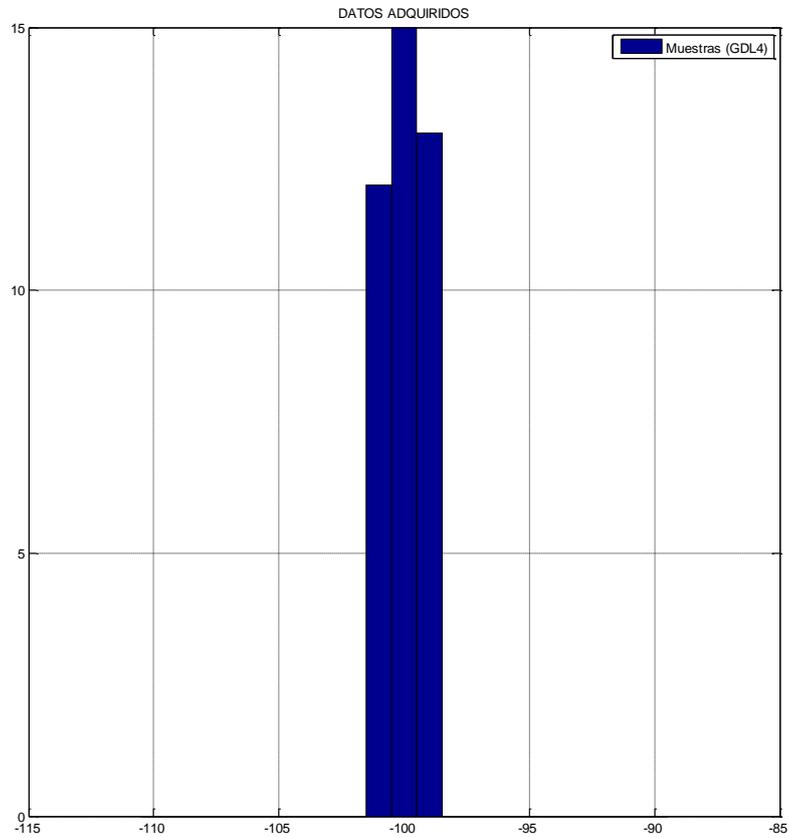


Proceso 4 GRADO-4

Punto Final GD1=50, GD2=30, GD3=80, GD4=-100, GD5=-40

Media	-99.9533	Exactitud.	0.0467
Desviación Estándar	0.6465	Precisión.	0.6465

GDL4

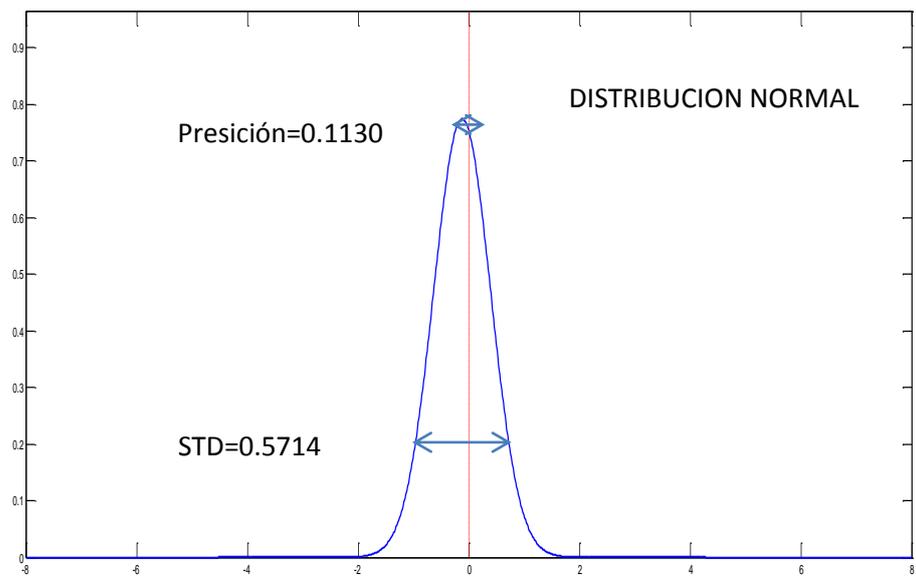
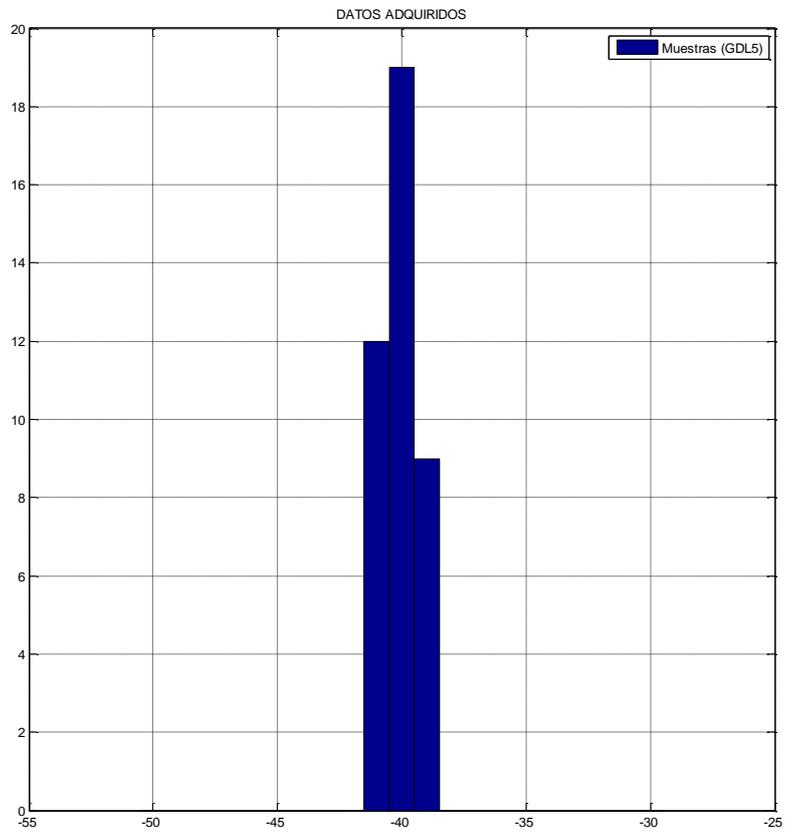


