

# **ESCUELA POLITÉCNICA DEL EJÉRCITO**

## **CARRERA DE INGENIERÍA MECATRÓNICA**

### **“CONTROL DE MOVIMIENTO DEL MANIPULADOR DE CINCO GRADOS DE LIBERTAD CRS-A255 POR ESQUELETIZACIÓN”**

#### **PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN MECATRÓNICA**

**REALIZADO POR:**

**SYLVIA NATHALY REA MINANGO**

**DIRECTOR: ING. ALEJANDRO CHACÓN**

**CODIRECTOR: ING. LUIS ECHEVERRÍA**

**Sangolquí, 2012 – 11 - 28**

## **CERTIFICACIÓN DE LA ELABORACIÓN DEL PROYECTO**

El proyecto **“CONTROL DE MOVIMIENTO DEL MANIPULADOR DE CINCO GRADOS DE LIBERTAD CRS-A255 POR ESQUELETIZACIÓN”**, fue realizado en su totalidad por Sylvia Nathaly Rea Minango, como requerimiento parcial para la obtención del título de Ingeniera en Mecatrónica.

---

**Ing. Luis Alejandro Chacón**  
**Chacón Encalada**  
**DIRECTOR**

---

**Ing. Luis Manuel**  
**Echeverría Yáñez**  
**CODIRECTOR**

Sangolquí, 2012-11-28

## **LEGALIZACIÓN DEL PROYECTO**

**“CONTROL DE MOVIMIENTO DEL MANIPULADOR DE CINCO GRADOS DE  
LIBERTAD CRS-A255 POR ESQUELETIZACIÓN”**

**ELABORADO POR:**

---

Sylvia Nathaly  
Rea Minango

**CARRERA DE INGENIERÍA MECATRÓNICA.**

---

Director de la Carrera de Ingeniería Mecatrónica.

**Ing. Hernán Vinicio Lara Padilla.**

Sangolquí, 2012-11-28

# CARRERA DE INGENIERÍA MECATRÓNICA

## AUTORIZACIÓN

Yo, Sylvia Nathaly Rea Minango

Autorizo a la Escuela Politécnica del Ejército la publicación, en la biblioteca virtual de la institución del proyecto de grado titulado: “CONTROL DE MOVIMIENTO DEL MANIPULADOR DE CINCO GRADOS DE LIBERTAD CRS-A255 POR ESQUELETIZACIÓN”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

---

Sylvia Nathaly  
Rea Minango

Sangolquí, 2012-11-28

## DEDICATORIA

Dedico este proyecto a mi padre, Luis, por sus consejos y el apoyo que siempre me brindó para cumplir mis metas, por ser hasta el último día mi más grande ejemplo de integridad, constancia y valor; a mi madre, Sylvia, por el esfuerzo y sacrificio para sacar adelante a sus hijas, por el cariño que cada día nos entrega.

Ustedes son mi mayor tesoro y mi más grande inspiración.

***Sylvia Nathaly Rea Minango***

## **AGRADECIMIENTOS**

A mi padre, Luis, por su dedicación y apoyo a lo largo de mis estudios, por ser la persona que me enseñó que la disciplina, la perseverancia y la honestidad son las herramientas más importantes para la vida profesional.

A mi madre, Sylvia, por su comprensión y esfuerzo diario, por su apoyo constante e incondicional en los momentos más difíciles.

A mi hermana, Carolina, por su comprensión y paciencia.

A todos mis familiares, por ser el pilar más fuerte cuando estuve más débil.

A mis profesores, por enseñarme más que profesionales hay que ser seres humanos de calidad.

A José Carlos, Diana y Danny V., por sus buenos consejos y por ser siempre amigos incondicionales.

A Richard, por sus consejos, su comprensión y su ayuda a lo largo de este proyecto, y por la alegría y el cariño que me da cada día.

A todos mis amigos y compañeros de carrera, porque me enseñaron que las más grandes victorias son las que se logran frente a las mayores adversidades.

***Sylvia Nathaly Rea Minango***

# ÍNDICE DE CONTENIDO

CERTIFICACIÓN DE LA ELABORACIÓN DEL PROYECTO.....	ii
LEGALIZACIÓN DEL PROYECTO .....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA .....	v
AGRADECIMIENTOS.....	vi
ÍNDICE DE CONTENIDO.....	vii
ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS .....	xiii
ÍNDICE DE ANEXOS .....	xvii
RESUMEN .....	xix
<b>CAPÍTULO 1 .....</b>	<b>1</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1 ANTECEDENTES .....	1
1.2 JUSTIFICACIÓN .....	4
1.3 ALCANCE DEL PROYECTO .....	6
1.4 OBJETIVOS .....	6
1.4.1 GENERAL .....	6
1.4.2 ESPECÍFICOS.....	7
1.5 METODOLOGÍA .....	7
<b>CAPITULO 2 .....</b>	<b>9</b>

<b>MARCO TEÓRICO .....</b>	<b>9</b>
2.1 GENERALIDADES DEL SISTEMA.....	9
2.2 REQUERIMIENTOS DE LA ETAPA DE ADQUISICIÓN DE IMÁGENES	11
2.2.1 DISPOSITIVOS .....	11
2.2.1.1 Cámaras Web .....	12
2.2.1.2 Cámaras estereoscópicas.....	13
2.2.1.3 Sensor Kinect.....	15
2.2.2 SELECCIÓN DE DISPOSITIVOS.....	16
2.3 REQUERIMIENTOS DE LA ETAPA DE PROCESAMIENTO .....	19
2.3.1 ESQUELETIZACIÓN .....	20
2.3.1.1 Definición .....	20
2.3.1.2 Historia.....	21
2.3.1.3 Aplicaciones .....	23
2.3.1.4 Limitaciones .....	24
2.3.2 ALGORITMOS DE ESQUELETIZACIÓN .....	25
2.3.2.1 Algoritmo Zhang-Suen .....	26
2.3.2.2 Algoritmo de Pavlidis.....	29
2.3.2.3 Algoritmo de Holt.....	31
2.3.3 SELECCIÓN DEL ALGORITMO DE ESQUELETIZACION .....	34
2.4 REQUERIMIENTOS DE LA ETAPA DE COMUNICACIONES .....	35
<b>CAPITULO 3 .....</b>	<b>37</b>
<b>DESCRIPCIÓN FÍSICA DEL SISTEMA.....</b>	<b>37</b>
3.1 COMPONENTES FÍSICOS DEL SISTEMA .....	37
3.1.1 SISTEMA DE VIDEO.....	38
3.1.1.1 Distribución de las Cámaras .....	39
3.1.1.1.1 Ángulos de Visión .....	41
3.1.1.1.2 Distancia de Captura.....	42
3.1.2 ILUMINACIÓN .....	45
3.1.3 UNIDAD DE PROCESAMIENTO.....	45



3.1.4	MANIPULADOR ROBÓTICO .....	47
3.1.4.1	Características de Integración.....	48
3.1.4.1.1	Analogía Morfológica .....	49
3.1.4.1.2	Restricciones De Movimiento.....	52
<b>CAPITULO 4</b>	.....	<b>54</b>
<b>SOFTWARE DEL SISTEMA</b>	.....	<b>54</b>
4.1	REQUERIMIENTOS DE SOFTWARE .....	54
4.2	COMPONENTES DEL SOFTWARE.....	55
4.2.1	COMUNICACIÓN ENTRE LOS DISPOSITIVOS.....	56
4.2.2	INTERFAZ GRÁFICA .....	58
4.2.3	PROGRAMA PARA EL CONTROL DEL MANIPULADOR .....	61
4.2.3.1	Software para Desarrollo.....	62
4.2.3.2	Subprocesos del Programa para el Control del Manipulador .....	63
4.2.3.2.1	Inicialización.....	65
4.2.3.2.2	Adquisición y tratamiento de la imagen.....	67
4.2.3.2.3	Esqueletización .....	72
4.2.3.2.4	Aproximación y medición de ángulos.....	76
4.2.3.2.5	Control del gripper.....	80
4.2.3.2.6	Aplicación de restricciones y Ajuste de ángulos .....	82
4.2.3.2.7	Envío de datos .....	85
4.2.3.3	Estructura del Programa de Control del Manipulador .....	87
<b>CAPITULO 5</b>	.....	<b>91</b>
<b>PRUEBAS Y RESULTADOS</b>	.....	<b>91</b>
5.1	PRUEBAS .....	91
5.1.1	METODOLOGÍA DE LA PRUEBA Y VARIABLES A MEDIR.....	91
5.1.2	EJECUCIÓN DE LA PRUEBA.....	93
5.1.2.1	Posición 1.....	95
5.1.2.2	Posición 2.....	96
5.1.2.3	Posición 3.....	97

5.1.2.4	Posición 4.....	98
5.1.2.5	Posición 5.....	99
5.1.2.6	Posición 6.....	100
5.1.2.7	Posición 7.....	101
5.1.2.8	Posición 8.....	102
5.2	ANÁLISIS DE RESULTADOS.....	103
<b>CAPITULO 6 .....</b>		<b>108</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>108</b>
6.1	CONCLUSIONES.....	108
6.2	RECOMENDACIONES .....	111

## ÍNDICE DE TABLAS

Tabla 2.1	Evaluación de dispositivos.....	17
Tabla 2.2	Evaluación de algoritmos de esqueletización .....	34
Tabla 3.1	Especificaciones WebCam Omega .....	39
Tabla 3.2	Ubicación de las cámaras.....	44
Tabla 3.3	Características Luminaria Halógena.....	45
Tabla 3.4	Especificaciones del computador .....	46
Tabla 3.5	Especificaciones del manipulador CRS-A255 .....	47
Tabla 3.6	Restricciones de movimiento .....	51
Tabla 3.7	Referencias para el movimiento del operador .....	51
Tabla 4.1	Parámetros de comunicación con el manipulador .....	57
Tabla 4.2	Secuencia de inicialización.....	66
Tabla 4.3	Casos para medición de ángulos.....	79
Tabla 4.4	Respuesta del comando W2.....	84
Tabla 5.1	Metodología para le ejecución de la prueba .....	92
Tabla 5.2	Datos obtenidos para posición 1.....	95
Tabla 5.3	Resultados para posición 1 .....	95
Tabla 5.4	Datos obtenidos para posición 2.....	96
Tabla 5.5	Resultados para posición 2 .....	96
Tabla 5.6	Datos obtenidos para posición 3.....	97
Tabla 5.7	Resultados para posición 3 .....	97
Tabla 5.8	Datos obtenidos para la posición 4.....	98
Tabla 5.9	Resultados para posición 4 .....	98
Tabla 5.10	Datos obtenidos para la posición 5 .....	99
Tabla 5.11	Resultados para posición 5 .....	99

Tabla 5.12	Datos obtenidos para posición 6 .....	100
Tabla 5.13	Resultados para posición 6 .....	100
Tabla 5.14	Datos obtenidos para posición 7 .....	101
Tabla 5.15	Resultados para posición 7 .....	101
Tabla 5.16	Resultados obtenidos para posición 8.....	102
Tabla 5.17	Resultados para posición 8 .....	102
Tabla 5.18	Resultados de interés de las pruebas .....	103
Tabla 5.19	Características generales de la plataforma .....	107

## ÍNDICE DE FIGURAS

Figura 1.1	Aplicaciones de Visión artificial. (a) Ensamble de piezas robotizado. (b) Control de calidad multiobjeto. (c) Interacción robótica. (d) Identificación de imperfecciones en pintura de automóviles. ....	2
Figura 1.2	Esquema de relaciones entre visión por computadora y otras áreas afines.....	3
Figura 2.1	Elementos del sistema .....	9
Figura 2.2	Correspondencia operador – manipulador CRS-A255 .....	10
Figura 2.3	Cámara Web Logitech.....	12
Figura 2.4	Cámara estereoscópica 120 tri-lens stereo.....	13
Figura 2.5	Percepción de profundidad por disparidad retinal. ....	14
Figura 2.6	Sensor Kinect para Xbox 360.....	15
Figura 2.7	Esquema del método de captura del movimiento de Kinect.....	16
Figura 2.8	Proyección Ortogonal.....	18
Figura 2.9	(a) Patrón original. (b) Esqueleto obtenido.....	21
Figura 2.10	El esqueleto de un rectángulo definido en términos de círculos bi-tangenciales. ....	21
Figura 2.11	Línea de tiempo de autores de trabajos sobre esqueletización .....	23
Figura 2.12	Conjunto de 3x3 píxeles para aplicación del algoritmo Zhang-Suen	26
Figura 2.13	Diagrama de flujo del algoritmo Zhang-Suen.....	28
Figura 2.14	Numeración de píxeles para algoritmo de Pavlidis.....	29
Figura 2.15	Plantillas para aplicación de algoritmo de Pavlidis .....	29
Figura 2.16	Diagrama de flujo del Algoritmo de Pavlidis .....	30
Figura 2.17	Ejemplo de defectos resueltos por el algoritmo de Holt .....	31
Figura 2.18.	Casos para el algoritmo de Holt. (a) Escalera con orientación norte (b) Escalera con orientación sur.....	32

Figura 2.19.	Denominación de los pixeles para el algoritmo de Holt.....	32
Figura 2.20	Diagrama de flujo del algoritmo de Holt .....	33
Figura 2.21	Relación de ángulos entre brazo humano - esqueleto .....	35
Figura 2.22	Etapas trascendentes de la comunicación con el manipulador.....	36
Figura 3.1	Hardware del sistema Hardware .....	38
Figura 3.2	Esquema de ubicación de las cámaras.....	40
Figura 3.3	Cálculo de los ángulos de visión cámara web.....	42
Figura 3.4	Ubicación de las cámaras y áreas captadas .....	44
Figura 3.5	Articulaciones del manipulador CRS-A255. (a) Vista lateral (b) Vista superior .....	49
Figura 3.6	Arco de movimiento de las articulaciones humanas de interés (a) Cintura, (b) hombro, (c) codo, (d) muñeca .....	50
Figura 3.7	Esquema de ángulos y distancias para la cinemática directa del manipulador CRS-A255 .....	53
Figura 4.1	Relación entre los componentes de la plataforma .....	55
Figura 4.2	Componentes del software.....	56
Figura 4.3	Comunicación entre los dispositivos .....	57
Figura 4.4	Elementos esenciales de la interfaz gráfica .....	59
Figura 4.5	Interfaz gráfica desarrollada.....	60
Figura 4.6	Software de desarrollo .....	62
Figura 4.7	Diagrama de flujo del software .....	64
Figura 4.8	Diagrama de flujo de la etapa de inicialización .....	65
Figura 4.9	Diagrama de flujo de la etapa de adquisición y tratamiento de imágenes.....	68
Figura 4.10	Imágenes sin procesamiento. (a) Imagen frontal, (b) Imagen superior .....	70
Figura 4.11	Filtros aplicados en la imagen frontal. (a) Filtro de color de piel, (b) Filtro de color de piel + Suavizado .....	71

Figura 4.12	Región de interés (a) Imagen original, (b) ROI, (c) ROI con filtros...	72
Figura 4.13	Diagrama de flujo de la etapa de esqueletización.....	73
Figura 4.14	Imágenes esqueletizadas (a) superior, (b) frontal .....	74
Figura 4.15	Grados de libertad. (a) sólido, (b) sólido esqueletizado. ....	75
Figura 4.16	Diagrama de flujo de la etapa de aproximación y medición de ángulos.....	77
Figura 4.17	Esqueleto aproximado (a) Imagen superior, (b) Imagen frontal .....	78
Figura 4.18	Diagrama de flujo de la etapa de control del gripper .....	81
Figura 4.19	Diagrama de la etapa de aplicación de restricciones y ajuste de ángulos.....	83
Figura 4.20	Diagrama de flujo de la etapa de envío de datos .....	85
Figura 4.21	Estructura de la sentencia JOINT .....	86
Figura 4.22	Diagrama de clases del software .....	89
Figura 4.23	Dependencias del software .....	90
Figura 5.1	Configuración de la Plataforma de Control de Movimiento por Esqueletización para las pruebas. (a)Unidad de procesamiento ejecutando la aplicación; (b) Sistema de video e iluminación; (c) Ejecución de mediciones manuales en el Manipulador CRS-A255..	94
Figura 5.2	Goniómetro usado durante las pruebas .....	94
Figura 5.3	Posición 1 adoptada por el operario y el manipulador .....	95
Figura 5.4	Postura 2 adoptada por el operador y manipulador .....	96
Figura 5.5	Postura 3 adoptada por el operador y el manipulador .....	97
Figura 5.6	Postura 4 adoptada por el operador y manipulador .....	98
Figura 5.7	Postura 5 adoptada por el operador y manipulador .....	99
Figura 5.8	Postura 6 adoptada por el operador y el manipulador .....	100
Figura 5.9	Postura 7 adoptada por el operador y manipulador .....	101
Figura 5.10	Postura 8 adoptada por el operador y manipulador .....	102

Figura 5.11	Representación gráfica de las posturas con mayor error respecto al manipulador: (a) Postura 1, de acuerdo a los datos presentados en la Tabla 5.2; (b) Postura 4, de acuerdo a los datos de la Tabla 5.8; (c) Postura 5, de acuerdo a los datos de la Tabla 5.10.....	106
Figura 6.1	Esqueleto no céntrico.....	109
Figura 6.2	Pérdida en aproximación de líneas .....	110
Figura 6.3	Resultados aleatorios de esqueletización .....	111



## ÍNDICE DE ANEXOS

<b>ANEXO A</b> .....	¡Error! Marcador no definido.
ESPECIFICACIONES DEL MANIPULADOR CRS-A255	¡Error! Marcador no definido.
<b>ANEXO B</b> .....	¡Error! Marcador no definido.
COMANDOS RAPL-II.....	¡Error! Marcador no definido.
<b>ANEXO C</b> .....	¡Error! Marcador no definido.
MANUAL DE USUARIO Y GUÍA DE INSTALACIÓN	¡Error! Marcador no definido.
VISIÓN GENERAL.....	¡Error! Marcador no definido.
Componentes .....	¡Error! Marcador no definido.
INSTALACIÓN DE LOS ELEMENTOS .....	¡Error! Marcador no definido.
PREPARACIÓN DEL SOFTWARE.....	¡Error! Marcador no definido.
OPERACIÓN DE LA PLATAFORMA .....	¡Error! Marcador no definido.
Iniciar la Plataforma.....	¡Error! Marcador no definido.
Ajustar los parámetros de la imagen .....	¡Error! Marcador no definido.
Controlar el movimiento del manipulador .....	¡Error! Marcador no definido.
Cierre de la aplicación .....	¡Error! Marcador no definido.
RESOLUCIÓN DE PROBLEMAS .....	¡Error! Marcador no definido.
Fallo de comunicación con el manipulador....	¡Error! Marcador no definido.
Fallo de la aplicación durante la ejecución ....	¡Error! Marcador no definido.
El manipulador no responde a los comandos	¡Error! Marcador no definido.
Después de un paro de emergencia.....	¡Error! Marcador no definido.
<b>ANEXO D</b> .....	¡Error! Marcador no definido.
DOCUMENTACIÓN DEL SOFTWARE .....	¡Error! Marcador no definido.
Control De Movimiento Del Manipulador CRS-A255 por Esqueletización	¡Error! Marcador no definido.

LISTA DE CLASES .....	¡Error! Marcador no definido.
LISTA DE ARCHIVOS.....	¡Error! Marcador no definido.
REFERENCIA DE LAS CLASES .....	¡Error! Marcador no definido.
Clase Ui_MainWindow .....	¡Error! Marcador no definido.
Clase MainWindow .....	¡Error! Marcador no definido.
Clase Ui::MainWindow .....	¡Error! Marcador no definido.
Clase comunicación .....	¡Error! Marcador no definido.
Clase control.....	¡Error! Marcador no definido.
Clase esqueletizacion.....	¡Error! Marcador no definido.
Clase hiloCamaras .....	¡Error! Marcador no definido.
Clase medicion .....	¡Error! Marcador no definido.
REFERENCIA DE LOS ARCHIVOS .....	¡Error! Marcador no definido.
comunicacion.cpp.....	¡Error! Marcador no definido.
comunicacion.h.....	¡Error! Marcador no definido.
control.cpp .....	¡Error! Marcador no definido.
control.h.....	¡Error! Marcador no definido.
esqueletizacion.cpp .....	¡Error! Marcador no definido.
esqueletizacion.h.....	¡Error! Marcador no definido.
hilocamaras.cpp .....	¡Error! Marcador no definido.
hilocamaras.h .....	¡Error! Marcador no definido.
main.cpp .....	¡Error! Marcador no definido.
mainwindow.cpp .....	¡Error! Marcador no definido.
mainwindow.h.....	¡Error! Marcador no definido.
medicion.cpp .....	¡Error! Marcador no definido.
medicion.h .....	¡Error! Marcador no definido.
moc_hilocamaras.cpp.....	¡Error! Marcador no definido.
moc_mainwindow.cpp .....	¡Error! Marcador no definido.
ui_mainwindow.h .....	¡Error! Marcador no definido.

## RESUMEN

El presente proyecto, denominado “Control de Movimiento del Manipulador de Cinco Grados de Libertad CRS-A255 por Esqueletización”, plantea el diseño e implementación de una plataforma, cuyo objetivo es controlar el movimiento de un manipulador robótico a través de la imitación los movimientos del brazo humano, sin la intervención de controles mecánicos o el envío manual de comandos en una terminal, para lo cual se vale de una técnica de visión artificial llamada esqueletización.

La utilización de la plataforma planteada ofrece una opción para el control del manipulador CRS-A255, ubicado en el Laboratorio CIM 2000, de la Escuela Politécnica del Ejército, sin limitar al operador en cuanto a movimientos y sin requerir conocimientos avanzados de programación, además de evadir el obstáculo de la incompatibilidad con sistemas modernos, pues al robot se le envía únicamente comandos por su puerto de comunicaciones, a través de un computador.

La plataforma está conformada por un manipulador CRS-A255 y dos cámaras de visión dispuestas en planos perpendiculares enfocadas hacia el usuario para la adquisición de los movimientos; comandados por una aplicación desarrollada en lenguaje C++, en base a librerías de código abierto.

Al final de este proyecto se consigue un control de movimiento del manipulador con una diferencia de aproximadamente  $5^\circ$  respecto a la posición del operador, mediante comandos de movimientos individuales para cada articulación, a una velocidad de trabajo del 40%, permitiendo un control más rápido que el realizado mediante el control manual del manipulador.

Se concluye que la técnica de esqueletización es potente en cuanto a compresión y pre-procesamiento de imágenes; sin embargo su utilización en un sistema de control de movimiento que requiere de alta precisión, posee grandes limitaciones.

# CAPÍTULO 1

## INTRODUCCIÓN

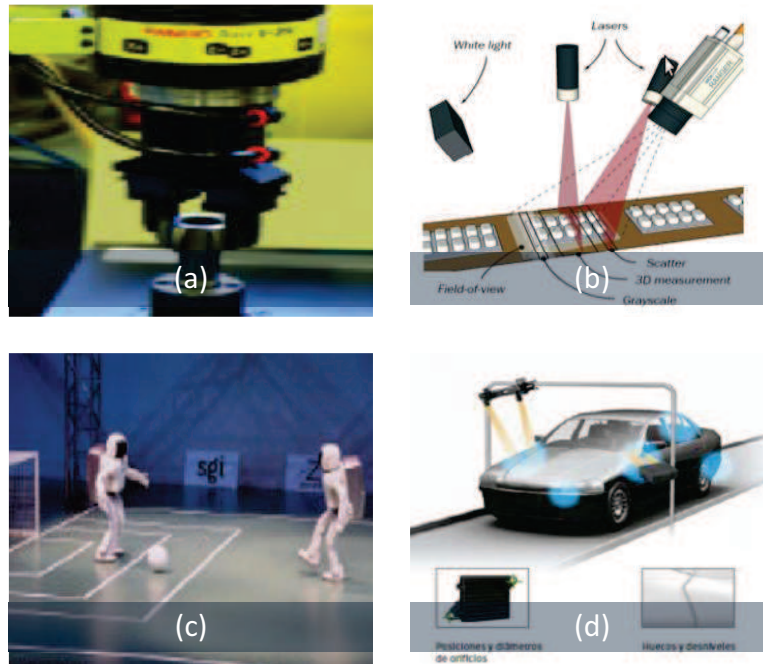
### 1.1 ANTECEDENTES

La visión es uno de los mecanismos sensoriales de percepción más importantes que tiene el ser humano y la mayoría de los organismos biológicos; es utilizada para el desenvolvimiento eficiente dentro del entorno y para la detección de objetos determinados por medio de sus características físicas.

En la década de los 60s el Instituto Tecnológico de Massachusetts (MIT) inicia el estudio de la visión artificial, lo cual implicó no solo captar imágenes a través de una cámara, sino también, comprender el significado de estas imágenes, motivo por el cual se llegó a concluir que la interpretación automática de imágenes supone un gran desafío para la visión artificial a diferencia de la visión humana.

David Marr define a la visión en 1982 como “un proceso que produce, a partir de las imágenes del mundo exterior, una Descripción que es útil para el observador y que no tiene información irrelevante”<sup>1</sup>; en base a este concepto el campo de la visión artificial ha desarrollado aplicaciones a nivel educativo, comercial e industrial, como se muestra en la Figura 1.1.

El desarrollo tecnológico ha permitido el perfeccionamiento de estas aplicaciones, desde la digitalización de imágenes, la creación automática de imágenes panorámicas a partir de varias fotos, hasta sistemas más complejos como la detección y reconocimiento de rostros, visión estereoscópica, sistemas de detección de peatones y vehículos para la conducción asistida, e incluso la búsqueda de información en internet a partir de imágenes y no únicamente de texto.

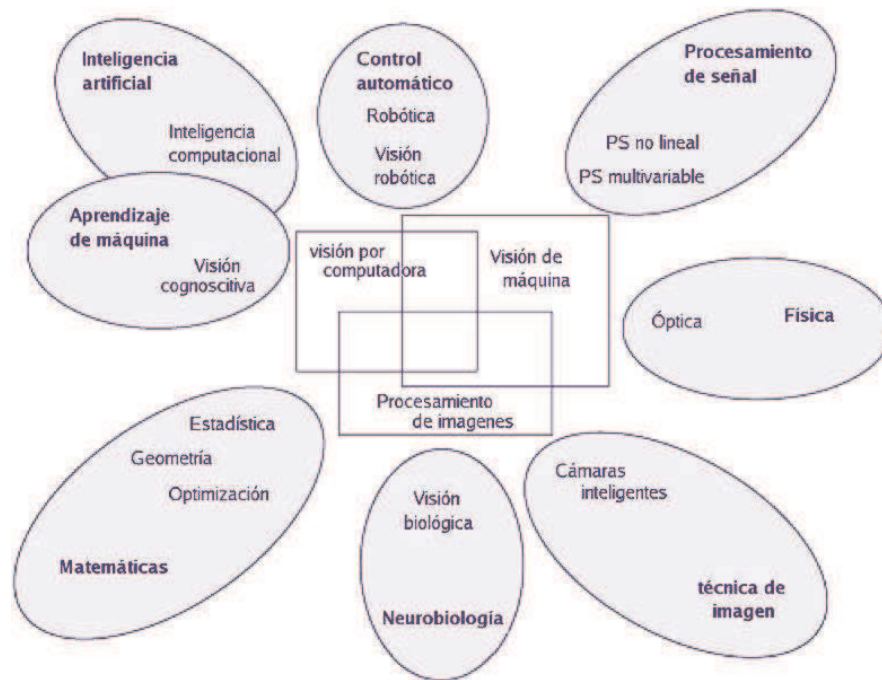


**Figura 1.1 Aplicaciones de Visión artificial. (a) Ensamble de piezas robotizado. (b) Control de calidad multiobjeto. (c) Interacción robótica. (d) Identificación de imperfecciones en pintura de automóviles.**

FUENTE: SOBRADO, E. Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. Pontificia Universidad Católica del Perú. Perú. 2003

Uno de los aportes de la visión artificial que constituirá un hito en este campo es el “Proyecto Natal”, mejor conocido como el sistema para videojuegos Kinect de Microsoft, el cual ha permitido por primera vez la interacción entre jugador y consola sin la necesidad de mandos, valiéndose de un completo sistema de visión estereoscópica. El dispositivo cuenta con una cámara RGB, un sensor de profundidad, un micrófono de múltiples matrices y un procesador personalizado ejecutor del software patentado, que proporciona captura de movimiento de todo el cuerpo en tres dimensiones, reconocimiento facial y capacidad de reconocimiento de voz. A partir de este sistema se ha planteado la posibilidad de adaptarlo para comandar elementos más complejos.

Al ser un campo muy amplio, la visión artificial puede relacionarse con varias ramas de la ingeniería, como se muestra en la Figura 1.2.



**Figura 1.2 Esquema de relaciones entre visión por computadora y otras áreas afines.**

FUENTE: Visión artificial. Wikipedia.org. 2009

Para alcanzar la integración entre visión artificial y otros sistemas se ha precisado dotar de todas las capacidades de la visión humana a los sistemas artificiales. Estos esfuerzos han dado como resultado varias técnicas enfocadas en el reprocesado, compresión, filtrado y segmentación de imágenes para su mejor comprensión.

Una técnica que ha tenido grandes resultados tanto en compresión como en segmentación es la esqueletización, la cual consiste en la reducción de objetos de una imagen, a un esqueleto con grosor de un pixel, el cual mantiene las características del original restándole peso permitiendo un mejor manejo del mismo. La esqueletización, hasta el presente día, ha sido aplicada mayormente para reconocimiento de patrones y gestos, identificación de defectos físicos y verificación de huellas digitales o firmas.

La robótica es una de las disciplinas más favorecidas de los avances en visión artificial y al mismo tiempo la que mayor problema ha tenido al integrarla en sus

sistemas pues, mientras la información recibida a través de sensores se puede interpretar con relativa facilidad y entra a formar parte de la Descripción del modelo de universo empleada por el robot para tomar decisiones, la percepción de las imágenes captadas y su interpretación correcta es una labor muy compleja pero brinda una mejor apreciación del entorno.

La combinación de la visión artificial con sistemas robóticos ha originado el desarrollo de aplicaciones como el control de sistemas teleoperados, sistemas animatrónicos, la implementación de sistemas de rehabilitación física, así como sistemas de asistencia para personas con algún tipo de discapacidad.

La Escuela Politécnica del Ejército cuenta desde 1995 con varios manipuladores CRS modelo A255, los cuales han sido utilizados para el desarrollo de proyectos que relacionan visión artificial. Entre los proyectos más destacados desarrollados en esta institución se pueden citar: Implementación de la estación de Control de Calidad en el laboratorio CIM-2000 (1995) y su posterior modificación (2000), en la cual se ha utilizado este manipulador y una cámara para realizar el control de calidad en dos dimensiones de las piezas dentro del escenario de una línea de producción; Diseño e Implementación de un Sistema de Visión Estereoscópica para el Posicionamiento del Brazo Robótico CRS-A255 (2011), en la cual se utilizaron dos cámaras para crear visión estereoscópica para permitir el seguimiento de un objeto determinado con el manipulador.