Proceso 4 GRADO-5

Punto Final GD1=50, GD2=30, GD3=80, GD4=-100, GD5=-40

Media	-40.1130	Exactitud.	0.1130
Desviación Estándar	0.5714	Precisión.	0.5714

GDL5

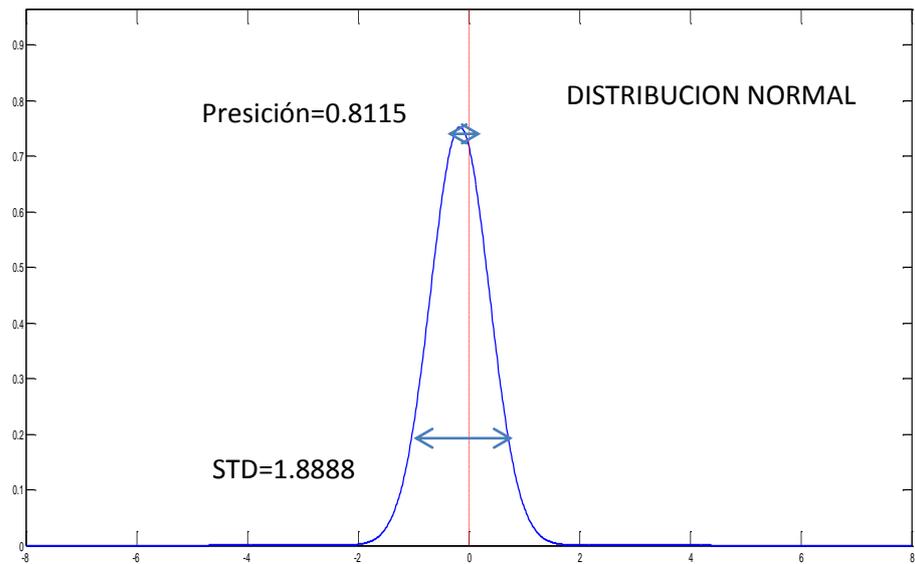
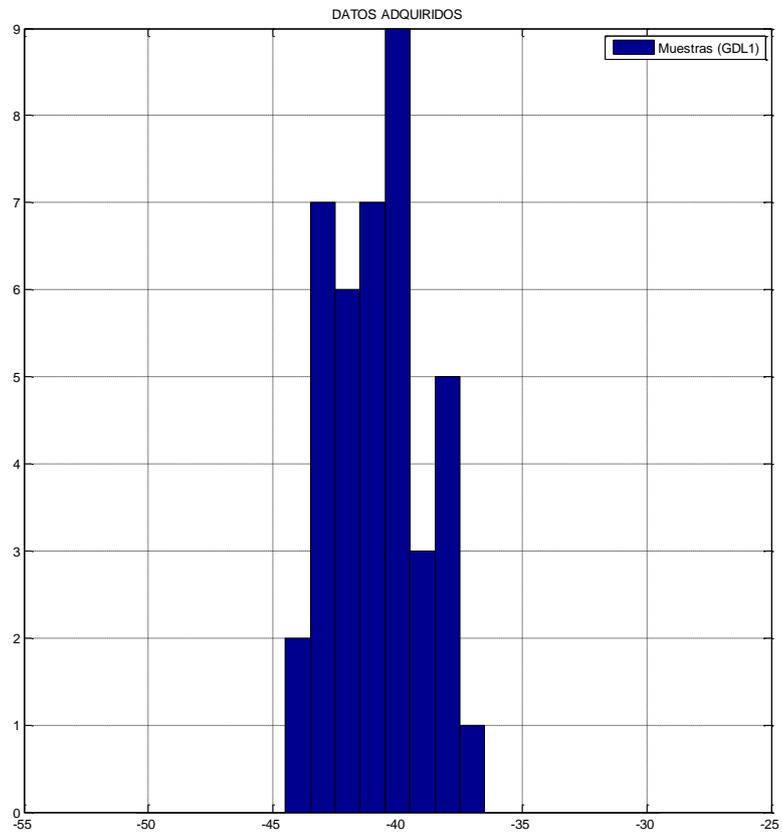


Proceso 5 GRADO-1

Punto Final GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30

Media	-40.8115	Exactitud.	0.8115
Desviación Estándar	1.8888	Precisión.	1.8888

GDL1

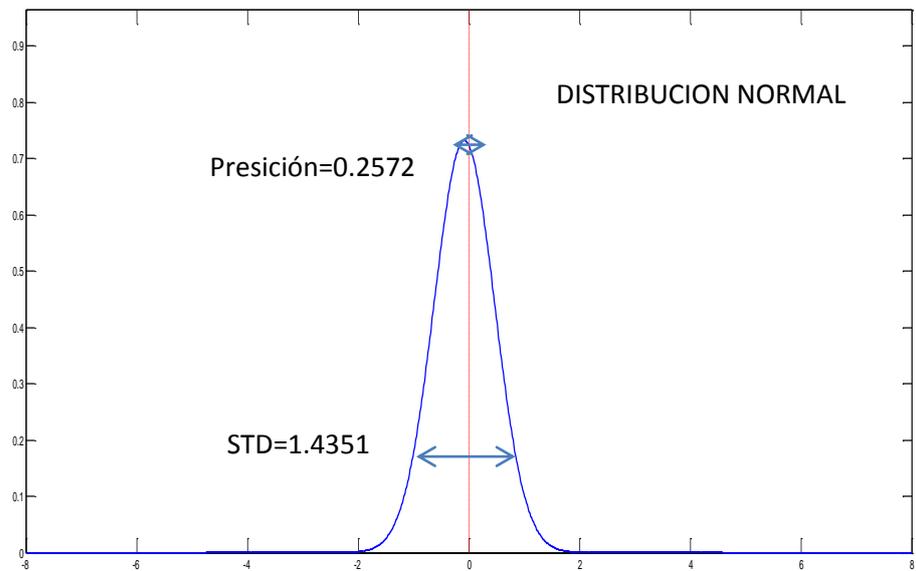
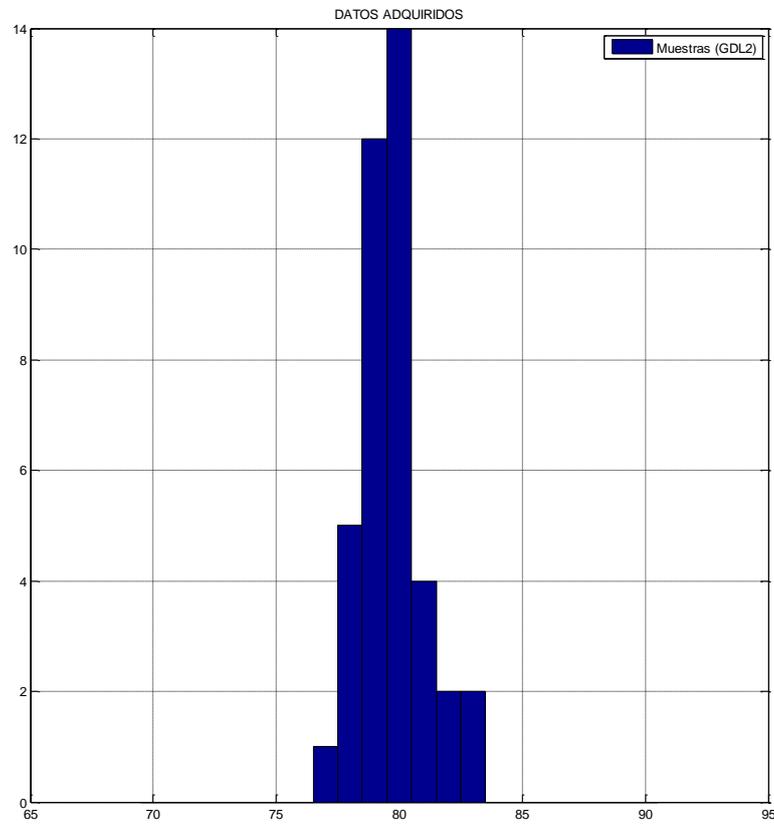


Proceso 5 GRADO-2

Punto Final GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30

Media	79.7428	Exactitud.	0.2572
Desviación Estándar	1.4351	Precisión.	1.4351

GDL2

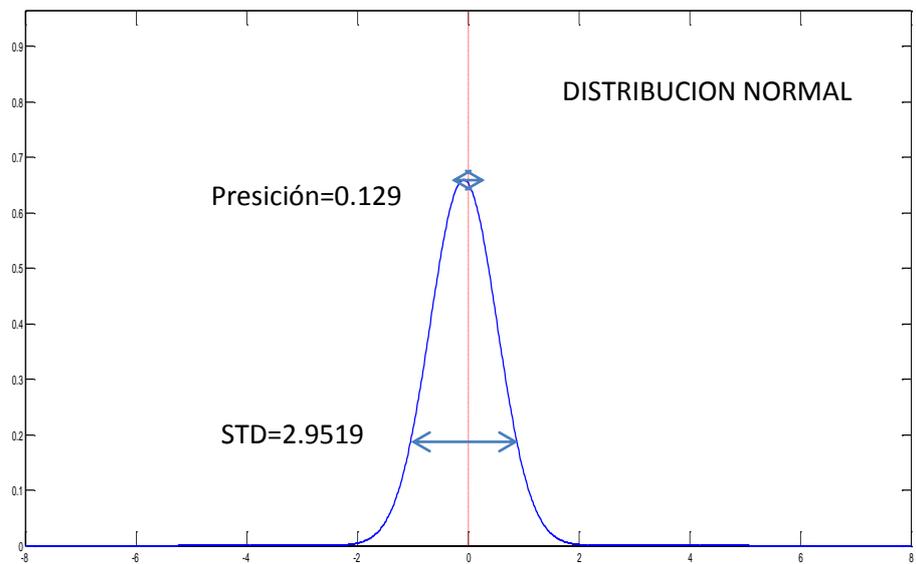
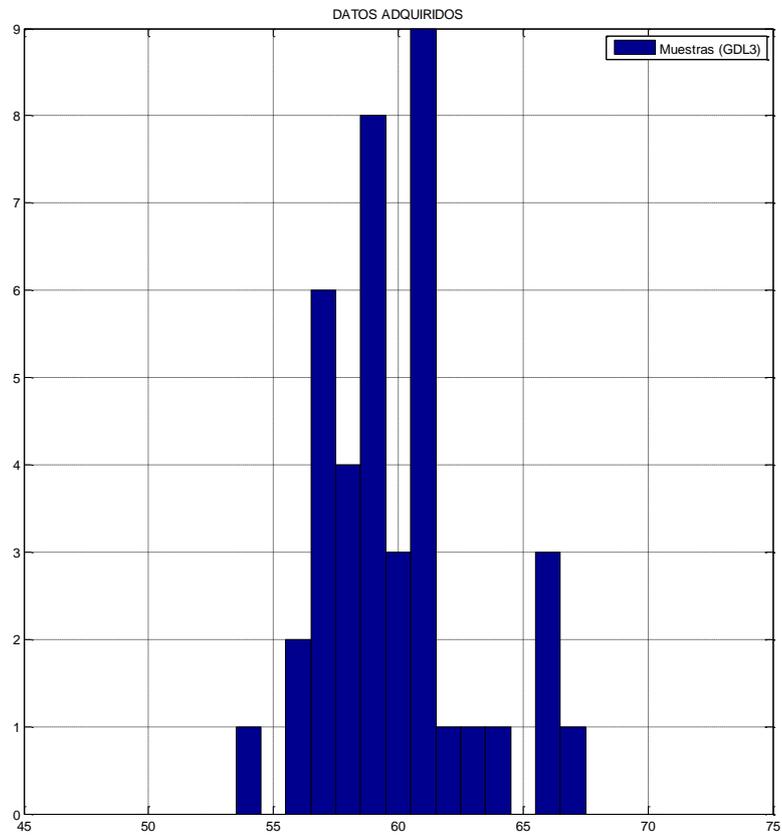