El presente proyecto consiste en un sistema que controla los movimientos del manipulador robótico CRS-A255 a través de los movimientos del brazo humano, sin la necesidad de usar controles u otro tipo de periféricos, ayudándose para ello de la visión artificial, especialmente de la técnica de esqueletización.

## **1.2 JUSTIFICACIÓN**

El lenguaje es el medio que utiliza el hombre para gobernar las máquinas controladas por sistemas informáticos, por lo cual, su correcta adaptación con la tarea a realizar y sencillez de manejo, son factores determinantes del rendimiento

obtenido en los robots. Actualmente existen tres maneras de comunicarse con un robot: reconocimiento de voz, enseñanza y repetición; y, lenguajes de programación de alto nivel.

A pesar de los avances realizados en cada uno de ellos, sus limitaciones aún son importantes. En el caso del reconocimiento de voz la dificultad radica en el filtraje del ruido ambiental y la dificultad para describir una tarea específica a realizar; por otra parte, en la enseñanza y repetición la desventaja es el uso de dispositivos externos para el control del robot; finalmente, en el caso de lenguajes de programación de alto nivel el problema radica en la diversidad de los mismos, pues cada lenguaje se aplica a un determinado controlador, además de que es necesario un conocimiento a nivel medio o avanzado de programación para comandar al robot.

El presente proyecto de grado aporta una opción para el control de manipuladores bajo el método de enseñanza y repetición en el cual el dispositivo de control externo sea sustituido por un sistema de visión artificial.

Los principales inconvenientes con los dispositivos de control externos como palancas de juego (joystick) o controles a distancia (teach pendant) son la limitación de movimientos del operario y la falta de una verdadera interacción con el objeto pues sus movimientos son bruscos y no pueden alterarse por la naturaleza del objeto a manipular, convirtiéndose en grandes restricciones sobre todo para sistemas robóticos teleoperados.

Para solventar estos problemas se han creado varios controles que ofrecen mayor libertad al operador, desde estructuras con luces infrarrojas pasando por guantes sensorizados y por último la adaptación de Kinect, lo cual ha brindado nuevas perspectivas en este campo. Sin embargo, con el nacimiento de nuevos controles cada vez más sofisticados aparece un nuevo obstáculo: la incompatibilidad de éstos con las plataformas robóticas, especialmente las antiguas, como es el caso del manipulador CRS-A255.

La utilización del sistema de visión artificial planteado en este proyecto ofrece una opción de control para este manipulador, sin limitar al operador en cuanto a movimientos y sin requerir conocimientos avanzados de programación, además de evadir el obstáculo de la incompatibilidad, pues al robot se le envía únicamente



comandos por su puerto de comunicaciones a través de un computador, encargado de realizar el procesamiento previo.

Adicionalmente, ese proyecto abre las puertas para nuevos temas de investigación en visión artificial que pueden ser enriquecidos con visión estereoscópica o realidad aumentada aplicada a sistemas robóticos.

Valiéndose de que, la Escuela Politécnica del Ejército ha impulsado el desarrollo de proyectos de investigación en todas las áreas y de que el campo de la visión artificial es un tema abierto, se plantea este proyecto como base para una nueva línea de investigación.

### **1.3 ALCANCE DEL PROYECTO**

El presente proyecto desarrolló un programa en lenguaje C y C++ que permite controlar los movimientos del manipulador CRS-A255 hasta el 30% de su velocidad, a través de la imitación de los movimientos del brazo humano, bajo la técnica de esqueletización.

El sistema está conformado por un manipulador CRS-A255 y dos cámaras de visión dispuestas en planos perpendiculares enfocadas hacia el usuario para la adquisición de los movimientos.

El código del programa se desarrolló en base a librerías de código abierto ya existentes; y, dispone de una interfaz gráfica para su utilización.

Se generó, adicionalmente, un manual de usuario del sistema, las instrucciones de montaje de los elementos y se realizaron pruebas para determinar las condiciones óptimas de funcionamiento del sistema.

### **1.4 OBJETIVOS**

#### **1.4.1 GENERAL**

Implementar una plataforma de control de movimiento para el manipulador CRS-A255 mediante visión artificial por esqueletización.

#### **1.4.2 ESPECÍFICOS**

- Desarrollar un método de esqueletización para el brazo humano mediante librerías de visión artificial
- Controlar el movimiento de los eslabones del manipulador CRS-A255 mediante comandos de movimiento continuo
- Integrar el sistema de visión artificial con el manipulador
- Determinar las condiciones bajo las cuales la plataforma es capaz de controlar el manipulador.

#### **1.5 METODOLOGÍA**

Para alcanzar los objetivos planteados en este proyecto se utilizará como base la metodología científica basada en el análisis, diseño, implementación y evaluación.

- En la fase de análisis se estudiarán las posibles alternativas para el sistema, así como las configuraciones, componentes y limitaciones de cada una. Se escogerá la que mejor rendimiento y prestaciones ofrezca.
- Sobre la base de la alternativa elegida, en la etapa de diseño, se determinará la forma de aplicación de la esqueletización en el sistema de visión artificial y la comunicación con el manipulador CRS-A255. Se determinarán las interacciones apropiadas entre estos componentes que permiten alcanzar los objetivos del sistema planteado.
- En la fase de implementación, se desarrollarán los componentes de visión artificial por esqueletización y de comunicaciones, a través de la programación orientada a objetos. Se integrarán estos componentes en un solo sistema capaz de controlar el movimiento del manipulador CRS-A255.

- Finalmente, en la fase de evaluación se valorará la efectividad y eficiencia del sistema a través de pruebas que además permitan identificar las condiciones apropiadas de funcionamiento del mismo.

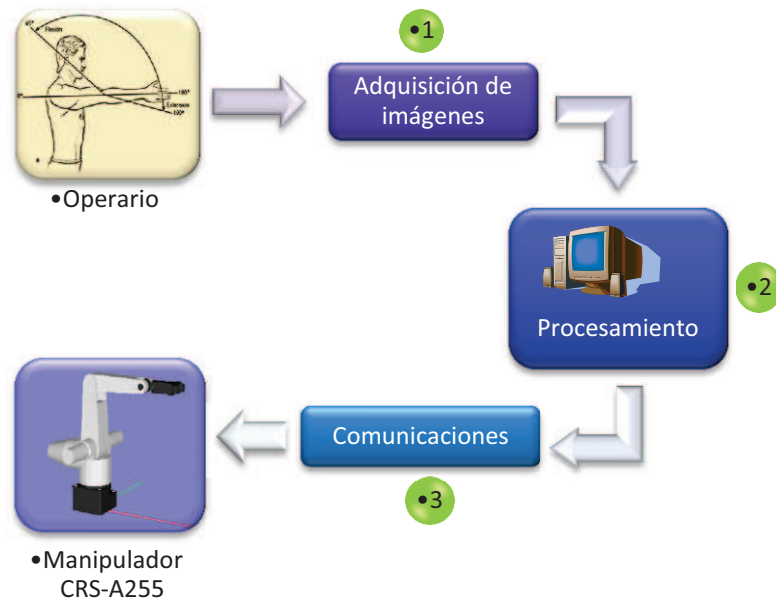
## CAPITULO 2

### MARCO TEÓRICO

#### 2.1 GENERALIDADES DEL SISTEMA

Como se mencionó en el Capítulo 1, el producto final de este proyecto es un sistema que permite a un manipulador robótico imitar los movimientos del brazo humano sin la intervención de controles mecánicos o el envío manual de comandos en una terminal.

Para cumplir con este objetivo se planteó la utilización de la visión artificial y en base a ello se procedió a dividir al sistema en tres etapas, como puede verse en la Figura 2.1.



**Figura 2.1 Elementos del sistema**

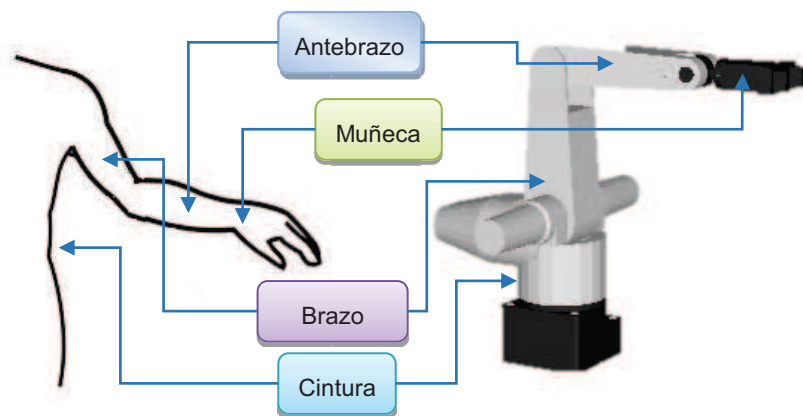
FUENTE: Elaboración propia

La etapa de adquisición de imágenes es el pilar del sistema pues en ella se obtienen los datos necesarios de movimiento que el robot debe realizar para imitar al brazo humano.

El procesamiento es la parte del sistema en la cual se aplica la técnica de esqueletización y se obtienen los ángulos que cada articulación debe moverse para asemejarse a la posición del brazo del operario en ese instante. Esta etapa se realiza en un computador externo.

Finalmente, el computador envía al manipulador los datos de los movimientos a realizar.

Es importante aclarar que, la forma en la cual se planteó controlar el movimiento del manipulador fue relacionando las articulaciones del mismo con su correspondiente articulación humana, por ejemplo: corresponder el giro de la cintura del operador con el giro de la cintura del manipulador y el movimiento de la muñeca humana con la muñeca del manipulador. Esta correspondencia se muestra en la Figura 2.2



**Figura 2.2 Correspondencia operador – manipulador CRS-A255**

FUENTE: Elaboración propia

Para cada una de las etapas se analizaron varias opciones, las cuales son presentadas a lo largo de este capítulo, así como los parámetros considerados para la elección de la alternativa a utilizar de acuerdo con los requerimientos del sistema, sus potenciales resultados y su rendimiento.

## **2.2 REQUERIMIENTOS DE LA ETAPA DE ADQUISICIÓN DE IMÁGENES**

En toda aplicación que utilice técnicas de procesamiento de imágenes como la esqueletización, el elemento principal a considerar será el dispositivo con el cual la imagen es adquirida, pues de éste dependerá la resolución y profundidad de la misma así como su calidad de color.

Además, se debe considerar la aplicación, pues de acuerdo a ésta será necesario trabajar con una o varias imágenes del mismo objeto las cuales aporten datos en varios planos, es decir, posición en X y Y, y de ser necesario profundidad.

En el caso específico de este proyecto, los requisitos fundamentales que deben cumplir las imágenes son:

- Cubrir un campo de visión, lo suficientemente amplio para captar los movimientos del operador.
- Aportar los datos suficientes para el movimiento de las articulaciones del manipulador.
- Poseer un formato de fácil conversión para agilizar el procesamiento.
- Presentar una resolución suficiente para identificar los elementos en la imagen
- Mostrar colores análogos a los presentes en la escena del operador
- En cuanto al dispositivo: debe ser altamente portátil, poseer una interfaz de comunicación sencilla y adquirir las imágenes a gran velocidad.

De acuerdo con estos requisitos, se analizaron los posibles dispositivos a utilizar y la medida en que éstos generarían imágenes válidas para el sistema.

### **2.2.1 DISPOSITIVOS**

En la actualidad, el mercado ofrece una variedad de cámaras de todo tipo y características, además de dispositivos especializados creados por la industria de

los videojuegos para lograr una interacción con el usuario por medio de imágenes y movimiento. Los dispositivos más utilizados y disponibles en el mercado son:

- Cámaras web
- Cámaras estereoscópicas
- Sensor Kinect

### 2.2.1.1 Cámaras Web

La primera alternativa planteada para incluir como dispositivo para la adquisición de imágenes fue la cámara web (Figura 2.3), la cual es utilizada normalmente para enviar imágenes a páginas en internet o aplicaciones para chat y videoconferencias, a través de una conexión RS-232 o USB.



**Figura 2.3 Cámara Web Logitech.**

FUENTE: Megahard (2010). Productos. Recuperado de:  
[http://www.megahardpcs.com.ar/index.php?p=1\\_2\\_Productos](http://www.megahardpcs.com.ar/index.php?p=1_2_Productos)

Esta cámara adquiere la luz de la imagen a través de un único lente frontal; la misma atraviesa un filtro RGB que descompone la luz en tres colores: rojo, verde y azul. Posteriormente, estos rayos atraviesan un chip sensible a la luz el cual los codifica como una secuencia de 0's y 1's; éstos son enviados al computador e interpretados por el software adecuado como una imagen.

Los chips sensibles a la luz utilizados en estas cámaras pueden ser dispositivos de carga acoplada (CCD) o un chip semiconductor complementario de metal-óxido (CMOS). Las dos tecnologías funcionan a base de una matriz de celdas fotosensibles, sin embargo en los CCD la información de cada una de las celdas es enviada a través del chip hacia una esquina del arreglo y codificada mediante

un conversor análogo-digital, mientras en los sensores fotosensibles tipo CMOS la información se codifica en cada celda, eliminando la circuitería adicional.<sup>2</sup>

El principal motivo para considerar las cámaras web como alternativa es que fueron concebidas para transmitir imágenes “en tiempo real”. En realidad, estas cámaras obtienen fotogramas de 15 a 30 veces por segundo por lo cual la imagen se refresca permanentemente dando la ilusión de movimiento continuo, pero con una resolución y calidad de color menor a otras cámaras digitales existentes.

### 2.2.1.2 Cámaras estereoscópicas



**Figura 2.4 Cámara estereoscópica 120 tri-lens stereo**

FUENTE: Ouna. (2007). Cámara digital 3D de 3D World: 120 Tri-lensStereo [Artículo].

Recuperado de: <http://ounae.com/camara-digital-3d-de-3d-world-120-tri-lens-stereo/>

La segunda alternativa considerada es la cámara estereoscópica (Figura 2.4), un tipo especial de cámara con dos o más lentes que capturan un fotograma, cada uno, simultáneamente, simulando la visión binocular humana y haciendo posible la obtención de imágenes con efectos tridimensionales, conocida como fotografía estereoscópica.

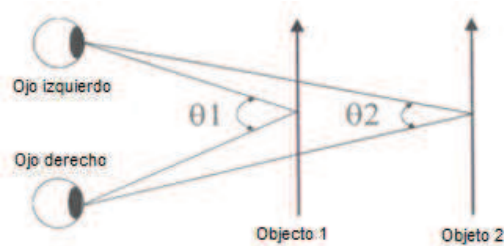
Las cámaras estereoscópicas utilizan, como se mencionó anteriormente, el mismo método de la visión humana al formar la sensación de profundidad a partir de imágenes tomadas desde dos posiciones, por lo cual se analizó aplicarlas al proyecto ya que facilitarían la adquisición de la posición del brazo humano tridimensionalmente. Esto es posible en estas cámaras gracias a dos cualidades:

- Disparidad retinal



- Paralaje estereoscópico

Al observar un objeto, el ser humano es capaz de captar la proximidad o lejanía del mismo debido a que la imagen formada en la retina derecha e izquierda no es la misma, el objeto se encuentra a una distancia horizontal diferente en cada una, fenómeno conocido como **disparidad retinal**. El cerebro humano toma las imágenes formadas en la retina derecha e izquierda y las interpreta como un espacio tridimensional; mientras mayor es la diferencia en un punto de la imagen aquel punto se encontrará más cerca (Figura 2.5).



**Figura 2.5 Percepción de profundidad por disparidad retinal.**

FUENTE: Bustacara, C. (2010). Estereoscopia o visión 3D [Documento]. Pontificia Universidad Javeriana. Recuperado de: <http://sophia.javeriana.edu.co/~cbustaca/Realidad%20Virtual/material/clase14/estereoscopia.pdf>

Por otro lado, el **paralaje estereoscópico** es el mecanismo con el cual se intenta reproducir la disparidad retinal, de forma que el observador obtenga imágenes tridimensionales a partir de dos imágenes bidimensionales. En este caso, la distancia horizontal de un punto se toma en el plano de proyección en lugar de la retina y el cálculo de la profundidad puede realizarse tomando puntos de referencia en la imagen.

Actualmente existen varios modelos de cámaras estereoscópicas, tanto para tomar fotogramas como para obtener imágenes en movimiento, sin embargo sus precios son muy elevados comparados con otras cámaras digitales y su presencia en el mercado es escasa.

### 2.2.1.3 Sensor Kinect

Desde su lanzamiento comercial, el sensor Kinect ha sido utilizado en diversos proyectos de investigación en el campo de la visión artificial y robótica, por lo cual también se lo consideró para este proyecto.

El sensor Kinect (Figura 2.6) es en realidad un conjunto de dispositivos como procesadores, cámaras, sensores infrarrojos y micrófonos, trabajando conjuntamente para obtener un modelo tridimensional del ambiente en el cual se encuentra, de tal forma que el usuario pueda interactuar con una máquina sin la necesidad de un mando tangible.



**Figura 2.6 Sensor Kinect para Xbox 360**

FUENTE: VARIOS (2011). Kinect para Windows a principios de año [Artículo]. Xbox Perú.

Recuperado de: <http://xboxperu.com/?p=1019>

Al igual que las cámaras estereoscópicas, Kinect fue considerado en este proyecto por su capacidad para calcular profundidad. Esto lo realiza gracias a la combinación de una cámara infrarroja, una cámara VGA y mediante la utilización de la técnica conocida como codificación por luz (light coding), como se muestra en la Figura 2.7.

La codificación por luz es una técnica en la cual se emiten permanentemente haces de luz infrarroja, los cuales generan un patrón puntos en el espacio, los cuales rebotan sobre el entorno y son recibidos por un sensor CMOS como haces de luz en mayor o menor grado de color dependiendo de la distancia al objeto encontrado. Este patrón de colores es analizado dando como resultado una imagen de profundidad por fotograma en la que los objetos más alejados se presentan en color gris.



**Figura 2.7 Esquema del método de captura del movimiento de Kinect**

FUENTE: Imagen basada en: VARIOS. (2011). Kinect al traguadoo. Ed è già in pezzi! [Blog]. *Technology.it*. Recuperado de: <http://www.tech-nology.it/web-news/kinect-al-traguadoo-ed-e-gia-in-pezzi/>

A pesar de que Kinect es capaz de calcular profundidad, su capacidad es limitada respecto a las cámaras estereoscópicas, pues el rango de detección del sensor infrarrojo es muy corto.

De acuerdo a lo presentado, se puede apreciar que tanto la resolución como la calidad de color de Kinect es similar a la de las cámaras web, su funcionalidad semejante a las cámaras estereoscópicas, y su precio está en un nivel intermedio entre las dos.

## 2.2.2 SELECCIÓN DE DISPOSITIVOS

Una vez presentados los posibles dispositivos para la adquisición de imágenes, éstos fueron evaluados de acuerdo a los parámetros establecidos anteriormente. Los resultados de esta evaluación se muestran en la Tabla 2.1.

**Tabla 2.1 Evaluación de dispositivos**

<b>Parámetro</b>	<b>Cámara Web</b>	<b>Cámara Estereoscópica</b>	<b>Kinect</b>
<b>Campo de visión</b>	Medio	Mayor	Medio
<b>Aporta todos los datos de movimiento necesarios</b>	Con 2 o más dispositivos	Basado en 2 imágenes	Sensor IR
<b>Formato de imagen fácilmente manejable</b>	✓		✓
<b>Resolución</b>	Media	Mayor	Media
<b>Calidad de color</b>	Media	Media	Media
<b>Dispositivo portátil</b>	✓	✓	✓
<b>Interfaz de comunicación sencilla</b>	✓		✓
<b>Adquisición de imágenes a gran velocidad</b>	✓		✓
<b>Costo</b>	Menor	Mayor	Medio

FUENTE: Elaboración propia

Como se puede apreciar, la cámara web y el sensor Kinect son los dispositivos que más se ajustan a los requerimientos del sistema, pues sus características son similares.

Debido al planteamiento para el control del manipulador robótico, la cámara web resultó ser el dispositivo más adecuado para ser incluido en el sistema pues las cámaras estereoscópicas y el sensor Kinect se ajustan mejor a un control por coordenadas del punto final del manipulador y no a un control articulación a

articulación, por su funcionalidad de cálculo de profundidad propio de aquellos dispositivos.

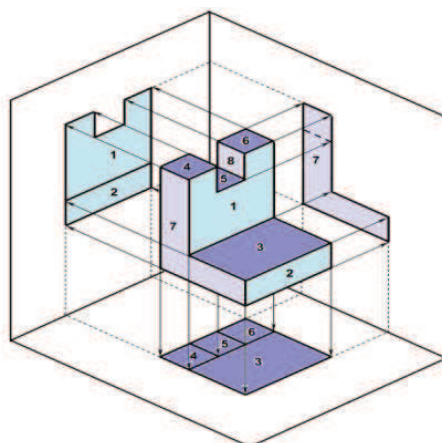
Al trabajar con una cámara web todos los datos son obtenidos a partir de las imágenes; adicionalmente, el costo de los dispositivos y la facilidad de manejo y comunicación fueron importantes factores de decisión.

Como se mencionó anteriormente, las cámaras web no son capaces de obtener por si solas datos de profundidad, sin embargo se analizaron varias configuraciones, las cuales hacen posible la adquisición de todos los datos necesarios con la introducción de una o dos cámaras extras.

Debido a que se requiere capturar el movimiento de la cintura del operador así como el de su brazo, se establecieron distintas configuraciones para lograr este objetivo con cámaras convencionales, las cuales son:

- Dos o más cámaras paralelas
- Cámara móvil
- Cámara con juego de espejos
- Dos o más cámaras en planos mutuamente perpendiculares

De estas opciones, la configuración que más se ajusta a los requerimientos del sistema es la de **cámaras ubicadas en planos mutuamente perpendiculares**, como se muestra en la Figura 2.8.



**Figura 2.8 Proyección Ortogonal.**

FUENTE: CASTELA, P. (2010). Vistas de un objeto [Documento]. España. Recuperado de:  
<http://www.pcastela.es/documentos/dibujo/Vistas.pdf>

Esta técnica es utilizada por el dibujo técnico para representar un objeto tridimensional mediante sus proyecciones en planos mutuamente perpendiculares

Esta configuración presenta dos ventajas:

- Cada imagen contiene la ubicación de un punto con relación a dos ejes distintos, es decir, entre las dos imágenes se tienen las coordenadas del punto en los tres ejes.
- Las imágenes pueden ser utilizadas de forma independiente y pueden detectarse detalles desde otro punto de vista.

Orientada al proyecto, esta configuración permite la adquisición de dos imágenes distintas: una desde una vista superior con la cual se obtiene el movimiento de la cintura; y, otra con una vista frontal del usuario en la que se captura el brazo completo.

## **2.3 REQUERIMIENTOS DE LA ETAPA DE PROCESAMIENTO**

Una vez seleccionado el dispositivo para adquirir las imágenes, se procedió a estudiar la forma de procesamiento de las mismas.

En el apartado anterior se establecieron los motivos por los cuales el sistema trabaja con dos cámaras web, teniéndose como entrada dos imágenes: una con una vista superior del operario y otra con vista frontal. Éstas son la base de la cual se extraen los movimientos que debe realizar el manipulador, por ello se debe considerar los siguientes aspectos al momento de su procesamiento:

- Se debe obtener los movimientos del operario, enfocándose a las articulaciones deseadas, es decir posición del brazo, antebrazo, muñeca y cintura
- La posición de las articulaciones debe mantenerse a lo largo del procesamiento de manera que puedan extraerse fácilmente los datos para el movimiento del manipulador robótico, es decir se debe mantener la geometría presente en la imagen, así como su topología.
- La velocidad de procesamiento de las imágenes debe ser alta, pues es necesario un control del manipulador en tiempo real.

- El sistema debe ser capaz de ejecutarse desde un computador personal, por lo que el consumo de recursos en la etapa de procesamiento debe ser moderado.
- La información arrojada durante el procesamiento de la imagen debe estar acorde con la morfología del manipulador robótico.

La técnica contemplada en este proyecto, como parte del procesamiento de la imagen, es la esqueletización. A continuación se analizará su concepto y sus Métodos de implementación más conocidos, lo cual permitió definir la estrategia a seguir en esta etapa del proyecto.

### 2.3.1 ESQUELETIZACIÓN

A medida que los dispositivos fotográficos incrementaron la resolución de las imágenes capturadas, fue necesario buscar formas viables de comprimir la información de las mismas. Una de estas formas se basa en la extracción de las características relevantes de la imagen, adelgazándola hasta obtener un patrón binario con menor cantidad de pixeles representativos, formando líneas, denominada esqueletización.

#### 2.3.1.1 Definición

La esqueletización es una técnica que reduce una imagen a un conjunto de líneas y curvas conectadas, de un pixel de grosor, las cuales mantienen la geometría y topología de la imagen original. El patrón arrojado como resultado la aplicación de esta técnica se ha denominado esqueleto, como muestra la Figura 2.9.

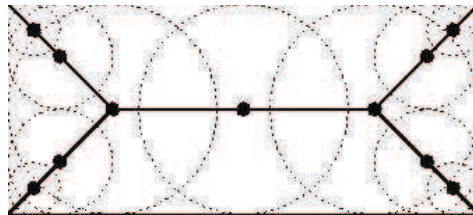


### Figura 2.9 (a) Patrón original. (b) Esqueleto obtenido

FUENTE: GALLEGOS, G. (2005), Convergencia multi-reticular de propiedades de los esqueletos continuos y discretos generados mediante adelgazamiento [Disertación de tesis]. Centro De Investigación y de Estudios Avanzados Del Instituto Politécnico Nacional. México.

Varios autores han desarrollado conceptos específicos del término “esqueleto”:

- “.. una representación de un patrón través de una colección de arcos y curvas finas.”
- “..una serie de líneas de un pixel de grosor que pasan por el centro de la región estudiada”<sup>3</sup>
- “..el lugar donde convergen los centros de círculos bi-tangenciales que entran de manera completa dentro de la región considerada”, como se muestra en la Figura 2.10.



### Figura 2.10 El esqueleto de un rectángulo definido en términos de círculos bi-tangenciales.

FUENTE: FISHER, R., PERKINS, S., WALKER, A., WOLFART, E. (2004). Skeletonization/Medial Axis Transform [Artículo]. *Image Processing Learning Resources, HIPR2 Group*. Recuperado de <http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm>

#### 2.3.1.2 Historia

La técnica de esqueletización se fundamenta en el adelgazamiento de las imágenes, tiene su origen en la década de 1950. El primer paso hacia a esqueletización se produjo en el año 1955 cuando Gerald Dineen afirma que al calcular la media sobre una ventana de píxeles con un nivel de umbral muy alto, el resultado es una imagen más delgada.<sup>4</sup>



El primer algoritmo formalmente estructurado sobre adelgazamiento fue presentado en 1966 por Rutovitz. Consistía en un procesamiento paralelo de los píxeles que formaban parte de la imagen y los marcaba como eliminables tras cumplir con ciertas reglas; sin embargo el resultado arrojado por este algoritmo no preservaba la topología del patrón original y en casos específicos eliminaba todo el patrón.<sup>5</sup>

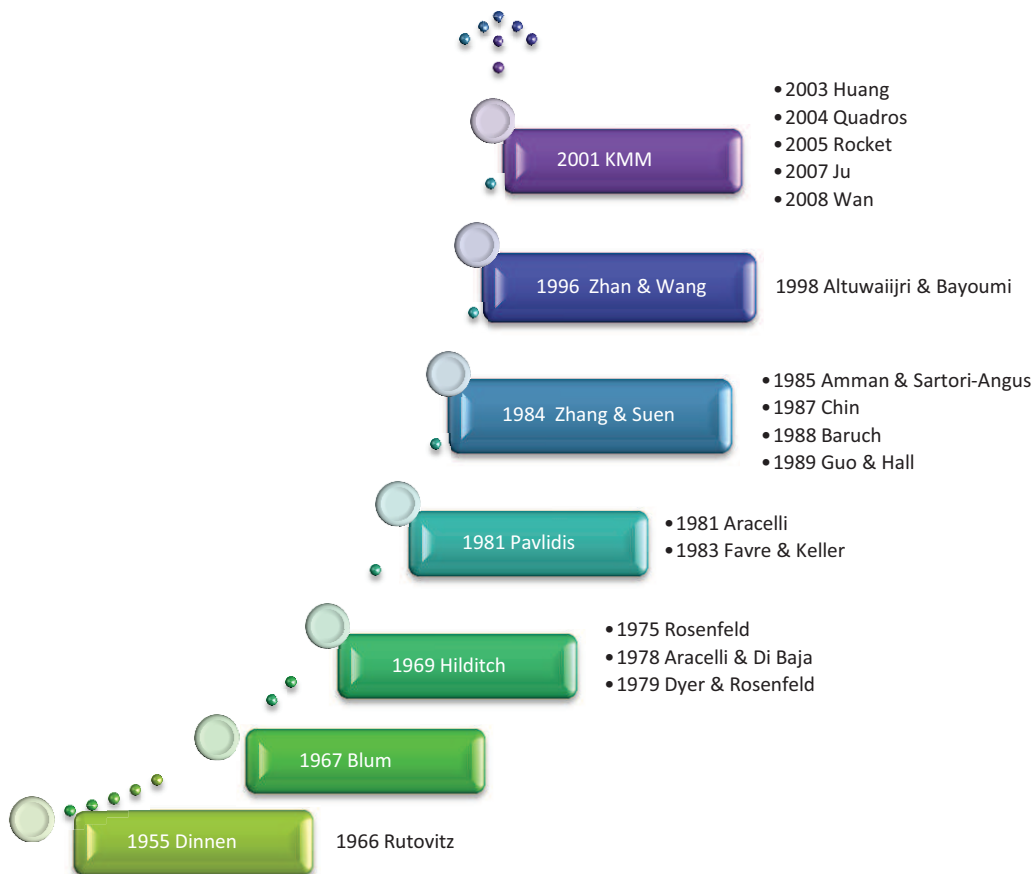
En 1969, Hilditch propone una metodología para obtener el esqueleto de una imagen, con base en la definición presentada en 1967 por Harry Blum. Este método consistía en un análisis paralelo-secuencial de un grupo de píxeles de la imagen, pero al igual que el caso anterior, en determinadas configuraciones eliminaba todos los píxeles de la imagen.

Pavlidis, en 1981, estructura uno de los algoritmos más conocidos para la obtención del esqueleto de una imagen que consiste en un análisis paralelo de los píxeles a partir del borde de la imagen y la comparación de éstos con plantillas específicas. Este algoritmo provee resultados válidos y el esqueleto conserva la topología original.

En 1984, Zhang y Suen proponen un algoritmo de esqueletización basado en el algoritmo de Hilditch, cuya implementación es mucho más sencilla y consta de dos sub-iteraciones por cada iteración, lo cual mejora los resultados en el esqueleto obtenido.

Todos los algoritmos mencionados fueron desarrollados sobre patrones fijos y en la actualidad han sido adaptados para procesar imágenes en movimiento y conseguir una esqueletización en tiempo real e incluso esqueletos en tres dimensiones.

Una representación cronológica de los trabajos en esqueletización y adelgazamiento de imágenes se muestra en la Figura 2.11.



**Figura 2.11 Línea de tiempo de autores de trabajos sobre esqueletización**

FUENTE: Elaboración propia en base a: SAEED, K., TABE, DZKI, M., RYBNIK, M. & ADAMSKI, M. (2010). K3M: A Universal Algorithm for Image Skeletonization and Review of Thinning Techniques. Int. J. Appl. Math. Comput. Sci. Volumen 20 (2) 317-335. DOI: 10.2478/v10006-010-0024-4

### 2.3.1.3 Aplicaciones

Por las ventajas que ofrecen los algoritmos de esqueletización para la compresión de datos y la extracción de características de las imágenes, esta técnica ha tenido gran acogida en varias áreas, siendo las principales aplicaciones:

- Reconocimiento de caracteres (OCR)
- Verificación de firmas y textos manuscritos
- Identificación de huellas dactilares
- Reconocimiento de células en cultivos biológicos

- Detección de aneurismas y obstrucciones en vías circulatorias
- Interacción por movimiento

#### **2.3.1.4 Limitaciones**

Los diferentes Métodos de esqueletización afrontan tres inconvenientes críticos en lo referente al esqueleto resultante:

- La conectividad de la región, de manera que se preserve la topología de la imagen original<sup>6</sup>
- La preservación de los puntos finales, para obtener la mayor semejanza geométrica posible
- Cada componente del esqueleto debe estar centrado respecto a la región a la cual representa.

La solución a estos inconvenientes no ha sido encontrada pues hasta el momento no se ha desarrollado un algoritmo de esqueletización completamente exacto; sin embargo se están utilizando técnicas auxiliares para tratar la imagen previa la esqueletización, por ejemplo la aplicación de umbrales, de manera que la imagen binaria conserve sus características principales después de la aplicación del algoritmo.

Otro inconveniente de la esqueletización es el rendimiento presentado por los diferentes algoritmos. Debido a que la esqueletización se realiza de manera iterativa, se requiere un tiempo de procesamiento directamente proporcional a la resolución de la imagen, es decir, mientras mejor sea la resolución de la imagen de entrada el algoritmo tardará más en obtener el esqueleto de la misma, lo cual representa una gran desventaja cuando se tratan imágenes en movimiento si no se cuenta con un procesador de altas prestaciones.

Por estas razones la esqueletización fue utilizada mayormente para trabajar con patrones fijos y de mediana resolución, aplicaciones en las cuales el rendimiento no es trascendental. No obstante, a partir 2009 la técnica de esqueletización ha tomado fuerza por la similitud con la forma de interacción con el usuario del sensor Kinect de Microsoft, que permite trabajar con imágenes en movimiento

para su aplicación en videojuegos basándose en el movimiento del cuerpo humano, sin necesidad de controles.

Actualmente se están buscando formas más óptimas para obtener el esqueleto de una imagen e incluso Métodos de esqueletización para cuerpos en 3 dimensiones.

### 2.3.2 ALGORITMOS DE ESQUELETIZACIÓN

Por las ventajas que ofrece la esqueletización como etapa previa al procesamiento e imágenes, se han planteado varios algoritmos para la obtención de esqueletos de una imagen. Estos algoritmos se han clasificado en dos categorías:

- Secuenciales
- Paralelos

Los **algoritmos secuenciales** inician su análisis con un punto del borde del patrón y en base a él buscan, de forma ordenada, puntos contiguos sobre el mismo para compararlos con plantillas y definir si el pixel es borrado o se mantiene; el proceso se realiza iteración tras iteración hasta obtener líneas y curvas delgadas que representen al patrón original. Este tipo de algoritmos requieren alto tiempo de procesamiento y un método de seguimiento de contorno para su óptimo funcionamiento.

En lo referente a los **algoritmos paralelos**, éstos analizan simultáneamente cada píxel de la imagen, y de acuerdo al cumplimiento de determinadas reglas, sean eliminados o conservados en cada iteración. Debido al procesamiento simultáneo pueden eliminarse erróneamente pixeles esenciales si no se ha asignado un orden de borrado determinado. Este tipo de algoritmos tiene como ventaja la velocidad de procesamiento y el trabajo en forma independiente.

Tanto los algoritmos paralelos como secuenciales necesitan varias iteraciones para alcanzar una esqueletización completa y los resultados obtenidos son diferentes en cada caso.

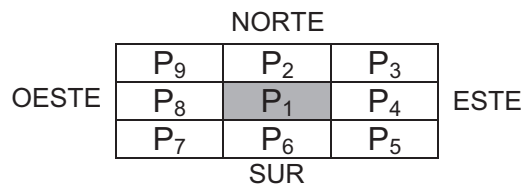
A continuación se describen los algoritmos de esqueletización que fueron considerados para la implementación del proyecto:

- Algoritmo paralelo de Zhang-Suen
- Algoritmo secuencial-paralelo de Pavlidis
- Algoritmo de post-procesamiento de Holt

### 2.3.2.1 Algoritmo Zhang-Suen<sup>7</sup>

Dentro de los algoritmos de esqueletización, uno de los más conocidos es el presentado por T. Y. Zhang y C. Y. Suen en el artículo “A fast parallel algorithm for thinning digital patterns”.

Este algoritmo toma una imagen binaria<sup>a</sup> y la analiza en ventanas de 3x3 pixeles para cada iteración (Figura 2.12), ejecutando realmente dos sub-iteraciones por cada una, con lo cual se eliminan todos los pixeles del contorno de la imagen o patrón, excepto aquellos que son parte del esqueleto.



**Figura 2.12 Conjunto de 3x3 pixeles para aplicación del algoritmo Zhang-Suen**

FUENTE: Elaboración propia basada en: ZHANG, T., SUEN, C. (1984). A fast parallel algorithm for thinning digital patterns [Artículo]. Communications of the ACM, volumen 27, número 3. Estados Unidos.

El píxel P<sub>1</sub> es eliminado si cumple con las siguientes condiciones:

1.  $2 \leq B(P_1) \leq 6$

---

<sup>a</sup> *Imagen binaria* es una matriz cuyos elementos pueden tener el valor de 1 o 0 solamente, donde los pixeles que forman el patrón tienen el valor de 1.

2.  $A(P_1) = 1$
3.  $P_2 * P_4 * P_6 = 0$
4.  $P_4 * P_6 * P_8 = 0$

Para la segunda sub-iteración solo se modifican las dos últimas condiciones:

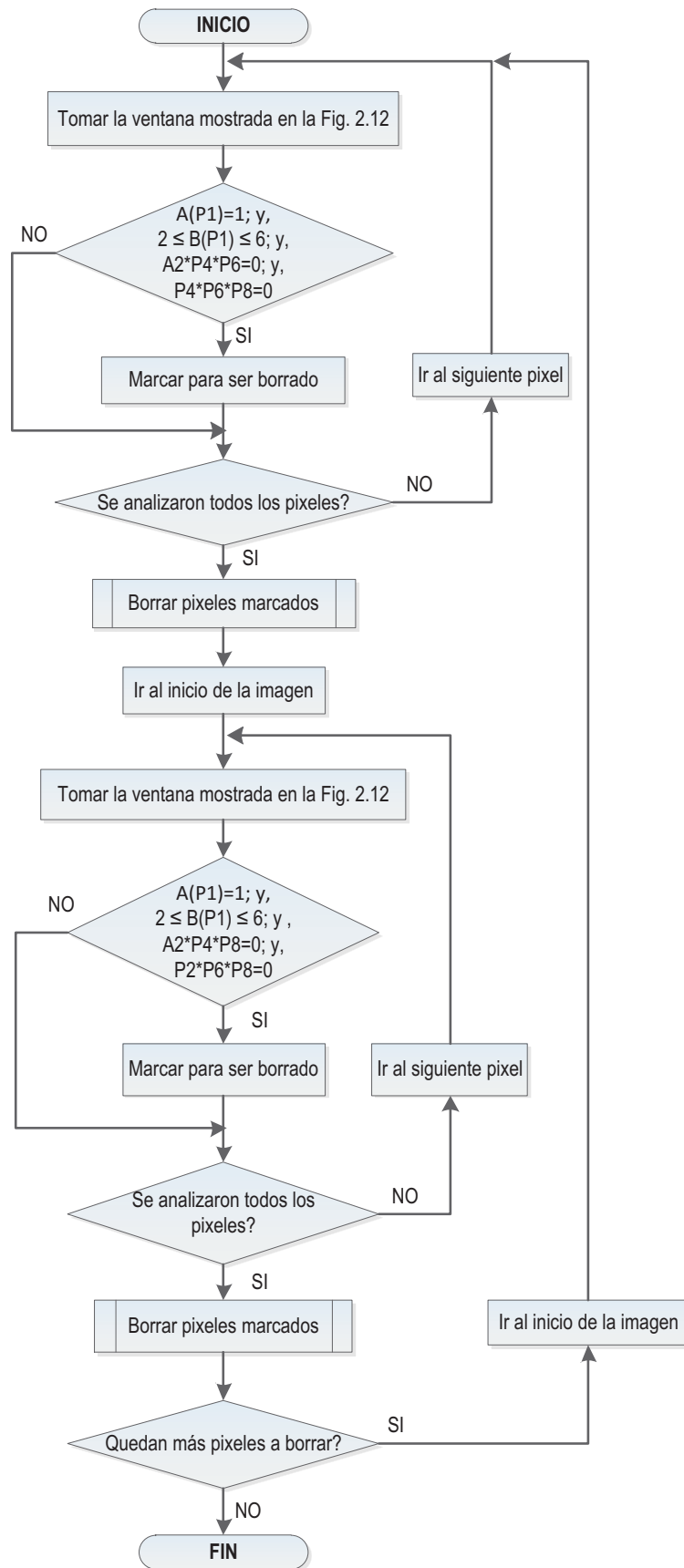
- 3'.  $P_2 * P_4 * P_8 = 0$
- 4'.  $P_2 * P_6 * P_8 = 0$

La primera condición exige que el número de vecinos del pixel  $P_1$  cuyo valor sea 1 esté entre 2 y 6, con lo cual el algoritmo asegura la permanencia de los pixeles que sean puntos finales o centrales de la imagen.

La segunda condición requiere que desde el pixel  $P_2$  hasta  $P_9$  se produzca el cambio de un pixel con valor 0 a un pixel con valor 1 únicamente una vez.

La tercera y cuarta condición en la primera sub-iteración tienen solución cuando  $P_4=0$ ,  $P_6=0$  o bien  $P_2$  o  $P_8$  sean 0. Si  $P_4=0$  el pixel  $P_1$  es parte del límite Este de la imagen, si  $P_6=0$  es parte del límite sur; y, si  $P_2$  o  $P_8$  son 0 forma parte de la esquina Noroeste de la imagen. En la segunda sub-iteración, si  $P_2=0$  el pixel forma parte del límite norte, si  $P_8=0$  del límite oeste; y, si  $P_4$  o  $P_6$  son 0 forma parte de la esquina sureste.

El algoritmo de Zhang-Suen se muestra en la Figura 2.13.



**Figura 2.13 Diagrama de flujo del algoritmo Zhang-Suen**

FUENTE: Elaboración propia

Las dos sub-iteraciones son aplicadas en cada pixel de la imagen logrando así obtener el esqueleto de la misma al final del proceso.

### 2.3.2.2 Algoritmo de Pavlidis<sup>8</sup>

Theo Pavlidis propuso un algoritmo secuencial en su artículo “A thinning algorithm for discrete binary images” en 1980. El proceso toma como entrada una imagen binaria y analiza una ventana de 3x3 pixeles alrededor del mismo, numerándolos como se muestra en la Figura 2.14.

3	2	1
4	P	0
5	6	7

**Figura 2.14 Numeración de pixeles para algoritmo de Pavlidis**

FUENTE: DEL RAYO, M. (2004). Adelgazamiento en imágenes digitales de dimensión dos modeladas por complejos celulares cuadráticos [Disertación de tesis]. Centro De Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. México.

Los pixeles a tomar en cuenta son {0,2,4,6}, ya que si alguno de ellos tiene un valor de 0 el pixel p se puede considerar un pixel de borde.

La base del algoritmo son las plantillas mostradas en la Figura 2.15 y sus respectivas rotaciones de 90°. En las dos primeras plantillas, al menos un pixel marcado como A o B debe tener el valor de 1. Para la tercera plantilla por lo menos un pixel marcado como C debe ser tener el valor de 1.

A	A	A
0	p	0
B	B	B

A	A	A
A	P	0
A	0	2

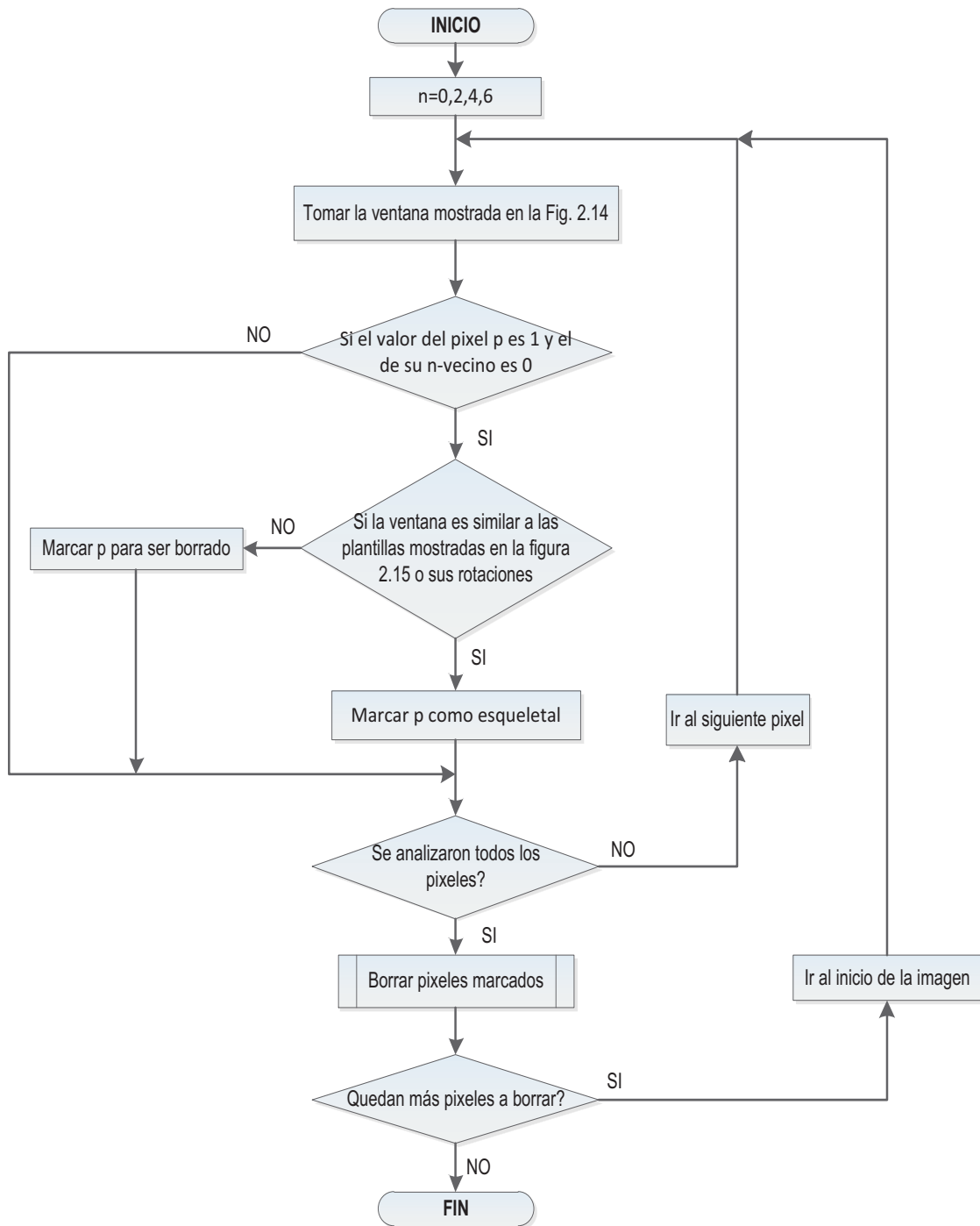
A	A	C
0	2	2
B	B	C

**Figura 2.15 Plantillas para aplicación de algoritmo de Pavlidis**

FUENTE: DEL RAYO, M. (2004). Adelgazamiento en imágenes digitales de dimensión dos modeladas por complejos celulares cuadrático [Disertación de tesis]. Centro De Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. México.



Los pasos necesarios para la implementación del algoritmo de Pavlidis se muestran en la Figura 2.16.

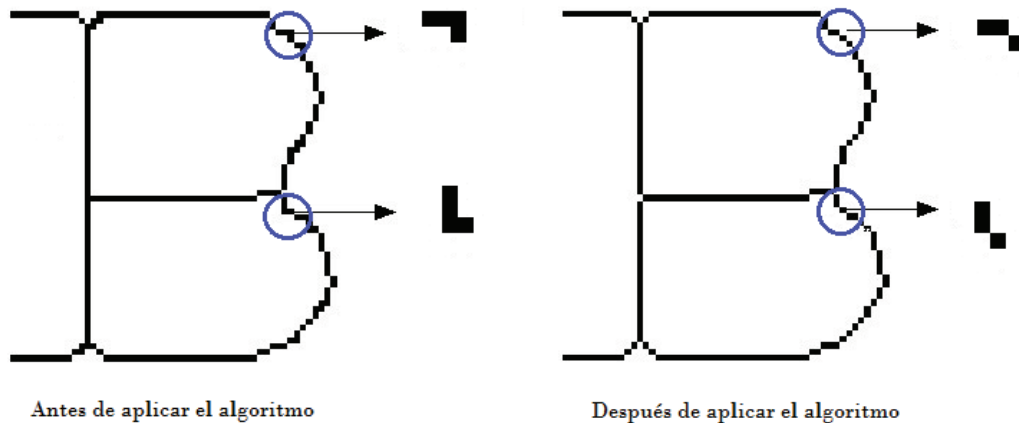


**Figura 2.16 Diagrama de flujo del Algoritmo de Pavlidis**

FUENTE: Elaboración propia

### 2.3.2.3 Algoritmo de Holt

Con el objetivo de obtener un esqueleto totalmente delgado se plantearon varios algoritmos denominados de post-procesamiento, los cuales tienen como meta la eliminación de ciertos píxeles que han quedado luego del proceso de esquelización; es decir, simplifican aún más los algunos patrones restantes, como se muestra en la Figura 2.17. Uno de estos algoritmos es el algoritmo planteado por Christopher Holt en 1987, en el artículo “An Improved Parallel Thinning Algorithm”, basado en las reglas de esquelización planteadas por Zhang y Suen.



**Figura 2.17 Ejemplo de defectos resueltos por el algoritmo de Holt**

FUENTE: FACON, J. (2002). Algoritmo de Afinamiento de Holt. Pontificia Universidade Católica do Paraná. Recuperado de: <http://www.ppgia.pucpr.br/~facon/Afinamento/AfinamentoHolt3.pdf>

Holt afirma que la mitad de los píxeles que forman una escalera pueden ser eliminados sin afectar la topología del patrón original, dividiendo las configuraciones posibles en dos casos: remoción de escalera con orientación norte y remoción de escalera con orientación sur, los cuales se muestran en la Figura 2.18. Al igual que en el algoritmo de Zhang-Suen, el análisis se realiza en una ventana de 3x3 píxeles.

0	1	x
1	1	x
x	x	0

x	1	0
x	1	1
0	x	x

0	x	x
X	1	1
X	1	0

x	x	0
1	1	x
0	1	x

(a) (b)

**Figura 2.18. Casos para el algoritmo de Holt. (a) Escalera con orientación norte (b) Escalera con orientación sur**

FUENTE: ANAK, A. (2005). Enhancement of parallel thinning algorithm for handwritten characters using neural network. Universiti Teknologi Malaysia. [Disertación de tesis] Recuperado de: <http://eprints.utm.my/3796/1/AdelineEngkamatMCD2005TTT.pdf>

Para cada caso se planteó una expresión, la cual al cumplirse asegura la permanencia del pixel analizado, caso contrario es borrado. La expresión para la remoción de escalera con orientación norte es:

$$v(C) \wedge \sim(v(N) \wedge ((v(E) \wedge \sim v(NE) \wedge \sim v(SO) \wedge (\sim v(O) \vee \sim v(S)) \vee (v(O) \wedge \sim v(NO) \wedge \sim v(SE) \wedge (\sim v(E) \vee \sim v(S)))))))$$

Donde:  $\sim$  simboliza no; la función  $v(C)$  da el valor de pixel (1 si es negro y 0 si es blanco); las letras N, E, S, O, NE, SE, SO, NO representan los pixeles en una dirección particular a partir del pixel centro  $C^9$ , como se aprecia en la Figura 2.19.

NO	N	NE
O	C	E
SO	S	SE

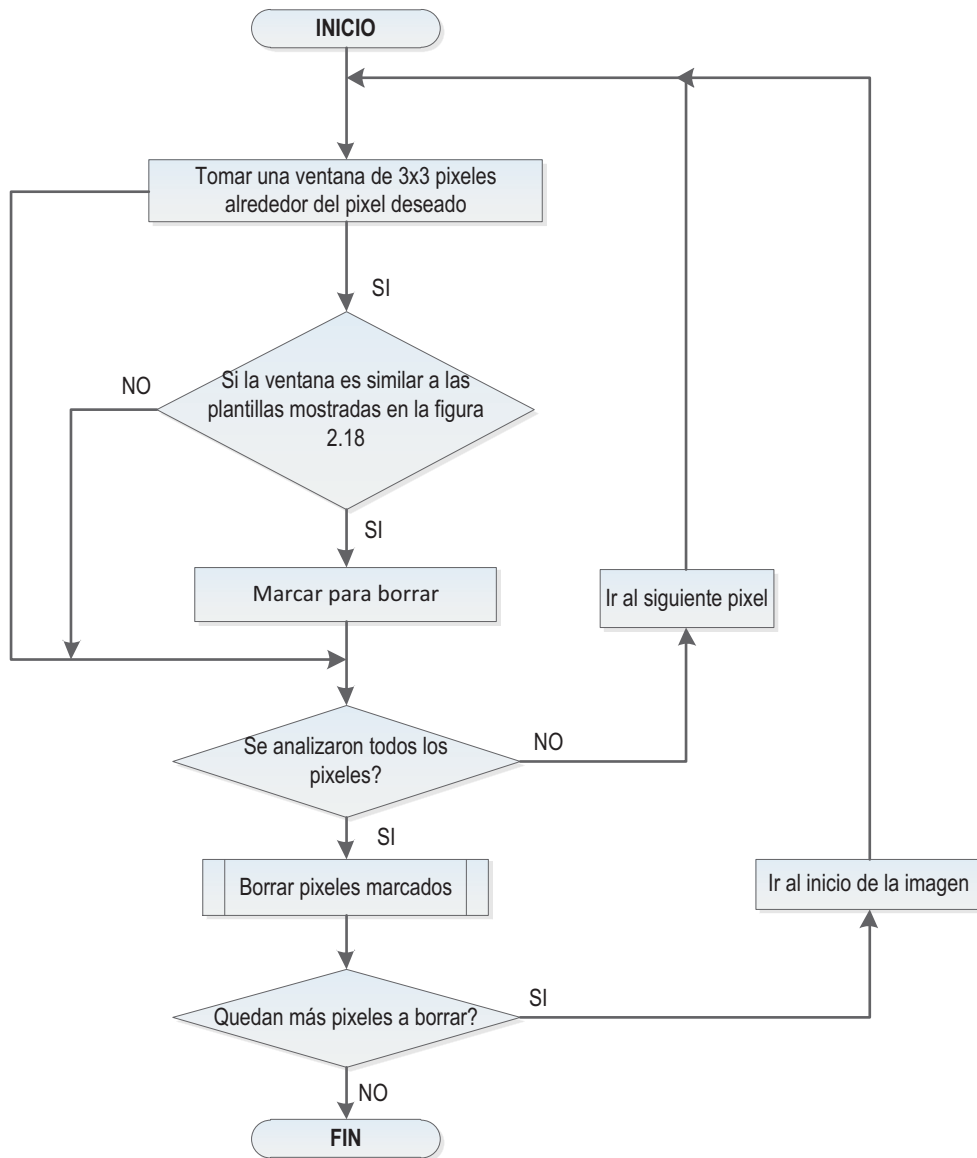
**Figura 2.19. Denominación de los pixeles para el algoritmo de Holt**

FUENTE: ANAK, A. (2005). Enhancement of parallel thinning algorithm for handwritten characters using neural network. Universiti Teknologi Malaysia. [Disertación de tesis] Recuperado de: <http://eprints.utm.my/3796/1/AdelineEngkamatMCD2005TTT.pdf>

La expresión para el caso de remoción de la escalera con orientación sur, es la misma que la de orientación norte con la única diferencia de intercambiar las letras N por S.

$$v(C) \wedge \sim(v(S) \wedge ((v(E) \wedge \sim v(NE) \wedge \sim v(SO) \wedge (\sim v(O) \vee \sim v(N)) \vee (v(O) \wedge \sim v(NO) \wedge \sim v(SE) \wedge (\sim v(E) \vee \sim v(N)))))))$$

El proceso que utiliza este algoritmo para eliminar los detalles en la imagen se muestra en la Figura 2.20.



**Figura 2.20 Diagrama de flujo del algoritmo de Holt**

FUENTE: Elaboración propia

Este algoritmo se ejecuta una sola vez en toda la imagen, la velocidad de procesamiento es mayor que la del algoritmo de Zhang-Suen y sus resultados arrojan esqueletos de un solo pixel de grosor y patrones simplificados, por lo cual es un algoritmo con gran aceptación.

### 2.3.3 SELECCIÓN DEL ALGORITMO DE ESQUELETIZACION

De acuerdo con los requerimientos del proyecto, se analizaron los algoritmos planteados, considerando que los algoritmos de Zhang-Suen y de Pavlidis actúan en la parte de pre-procesamiento, mientras el algoritmo de Holt constituye una herramienta de post-procesamiento de la imagen resultante de los dos anteriores.

La evaluación de estos algoritmos se muestra en la Tabla 2.2.

**Tabla 2.2 Evaluación de algoritmos de esqueletización**

<b>Parámetro</b>	<b>Zhang-Suen</b>	<b>Pavlidis</b>
<b>Tipo de ejecución</b>	Paralelo	Mixto (Secuencial-paralelo)
<b>Conservación de la geometría</b>	Buena	Muy buena
<b>Conservación de la topología</b>	Muy Buena	Muy buena
<b>Velocidad de procesamiento</b>	Mayor	Menor
<b>Consumo de recursos</b>	Menor	Mayor

FUENTE: Elaboración propia

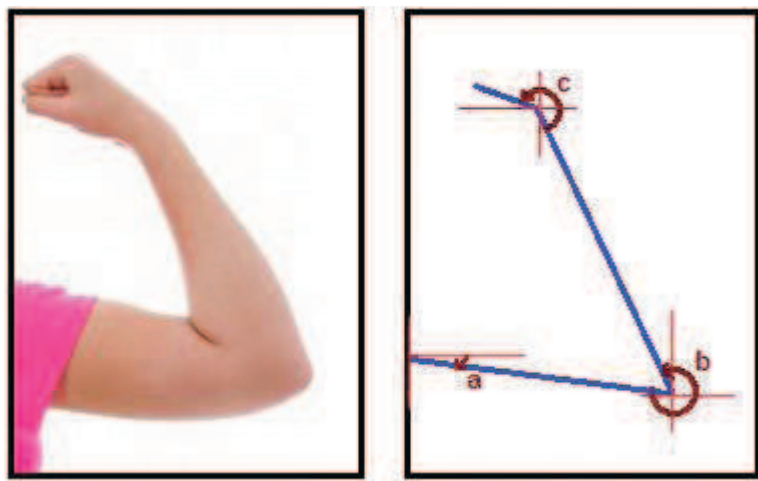
Como se puede apreciar, el algoritmo de Zhang-Suen se constituyó en la mejor opción para incluir en el sistema, pues el algoritmo de Pavlidis presenta una velocidad de procesamiento menor y mayor demanda de recursos debido a la utilización previa de un algoritmo de seguimiento de contornos, como se mencionó anteriormente. Cabe resaltar que en ambos casos, el algoritmo de Holt es una alternativa para definir mejor el esqueleto resultante, de forma que las

etapas posteriores sean realizadas de mejor manera, y su inclusión no altera de manera importante el tiempo de procesamiento.

Se debe tomar en cuenta el tamaño de las imágenes a esqueletizar pues es un factor determinante en la velocidad de procesamiento, por lo cual la mejor opción es escalar la imagen a un tamaño que permita la obtención de sus datos en tiempo real.

El esqueleto obtenido es el primer paso en el procesamiento de la imagen, luego se procederá a calcular los ángulos de inclinación de cada una de las articulaciones; y, por último se ajustarán estos ángulos para comunicarlos al manipulador robótico, el cual efectúa el movimiento finalmente.

La obtención de los datos de movimiento se la realiza en base a las líneas que forman los esqueletos de cada una de las imágenes, como se muestra en la Figura 2.21.



**Figura 2.21 Relación de ángulos entre brazo humano - esqueleto**

FUENTE: Elaboración propia

## 2.4 REQUERIMIENTOS DE LA ETAPA DE COMUNICACIONES

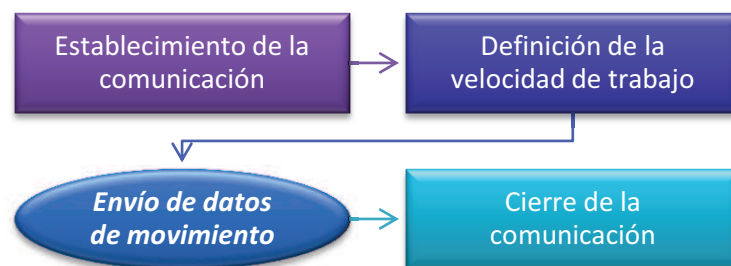
Una vez obtenidos los datos de movimiento para el manipulador, solo resta comunicarlos al manipulador para que éste ejecute dicho movimiento. Para lograr

esto, el sistema debe comunicarse de manera eficiente y veloz con el manipulador robótico, pues éste debe ser capaz de seguir el ritmo de movimiento hasta donde le sea posible al operar al 30% de su velocidad.