Proceso 5 GRADO-3

Punto Final GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30

Media	59.8710	Exactitud.	0.129
Desviación Estándar	2.9519	Precisión.	2.9519

GDL3

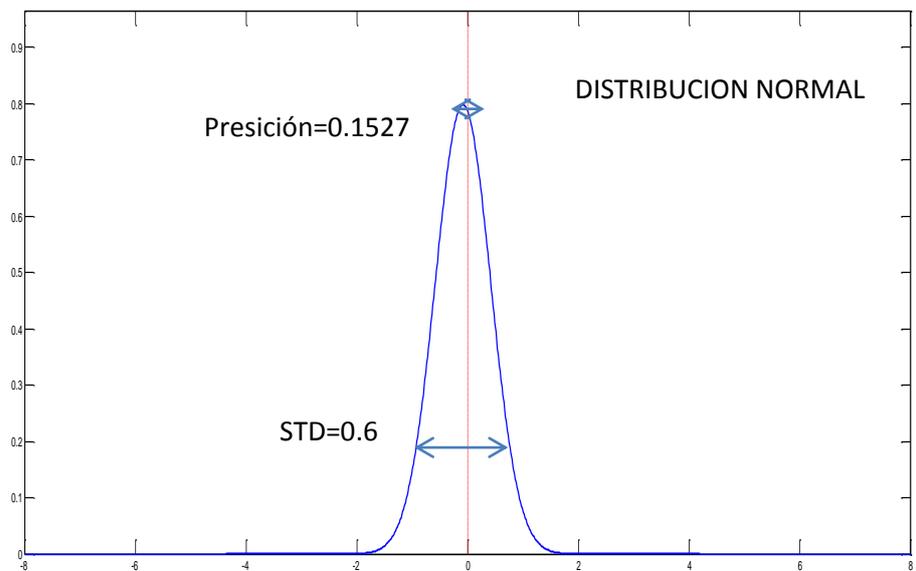
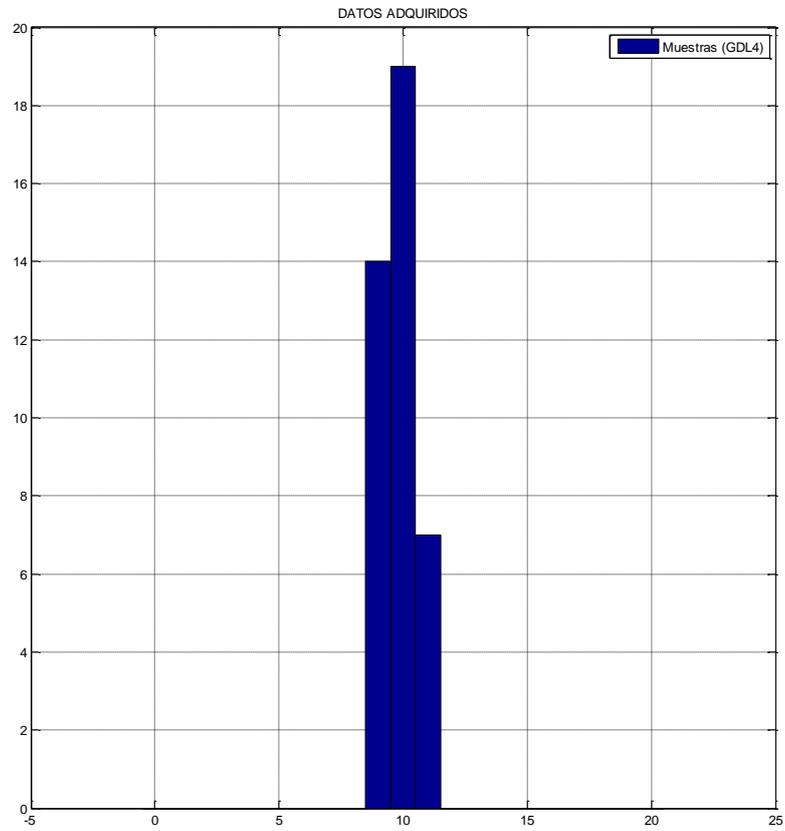


Proceso 5 GRADO-4

Punto Final GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30

Media	9.8473	Exactitud.	0.1527
Desviación Estándar	0.6	Precisión.	0.6

GDL4

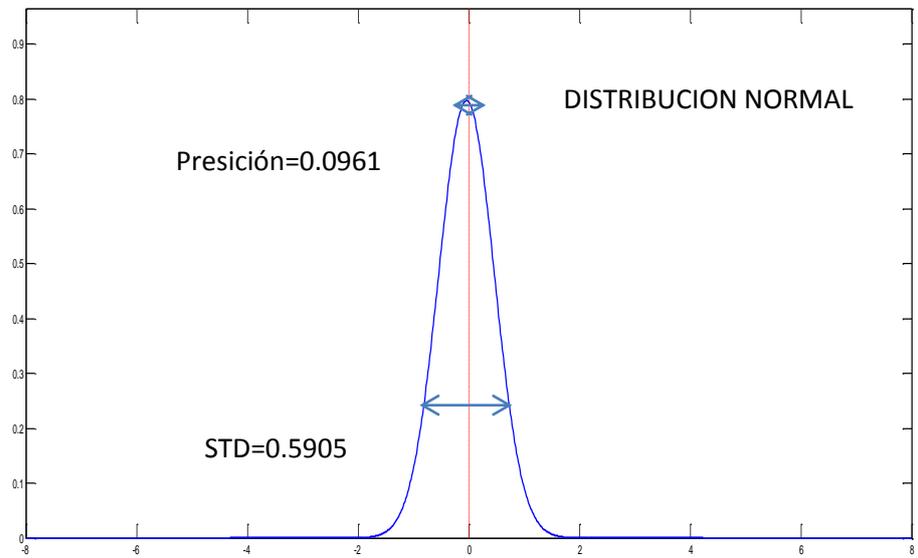
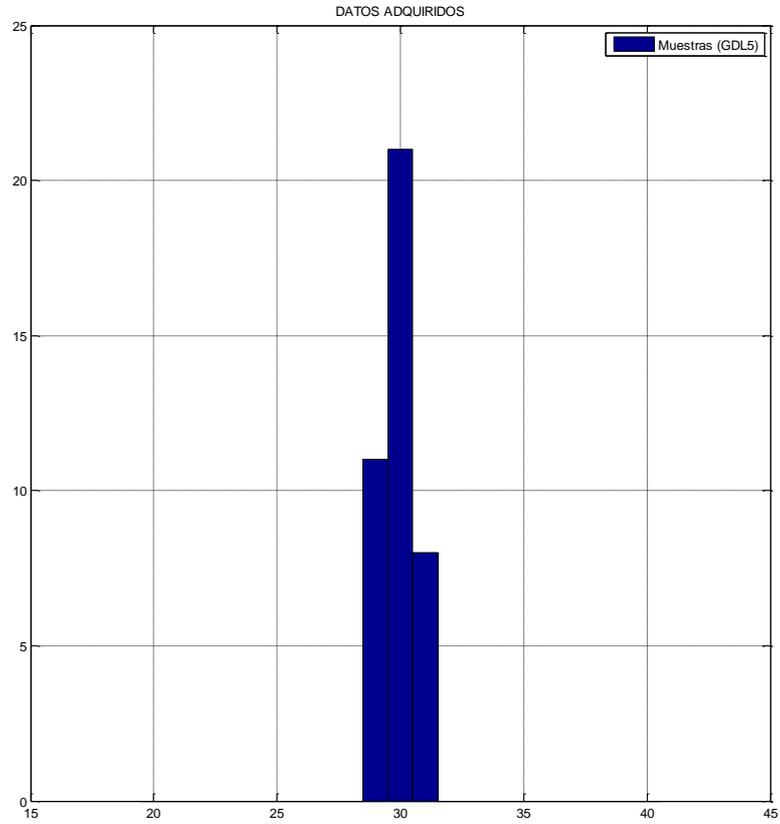