Debe considerarse que, el sistema debe enviar los datos de cada una de las articulaciones, pues debe imitar el movimiento completo y no únicamente alcanzar una posición determinada con el efector final, por esta razón se necesita enviar un comando por cada articulación a desplazar.

Asimismo, previo al envío de comandos el sistema debe establecer la comunicación con el manipulador, con el fin de inicializarlo y establecer la velocidad de su desplazamiento.

Las etapas importantes de la comunicación con el manipulador se muestran a continuación, en la Figura 2.22.



**Figura 2.22 Etapas trascendentes de la comunicación con el manipulador**

FUENTE: Elaboración propia

Para establecer la comunicación con el manipulador se ha planteado utilizar el puerto de comunicaciones propio de éste, el cual será analizado a profundidad en el Capítulo 4.

## **CAPITULO 3**

### **DESCRIPCIÓN FÍSICA DEL SISTEMA**

Como ya se ha mencionado en el Capítulo 2.1, el sistema planteado en este proyecto tiene como objetivo lograr que el manipulador robótico CRS-A255 imite los movimientos de un operador sin intervención de mandos externos, mediante dispositivos para adquisición de imágenes, un equipo para procesamiento y una interfaz de comunicación.

En este capítulo se detallan los dispositivos a utilizar en cada etapa del sistema así como su distribución en el área de trabajo. La Descripción de la plataforma se realizará en cuatro partes:

- Sistema de video
- Iluminación
- Unidad de procesamiento
- Brazo manipulador

Al final del capítulo se presentan las consideraciones físicas necesarias para la integración del sistema.

#### **3.1 COMPONENTES FÍSICOS DEL SISTEMA**

En la Figura 3.1 se puede apreciar un esquema general de los elementos que componen el sistema, los cuales son descritos a lo largo este capítulo.





**Figura 3.1 Hardware del sistema Hardware**

FUENTE: Elaboración propia

### 3.1.1 SISTEMA DE VIDEO

Por los motivos expuestos en el Capítulo 2.2.2, la cámara web es el dispositivo que más se ajusta a los requerimientos del sistema.

Actualmente existen en el mercado un buen número de cámaras USB con diferentes resoluciones, y velocidad de captura de imágenes. Sin embargo la resolución máxima de la cámara solo puede ser apreciada en imágenes estáticas, muchas veces capturadas por un software específico del fabricante, mientras las imágenes en tiempo real tienen una resolución universal de 640x480 pixeles, por lo cual la mayor parte de las cámaras web comerciales son aplicables a este sistema.

La cámara web utilizada en este proyecto posee las características mostradas en la Tabla 3.1.

**Tabla 3.1 Especificaciones WebCam Omega**

<b>Característica</b>	<b>Valor</b>
<b>Marca/Modelo</b>	Omega 361148
<b>Tipo de sensor</b>	CMOS
<b>Resolución en tiempo real</b>	640x480 pixeles
<b>Velocidad de adquisición de imágenes</b>	30 fps <sup>b</sup>
<b>Interfaz de comunicaciones</b>	USB 2.0
<b>Administración de color</b>	RGB
	

FUENTE: Elaboración propia basada en: VARIOS (2011). Ítem No. 361148 [Catálogo de productos]. *WebCams*. Recuperado de: <http://www.omega.com.tw/webcam-361148.asp#>

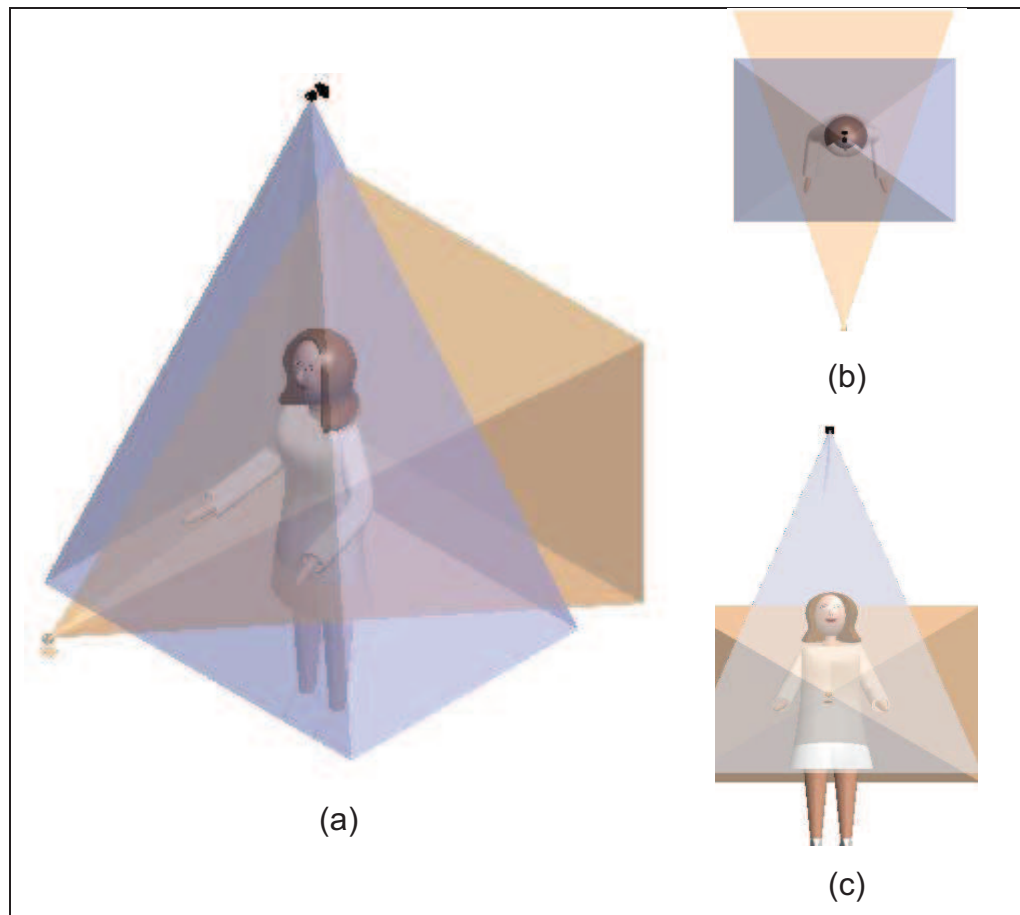
La ventaja de este modelo es la facilidad para rotar libremente sobre su propia base, además maneja un controlador genérico y su costo es bastante accesible.

### **3.1.1.1 Distribución de las Cámaras**

---

<sup>b</sup> fps: de las siglas en inglés *frames per second*, cuadros por segundo

De acuerdo a lo establecido en el Capítulo 2.2.2, las cámaras se ubicarán de tal forma que se obtenga una vista superior y una frontal del operador, como se observa en la Figura 3.2 (a).



**Figura 3.2 Esquema de ubicación de las cámaras**  
**(a) Vista Isométrica, (b) Vista superior, (c) Vista frontal**

FUENTE: Elaboración propia

La cámara superior captura el movimiento de la cintura; sin embargo, ya que no es posible capturar el giro de ésta por la técnica de esqueletización y al nivel mismo de la cintura, se utiliza como referencia la región supraclavicular para detectar este movimiento (Figura 3.2 (b)). Esta región está comprendida entre el cuello y la articulación del hombro, puede distinguirse claramente desde una vista superior y refleja exactamente el movimiento realizado por la cintura.

La posición de las partes restantes del miembro superior reflejadas en esta vista no es considerada para el sistema pues el manipulador robótico no posee las mismas características de movimiento del brazo humano, por lo que estos datos no son relevantes.

Por otro lado, la cámara frontal provee una visión del brazo, el antebrazo y la mano dentro de la capacidad de movimiento del manipulador robótico (Figura 3.2 (c)). De esta forma pueden calcularse los ángulos de cada porción del miembro superior con su porción adyacente y el plano horizontal.

Pese a que se ha definido la ubicación general de cada cámara, también es necesario definir el área a cubrir por cada cámara, para lo cual es necesario especificar la distancia desde el operador a la cámara. Esto puede obtenerse a partir de los ángulos de visión del modelo de cámara web usada.

#### 3.1.1.1.1 Ángulos de Visión

“El ángulo de visión es el arco en grados que percibe la cámara de un entorno real de 360 grados.”<sup>10</sup> Generalmente se proveen tres ángulos básicos de visión:

- Horizontal
- Vertical
- Diagonal

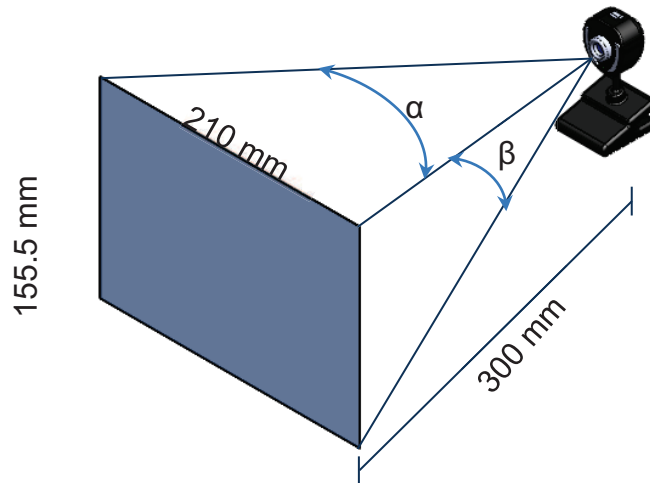
La relación existente entre el ángulo de visión, el campo de visión y la distancia a la cual se encuentra la cámara, se muestra en la Ecuación 3.1.

$$G = 2 \times d \times \tan(\alpha / 2) \qquad \text{Ecuación 3.1}$$

Donde:

- G= campo de visión
- d= distancia
- $\alpha$ = ángulo de visión

Para el modelo específico de cámara web utilizado en este proyecto, el fabricante no provee estos datos, por lo cual se los calculó de forma experimental a partir de la configuración mostrada en la Figura 3.3, utilizando una distancia conocida y midiendo el campo de visión cubierto.



**Figura 3.3 Cálculo de los ángulos de visión cámara web**

FUENTE: Elaboración propia

Se calcularon los ángulos de visión horizontal ( $\alpha$ ) y vertical ( $\beta$ ), obteniéndose:

$$\alpha = 2 \times \text{Tan}^{-1} \left( \frac{G}{2 \times d} \right) \quad \text{Ecuación 3.2}$$

$$\alpha = 2 \times \text{Tan}^{-1} \left( \frac{210 \text{ mm}}{2 \times 300 \text{ mm}} \right) = 38.58^\circ$$

$$\beta = 2 \times \text{Tan}^{-1} \left( \frac{155 \text{ mm}}{2 \times 300 \text{ mm}} \right) = 28.97^\circ$$

Con los resultados arrojados, es posible modificar las condiciones y calcular el campo de visión a cualquier distancia o viceversa.

#### 3.1.1.1.2 Distancia de Captura

Para este proyecto en específico, la distancia de captura más conflictiva es la correspondiente a la cámara frontal ya que todo el movimiento del miembro superior debe ser apreciado en las imágenes obtenidas por ella.

Con el afán de calcular esta distancia, se tomó en cuenta que, según los cánones de la antropometría, “la longitud de los brazos extendidos de una persona es aproximadamente igual a su altura”<sup>11</sup>, por lo cual se tomó como aproximación válida:

$$\text{Longitud de un brazo} \approx \frac{\text{Altura de la persona}}{2} \quad \text{Ecuación 3.3}$$

Para el sistema es esencial captar por completo la imagen del miembro superior extendido, es decir, el campo de visión relacionado al ángulo de visión horizontal debe ser igual o superior a la expresión expuesta en la Ecuación 3.3, cuya introducción en la Ecuación 3.1, arroja:

$$d \geq \frac{\text{Altura de la persona}}{4 \times \tan(\alpha/2)} \quad \text{Ecuación 3.4}$$

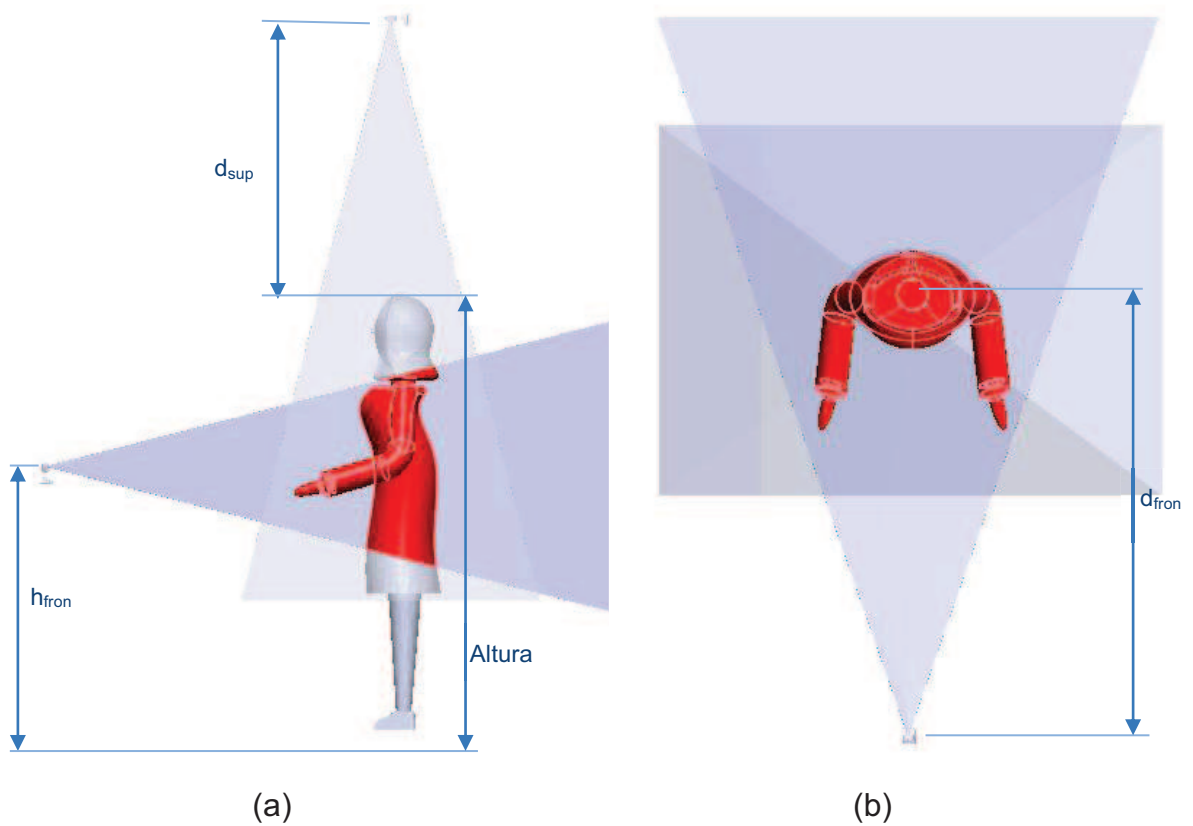
Adicionalmente, de forma experimental se definió que la cámara frontal debe ubicarse a la altura del pecho; la cámara superior debe colocarse sobre la persona y a una distancia mínima de un metro sobre la misma, de forma que la imagen capturada por ambas cámaras cubra totalmente el campo de movimiento requerido.

A partir de las expresiones propuestas y las medidas definidas experimentalmente, se precisaron las áreas del cuerpo capturadas por las cámaras, así como su ubicación, como se muestra en la Tabla 3.2 y en la Figura 3.4.

**Tabla 3.2 Ubicación de las cámaras**

Medida	Valor (mm)
Altura operador	1520
$d_{sup}$	1000
$x_{sup}$	(Sobre la persona)
$d_{fron}$	1085.7
$h_{fron}$	950

FUENTE: Elaboración propia



**Figura 3.4 Ubicación de las cámaras y áreas captadas**

**(a) Vista Lateral, (b) Vista Superior**

FUENTE: Elaboración propia

### 3.1.2 ILUMINACIÓN

Pese a la efectividad de la cámara para adquirir la imagen del área de trabajo, se hizo necesaria la inclusión de una fuente de luz adicional que contrarreste los cambios de luz ambiental a lo largo del día.

Debido a que el proyecto se desarrolla en un ambiente cerrado con luminarias de luz blanca en el techo y entradas de luz natural, se optó por utilizar una fuente de luz fuerte, la cual conserve su intensidad con el paso del tiempo, además de concentrarla en un área específica. Esto se logró mediante un reflector de luz halógena, con las características que se muestran en la Tabla 3.3.

**Tabla 3.3 Características Luminaria Halógena**

Característica	Valor
Potencia	150 W
Voltaje de alimentación	110V
Flujo luminoso	2400 lm
	

FUENTE: VARIOS (2011) Quartz Halogen lamp Tube R7s 150W [Catálogo de productos] *Halogen Lamps*. Recuperado de: [http://chinayuehang.en.alibaba.com/product/539961477-209296942/Quartz\\_Halogen\\_lamp\\_Tube\\_R7s\\_150W\\_300W\\_500W\\_1000W.html](http://chinayuehang.en.alibaba.com/product/539961477-209296942/Quartz_Halogen_lamp_Tube_R7s_150W_300W_500W_1000W.html)

### 3.1.3 UNIDAD DE PROCESAMIENTO



El núcleo de la plataforma lo conforma el elemento que realiza el procesamiento de las imágenes adquiridas. Como se determinó en el Capítulo 2.3.3, las imágenes son procesadas bajo el algoritmo de Zhang-Suen por lo cual se implementa el algoritmo de esqueletización en lenguaje C, determinando que este núcleo sea un computador externo.

Debido a la cantidad de acciones que deben ejecutarse durante el procesamiento (refiérase al Capítulo 4), se requiere que el computador posea recursos suficientes para desarrollarlas en tiempo real, de manera que el retraso por procesamiento sea mínimo. Por este motivo se utilizó un computador con procesador multinúcleo, cuyas características se muestra en la Tabla 3.4.

**Tabla 3.4 Especificaciones del computador**

<b>Característica</b>	<b>Valor</b>
<b>Marca</b>	Dell
<b>Modelo</b>	XPS 14z
<b>Procesador</b>	Intel® Core™ i5-2430M
<b>Velocidad del procesador</b>	2.40 GHz
<b>Memoria RAM</b>	4 Gb
<b>Disco Duro</b>	500 Gb
<b>Pantalla</b>	True Life WLED 14 pulg. Hi-Def (720p)
<b>Interfaz USB</b>	1 puerto USB 2.0 1 puerto USB 3.0
<b>Sistema Operativo</b>	Windows XP Service Pack 2 (32 bits)



FUENTE: Dell Inc. (2011). Especificaciones técnicas XPS 14z. Recuperado de:  
<http://www.dell.com/ec/p/xps-l412z/pd>

### 3.1.4 MANIPULADOR ROBÓTICO

La plataforma se ha desarrollado específicamente para el manipulador CRS-A255, con un controlador C500c. Este manipulador robótico fue diseñado para su uso en educación y automatización de laboratorios, principalmente para manejo de materiales ligeros.

Las principales características de este manipulador se muestran en la Tabla 3.5.

**Tabla 3.5 Especificaciones del manipulador CRS-A255**

<b>Característica</b>	<b>Valor</b>
<b>Fabricante</b>	Thermo CRS Robotics
<b>Modelo</b>	A255
<b>Número de ejes</b>	5
<b>Peso</b>	17 Kg
<b>Alcance</b>	559 mm (sin efector final)
<b>Mecanismo</b>	Motores electromecánicos con motores

	incrementales en cada articulación
<b>Controlador</b>	Thermo CRS Robotics C500C
<b>Microprocesador</b>	133 MHz i486DX (procesador del sistema) 60 MHz TMS320C31 DSP (control de movimiento)
<b>Voltaje</b>	110 V
<b>Potencia</b>	1000 W
	

FUENTE: *A255 System User Guide*, Thermo CRS, 2002

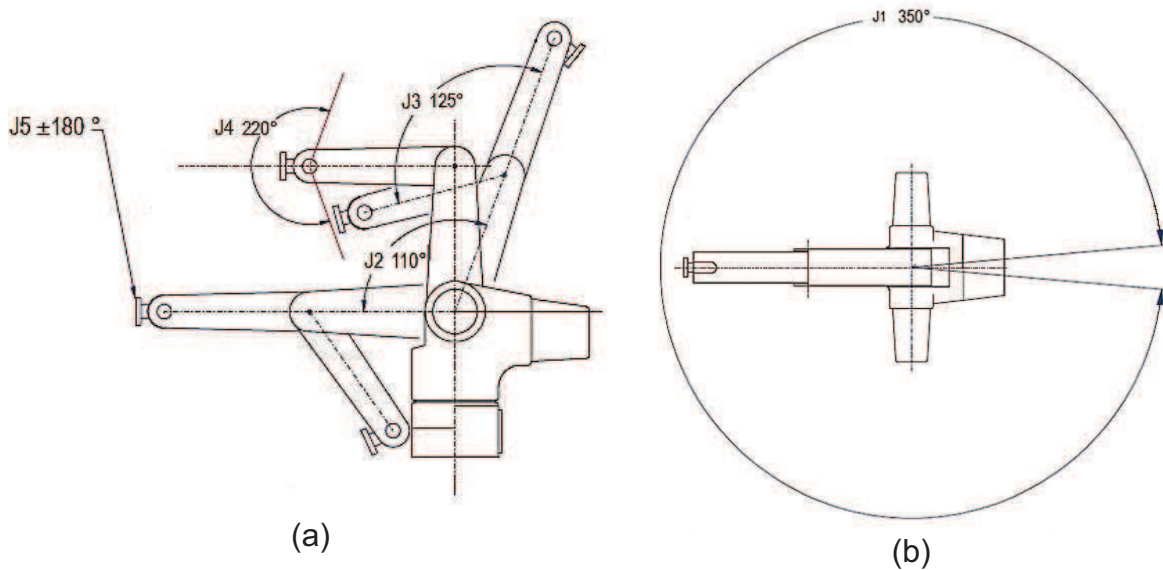
Para mayor detalle de las características cinemáticas del manipulador, consultar el Anexo A.

#### 3.1.4.1 Características de Integración

Debido a la naturaleza de la plataforma es necesario considerar ciertos aspectos relativos al hardware que posibilitan su interacción con el operador humano. Es necesario puntualizar las restricciones físicas que poseen, tanto el manipulador como el operador, durante el movimiento, así como las analogías que se utilizan para comandar cada articulación.

### 3.1.4.1.1 Analogía Morfológica

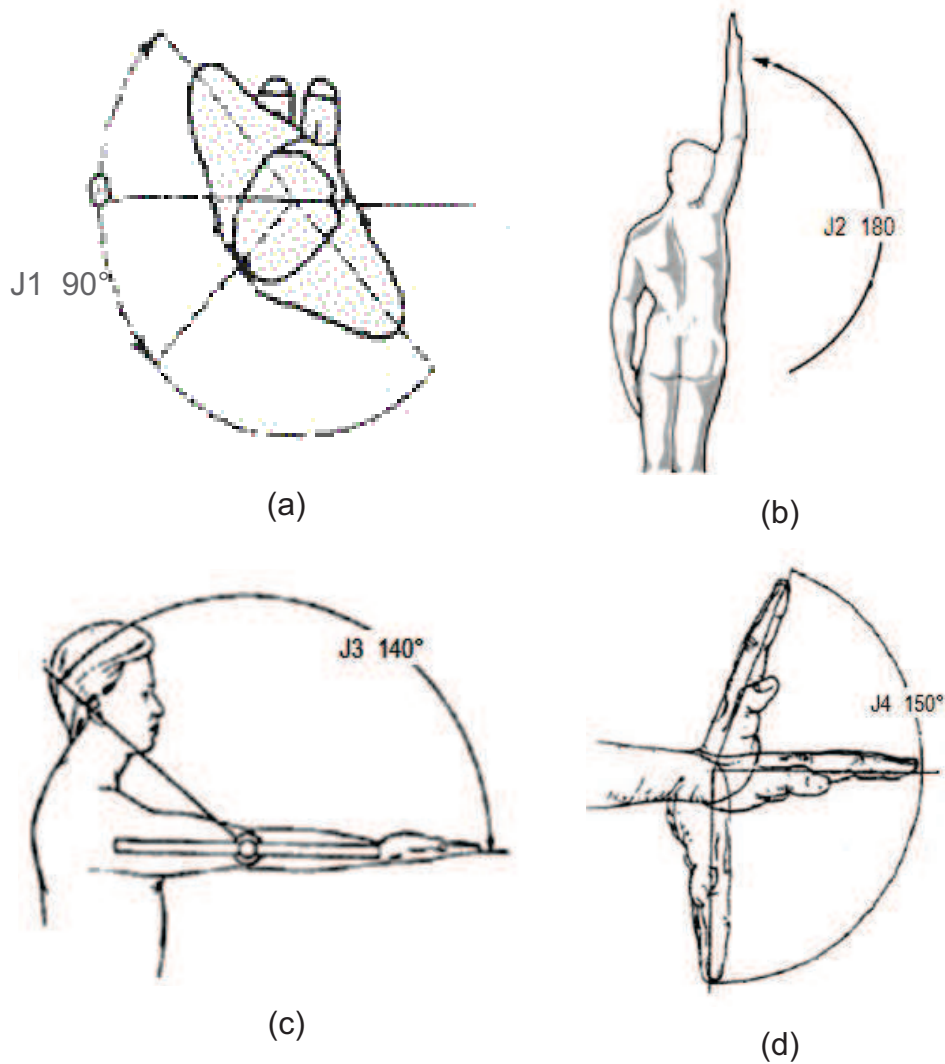
Como se mencionó en el Capítulo 2.1, el control de movimiento que realiza la plataforma se basa en la correspondencia de los movimientos humanos con los movimientos del manipulador. Para lograr una correspondencia que no requiera exceder el máximo esfuerzo del operador, se debe considerar el mecanismo de movimiento del robot.



**Figura 3.5 Articulaciones del manipulador CRS-A255. (a) Vista lateral (b) Vista superior**

FUENTE: *A255 System User Guide*, Thermo CRS, 2002

En la Figura 3.5 se aprecia el arco de movimiento de las articulaciones del manipulador robótico; en contraparte, la Figura 3.6 muestra los arcos de las articulaciones humanas respectivas.



**Figura 3.6 Arco de movimiento de las articulaciones humanas de interés  
(a) Cintura, (b) hombro, (c) codo, (d) muñeca**

FUENTE: (a)<sup>12</sup>; (b)<sup>13</sup>; (c)<sup>14</sup>; (d)<sup>15</sup>

Como se observa, el movimiento del operador humano difiere significativamente en rangos, con respecto al manipulador robótico, por lo que se establecen restricciones para el movimiento libre. Para ello se tomó en cuenta el movimiento ergonómico del operador y el rango de movimiento del manipulador sin que éste se vea comprometido. Las restricciones establecidas se muestran en la Tabla 3.6.

**Tabla 3.6 Restricciones de movimiento**

Articulación	Rango de Movimiento		Rango Restringido
	CRS-A255	Humano	
<b>J1: Cintura</b>	350°	90°	+45° / - 45°
<b>J2: Hombro</b>	110°	180°	+10° / +110°
<b>J3:Codo</b>	125°	140°	-45° / J2 <sup>c</sup>
<b>J4: Muñeca</b>	220°	150°	-90°+J3 / +90°+J3 <sup>d</sup>
<b>J5: Muñeca en rotación</b>	180°	180°	-180° / +180°

FUENTE: Elaboración propia

Adicionalmente, el movimiento, ergonómicamente cómodo, del ser humano en lo que respecta a la articulación J2 (hombro) se realiza desde la extensión completa del brazo, a la altura de la clavícula, hacia abajo, mientras que en el manipulador se considera desde la extensión completa del brazo hacia arriba, por lo cual se invierte el movimiento, de manera que el operario pueda manejar el manipulador sin recurrir a movimientos forzados. Esto provoca que un movimiento descendente del brazo se traduzca en un movimiento ascendente para el manipulador. Por este motivo, se establecieron referencias específicas para la medición de los ángulos de cada articulación, las cuales se muestran en la Tabla 3.7.

**Tabla 3.7 Referencias para el movimiento del operador**

Articulación	Referencia (0°)	Movimiento Positivo
<b>J1: Cintura</b>	Operador totalmente de frente	Giro hacia la derecha

<sup>c</sup> J2 se refiere al valor que tomará la articulación J2 para la posición deseada.

<sup>d</sup> J3 se refiere al valor que tomará la articulación J3 para la posición deseada

<b>J2: Hombro</b>	Brazo derecho totalmente extendido, paralelo a la horizontal	Movimiento hacia abajo
<b>J3: Codo</b>	Extendido, paralelo a la horizontal	Movimiento hacia abajo
<b>J4: Muñeca</b>	Totalmente extendida, paralela a la horizontal	Movimiento hacia abajo
<b>J5: Muñeca en rotación</b>	Palma de la mano hacia abajo, paralela a la horizontal	Giro hacia la derecha

FUENTE: Elaboración propia

### 3.1.4.1.2 Restricciones De Movimiento

Con base en las referencias y los ángulos máximos de movimiento del robot, se limita a un espacio de trabajo para el robot, sin embargo, también se debe tomar en cuenta el ambiente donde éste se desenvuelve, de manera que no afecte a ningún elemento presente en el área de trabajo.

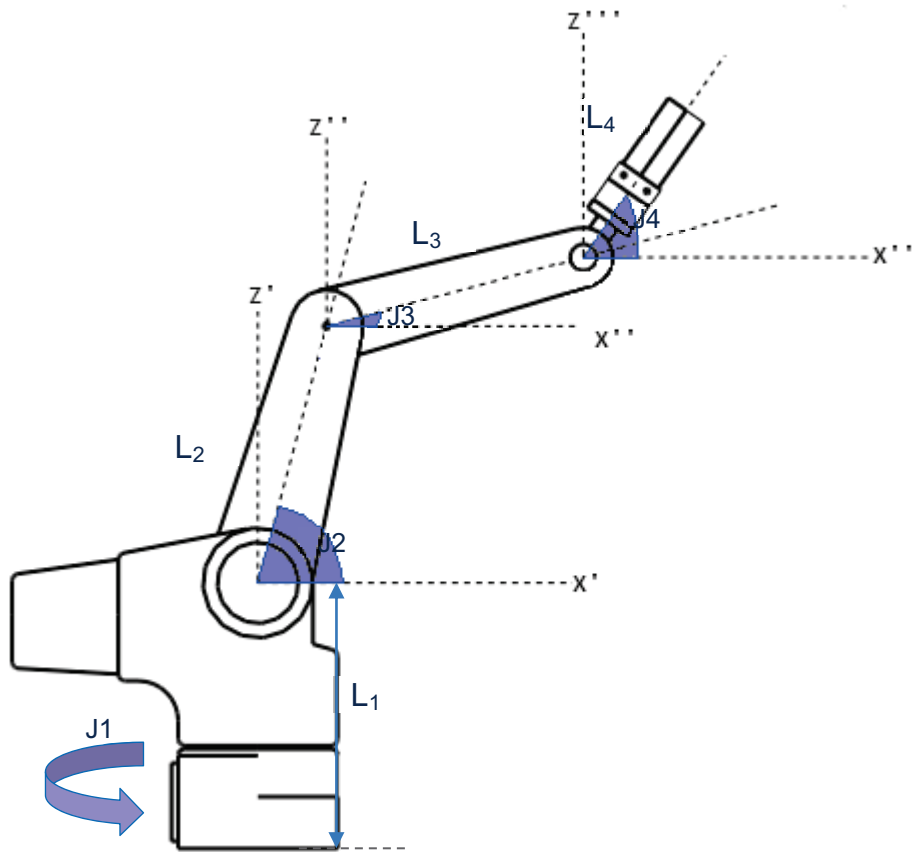
En el caso del manipulador CRS-A255 utilizado en este proyecto, se encuentra ubicado sobre una mesa de laboratorio y muy cercano a dos áreas de trabajo ubicadas una a cada lado del manipulador, razón por la cual se establece una restricción adicional de movimiento con referencia a la posición del efector final, de manera que no se acerque a ninguno de los elementos mencionados.

Para la construcción de esta restricción se utiliza la expresión de cinemática directa del manipulador, limitando su avance en Z hasta una altura de 200 mm sobre la mesa en la que se encuentra.

Tomando como referencia la Figura 3.7, se puede establecer la relación mostrada en la Ecuación 3.5.

$$z = l_1 + l_2 \sin(J_2) + l_3 \sin(J_3) + l_4 \sin(J_4)$$

**Ecuación 3.5**



**Figura 3.7 Esquema de ángulos y distancias para la cinemática directa del manipulador CRS-A255**

FUENTE: Elaboración propia

Con la Ecuación 3.5 limitada a 200 mm en el eje Z, se asegura que el manipulador no golpee la superficie donde se encuentra ubicado. Adicionalmente, las limitaciones de giro de la articulación J1 garantizan que el manipulador tampoco alcance las dos áreas de trabajo adyacentes a él.

Con las consideraciones relativas al hardware referidas en este capítulo, se procede a la creación del software de la plataforma, encargado del procesamiento de imágenes y cálculo de ángulos, el cual es abordado en el Capítulo 4 del presente trabajo.



# CAPITULO 4

## SOFTWARE DEL SISTEMA

En el Capítulo 3 se analizaron los componentes físicos de la plataforma de control de movimiento por esqueletización; en este capítulo se analizará el software que comanda todos aquellos dispositivos y su integración con los mismos.

Este capítulo se desarrollará en dos partes:

- Requerimientos de software
- Componentes del software

Adicionalmente, al final de este capítulo se presentará la estructura interna del programa desarrollado.

### 4.1 REQUERIMIENTOS DE SOFTWARE

Como se definió en el Capítulo 3.1, la plataforma lo conforman cuatro elementos:

- Sistema de video
- Iluminación
- Unidad de procesamiento
- Manipulador robótico

Para cumplir con el objetivo planteado en este proyecto, estos dispositivos deben estar integrados por medio de una aplicación de software que no solamente ejecute el algoritmo de esqueletización seleccionado en el Capítulo 2.3.3, sino también, todas las operaciones necesarias para la adquisición y envío de datos al manipulador. La Figura 4.1 muestra las operaciones requeridas por cada elemento de la plataforma en términos de software.



**Figura 4.1 Relación entre los componentes de la plataforma**

FUENTE: Elaboración propia

En base a las necesidades de los dispositivos y a la naturaleza de la plataforma abordada en este proyecto, se han definido los siguientes requerimientos en cuanto a la aplicación a desarrollar:

- Debe procesar en tiempo real las imágenes que recibe del sistema de video de la plataforma.
- Implementar el algoritmo de esqueletización de Zhang-Suen, en lenguaje C, como se estableció en el Capítulo 2.3.3
- Mantener comunicación con el manipulador robótico en todo momento.
- Calcular y enviar datos de movimiento al manipulador en el menor tiempo posible.
- Poseer una interfaz de usuario, permitiendo observar los datos obtenidos y las imágenes provenientes del sistema de video, así como configurar algunos parámetros de procesamiento y dar información sobre los parámetros de comunicación.
- Finalmente, debe ejecutarse sobre sistema operativo Windows XP, debido a las características de la unidad de procesamiento, establecidas en el Capítulo 3.1.3

## 4.2 COMPONENTES DEL SOFTWARE

En respuesta a los requerimientos mencionados en la sección anterior, se han planteado tres componentes para el componente de software de la plataforma, los cuales se muestran en la Figura 4.2.



**Figura 4.2 Componentes del software**

FUENTE: Elaboración propia

Estos componentes son abordados ampliamente en las secciones posteriores de este capítulo, así como las librerías utilizadas en cada uno de ellos.

#### **4.2.1 COMUNICACIÓN ENTRE LOS DISPOSITIVOS**

Como se determinó en el Capítulo 3, la plataforma está conformada por cuatro elementos esenciales, los cuales se comunican con la unidad de procesamiento para llevar a cabo las funciones que le permitirán controlar el movimiento del manipulador robótico.

De acuerdo a las especificaciones técnicas de cada uno de los dispositivos, esta comunicación se la realiza mediante diferentes interfaces, como se muestra en la Figura 4.3.



**Figura 4.3 Comunicación entre los dispositivos**

FUENTE: Elaboración propia

En el caso del sistema de video, la interfaz USB 2.0 utilizada, está determinada por las cámaras web que lo conforman. El estándar de comunicaciones USB (Universal Serial Bus) 2.0 permite una comunicación 100 veces más rápida a un puerto serial estándar (480Mb/s)<sup>16</sup>, lo cual en el caso específico de este proyecto, favorece a la adquisición de imágenes en tiempo real. Adicionalmente, brinda la funcionalidad *plug-and-play*<sup>e</sup>, lo cual facilita el reconocimiento de los dispositivos a través de drivers genéricos para los mismos.

Para el manipulador robótico, la interfaz RS-232 la establece el controlador c500c, correspondiente al robot CRS-A255. Al tratarse de comunicación serial, requiere de la aplicación de ciertos parámetros para la comunicación efectiva entre el controlador y la unidad de procesamiento de la plataforma; estos parámetros se muestran en la Tabla 4.1.

**Tabla 4.1 Parámetros de comunicación con el manipulador**

Parámetro	Valor
Velocidad del puerto	38400
Bits de salida	8
Paridad	Ninguna
Bits de parada	1

FUENTE: Elaboración propia

<sup>e</sup> *Plug-and-play*: se refiere a los dispositivos que únicamente requieren conectar para usar.

El controlador puede trabajar bajo dos modalidades:

- ACI, que permite transmitir datos, cargar programas y acceder a variables en la memoria del controlador.
- ASCII, permite el funcionamiento del host como una terminal, pudiendo ejecutar comandos RAPL-II directamente, así como ejecutar programas en memoria. Para mayor información sobre comandos RAPL-II diríjase al Anexo B.

En el caso específico de esta plataforma, se trabajará bajo modo ASCII por lo cual se requieren secuencias de inicialización específicas, así como de envío y lectura de datos. Estas secuencias se explicarán en las secciones 4.2.3.2.1, 4.2.3.2.6 y 4.2.3.2.7.

En lo referente al software, la comunicación serial exige un manejo de puertos por parte de la aplicación, para lo cual se ha planteado la utilización de una librería de código abierto, especializada en comunicación serial en lenguaje C, denominada *Linux Communication*.

**LnxCmm** (*Linux Communication*) es una librería originalmente creada para la comunicación serial bajo sistema operativo Linux. Fue desarrollada enteramente en lenguaje C, con acceso total a su código de programación.

Actualmente, la librería brinda funciones para manejo de puerto serial, puertos seriales emulados bajo USB (como es el caso de la plataforma), bajo sistema operativo Windows y Linux.<sup>17</sup>

En el caso específico de la plataforma desarrollada en este proyecto, esta librería se utilizará para establecer la comunicación, leer y enviar datos desde y hacia el controlador del manipulador robótico.

## 4.2.2 INTERFAZ GRÁFICA

Otra parte importante del software de la plataforma, abordada en este proyecto, es la interfaz gráfica, la cual permitirá el ajuste de ciertos parámetros de las imágenes, así como la visualización de los resultados del procesamiento de las imágenes.

Previo a su elaboración, se determinaron los elementos con los cuales debe contar para que el usuario pueda manejar la aplicación de manera sencilla, éstos se muestran en la Figura 4.4. Asimismo, se tomó en cuenta la interacción entre la interfaz y el programa desarrollado en lenguaje C, por ello fue necesario el uso de una biblioteca especializada en desarrollo de *GUIs*<sup>f</sup> compatible con este lenguaje.

INTERFAZ GRÁFICA		
<p style="text-align: center;"><b>IMÁGENES</b></p> <ul style="list-style-type: none"> <li>•A color (superior y frontal)</li> <li>•Esqueletizadas (superior y frontal)</li> </ul>	<p style="text-align: center;"><b>COMANDOS</b></p> <ul style="list-style-type: none"> <li>•Inicio de procesamiento cámara superior</li> <li>•Inicio de procesamiento cámara frontal</li> <li>•Inicio de comunicaciones</li> <li>•Parámetros para imágenes</li> <li>•Reinicio del robot</li> </ul>	<p style="text-align: center;"><b>INFORMACIÓN</b></p> <ul style="list-style-type: none"> <li>•Ángulos de cada articulación</li> <li>•Estado del gripper</li> <li>•Estado del robot</li> <li>•Parámetros de comunicación</li> </ul>

**Figura 4.4 Elementos esenciales de la interfaz gráfica**

FUENTE: Elaboración propia

Como respuesta a los requerimientos en lo referente a interfaz gráfica, se utiliza la biblioteca **Qt**, con **Qt Creator**<sup>9</sup>, para generarla.

Qt es una biblioteca multiplataforma, de código abierto, para el desarrollo de interfaces de usuario y programas en consola, desarrollada en lenguaje C++. Es utilizada en una variedad de entornos Gráficos para computadores y teléfonos inteligentes.

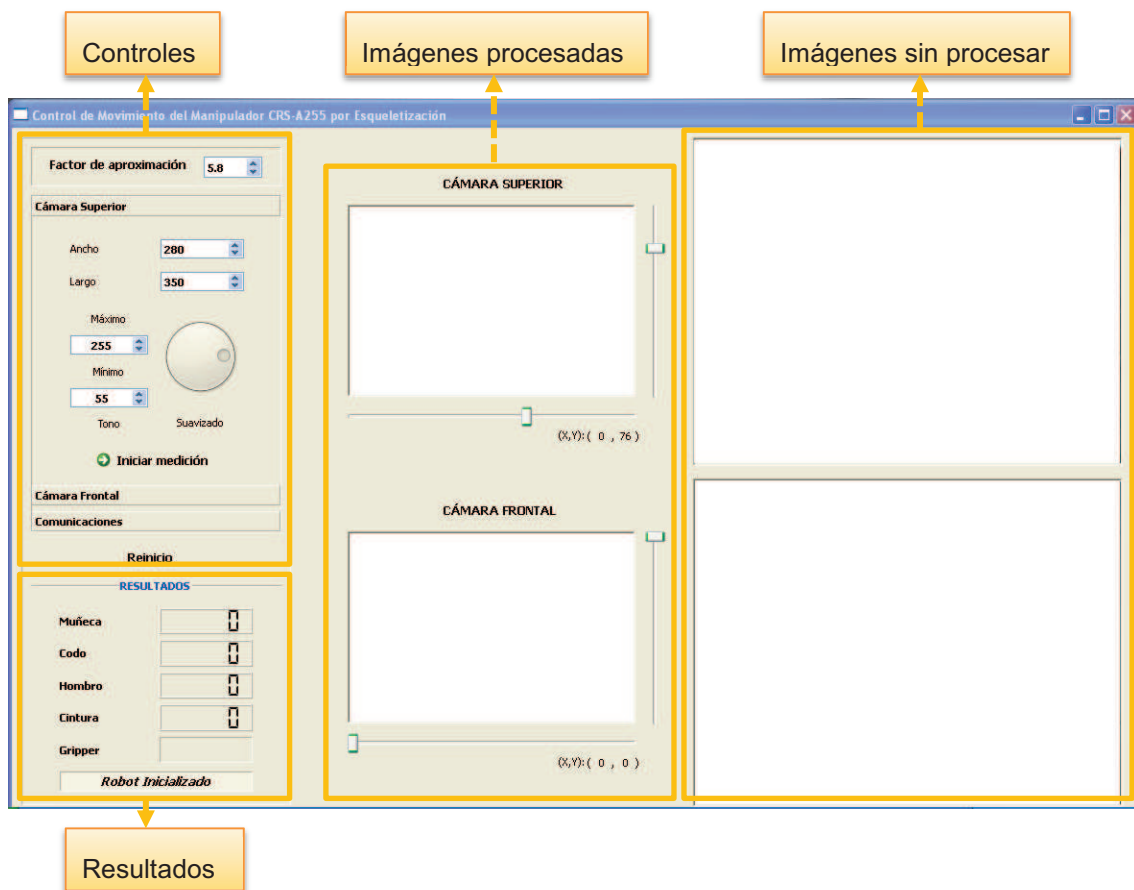
Una de las características más importantes en el funcionamiento de Qt es su trabajo bajo la modalidad de señales y conectores, lo que permite a la interfaz responder a eventos a través de la ejecución de acciones específicas determinadas por el programador.

<sup>f</sup> *GUI: Graphic user interface.* Interfaz gráfica de usuario.

<sup>9</sup> Qt Creator, es un entorno de desarrollo integrado, bajo licencia propietaria y GNU Lesser General Public License, versión 2.1. Incluye la herramienta Qt Designer. © 2012 Digia.

En el caso de la plataforma, el modo de funcionamiento de Qt facilita la interacción y aplicación de parámetros a las imágenes recibidas para su procesamiento. Además, su lenguaje nativo facilita la integración con la librería de comunicación.

El diseño de la interfaz gráfica se lo realizó mediante la herramienta Qt Designer, dando como resultado la interfaz mostrada en la Figura 4.5.



**Figura 4.5 Interfaz gráfica desarrollada**

FUENTE: Elaboración propia

Qt Creator genera un archivo de enlace entre los elementos Gráficos de la librería Qt y el código en lenguaje C. Adicionalmente, genera un meta-objeto, encargado de relacionar las señales con los conectores dentro del programa. Estos archivos fueron incluidos en la aplicación para manejar los eventos de la interfaz gráfica.

El manejo y las principales características de los elementos que conforman la interfaz gráfica se detallan en el Anexo C.

### 4.2.3 PROGRAMA PARA EL CONTROL DEL MANIPULADOR

El último componente de software lo constituye el programa para el control del manipulador, el cual se desarrolló en lenguaje C++.

Este programa constituye la parte central del presente proyecto, pues integra a todos los dispositivos para lograr un control de movimiento del manipulador robótico.

Al tratarse de una plataforma que utiliza una técnica de visión artificial, se hace necesario trabajar con una librería especializada en tratamiento de imágenes., la cual cumpla con los requerimientos mencionados en la sección 4.1, como lo es la librería OpenCV.

**OpenCV** (*Open Source Computer Vision*)<sup>h</sup> es una librería para visión computarizada en tiempo real, cuenta con funciones para lenguajes C/C++, Python y Java y se distribuye bajo licencia BSD<sup>18</sup>. Además, cuenta con tipos de datos propios de la librería para facilitar el procesamiento de los mismos.

Dentro de las funcionalidades más importantes de OpenCV se encuentran: funciones para procesamiento general de imágenes, segmentación, detección y seguimiento, transformación, descriptores geométricos y aprendizaje automático.

En el programa de control del manipulador se han utilizado varias funciones de OpenCV, versión 2.3, las cuales serán mencionadas a lo largo de las secciones 4.2.3.2.2 y 4.2.3.2.4.

Para información detallada acerca de las clases y funciones principales del presente programa, refiérase al Anexo D. Para información específica sobre la librería OpenCV, diríjase al Manual de Referencia de OpenCV.

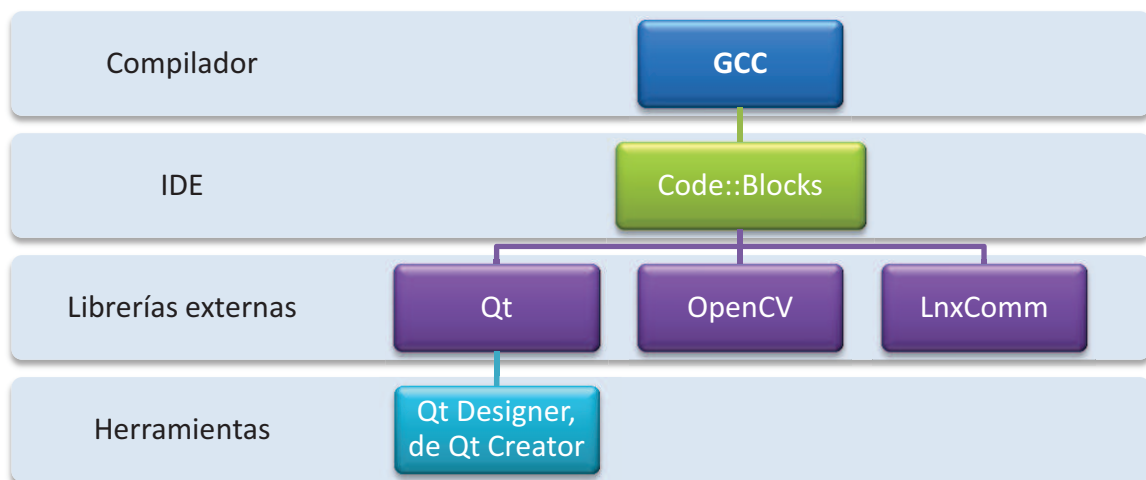
---

<sup>h</sup> *Open Source Computer Vision*: Visión Computarizada de Código Abierto



#### 4.2.3.1 Software para Desarrollo

Antes de describir el programa es necesario describir la integración de los elementos usados para su desarrollo. Como se ha mencionado a lo largo de este capítulo, ha sido necesaria la utilización de varias librerías especializadas. Cabe insistir en que estas librerías fueron escogidas por sus funcionalidades y su grado de compatibilidad con el lenguaje C; éstas se muestran en la Figura 4.6.



**Figura 4.6 Software de desarrollo**

- GCC (*GNU C Compiler*), es un compilador libre de lenguaje C y C++, distribuido por la *Free Software Foundation* bajo licencia GPL. Fue escogido por su compatibilidad con OpenCV.
- Code::Blocks, es un entorno de desarrollo integrado (*IDE*), libre y multiplataforma, para el desarrollo de programas en lenguaje C y C++, distribuido bajo licencia GPL. Este *IDE* se utilizó en el afán de manejar el desarrollo de la aplicación bajo herramientas libres.
- Librería Qt, es una librería libre y de código abierto, la cual fue utilizada para el desarrollo de la interfaz gráfica de esta plataforma, como se explicó en la sección 4.2.2.
  - Qt Designer, es una herramienta en modo Gráfico, propia de Qt para el diseño de estas interfaces.

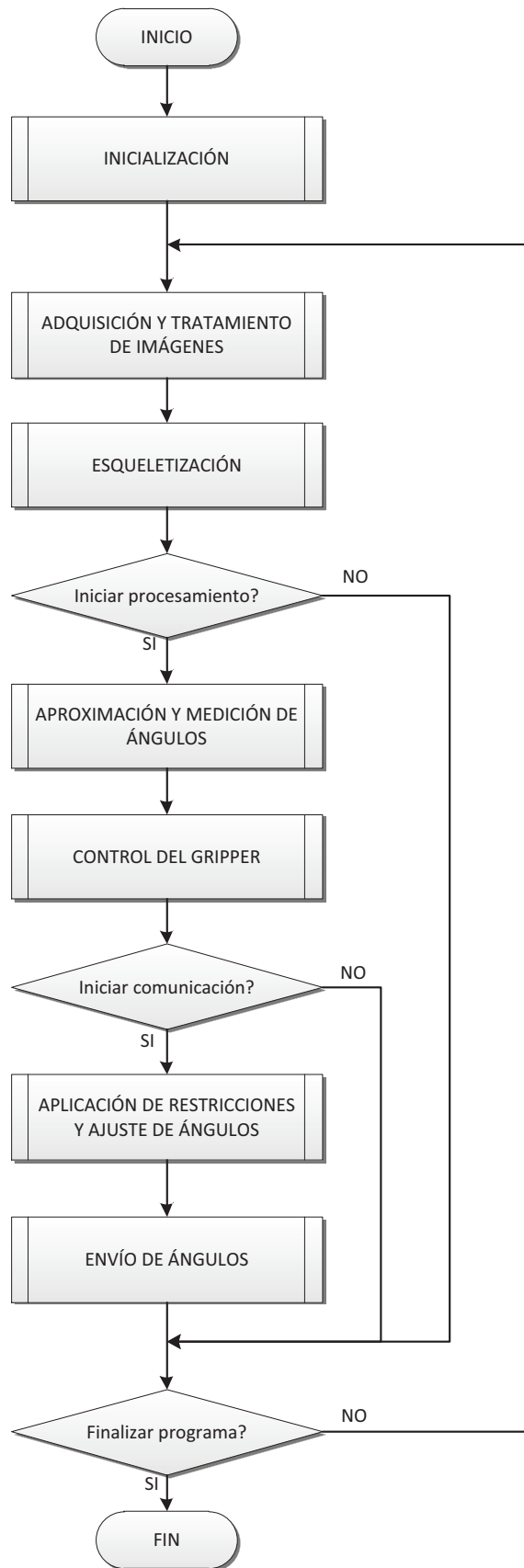
- OpenCV (*Open Computer Vision Library*), es una librería de código abierto para el tratamiento de imágenes. Ésta fue utilizada en la mayor parte del desarrollo de la aplicación.
- LnxComm (*Linux Communication*), fue la librería utilizada para el manejo del puerto de comunicaciones, como se explicó en la sección 4.2.1.

La integración de todos estos elementos se realizó de la siguiente manera:

- GCC fue agregado en Code::Blocks como herramienta de compilación para C++, a través de sus opciones de configuración.
- OpenCV se agregó a Code::Blocks manualmente, pues este último no posee una extensión que permita el uso inmediato de esta librería. Por este motivo se añadió cada una de sus dll al *IDE*, a través de sus opciones de configuración.
- En el caso de la integración de Qt con Code::Blocks, fue necesaria la generación de los conectores y señales en la herramienta de Qt, de modo que ésta los compile y modifique directamente los archivos de enlace. Estos archivos fueron añadidos al proyecto en Code::Blocks, así como las dll de Qt. Una vez importados los archivos, éstos pudieron ser modificados en el *IDE*.
- Para el caso de LnxComm, los archivos fuente de la librería fueron añadidos a la carpeta del proyecto y direccionados en el *IDE* para su reconocimiento.

#### **4.2.3.2 Subprocesos del Programa para el Control del Manipulador**

Como puede observarse en la Figura 4.7, el programa fue planteado como un ciclo en el cual interactúan los subprocesos en forma secuencial, cuyo funcionamiento será detallado a lo largo de esta sección, así como el resultado de cada uno de ellos.

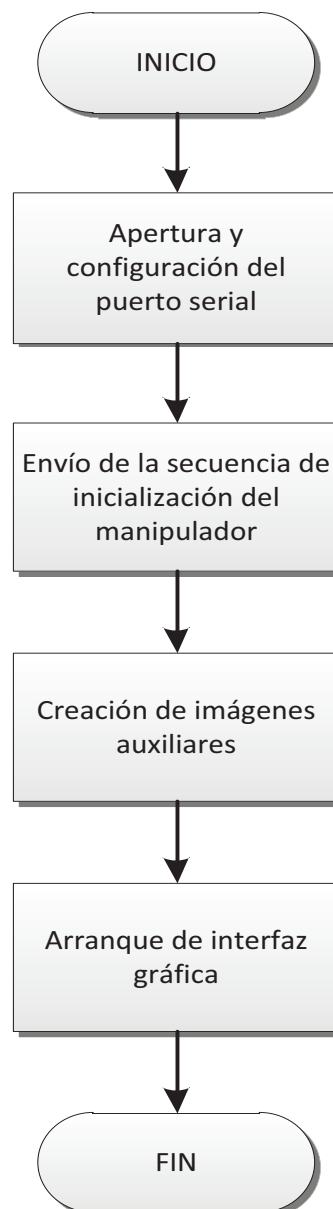


**Figura 4.7 Diagrama de flujo del software**

FUENTE: Elaboración propia

#### 4.2.3.2.1 Inicialización

El primer subproceso dentro del programa fue denominado inicialización. Éste se ejecuta solamente una vez, al inicio del programa y contiene todas las funciones necesarias para el arranque de la aplicación, como se puede observar en la Figura 4.8.



**Figura 4.8 Diagrama de flujo de la etapa de inicialización**

FUENTE: Elaboración propia

La primera etapa es la **apertura y configuración del puerto serial**, la cual se realiza con la ayuda de la librería LnxComm, sobre el puerto COM1, con los parámetros expuestos en la Tabla 4.1.

Como segunda etapa, se envía la **secuencia de inicialización del manipulador**. Esta etapa es importante para establecer la comunicación con el manipulador, así como para enviar comandos iniciales importantes para la plataforma.

Para obtener la secuencia de inicialización de la plataforma se utilizó un monitor de puerto serial, el cual interceptó la comunicación durante el arranque del manipulador con su terminal nativa, denominada RobCom. Los resultados obtenidos se muestran en la Tabla 4.2.

**Tabla 4.2 Secuencia de inicialización**

Secuencia	Petición	Respuesta
<b>Identificación maestro/ esclavo</b>	52 21 05	52 21 06
<b>Tipo de comunicación</b>	01 21 FF 00 48 00 01 00 00 00 00 03 69	06 02 00 03 00
<b>ACK</b>	06	04
<b>ACK</b>	04	
<b>Inicio de transmisión</b>	18	0D 0A 3E 3E

FUENTE: Elaboración propia, basada en <sup>19</sup>.

La secuencia adquirida se imita en el programa, logrando establecer la comunicación con el manipulador en modo ASCII. Al final de esta secuencia, se muestra un mensaje en pantalla indicando: *“Robot inicializado”* si la comunicación se realizó adecuadamente; caso contrario se muestra el error producido, el cual generalmente corresponde a una falla en la apertura del puerto serial.

Adicionalmente, se envían tres comandos RAPL-II para ayudar a la comunicación con la plataforma:

- NOHELP, para deshabilitar el autocompletado de comandos
- READY, para situar al manipulador en su posición, n inicial
- SPEED ##, para definir la velocidad de trabajo del robot, donde ## es un número entre 1 y 50.

Una vez establecida la comunicación, se procede a la **creación de las variables auxiliares** necesarias para la operación con imágenes. Este paso es necesario debido a que OpenCV posee variables del tipo *IplImage*, las cuales deben ser declaradas con un cierto tamaño de imagen, número de canales y profundidad. Para el caso específico de este programa se declaran nueve imágenes, las cuales serán usadas para almacenar el resultado de cada etapa de procesamiento y a su vez relacionarlas con las imágenes mostradas en la interfaz gráfica.

El último paso de la inicialización es el **arranque de la interfaz gráfica**. En este punto se inician las funciones básicas de Qt que permitirán mostrar los elementos Gráficos en la pantalla. También, se relacionan los conectores con sus respectivas señales y acciones a realizar para cada evento generado por el usuario en la interfaz gráfica.

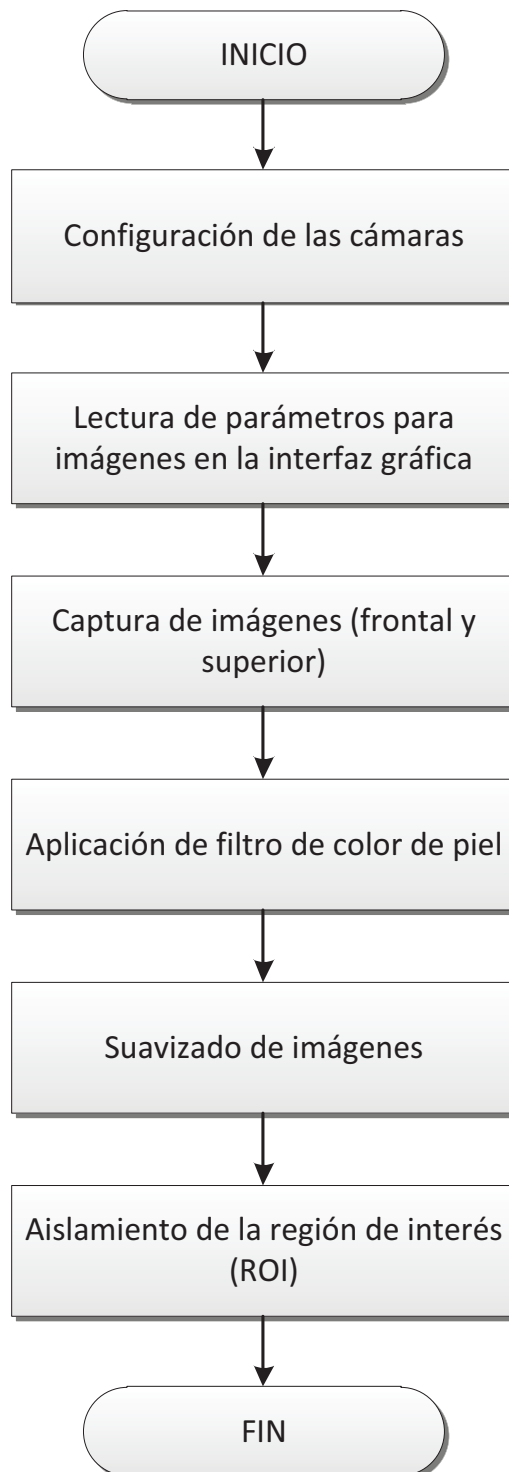
En resumen, como resultado del subproceso de inicialización se tiene:

- Puerto de comunicaciones abierto y configurado
- Comunicación con el manipulador establecida
- Manipulador inicializado
- Variables auxiliares declaradas
- Interfaz gráfica inicializada

#### 4.2.3.2.2 Adquisición y tratamiento de la imagen

El segundo subproceso del programa para el control del manipulador es la adquisición y tratamiento de las imágenes captadas por el sistema de video. En esta etapa se utilizó la librería OpenCV, así como funcionalidades específicas de Qt en base a las variables creadas en la inicialización.