Proceso 5 GRADO-5

Punto Final GD1=-40, GD2=80, GD3=60, GD4=10, GD5=30

Media	29.9039	Exactitud.	0.0961
Desviación Estándar	0.5905	Precisión.	0.5905

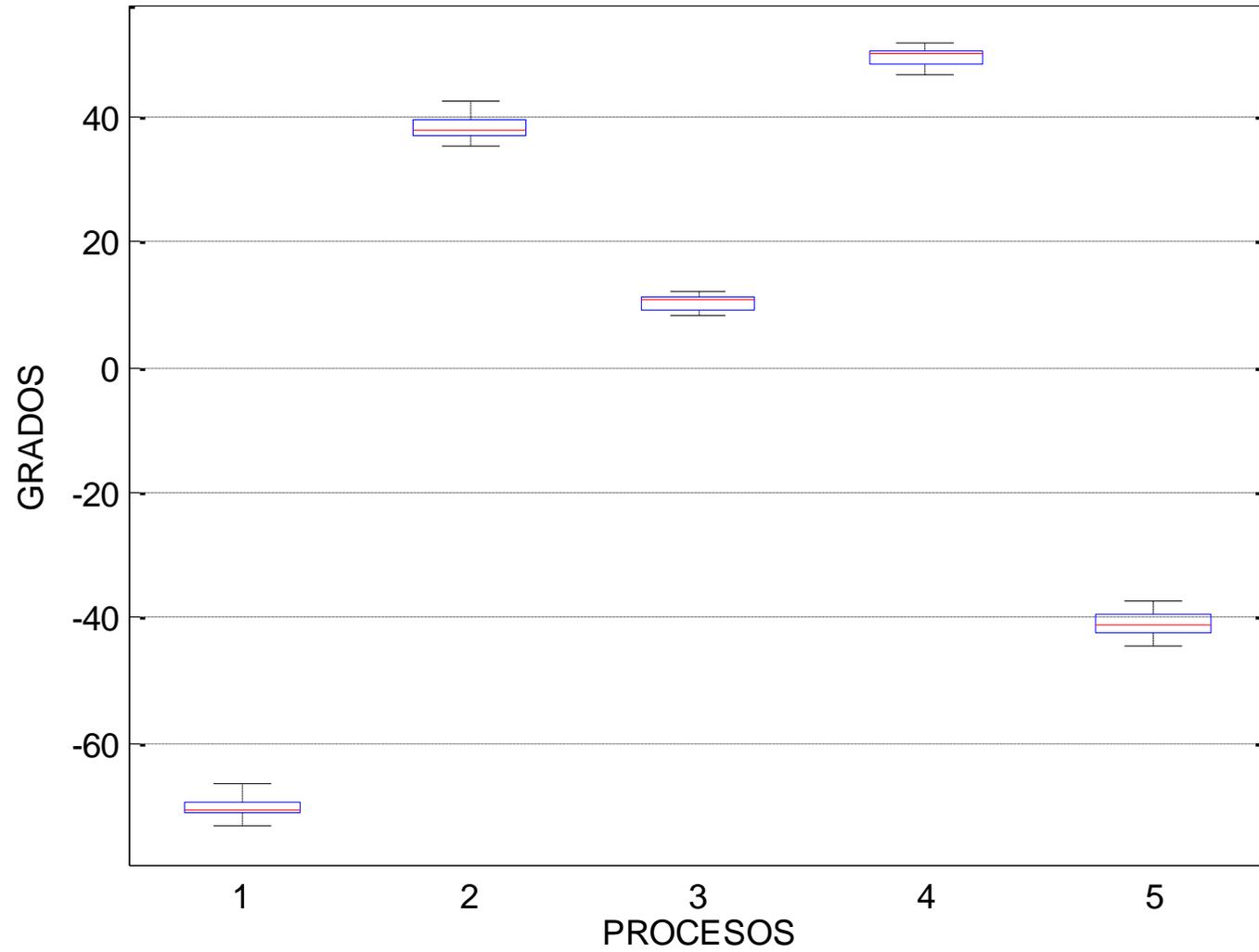
GDL5



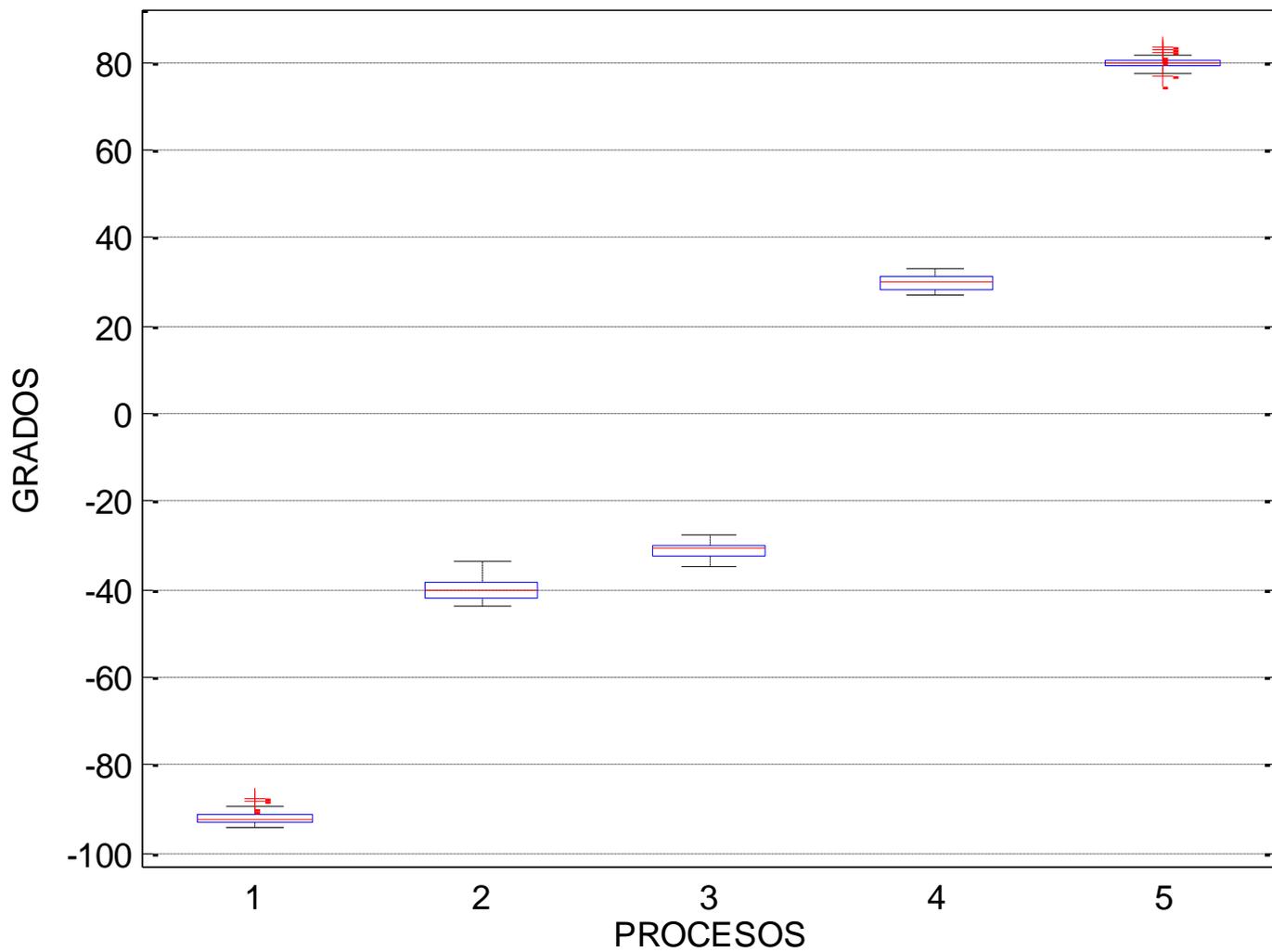
ANEXO C

**ANÁLISIS ESTADÍSTICO
COMPARATIVO DE LOS
EXPERIMENTOS REALIZADOS.**

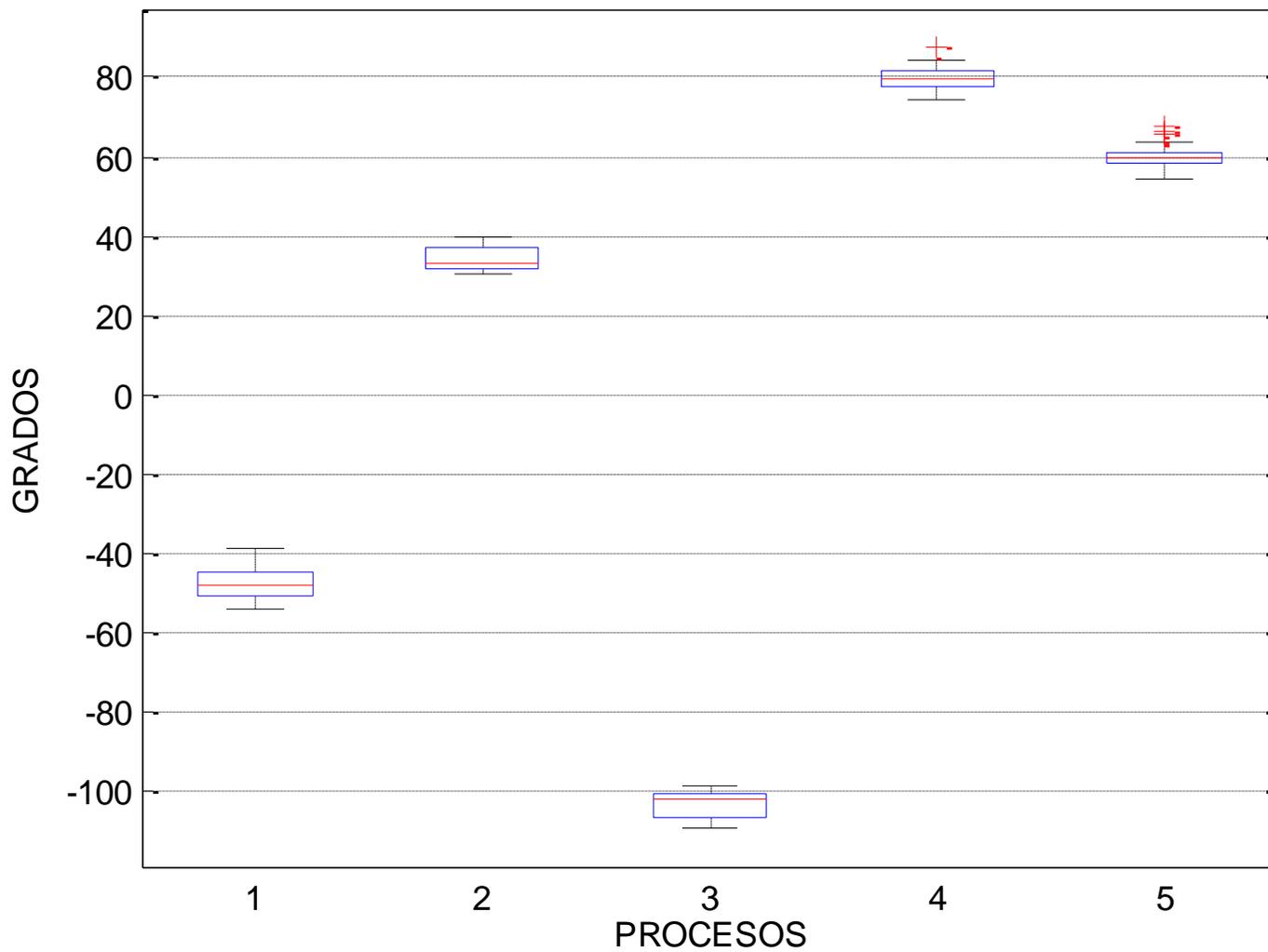
COMPARACION DE PROCESOS GDL1



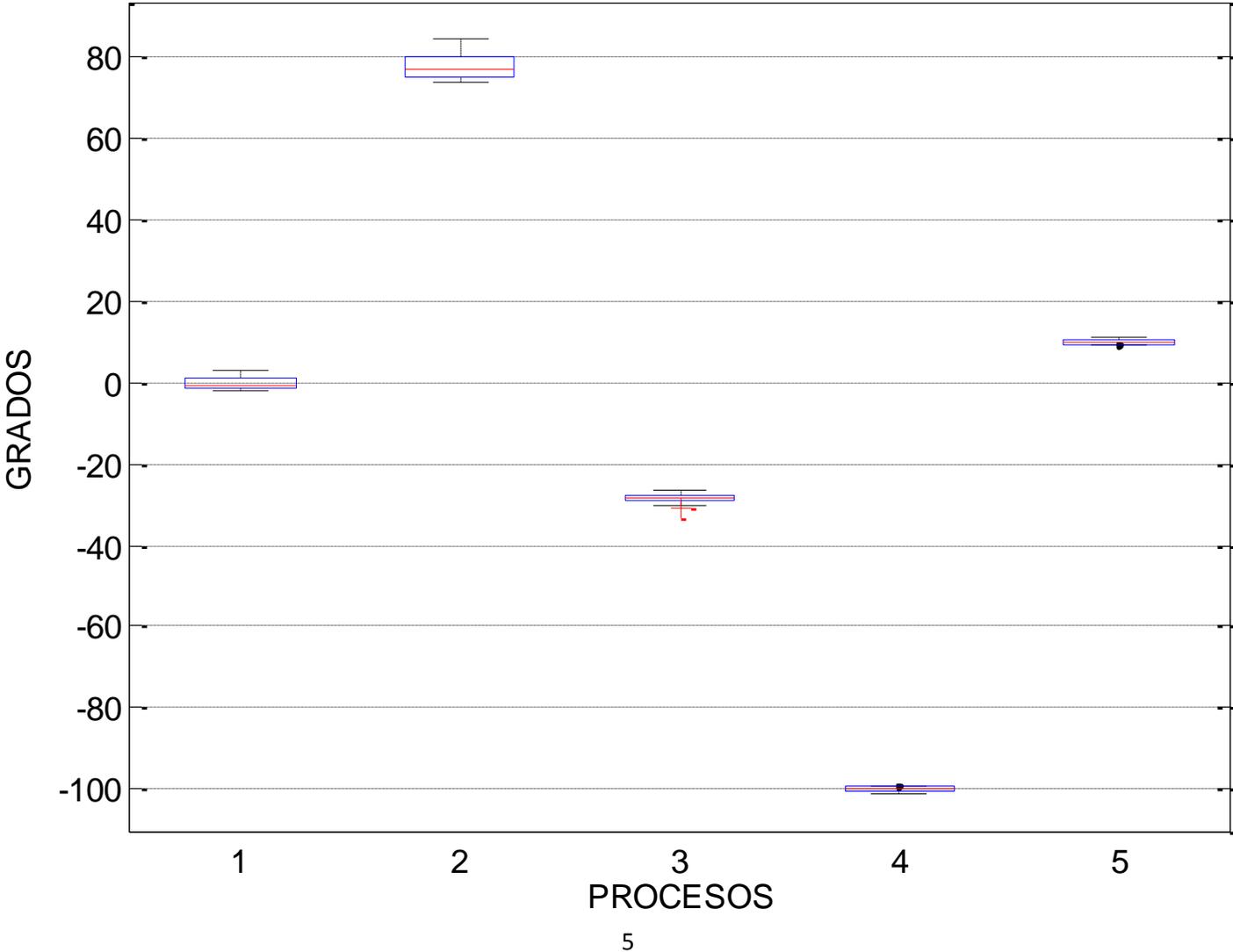
COMPARACION DE PROCESOS GDL2



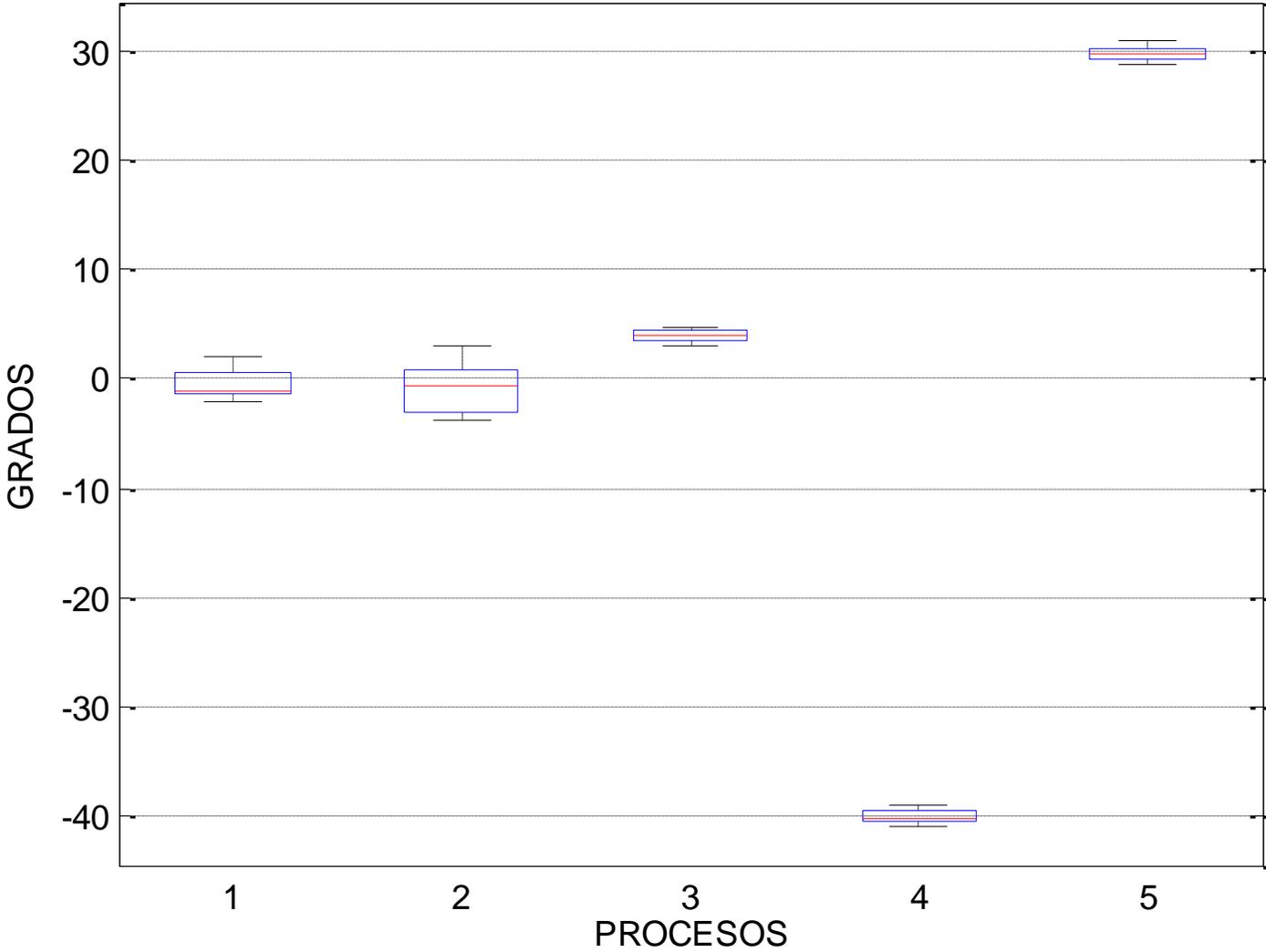
COMPARACION DE PROCESOS GDL3



COMPARACION DE PROCESOS GDL4



COMPARACION DE PROCESOS GDL5



REFERENCIAS BIBLIOGRAFICAS

- [1]. Barrientos A., Peñin L., Balaguer C., Aracil R. Fundamentos de Robótica, Universidad Politécnica de Madrid. McGraw-Hill España, 1997.
- [2]. Guzmán A., Cid J., Reyes F., Méndez J., Porras F., Villegas G. Interfaz del Brazo Robot PUMA 200 con la PC, Benemérita Universidad Autónoma de Puebla.
- [3]. FU K. S., González R. C., LEE C. S. G., Robótica: Control, detección, visión e inteligencia, Universidad Nacional de Educación a Distancia, McGraw-Hill España. 1997.