Las etapas que componen el subproceso abordado en esta sección se muestran en el diagrama de flujo de la Figura 4.9.



**Figura 4.9 Diagrama de flujo de la etapa de adquisición y tratamiento de imágenes**

FUENTE: Elaboración propia

La primera etapa en el subproceso es la **configuración de las cámaras** que componen el sistema de video. Este paso es esencial pues impone al dispositivo

de captura el tamaño en el cual debe adquirir la imagen para no generar conflictos con las variables tipo *IplImage* declaradas en la inicialización, mismo que debe ser de 640x480 pixeles.

El segundo paso es la **lectura de parámetros desde la interfaz gráfica**. Al iniciar el procesamiento de las imágenes se aplicarán una serie de filtros, los cuales requieren parámetros específicos establecidos por el usuario, proceso que inicia el instante mismo de la primera captura. Por este motivo, es necesario recoger los parámetros establecidos por defecto en la interfaz gráfica, para disponer de esta información antes de iniciar con la captura, evitando así errores en el arranque de la aplicación.

A continuación se realiza la **captura de imágenes** desde las dos cámaras ubicadas de acuerdo a lo expuesto en el Capítulo 3.1.1.1. En este punto, cabe mencionar que la configuración de las cámaras así como la captura en tiempo real de las mismas se realizó a través de un hilo (subproceso) propio de la librería Qt, el cual captura continuamente las imágenes del operador desde la vista superior y frontal y las envía al programa principal para ser procesadas allí.

Este hilo también realiza una conversión vital sobre la imagen: la transformación de la variable *IplImage* a *QImage*. Dicha conversión se realiza porque cuando la cámara captura la imagen a través de OpenCV, ésta es almacenada en un *IplImage* sobre la cual se ejecuta todo el procesamiento; sin embargo este tipo de dato no es aceptado por Qt, pues esta librería maneja imágenes a través de variables del tipo *QImage*, haciendo imposible mostrarla directamente al usuario en la interfaz. Para mayor información sobre esta transformación, refiérase a la Documentación del software disponible en el Anexo D de este documento.

Las imágenes obtenidas en esta etapa se muestran en la Figura 4.10.





**Figura 4.10** Imágenes sin procesamiento. (a) Imagen frontal, (b) Imagen superior

FUENTE: Elaboración propia

Con las imágenes base para cada una de las cámaras se procede a la **aplicación de un filtro de color de piel**. Este filtro fue implementado por la necesidad de eliminar la mayor cantidad de ruido posible de la imagen previa a la esqueletización, para lo cual fue necesario trabajar en el modelo de color HSV. A través de OpenCV y la función *cvInRange*, fue posible separar los píxeles de la imagen que tengan una coloración similar a la de la piel, dentro de los rangos:

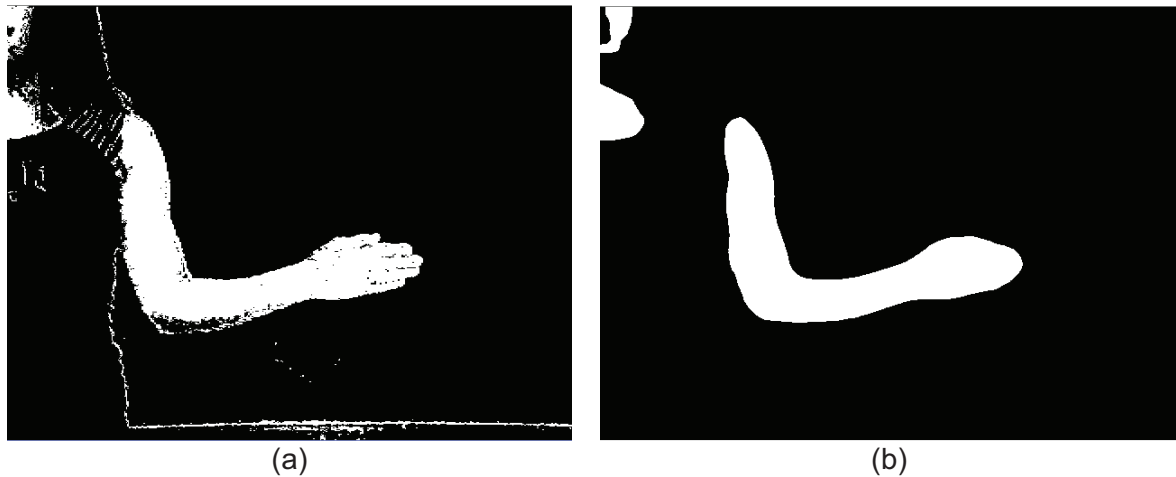
$$(H, S, V)_{min} = (0, satMin [55], 0)$$

$$(H, S, V)_{máx} = (50, satMax [255], 255)$$

Estos rangos fueron escogidos en base a las tonalidades más comunes de piel, sin embargo el usuario puede modificarlos en cierta medida en la interfaz gráfica. De esta forma, de la imagen original solo se consideran las partes que presenten esta coloración reduciendo la cantidad de píxeles a procesar y obteniéndose una imagen binaria al mismo tiempo.

Para mejorar el resultado de la aplicación del filtro se incluyó una fase de **suavizado** para las imágenes. El suavizado se realiza a través de una operación de OpenCV (*cvSmooth*), lo cual permite eliminar píxeles o grupos pequeños de píxeles de acuerdo a un tamaño específico, mismo que puede ser modificado por el usuario en la interfaz gráfica. Con este paso se logra una mayor definición de

los píxeles a procesar y la máxima eliminación posible de ruido. En la Figura 4.11 se muestra el resultado de la aplicación del filtro de color de piel y el suavizado.



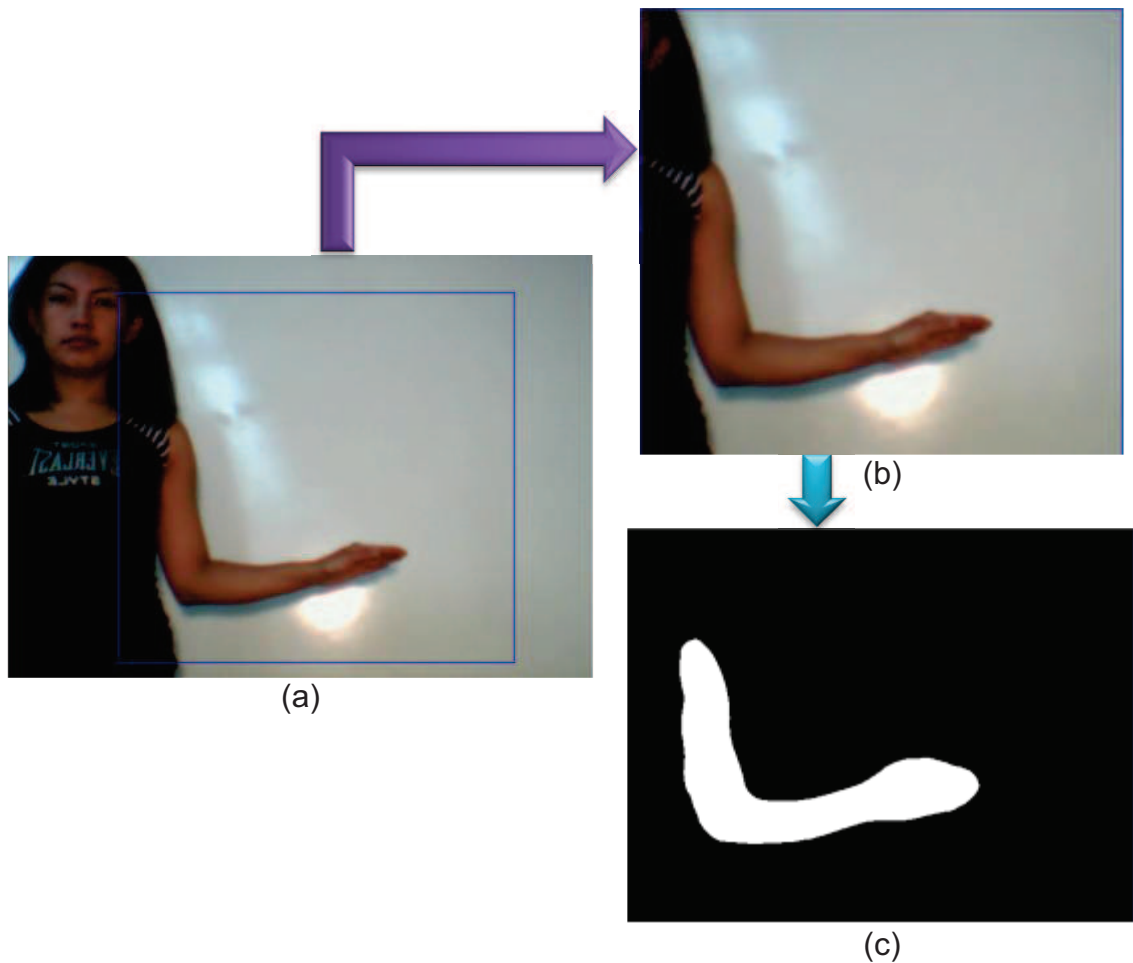
**Figura 4.11 Filtros aplicados en la imagen frontal. (a) Filtro de color de piel, (b) Filtro de color de piel + Suavizado**

FUENTE: Elaboración propia

Finalmente, para delimitar la zona de la imagen a procesar se especificó una **región de interés** (ROI) sobre la misma, mediante OpenCV. Esta región de interés permite al usuario restringir un espacio dentro del campo visual de la cámara para aplicar todas las operaciones de procesamiento. La posición así como el tamaño de esta región puede ser controlada por el usuario desde la interfaz gráfica. De esta manera se reduce aún más el tamaño de la imagen a procesar y se eliminan partes del cuerpo que puedan interferir en la esqueletización, como se muestra en la Figura 4.12.

Resumiendo, como producto de este subproceso se tienen dos imágenes por cada cámara:

- Una imagen a color, desplegada en tiempo real en la interfaz gráfica, que muestra al operador y su entorno
- Una imagen binaria más pequeña, en la cual se toman en cuenta únicamente los píxeles cuya coloración asemeja la piel y con menor cantidad de ruido.



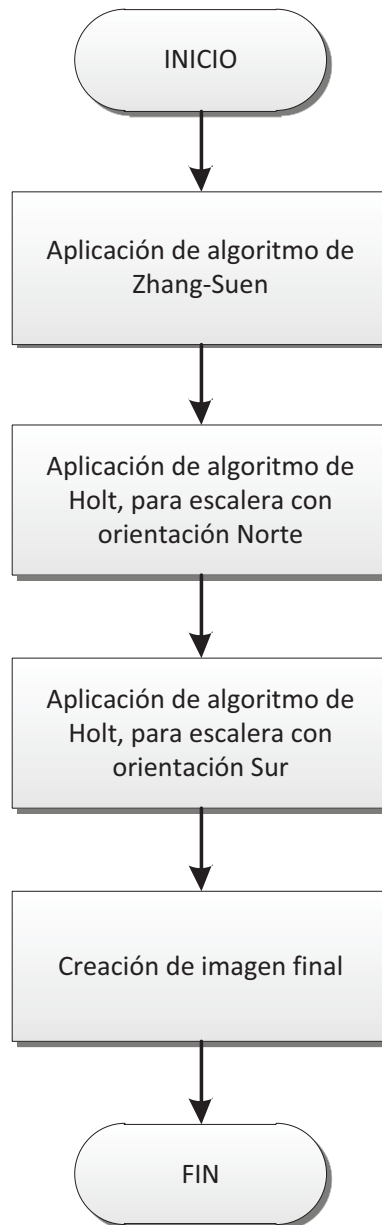
**Figura 4.12** Región de interés (a) Imagen original, (b) ROI, (c) ROI con filtros

FUENTE: Elaboración propia

La primera imagen servirá para orientar al operario, mientras la segunda imagen, pese a no ser mostrada, sirve de base para el subproceso de esqueletización.

#### 4.2.3.2.3 Esqueletización

El tercer subproceso planteado es el de esqueletización, cuyo núcleo son los algoritmos descritos en el Capítulo 2.3.3, los cuales son aplicados de acuerdo a la secuencia mostrada en la Figura 4.13.



**Figura 4.13 Diagrama de flujo de la etapa de esqueletización**

FUENTE: Elaboración propia

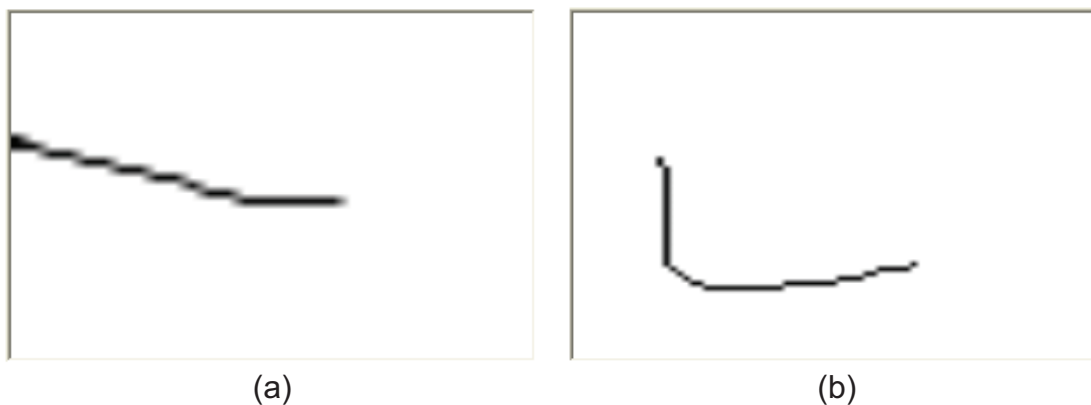
El subproceso de adquisición y tratamiento de imágenes arrojó una imagen binaria de la región de interés para el programa; por ejemplo, en el caso de la cámara frontal, el brazo completo del operador.

El proceso de esqueletización de esta imagen inicia con la aplicación del **algoritmo de Zhang-Suen**, detallado en el Capítulo 2.3.2.1. Toda la esqueletización se realiza mediante el análisis pixel a pixel de la imagen a través de funciones de la librería OpenCV, por lo cual fue necesario redimensionar la

imagen hasta un tamaño aproximado de 80x60 píxeles, de forma que la esqueletización no produzca retardos en la aplicación.

Obtenido el esqueleto primario se aplica un post-procesamiento, en dos etapas, bajo el **algoritmo de Holt**. En la primera etapa se eliminan los píxeles en forma de escalera con orientación norte y en la segunda, con orientación sur, como se explicó en el Capítulo 2.3.2.3.

Al término de la aplicación de los algoritmos de esqueletización se genera la imagen con el esqueleto final, la cual se muestra en la interfaz gráfica en tiempo real, como se observa en la Figura 4.14



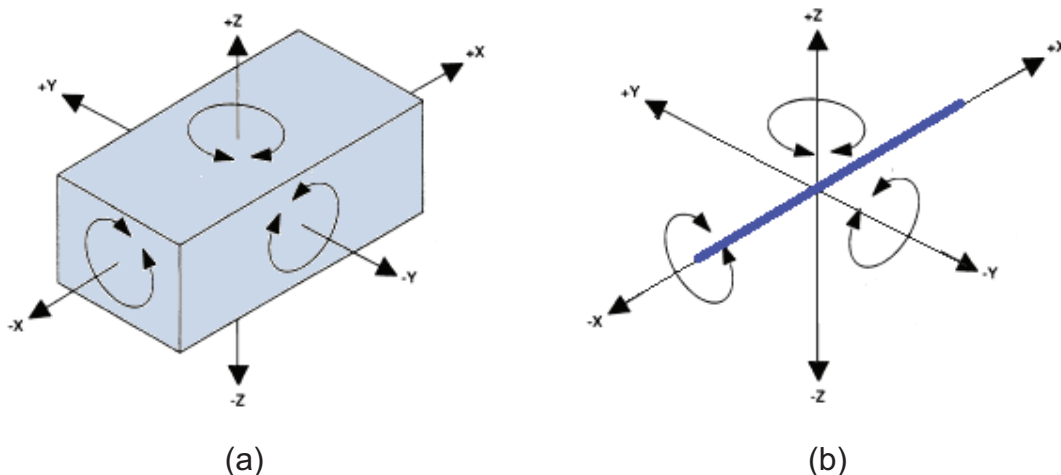
**Figura 4.14** Imágenes esqueletizadas (a) superior, (b) frontal

FUENTE: Elaboración propia

En este punto cabe realizar un análisis de los resultados de la esqueletización. Esta técnica de procesamiento de imágenes, busca reducir un objeto a una o varias líneas de un píxel de grosor, que conserven sus propiedades morfológicas. Al realizar esta reducción en una imagen bidimensional, se transforma al objeto en una línea en el plano, obteniéndose tres grados de libertad para la misma: dos de traslación y uno de rotación. En este proyecto, al utilizar dos cámaras ubicadas en planos perpendiculares, para un mismo objeto esqueletizado se obtiene un total de 5 grados de libertad: tres de traslación y dos de rotación.

Sin embargo, la rotación del objeto esqueletizado sobre su propio eje no es detectable. Tomando como referencia la Figura 4.15 (a), se puede apreciar que el sólido posee 6 grados de libertad claramente definidos: desplazamientos en X, Y

y Z, y rotaciones alrededor de X, Y y Z. Si a este objeto se lo esqueletiza, como se muestra en la Figura 4.15 (b), se podrá observar un desplazamiento en X, Y y Z, además de una rotación en Y y Z, pero no una rotación en X pues la línea no posee una referencia que denote un giro sobre su propio eje. Este caso se aplica en la muñeca humana, pues al ser esqueletizados conjuntamente el brazo y la mano forman una línea; si el usuario gira la muñeca sobre su propio eje, el resultado es el mismo, siendo imposible detectar su rotación a través de esta técnica. Dada esta limitación de la esqueletización, no se podrá medir y, por consiguiente, transmitir el giro de la muñeca sobre su propio eje, correspondiente a la articulación J5 del manipulador robótico, limitando el proyecto al control de movimiento de los restantes cuatro grados de libertad.



**Figura 4.15 Grados de libertad. (a) sólido, (b) sólido esqueletizado.**

FUENTE: (a) JERICO, Inc. (2010). Locating and Clamping Principles.[Página web]. Locators: Technical Information. Disponible en: <http://jerico.carrlane.net/catalog/index.cfm/29625071F0B221118070C1C513906103E0B05543B0B012009083C3B2853514059482013180B041D1E173C3B2853524B5B5E>; (b) Elaboración propia.

Retomando, el resultado del subproceso de esqueletización es, valga la redundancia, el esqueleto de la región de interés, es decir, una imagen formada por líneas y curvas interconectadas, de un pixel de grosor, que mantienen las propiedades morfológicas de la figura original. Sin embargo, las líneas y curvas del esqueleto aún no arrojan datos exactos de su inclinación o posición, ni son

identificadas directamente como una articulación específica del brazo, por lo cual es necesario enviarlas al subproceso de aproximación y medición de ángulos.

#### 4.2.3.2.4 Aproximación y medición de ángulos

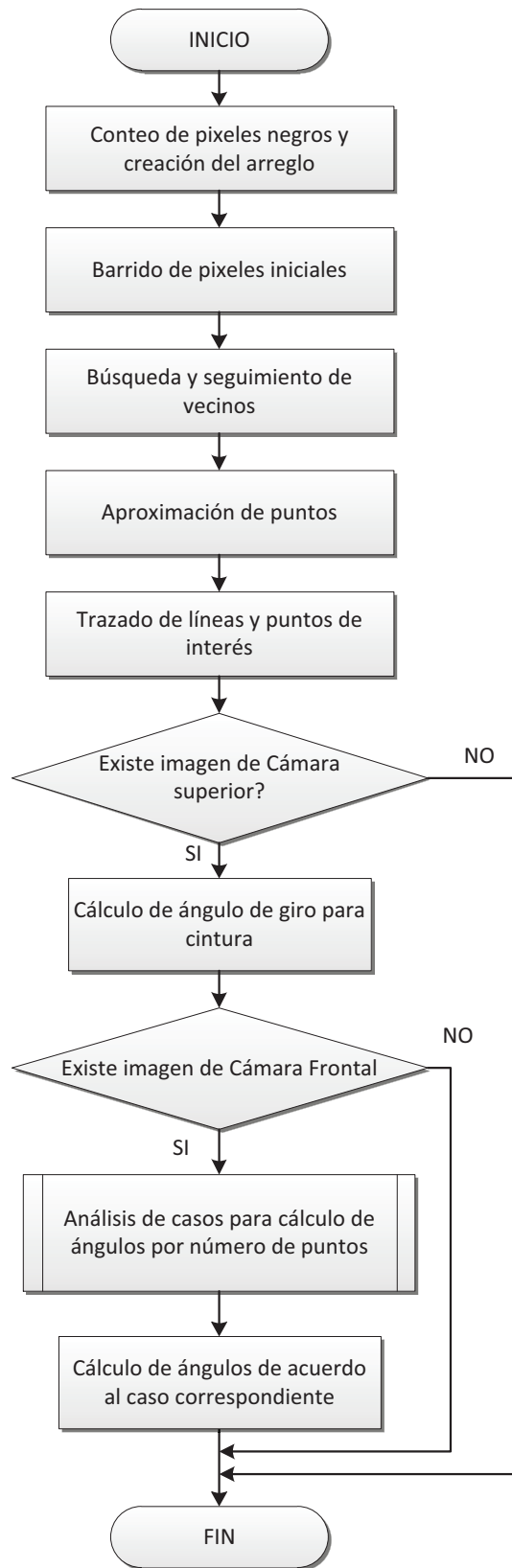
El cuarto subproceso de la aplicación es la aproximación y medición de ángulos, la cual se inicia a petición del usuario en la interfaz gráfica. Las etapas que componen este subproceso se muestran en la Figura 4.16.

La entrada para este subproceso es la imagen generada de la fase de esqueletización. Esta imagen al momento es un conjunto de puntos negros formando líneas y curvas interconectadas, sobre un fondo blanco, por lo que es necesario procesarlas para obtener datos válidos.

Para analizar la imagen de entrada, en una primera etapa se **obtiene el número de píxeles** negros presentes en la imagen y en base a él se crea y dimensiona un arreglo del tipo *CvMat*, el cual será utilizado posteriormente para almacenar los datos de cada pixel.

En el esqueleto de entrada se presentan líneas, es decir secuencias ordenadas de puntos, por ende el almacenamiento de los píxeles en el arreglo debe hacerse bajo el mismo concepto. Para ello se adoptó como estrategia realizar, como segunda etapa, un **barrido de píxeles** desde la esquina superior izquierda de la imagen. Este barrido buscará píxeles negros y los almacenará dentro del arreglo creado en la etapa anterior, en el mismo orden en el cual son desplegados en la imagen. Esta estrategia se utiliza únicamente sobre los primeros píxeles con el objetivo de crear una referencia de partida.

Con la referencia de partida se ejecuta una tercera etapa denominada **búsqueda de vecinos**. Esta función consiste en situarse sobre un punto de la línea y buscar un pixel adyacente a él que continúe dicha línea; al encontrarlo lo almacena dentro del arreglo y continúa el proceso, esta vez ubicándose sobre el último punto encontrado. De esta manera se asegura el almacenamiento de los puntos en el mismo orden de aparición en la línea y no de acuerdo a su posición en la imagen.



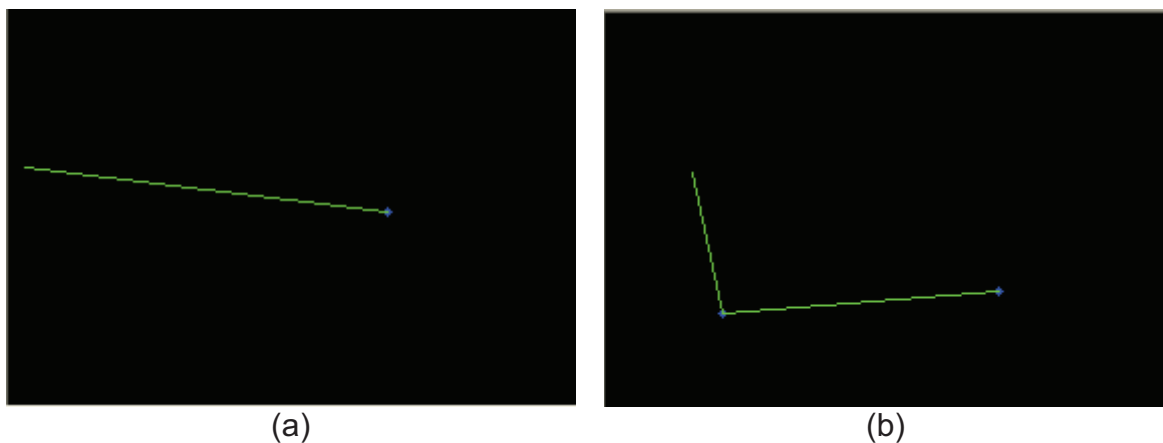
**Figura 4.16 Diagrama de flujo de la etapa de aproximación y medición de ángulos**

FUENTE: Elaboración propia



Al término de la búsqueda de vecinos, el arreglo contiene todos los píxeles y sus datos, de modo que pueden utilizarse fácilmente en la etapa de **aproximación de puntos**. Esta etapa consiste en la aplicación del algoritmo Douglas-Peucker<sup>20</sup>, a través de la función *cvApproxPoly* de OpenCV. Dicha función permite la reducción de los puntos guardados en el arreglo en base a una distancia máxima de tolerancia, especificada por el usuario en la interfaz gráfica; los puntos más allá de esta tolerancia no son tomados en cuenta, mientras aquellos se ubican dentro de la tolerancia se aproximan a una sola línea, obteniéndose solo dos puntos que la definen: el extremo inicial y final.

El proceso de aproximación se aplica en todo el arreglo, obteniéndose varias líneas y la ubicación de sus extremos, con lo cual es posible **graficarlas** y desplegarlas en la interfaz gráfica. La imagen resultante se muestra en la Figura 4.17.



**Figura 4.17 Esqueleto aproximado (a) Imagen superior, (b) Imagen frontal**

FUENTE: Elaboración propia

En base a las líneas y puntos que conforman el esqueleto aproximado, se procede al **cálculo de ángulos**, el cual fue dividido por casos, de acuerdo al número de líneas encontradas y a la cámara correspondiente, éstos se detallan en la Tabla 4.3.

**Tabla 4.3 Casos para medición de ángulos**

<b>Cámara</b>	<b>No. Puntos / No. Líneas<sup>i</sup></b>	<b>Descripción</b>	<b>Ángulos</b>
<b>Superior</b>	2 / 1	La línea representa el giro de la cintura del operador.	Cintura adopta el ángulo de inclinación de la línea
<b>Frontal</b>	2 / 1	El brazo se encuentra completamente extendido.	Hombro, codo y muñeca adoptan el ángulo de inclinación de la línea
	3 / 2	El codo se encuentra flexionado y la muñeca extendida.	Hombro: adopta ángulo de inclinación de la primera línea desde la izquierda.  Codo y muñeca: adoptan ángulo de inclinación de la segunda línea.
	3 / 2	La muñeca se encuentra flexionada y el codo extendido	Hombro y codo: adoptan ángulo de inclinación de la primera línea desde la izquierda.  Muñeca: Adopta ángulo de inclinación de la segunda línea.
	4 / 3	Tanto el codo como la muñeca están flexionados en alguna medida	Hombro: adopta ángulo de inclinación de la primera línea desde la izquierda.  Codo: adopta ángulo de inclinación de la segunda

<sup>i</sup> Los casos no contemplados son descartados pues no coinciden con el diseño del sistema.

línea.  Muñeca: adopta ángulo de inclinación de la tercera línea.
---

FUENTE: Elaboración propia

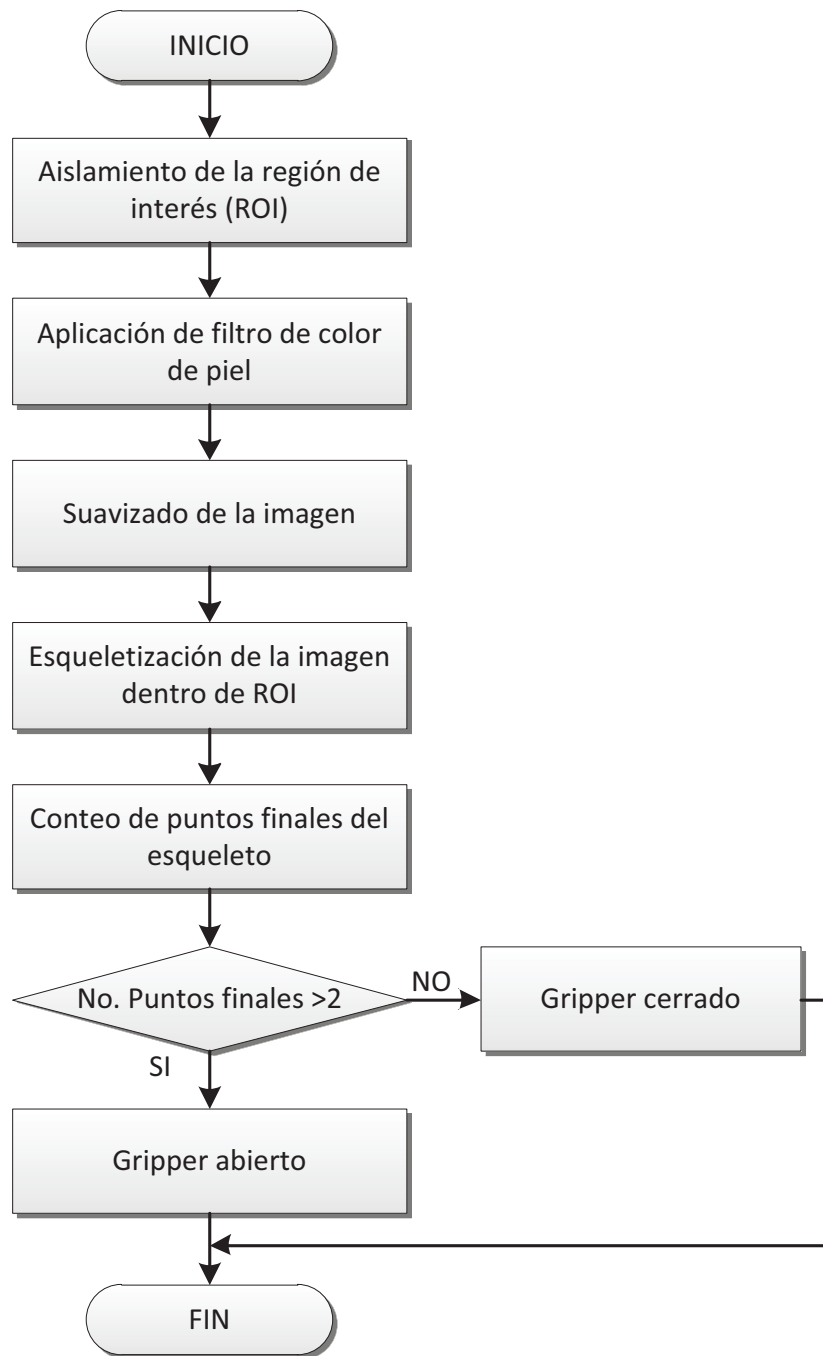
Para la cuantificación de los ángulos se utilizaron funciones trigonométricas de la librería *Math.h*, en base a los extremos inicial y final de las líneas encontradas. Los ángulos obtenidos son desplegados en la parte de “Resultados” de la interfaz gráfica.

Resumiendo, al final de este subproceso se obtiene:

- El esqueleto aproximado y datos de interés de las líneas que lo integran.
- Los ángulos de inclinación de cuatro articulaciones, que corresponden a cuatro grados de libertad del manipulador robótico.

#### 4.2.3.2.5 Control del gripper

Una vez que se adquieren los ángulos de las cuatro articulaciones se procede a ejecutar la detección del estado del gripper, para lo cual se ha planteado el proceso mostrado en la Figura 4.18.



**Figura 4.18 Diagrama de flujo de la etapa de control del gripper**

FUENTE: Elaboración propia

El motivo principal para ejecutar esta detección después de la medición de los ángulos es la disponibilidad de los datos de ubicación del punto final del esqueleto, es decir, la mano; en base a él se crea una **región de interés** sobre una copia de la imagen original de la cámara frontal. Se aísla esta región en su formato original debido a que se requiere aplicar los **filtros**, tanto de color de piel

como de suavizado, en su tamaño original para evitar la eliminación de detalles relevantes en esta etapa.

Con la región correspondiente a la mano aislada y filtrada, se procede a **esqueletizarla** de forma independiente bajo los mismos algoritmos que las imágenes anteriores, con la diferencia de que al ser una región de menor tamaño retuvo los detalles pequeños, como son los dedos. El esqueleto de esta región puede arrojar dos resultados generales:

- Una única línea, es decir un solo punto final, lo que indica que la mano está cerrada; o,
- Una línea inicial con derivaciones, es decir varios puntos finales, lo que indicaría que la mano está abierta ya que las derivaciones representan los dedos del operario.

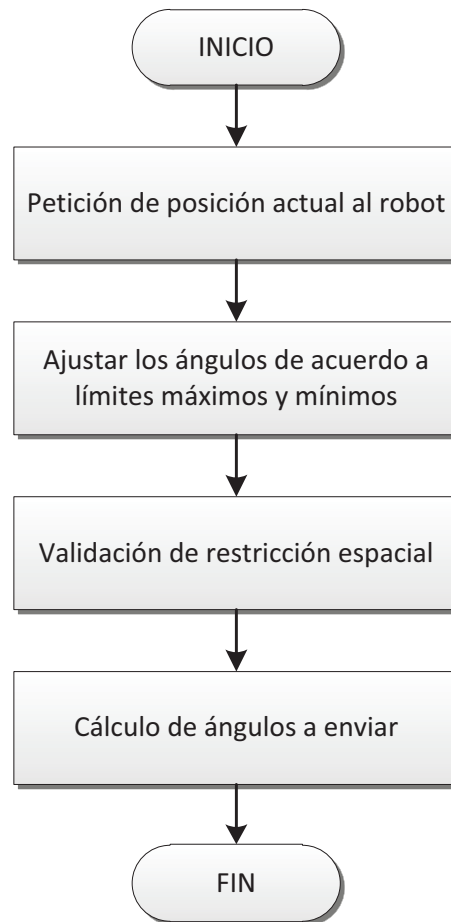
En este caso, para **contabilizar los puntos finales** no fue necesario el proceso de aproximación pues no es necesaria una información detallada, únicamente se cuentan los puntos que al pasar por la función de búsqueda de vecinos, durante la esqueletización, presenten solo un vecino adyacente; ello indica que son puntos finales.

Al detectar un solo punto final se procede a establecer el estado del gripper en “cerrado”, mientras que cuando se detectan 2 o más puntos finales se lo establece como “abierto”. Esta condición también se despliega en la interfaz gráfica, en la parte de “Resultados”.

El dato del estado del gripper es el producto final de este subproceso y será enviado al manipulador con los datos de las articulaciones en un subproceso posterior.

#### 4.2.3.2.6 Aplicación de restricciones y Ajuste de ángulos

Antes de enviar los ángulos al manipulador robótico es necesario realizar un ajuste de los mismos, para evitar el envío de datos erróneos o que podrían perjudicar al manipulador y su entorno. Este ajuste se realiza en varias etapas, como se muestra en la Figura 4.19.



**Figura 4.19 Diagrama de la etapa de aplicación de restricciones y ajuste de ángulos**

FUENTE: Elaboración propia

Este subproceso es la clave del control de movimiento del robot, pues asegura que los comandos enviados correspondan a movimientos válidos dada la posición actual del manipulador. Para ello se ha planteado una etapa inicial denominada **petición de la posición actual** del robot, la cual consiste en el envío del comando “W2” al controlador del mismo. La respuesta a esta solicitud radica en los datos de la posición actual, tanto en coordenadas como en ángulos de cada articulación, como se muestra en la Tabla 4.4.

**Tabla 4.4 Respuesta del comando W2**

Orden	Respuesta					
<b>W2</b>	CURRENT ACTUAL POSITION:					
	NAME	AX#1/6	AX#2/7	AX#3/8	AX#4	AX#5
	PULSES	-0000001	-0000018	+0001916	-0000001	-0000002
	NAME	JT#1/6	JT#2/7	JT#3/8	JT#4	JT#5
	JOINTS	-000.0050	+000.0900	+009.5799	+000.0225	-000.1350
	NAME	X	Y	Z	YAW	PITCH ROLL
	WORLD	+021.8605	-000.0019	+011.6807	-00.0050	- -
						00.0225 00.1350

FUENTE: Elaboración propia

De esta respuesta se aíslan los datos de interés que se encuentran en la línea *JOINTS*, los cuales serán utilizados posteriormente.

La siguiente etapa de este subproceso consiste en **ajustar los ángulos** obtenidos de la posición del operador dentro de los rangos máximos y mínimos de movimiento del manipulador robótico, mostrados en la Tabla 3.6, del Capítulo 3.1.4.1.1. El ajuste se realiza comparando el valor del ángulo obtenido en cada articulación con su ángulo máximo y mínimo permitido, y reemplazando su valor por el límite correspondiente en caso de que lo exceda.

Una vez ajustados los ángulos se procede a aplicar la **restricción espacial** de movimiento, aplicando la Ecuación 3.5 según lo detallado en el Capítulo 3.1.4.1.2; si para la posición del operador el manipulador cumple con la restricción espacial, los datos de los ángulos se envían a la siguiente etapa, caso contrario la operación termina allí y se reinicia todo el proceso desde la adquisición de imágenes.

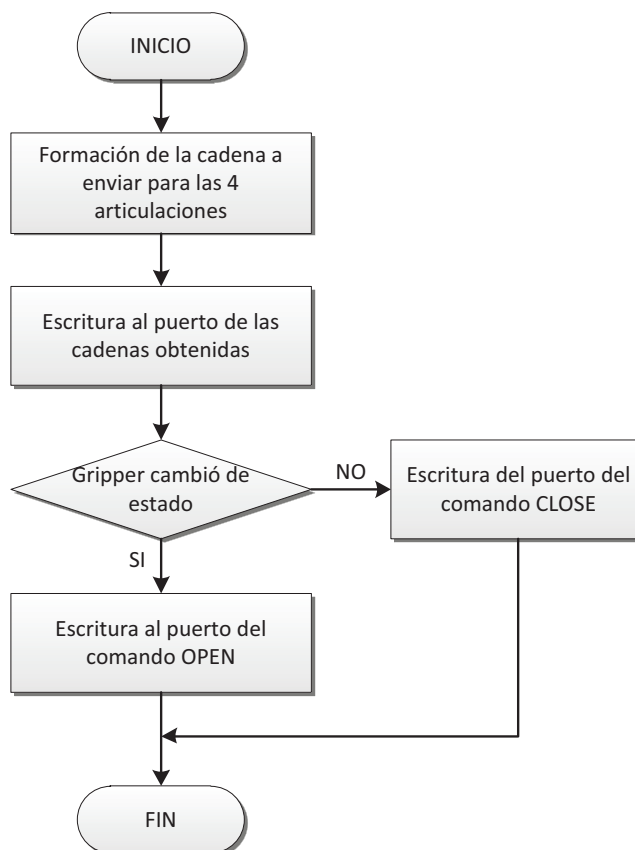
Finalmente, los ángulos obtenidos del operador, que fueron ajustados y cumplen con la restricción espacial, son utilizados para calcular los **ángulos a enviar** al manipulador, de acuerdo a la Ecuación 4.1.

$$\text{Ángulo a enviar} = \text{Posición deseada} - \text{Posición actual} \quad \text{Ecuación 4.1}$$

Como resultado de este subproceso se tiene:

- Ángulos a enviar para cuatro articulaciones
- Estado que debe adoptar el gripper del manipulador.

#### 4.2.3.2.7 Envío de datos



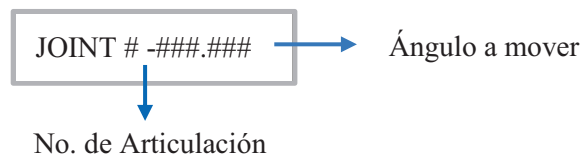
**Figura 4.20 Diagrama de flujo de la etapa de envío de datos**

FUENTE: Elaboración propia



El último subproceso que integra el programa para el control del manipulador es el envío de datos al controlador del manipulador, compuesto por las etapas mostradas en la Figura 4.20.

Tomando como base los ángulos a enviar, en la primera etapa del presente subproceso, **se estructuran las cadenas** que se enviarán al controlador del manipulador, bajo la sentencia *JOINT*, mostrada en la Figura 4.21.



**Figura 4.21 Estructura de la sentencia JOINT**

Para esta sentencia, el parámetro *No. de Articulación* se toma de acuerdo a la Tabla 3.6, del Capítulo 3.1.4.1.1, mientras el ángulo a moverse es el correspondiente a ella obtenido en el subproceso anterior.

Una vez estructuradas las cadenas, éstas son **enviadas** por comunicación serial, siempre y cuando el cambio en la posición de las articulaciones sea mayor a 4°. Esto se debe a que el proceso de esqueletización produce resultados diferentes en cada aplicación, lo cual se refleja en el proceso de aproximación como un aumento o disminución mínima en la medida de los ángulos, sin que necesariamente haya existido movimiento.

En el caso del **estado del gripper**, éste se envía mediante los comandos *OPEN* y *CLOSE*, únicamente si el programa detecta que ha existido un cambio respecto al estado anterior del mismo. El comando *OPEN* se utiliza para abrir el gripper, mientras el comando *CLOSE* es usado para cerrarlo.

Con el envío de datos al manipulador se termina el ciclo de ejecución del programa y se cumple el objetivo de la plataforma de control de movimiento, a través de las siguientes operaciones realizadas:

- Se adquirieron las imágenes desde las cámaras

- Se trataron las imágenes a través de filtros varios
- Las imágenes fueron esqueletizadas y se obtuvieron los ángulos de cada articulación
- Se ajustaron los ángulos de acuerdo a las capacidades del manipulador y se estableció el estado del gripper
- Por último, los datos de movimiento para el manipulador fueron enviados

En este punto, si el operador decide cerrar la aplicación se procede a parar la ejecución del hilo de adquisición de imágenes, se cierra el puerto serial y se restablece su configuración inicial; y, finalmente se destruye la interfaz gráfica, liberándose las variables que utiliza el programa y terminando por completo la aplicación.

#### **4.2.3.3 Estructura del Programa de Control del Manipulador**

El programa para el control del manipulador fue desarrollado en lenguaje C++, con interfaz Qt y la intervención de la librería OpenCV y LinxComm, como se mencionó en la sección 4.2.3. La programación se llevó a cabo en el IDE Code::Blocks con el compilador GCC.

Para la ejecución de todos sus subprocesos, el programa fue dividido en clases, las cuales están especializadas en actividades específicas, esto facilita la modificación del mismo en caso de aplicarlo sobre una plataforma robótica diferente. Estas clases son:

- *MainWindow*.- Esta clase es realmente el cuerpo principal del programa. Desde su archivo fuente se administra la intervención del resto de clases, se crean los objetos correspondientes a las mismas y se declaran las variables globales del programa.
- *esqueletizacion*.- En esta clase se encuentran los algoritmos de esqueletización usados y la función de conteo de puntos finales usada para la definición del estado del gripper.

- *medicion*.- Contiene las funciones de aproximación de líneas, medición de ángulos, diferenciación de casos para las cámaras superior y frontal, así como las operaciones matemáticas auxiliares para ello.
- *comunicacion*.- Tiene como base la librería LnxComm, en ella se encuentran todas las funciones que implican una comunicación entre el computador y el controlador del manipulador. Esta clase es la encargada de abrir, configurar y cerrar el puerto serial, formación de cadenas, escritura y lectura de datos, verificación del buffer del puerto; y, envío de la secuencia para la inicialización del manipulador robótico.
- *hiloCamaras*.- Esta clase es la encargada de adquirir las imágenes constantemente, además de enviarlas a la interfaz gráfica del programa. Tiene como base la librería Qt.
- *control*.- Dentro de esta clase se realiza el ajuste de ángulos para las cámaras superior y frontal, además de la aplicación de la restricción espacial del robot. También, en esta clase constan los valores de los rangos de movimiento del robot.

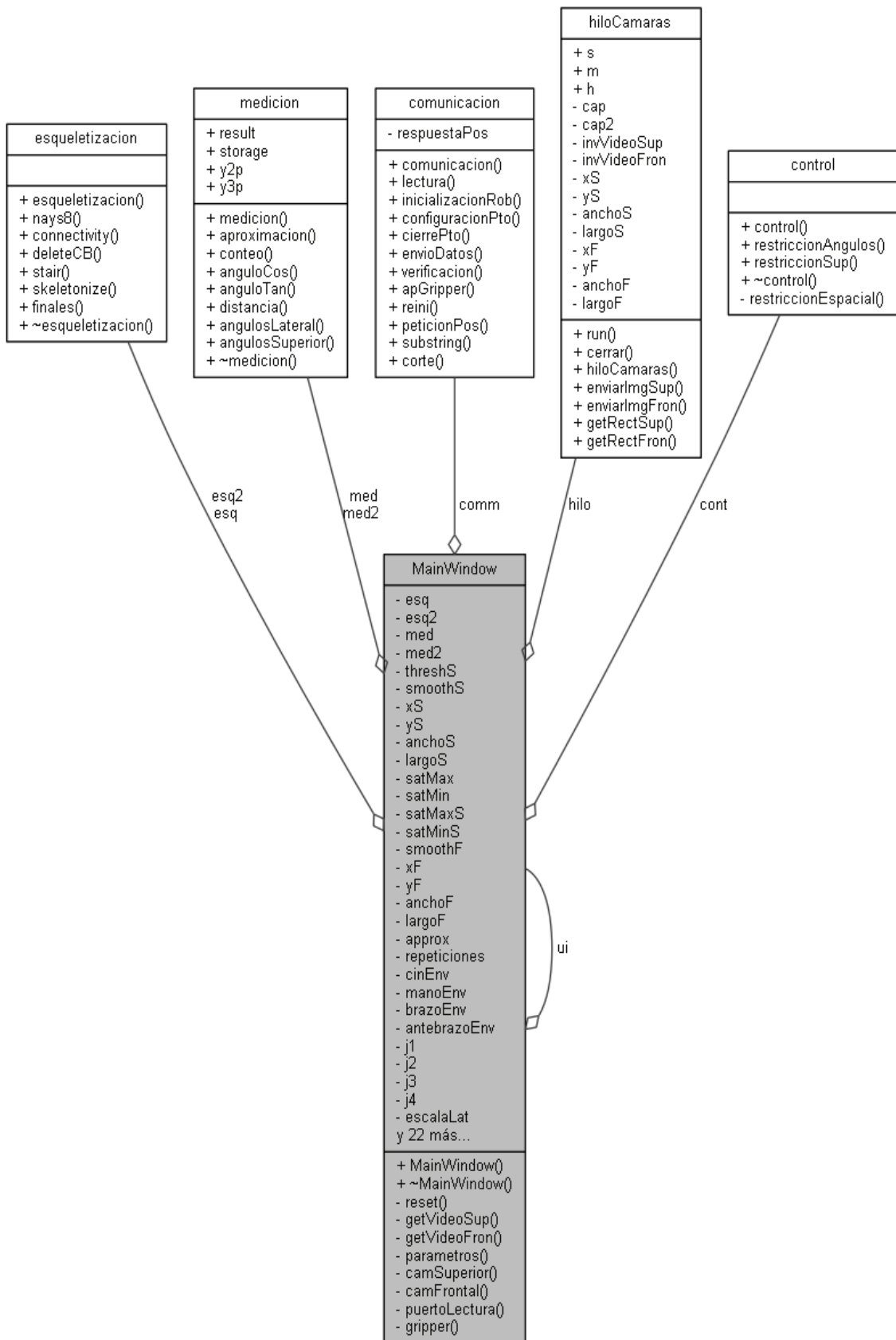
Estas clases se encuentran graficadas en el diagrama UML del software de la Figura 4.22.

Otro componente importante del programa es el archivo de enlace creado por Qt para la interfaz gráfica. Este archivo puede ser modificado directamente en el IDE o mediante la herramienta QtDesigner, para cambiar el aspecto y la posición de los diferentes elementos presentes en ella.

Adicionalmente los archivos *moc\_hiloCamaras* y *moc\_MainWindow* son esenciales para el manejo de eventos de la interfaz gráfica y la interrelación entre las señales y conectores.

Las dependencias de cada clase y de los archivos auxiliares se muestran en la Figura 4.23.

Para mayor información sobre el código fuente del programa, refiérase al Anexo D de este documento, en el que se encuentran documentadas a profundidad todas las funciones y variables usadas por el programa.



**Figura 4.22 Diagrama de clases del software**

FUENTE: Elaboración propia

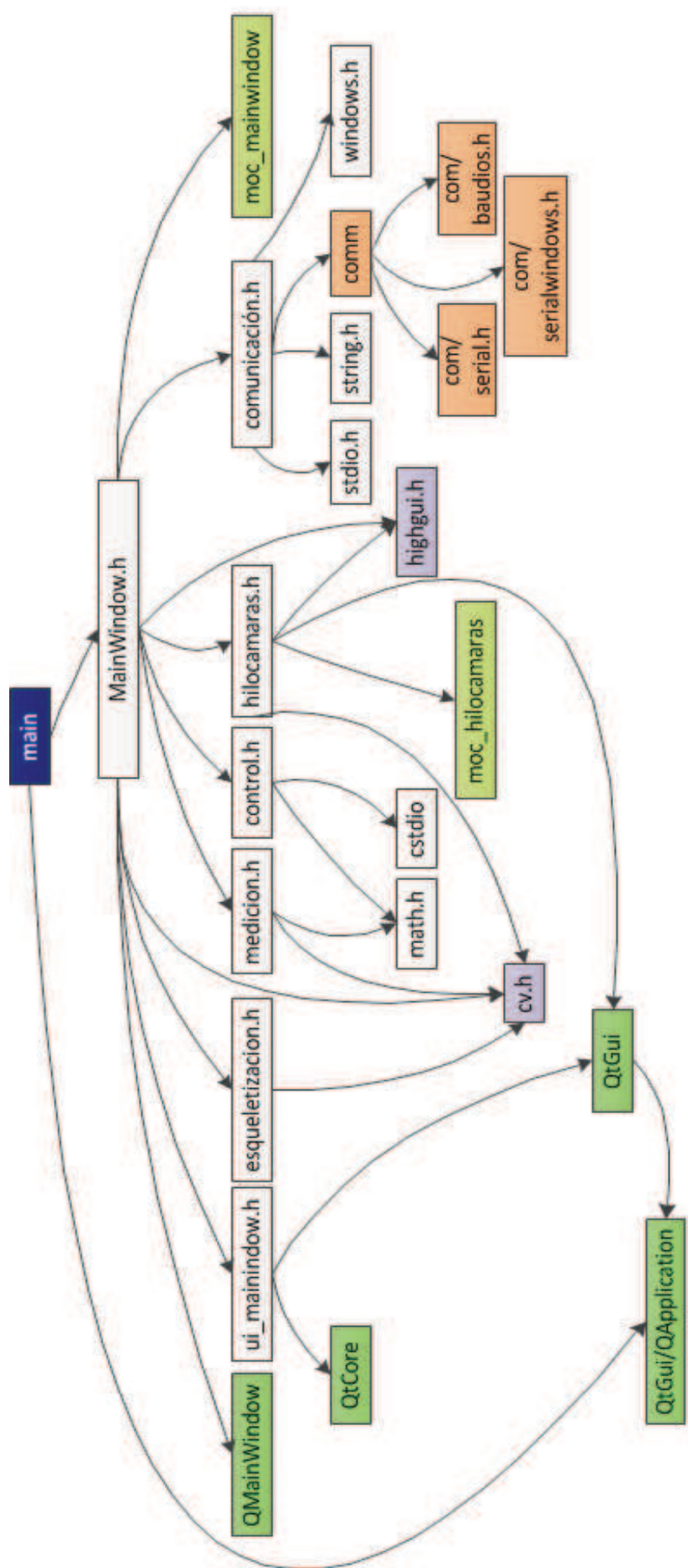


Figura 4.23 Dependencias del software

FUENTE: Elaboración propia

# **CAPITULO 5**

## **PRUEBAS Y RESULTADOS**

A lo largo de este documento se han detallado los elementos que componen la plataforma de control de movimiento por esqueletización, así como su estructura y funcionamiento. Este capítulo tratará sobre las pruebas realizadas sobre la plataforma para validar su funcionamiento y cuantificar la medida en la cual cumple con los objetivos propuestos en el Capítulo 1.

### **5.1 PRUEBAS**

Debido a que este proyecto trata sobre una plataforma, cuya meta es la reproducción exacta del movimiento de un operador, las variables que más relevancia tendrán para determinar su efectividad son:

- Exactitud en la reproducción del movimiento
- Precisión en la reproducción del movimiento

Para la cuantificación del error de estas variables, se aplicará una prueba de funcionamiento, cuyos datos obtenidos, al ser tabulados, arrojarán resultados que permitan evaluar integralmente a la plataforma.

En esta sección se detallará la ejecución de la prueba, así como los resultados obtenidos en ella. Estos datos serán analizados a profundidad más adelante.

#### **5.1.1 METODOLOGÍA DE LA PRUEBA Y VARIABLES A MEDIR**

Para la ejecución de la prueba se ha fijado como objetivo la medición de la exactitud y precisión en la reproducción del movimiento:

- La exactitud medirá qué tanto difiere la posición del manipulador de la posición que el operario de la plataforma adopta.
- La precisión arrojará cuánto varía la medida del ángulo de una de las articulaciones del manipulador para la misma posición del operador, en varias repeticiones.

Otro factor importante a determinar durante la prueba, es el error que existe entre el valor real y el valor teórico de un ángulo, es decir el valor mostrado en la interfaz gráfica.

En base a estas variables se estableció una metodología para la ejecución de la prueba, la cual se detalla en la Tabla 5.1.

**Tabla 5.1 Metodología para le ejecución de la prueba**

<b>Objetivo</b>	Cuantificar la exactitud y precisión en la reproducción del movimiento del operador por parte del manipulador.
<b>Técnica</b>	Comparación del ángulo real, medido en las articulaciones del operador, con el ángulo de las articulaciones del manipulador; y de éstos entre sí en varias repeticiones.
<b>Cuantificadores</b>	Exactitud: Error absoluto en la medición Precisión: Desviación estándar de los datos obtenidos
<b>Recursos</b>	<ul style="list-style-type: none"> <li>• Plataforma de Control de Movimiento por Esqueletización</li> <li>• Software CAD</li> <li>• Goniómetro</li> <li>• Cámara fotográfica</li> <li>• Operador</li> </ul>
<b>Metodología</b>	<ol style="list-style-type: none"> <li>1. Poner en marcha la plataforma de Control de Movimiento por Esqueletización</li> <li>2. Situar al operador dentro del campo de visión de la plataforma</li> <li>3. Ubicar una postura predefinida del operador en la cual participen las articulaciones de interés de su brazo.</li> <li>4. Tomar la medida del ángulo que forma cada articulación del</li> </ol>

brazo, con la ayuda del goniómetro

5. Capturar la medida de los ángulos obtenidos mediante la plataforma de Control de Movimiento por Esqueletización.
6. Tomar una fotografía del manipulador en la posición actual.
7. Registrar la medida de las articulaciones del manipulador para dicha posición.
8. Crear interferencia en la plataforma, de manera que se pierda la lectura anterior y repetir la prueba.
9. Realizar 5 repeticiones con la misma posición.
10. Obtener la medida del giro de la cintura del operador con el software CAD, utilizando la fotografía de la cámara superior.
11. Determinar la medida del giro de la cintura del manipulador, con ayuda del software CAD y la fotografía capturada.
12. Calcular el error absoluto entre las mediciones
13. Calcular la desviación estándar de los datos obtenidos.
14. Repetir el procedimiento completo con 7 posturas diferentes adicionales.

FUENTE: Elaboración propia

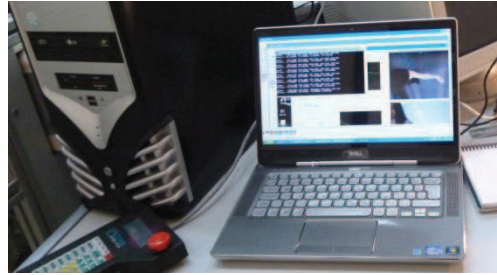
### 5.1.2 EJECUCIÓN DE LA PRUEBA

De acuerdo a la metodología planteada en la Sección 5.1.1, se ejecutaron las pruebas en 8 posiciones diferentes.

Las pruebas fueron realizadas en el laboratorio CIM-200 del Departamento de Eléctrica y Electrónica, en el que está ubicado el manipulador CRS-A255. En él se situaron todos los elementos de la plataforma, según la configuración analizada en el Capítulo 3.1.1.1.2, como se muestra en la Figura 5.1.

El instrumento de medición utilizado fue un goniómetro con rango de medición de 180° y apreciación de 1°, mostrado en la Figura 5.2.





(a)



(b)



(c)

**Figura 5.1 Configuración de la Plataforma de Control de Movimiento por Esqueletización para las pruebas. (a) Unidad de procesamiento ejecutando la aplicación; (b) Sistema de video e iluminación; (c) Ejecución de mediciones manuales en el Manipulador CRS-A255**

FUENTE: Elaboración propia



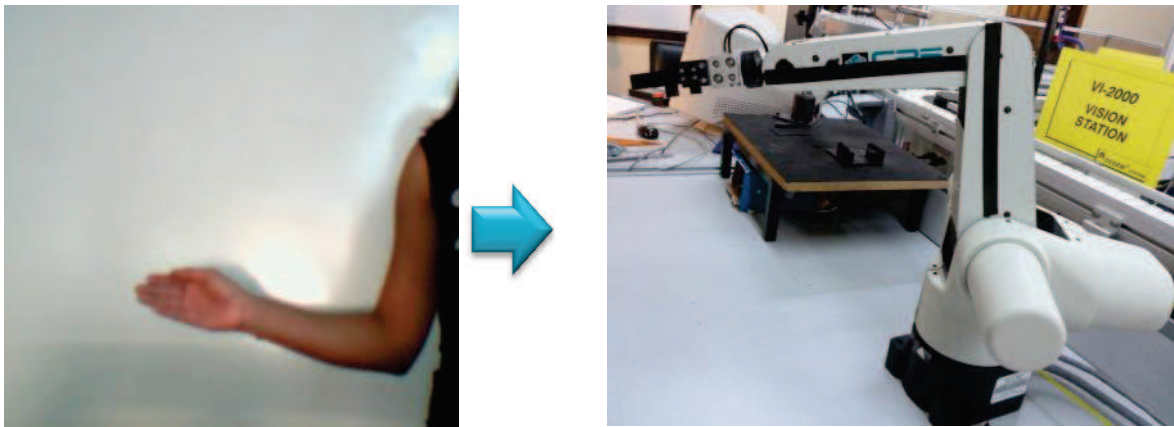
**Figura 5.2 Goniómetro usado durante las pruebas**

FUENTE: Elaboración propia

Para la ejecución de las pruebas, así como para el análisis de sus resultados, debe tomarse en cuenta la relación morfológica inversa que existe entre el brazo humano y el manipulador robótico, es decir, recordar que un movimiento del brazo hacia abajo se reflejará en un movimiento hacia arriba en el manipulador.

Los resultados obtenidos se muestran a lo largo de esta sección.

### 5.1.2.1 Posición 1



**Figura 5.3 Posición 1 adoptada por el operario y el manipulador**

FUENTE: Elaboración propia

**Tabla 5.2 Datos obtenidos para posición 1**

HUMANO	Codo		Hombro		Muñeca		Cintura	
Real	-10°		80°		N/A		-22°	
PLATAFORMA	MANIPULADOR				TEÓRICO			
	Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
1	-15°	75°	N/A <sup>x</sup>	-20°	-9,7°	79,1°	N/A	-20,6°
2	-17°	73°	N/A	-14°	-11°	76,5°	N/A	-18,4°
3	-17°	73°	N/A	-14°	-12°	70,2°	N/A	-14,9°
4	-17°	73°	N/A	-14°	-10°	76,5°	N/A	-17,4°
5	-16°	75°	N/A	-12°	-14°	72,3°	N/A	-13,4°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

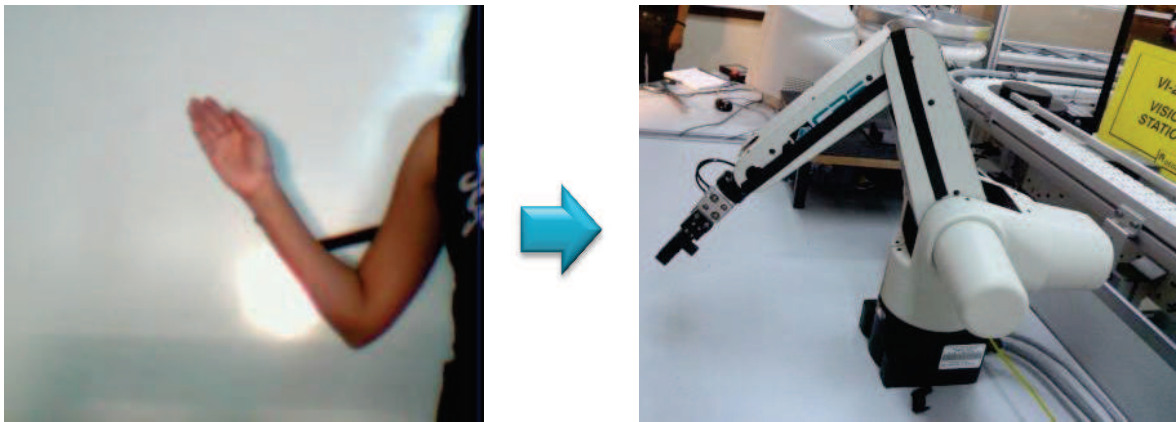
**Tabla 5.3 Resultados para posición 1**

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
-0,3°	0,9°	N/A	-1,4°	5°	5°	N/A	-2°
1°	3,5°	N/A	-3,6°	7°	7°	N/A	-8°
2°	9,8°	N/A	-7,1°	7°	7°	N/A	-8°
0°	3,5°	N/A	-4,6°	7°	7°	N/A	-8°
4°	7,7°	N/A	-8,6°	6°	5°	N/A	-10°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

<sup>x</sup> N/A = No aplica. Indica que esta articulación toma la medida de la articulación anterior pues está totalmente extendida respecto a ésta.