- [4]. Miguel Torres Torriti, Manipuladores Robóticos, Pontificia Universidad Católica de Chile, Tesis de la facultad de Ingeniería Eléctrica, Mayo 2006.
- [5]. Martínez P., Asmar P., Ibarra M., Restrepo M. Aprender a investigar: El proyecto de investigación. Instituto Colombiano para el fomento de la Educación Superior, ICFES. Santa Fe de Bogotá, 2009.
- [6]. Ronald W. Probabilidad y Estadística para Ingenieros, Prentice –Hall Hispanoamérica SA, Sexta Edición, México 1999.
- [7]. Ander-Egg, E Técnicas de Investigación (2da ed.). México, 1997.
- [8]. Escandón R., Ramón P., Delgado K., Peñafiel D.; Historia y evolución de la robótica Industrial. Universidad Politécnica Salesiana Cuenca Ecuador 2012.
- [9]. Umberto Eco, Técnicas y Procedimientos de investigación, estudio y escritura. Gedisa, S.A. España, 1988.
- [10]. ISO (International Organization for Standardization) 8373:2012, Robots and robotic devices.
- [11]. Robótica Industrial, Ingeniería de sistemas y Automática. Universidad Miguel Hernández. 2010.
- [12]. National Instruments. NI USB-621x DAQ, User Manual.
- [13]. Unimate; PUMA Mark III Robot. 700 Series Equipment Manual. 1986.
- [14]. Unimation Inc. PUMA Mark III Robot Equipment Manual 1986.

- [15]. Kuka Roboter GmbH; Principios de la programación de robots. Edición 19.12.2008
- [16]. Kuka Roboter GmbH; Especificaciones Robots KR16. Edición 06.2003.08
- [17]. Kuka Roboter GmbH; Especificaciones Robots KR5. Edición 02.2011.02
- [18]. Kuka Roboter GmbH; Manipuladores de cargas bajas KR16 & KR6-2.
- [19]. Kuka Roboter GmbH; KR C2 Manual, Edición 2005.
- [20]. Kuka Roboter GmbH; KUKA.Ethernet KRL XML 1.2, Versión 10.04.2012.
- [21]. Kuka Roboter GmbH; KUKA.RobotSensorInterface 2.3, Versión 13.05.2009.
- [22]. WAGO Kontakttechnik GmbH & Co. 750-306 KG,DeviceNet Fieldbus Coupler. Edición 18.02.2010.
- [23]. WAGO Kontakttechnik GmbH & Co. Modular I/O System DeviceNet 750-306, 750-806 Manual, Edición 1.0.0.
- [24]. Feedback Instruments, Mentor desktop robot, 1978.
- [25]. Texas Instruments Incorporated, L293D Quadruple half-h driver, 1990.

REFERENCIAS ELECTRÓNICAS

- [26]. <http://www.prsrobots.com/puma200.html>.
- [27]. <http://www.robothalloffame.org/inductees/03inductees/unimate.html>.
- [28]. <http://www.kuka-robotics.com/es/company/group/milestones/1973.htm>.
- [29]. <http://robots.epson.com>.
- [30]. <http://www.abb.com/robots>.
- [31]. <http://www.fanucrobotics.com>.
- [32]. <http://www.kawasakirobotics.com/>.
- [33]. <http://www.kuka-robotics.com/en/products/controllers/smartPAD/start.htm>.
- [34]. <http://www.reisrobotics.de/produkte/robot-control>.
- [35]. http://www.mitsubishi-automation.es/products/robots_content.html.

- [36]. <http://www.mathworks.com/products/pfo>.
- [37]. <http://www.mathworks.com/company/aboutus>.
- [38]. <http://www.mathworks.com/products/simulink/>
- [39]. <http://www.it-mentor.com.ar/servicios/capitacion.htm>.
- [40]. <http://www.mathworks.com/products/statistics/>.
- [41]. <http://www.kuka-robotics.com/en/company/>.
- [42]. <http://www.sintef.no/home/Information-and-Communication-Technology-IC/Applied-Cybernetics/>.