### 5.1.2.2 Posición 2



**Figura 5.4 Postura 2 adoptada por el operador y manipulador**

FUENTE: Elaboración propia

**Tabla 5.4 Datos obtenidos para posición 2**

HUMANO		Codo		Hombro		Muñeca		Cintura	
Real		-50° <sup>xi</sup>		60°		N/A		-16°	
PLATAFORMA		MANIPULADOR				TEÓRICO			
		Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
1		-48°	63°	N/A	-12°	-50,2°	64,4°	N/A	-9,7°
2		-48°	63°	N/A	-12°	-51,6°	58,2°	N/A	-11°
3		-48°	61°	N/A	-15°	-50,8°	60,3°	N/A	-13,7°
4		-48°	61°	N/A	-14°	-52,8°	58,2°	N/A	-15°
5		-47°	60°	N/A	-14°	-52,4°	59,4°	N/A	-17,6°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

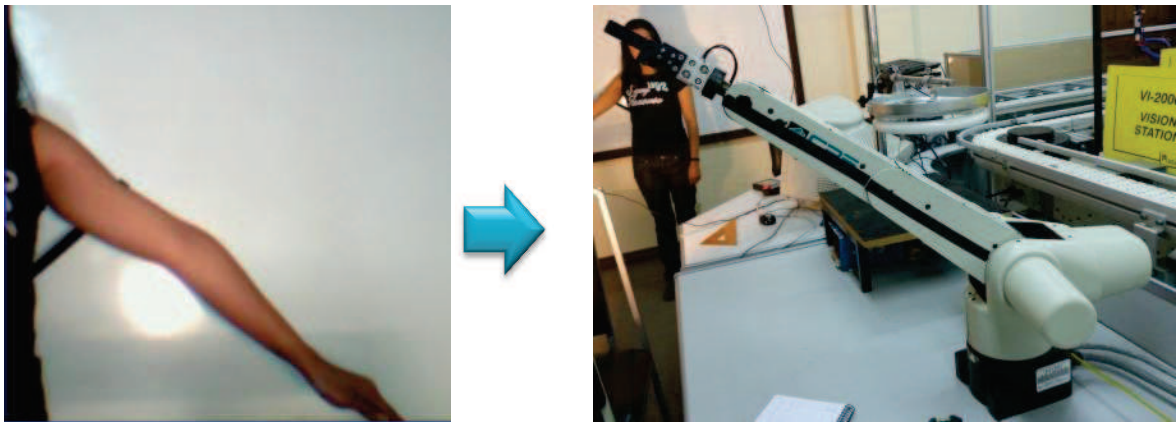
**Tabla 5.5 Resultados para posición 2**

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
0,2°	-4,4°	N/A	-6,3°	-2°	-3°	N/A	-4°
1,6°	1,8°	N/A	-5°	-2°	-3°	N/A	-4°
0,8°	-0,3°	N/A	-2,3°	-2°	-1°	N/A	-1°
2,8°	1,8°	N/A	-1°	-2°	-1°	N/A	-2°
2,4°	0,6°	N/A	1,6°	-3°	0°	N/A	-2°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

<sup>xi</sup> Por restricción de movimiento, este valor se toma automáticamente como -45° en la plataforma.

### 5.1.2.3 Posición 3



**Figura 5.5 Postura 3 adoptada por el operador y el manipulador**

FUENTE: Elaboración propia

**Tabla 5.6 Datos obtenidos para posición 3**

HUMANO		Codo		Hombro		Muñeca		Cintura	
Real		N/A		41°		N/A		-14°	
PLATAFORMA		MANIPULADOR				TEÓRICO			
		Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
1		N/A	40°	N/A	-13°	N/A	39,7°	N/A	-14°
2		N/A	40°	N/A	-14°	N/A	37,7°	N/A	-14°
3		N/A	40°	N/A	-13°	N/A	37,4°	N/A	-16,7°
4		N/A	40°	N/A	-15°	N/A	38°	N/A	-15,5°
5		N/A	40°	N/A	-13°	N/A	37,9°	N/A	-15,5°

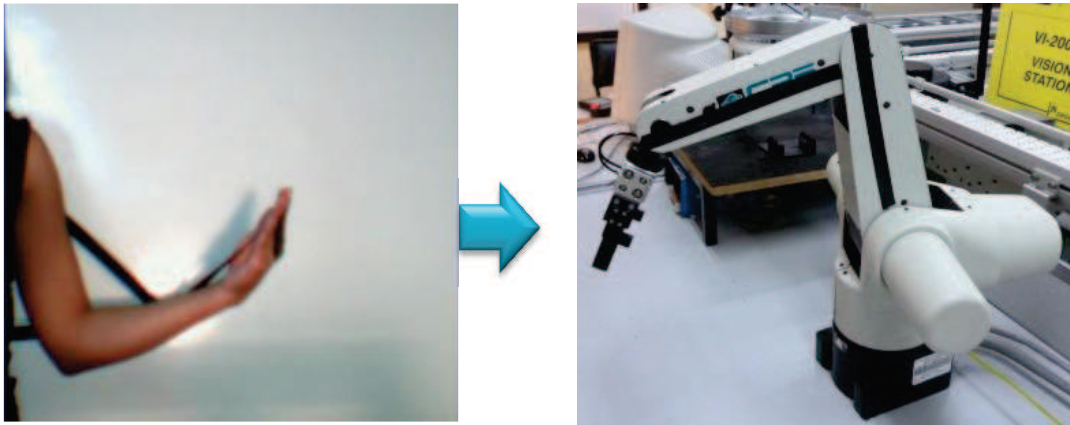
[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

**Tabla 5.7 Resultados para posición 3**

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
N/A	1,3°	N/A	0°	N/A	1°	N/A	-1°
N/A	3,3°	N/A	0°	N/A	1°	N/A	0°
N/A	3,6°	N/A	2,7°	N/A	1°	N/A	-1°
N/A	3°	N/A	1,5°	N/A	1°	N/A	1°
N/A	3,1°	N/A	1,5°	N/A	1°	N/A	-1°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

#### 5.1.2.4 Posición 4



**Figura 5.6** Postura 4 adoptada por el operador y manipulador

FUENTE: Elaboración propia

**Tabla 5.8** Datos obtenidos para la posición 4

HUMANO	Codo		Hombro		Muñeca		Cintura	
Real	-24°		76°		-69°		-15°	
PLATAFORMA	MANIPULADOR				TEÓRICO			
	Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
1	-19°	79°	-61°	-14°	-15,6°	76,8°	-59,5°	-16,7°
2	-17°	77°	-64°	-13°	-14°	75,5°	-54,5°	-14,4°
3	-17°	77°	-64°	-13°	-16,3°	73,3°	-56,3°	-12,9°
4	-17°	77°	-64°	-13°	-15,1°	74,6°	-54,5°	-11,3°
5	-17°	77°	-64°	-13°	-15,1°	74,6°	-57,3°	-10°

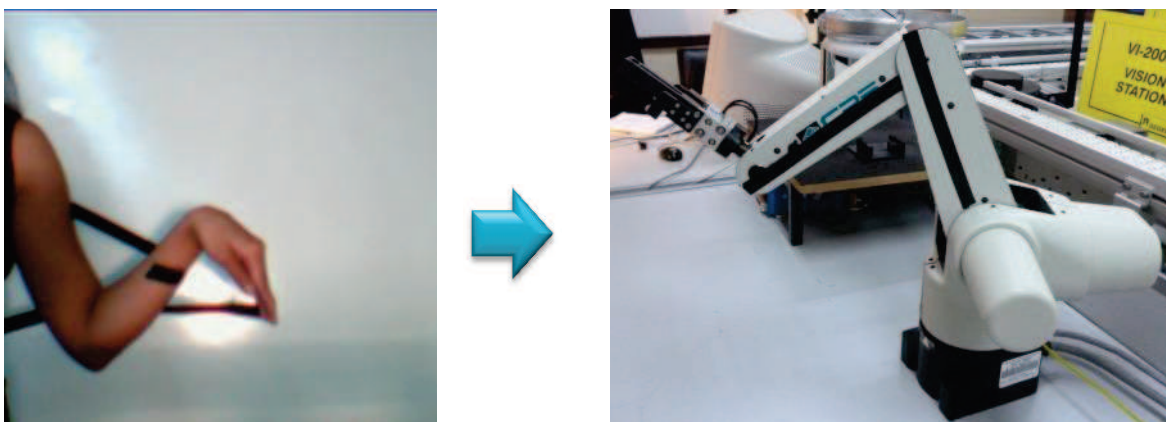
[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

**Tabla 5.9** Resultados para posición 4

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
-8,4°	-0,8°	-9,5°	1,7°	-5°	-3°	-8°	-1°
-10°	0,5°	-14,5°	-0,6°	-7°	-1°	-5°	-2°
-7,7°	2,7°	-12,7°	-2,1°	-7°	-1°	-5°	-2°
-8,9°	1,4°	-14,5°	-3,7°	-7°	-1°	-5°	-2°
-8,9°	1,4°	-11,7°	-5°	-7°	-1°	-5°	-2°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

### 5.1.2.5 Posición 5



**Figura 5.7 Postura 5 adoptada por el operador y manipulador**

FUENTE: Elaboración propia

**Tabla 5.10 Datos obtenidos para la posición 5**

HUMANO		Codo		Hombro		Muñeca		Cintura	
Real		-24°		72°		-114°		-22°	
PLATAFORMA	MANIPULADOR				TEÓRICO				
	Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura	
1	-24°	75°	-112°	-16°	-22,9°	74,1°	-120°	-19,3°	
2	-24°	75°	-112°	-15°	-25,6°	72,1°	-120°	-15,9°	
3	-24°	75°	-112°	-14°	-24,6°	72,1°	-117°	-16,7°	
4	-24°	75°	-112°	-14°	-27,6°	70,5°	-120°	-15,4°	
5	-24°	75°	-112°	-14°	-27,6°	69,9°	-120°	-15,4°	

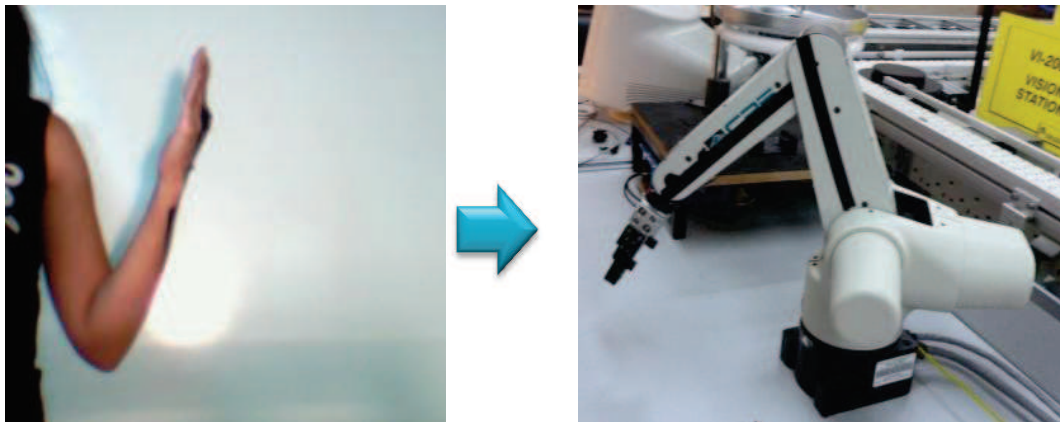
[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

**Tabla 5.11 Resultados para posición 5**

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
-1,1°	-2,1°	6°	-2,7°	0°	-3°	-2°	-4°
1,6°	-0,1°	6°	-6,1°	0°	-3°	-2°	-8°
0,6°	-0,1°	3°	-5,3°	0°	-3°	-2°	-9°
3,6°	1,5°	6°	-6,6°	0°	-3°	-2°	-9°
3,6°	2,1°	6°	-6,6°	0°	-3°	-2°	-9°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

### 5.1.2.6 Posición 6



**Figura 5.8 Postura 6 adoptada por el operador y el manipulador**

FUENTE: Elaboración propia

**Tabla 5.12 Datos obtenidos para posición 6**

HUMANO	Codo		Hombro		Muñeca		Cintura	
Real	$-48^{\circ}$ <sup>xii</sup>		$72^{\circ}$		N/A		$-28^{\circ}$	
PLATAFORMA	MANIPULADOR				TEÓRICO			
	Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
1	$-45^{\circ}$	$71^{\circ}$	N/A	$-21^{\circ}$	$-63,4^{\circ}$	$72,3^{\circ}$	N/A	$-32,2^{\circ}$
2	$-46^{\circ}$	$72^{\circ}$	N/A	$-22^{\circ}$	$-61,4^{\circ}$	$72,9^{\circ}$	N/A	$-31^{\circ}$
3	$-49^{\circ}$	$70^{\circ}$	N/A	$-20^{\circ}$	$-61,3^{\circ}$	$70,9^{\circ}$	N/A	$-31,5^{\circ}$
4	$-47^{\circ}$	$72^{\circ}$	N/A	$-24^{\circ}$	$-62,7^{\circ}$	$72,2^{\circ}$	N/A	$-33,3^{\circ}$

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

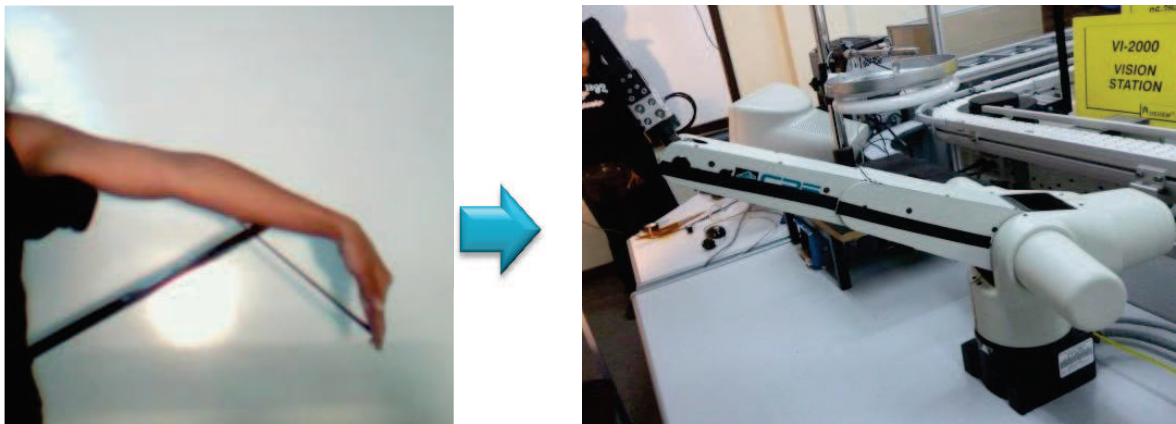
**Tabla 5.13 Resultados para posición 6**

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
$15,4^{\circ}$	$-0,3^{\circ}$	N/A	$4,2^{\circ}$	$-3^{\circ}$	$1^{\circ}$	N/A	$-7^{\circ}$
$13,4^{\circ}$	$-0,9^{\circ}$	N/A	$3^{\circ}$	$-2^{\circ}$	$0^{\circ}$	N/A	$-6^{\circ}$
$13,3^{\circ}$	$1,1^{\circ}$	N/A	$3,5^{\circ}$	$1^{\circ}$	$2^{\circ}$	N/A	$-8^{\circ}$
$14,7^{\circ}$	$-0,2^{\circ}$	N/A	$5,3^{\circ}$	$-1^{\circ}$	$0^{\circ}$	N/A	$-4^{\circ}$

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

<sup>xii</sup> Por restricción de movimiento, este valor se ajustará automáticamente a  $45^{\circ}$ .

### 5.1.2.7 Posición 7



**Figura 5.9 Postura 7 adoptada por el operador y manipulador**

FUENTE: Elaboración propia

**Tabla 5.14 Datos obtenidos para posición 7**

HUMANO		Codo		Hombro		Muñeca		Cintura	
Real		N/A		20°		65°		-16°	
PLATAFORMA		MANIPULADOR				TEÓRICO			
	Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura	
1	N/A	15°	62°	-14°	N/A	15,4°	57,5°	-13,4°	
2	N/A	15°	62°	-16°	N/A	18,8°	54°	-10,8°	
3	N/A	15°	62°	-15°	N/A	17,7°	54°	-13,4°	
4	N/A	15°	62°	-13°	N/A	17,7°	59,7°	-14,7°	
5	N/A	15°	62°	-13°	N/A	19,1°	54°	-14,7°	

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

**Tabla 5.15 Resultados para posición 7**

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
N/A	4,6°	7,5°	-2,6°	N/A	5°	3°	-2°
N/A	1,2°	11°	-5,2°	N/A	5°	3°	0°
N/A	2,3°	11°	-2,6°	N/A	5°	3°	-1°
N/A	2,3°	5,3°	-1,3°	N/A	5°	3°	-3°
N/A	0,9°	11°	-1,3°	N/A	5°	3°	-3°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia



### 5.1.2.8 Posición 8

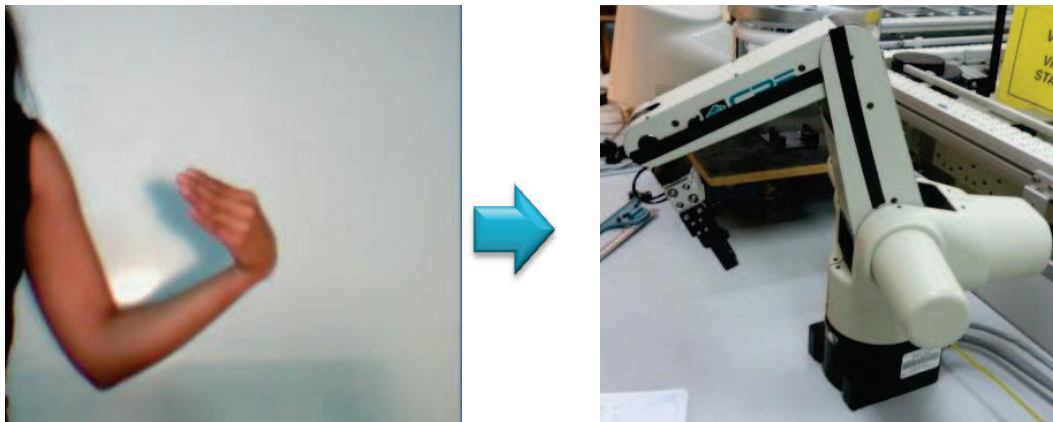


Figura 5.10 Postura 8 adoptada por el operador y manipulador

FUENTE: Elaboración propia

Tabla 5.16 Resultados obtenidos para posición 8

HUMANO	Codo		Hombro		Muñeca		Cintura	
Real	-36°		71°		40°		-22°	
PLATAFORMA	MANIPULADOR				TEÓRICO			
	Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
1	-35°	70°	40°	-18°	-32,5°	70,4°	45°	-24,3°
2	-35°	71°	41°	-14°	-33°	69,7°	34,7°	-15,9°
3	-32°	72°	43°	-13°	-31,3°	70,4°	45°	-10,8°
4	-32°	72°	43°	-13°	-34,3°	69,7°	45°	-14,7°
5	-36°	72°	37°	-13°	-33,7°	68,6°	35°	-13,4°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

Tabla 5.17 Resultados para posición 8

ERROR OPERADOR-V TEÓRICO				ERROR OPERADOR-MANIPULADOR			
Codo	Hombro	Muñeca	Cintura	Codo	Hombro	Muñeca	Cintura
-3,5°	0,6°	-5°	2,3°	-1°	1°	0°	-6°
-3°	1,3°	5,3°	-6,1°	-1°	0°	-1°	-7°
-4,7°	0,6°	-5°	-11,2°	-4°	-1°	-3°	-8°
-1,7°	1,3°	-5°	-7,3°	-4°	-1°	-3°	-8°
-2,3°	2,4°	5°	-8,6°	0°	-1°	3°	-8°

[Medidas en grados sexagesimales]. FUENTE: Elaboración propia

## 5.2 ANÁLISIS DE RESULTADOS

En la Tabla 5.18, se muestran los resultados más relevantes para el análisis de eficiencia de la plataforma de control de movimiento por esqueletización.

**Tabla 5.18 Resultados de interés de las pruebas**

Pos.	Característica	Unidades	Error Operador-Manipulador			
			Codo	Hombro	Muñeca	Cintura
1	Error absoluto promedio	grados sexagesimales	6,4	6,2	N/A	7,2
	Error relativo promedio	(%)	64	7,75	N/A	32,7
	Desviación estándar	grados sexagesimales	0,8	1	N/A	2,7
2	Error absoluto promedio	grados sexagesimales	2,2	1,6	N/A	2,6
	Error relativo promedio	(%)	4,4	2,7	N/A	16,3
	Desviación estándar	grados sexagesimales	0,4	1,2	N/A	1,2
3	Error absoluto promedio	grados sexagesimales	N/A	1	N/A	0,4
	Error relativo promedio	(%)	N/A	2,4	N/A	5,7
	Desviación estándar	grados sexagesimales	N/A	0	N/A	0,8
4	Error absoluto promedio	grados sexagesimales	6,6	1,4	5,6	1,8
	Error relativo promedio	(%)	27,5	1,8	8,1	12
	Desviación estándar	grados sexagesimales	0,8	0,8	1,2	0,4
5	Error absoluto promedio	grados sexagesimales	0	3	2	7,8
	Error relativo promedio	(%)	0	4,2	1,8	35,5
	Desviación estándar	grados sexagesimales	0	0	0	1,9
6	Error absoluto promedio	grados sexagesimales	1,2	0,6	N/A	6,3
	Error relativo promedio	(%)	3,3	0,8	N/A	17,9
	Desviación estándar	grados sexagesimales	1,3	0,8	N/A	1,5
7	Error absoluto promedio	grados sexagesimales	N/A	5	3	1,8

	Error relativo promedio	(%)	N/A	25	4,6	11,3
	Desviación estándar	grados sexagesimales	N/A	0	0	1,2
<b>8</b>	Error absoluto promedio	grados sexagesimales	2	0,3	0,6	7,3
	Error relativo promedio	(%)	5,6	1,1	5	35,5
	Desviación estándar	grados sexagesimales	1,7	0,8	2,2	0,8

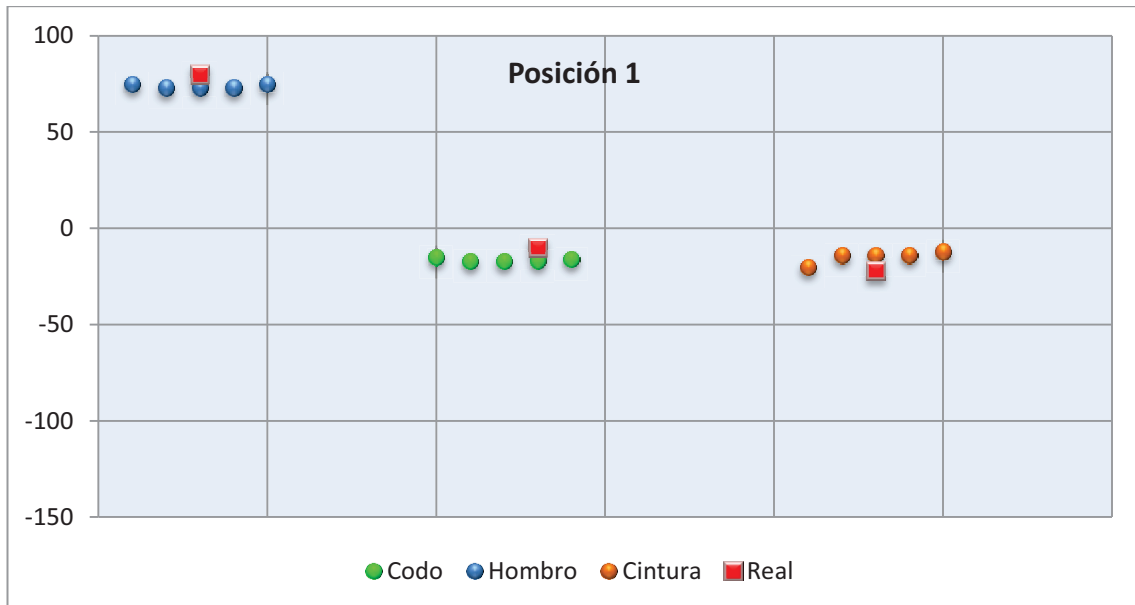
FUENTE: Elaboración propia

Como se puede observar, el error relativo muestra valores muy grandes en ciertas mediciones, sin embargo su correspondiente error absoluto no es un valor alto. Por ejemplo, tomando la posición 1, su valor original es de  $10^\circ$  y la medida del manipulador es de  $15^\circ$ , lo cual representa un error relativo del 50% pese a que la diferencia real es de  $5^\circ$ . En general la plataforma de control de movimiento muestra una diferencia de aproximadamente  $5^\circ$  entre la postura del operador y la del manipulador, pero en medidas pequeñas estos  $5^\circ$  son muy significativos, mientras que en otras posturas con ángulos mayores, esta diferencia representa un porcentaje muy bajo.

Adicionalmente, se debe considerar las restricciones de la plataforma para el envío de datos al manipulador. Como se mencionó en el Capítulo 4.2.3.2.7, para evitar movimientos cortos se asignó una limitación de **4 grados de diferencia mínima entre un movimiento y otro** para ser considerado válido y ser enviado al controlador. Analizando nuevamente la postura 1, esa limitación representa que en la medición de  $17^\circ$  pudo generarse un valor con error muy alto la primera vez, en la segunda este valor pudo bajar hasta  $13^\circ$ , pero debido a esta limitación el manipulador no corrigió su postura, manteniendo este error a lo largo de otras mediciones hasta que la diferencia sea mayor a  $4^\circ$ .

Por estas razones, para el análisis de los resultados se utilizará como base el error absoluto y no el error relativo, pues éste nos dará una mejor visión del comportamiento de la plataforma. Una mejor representación porcentual del error es el porcentaje del rango de movimiento que representa el error absoluto, tomando en cuenta que éste es de aproximadamente  $90^\circ$ .

En lo referente a la precisión, las posturas que presentaron un mayor error absoluto promedio son: la postura 1, con el mayor error en la medición del hombro; la postura 4, en el codo y muñeca; y, la postura 5 en la cintura; éstas se muestran en la Figura 5.11.



(a)



(b)



(c)

**Figura 5.11 Representación gráfica de las posturas con mayor error respecto al manipulador: (a) Postura 1, de acuerdo a los datos presentados en la Tabla 5.2; (b) Postura 4, de acuerdo a los datos de la Tabla 5.8; (c) Postura 5, de acuerdo a los datos de la Tabla 5.10.**

FUENTE: Elaboración propia

En general, se observa que las articulaciones más exactas y precisas en sus mediciones son aquellas cuyos segmentos del esqueleto son más largos, en este caso la medida del hombro y el codo, y aquellas con segmentos cortos, reflejarán mayor diferencia con respecto al operador.

Considerando los datos obtenidos de las pruebas, se puede afirmar que la plataforma de Control de Movimiento por Esqueletización presenta las características mostradas en la Tabla 5.19, las cuales son aceptables para facilitar el manejo del manipulador, pero deficientes para una aplicación que requiera precisión.

**Tabla 5.19 Características generales de la plataforma**

<b>Característica</b>	<b>Valor</b>
<b>Error absoluto promedio</b>	5 °
<b>Porcentaje correspondiente del rango de movimiento</b>	5,5%
<b>Desviación estándar promedio</b>	2 °

FUENTE: Elaboración propia

Cabe resaltar que al manejar el manipulador mediante su control manual los resultados obtenidos serían similares, pues por la antigüedad de éste, sus ejes presentan una ligera descalibración, generando errores sistemáticos y posturas igualmente imprecisas.

## CAPITULO 6

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 CONCLUSIONES

En base a los contenidos mostrados a lo largo de este documento, se concluye que:

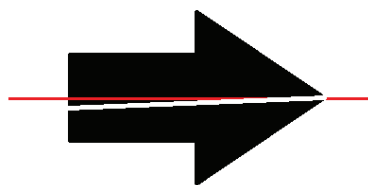
- Los algoritmos de esqueletización son técnicas poderosas de compresión y pre-procesamiento de imágenes; de acuerdo al análisis presentado en el Capítulo 2.3, el de Zhang-Suen es el más aplicable a la plataforma planteada. Dicho algoritmo, de la manera que fue implementado en este trabajo, es capaz de procesar las imágenes de hasta 80x60 pixeles en tiempo real; sin embargo, posee limitaciones en cuanto a centricidad y estabilidad del esqueleto generado.
- Se consigue un control de movimiento del manipulador mediante comandos de movimientos individuales para cada articulación y la aplicación de una velocidad del 40%, con lo cual se logra la imitación de la postura del operador en un tiempo aproximado de 30 segundos, lo que permite un control más rápido que el realizado mediante el control manual (*teach pendant*) del manipulador. No fue posible realizar un control de movimiento del manipulador mediante comandos de movimiento continuo, debido a que el controlador del mismo no dispone de esta opción.
- La integración del sistema de visión artificial con el manipulador CRS-A255 es posible gracias a la librería de comunicación serial y los ángulos calculados a partir del esqueleto generado. La analogía morfológica planteada en el capítulo 3.1.4 facilitó la correspondencia de movimiento del

operador y manipulador para las articulaciones de cintura, codo, hombro y muñeca. Empero, debido al carácter bidimensional de la técnica de esqueletización, no es posible controlar el giro axial de la muñeca de esta manera, por lo cual el control se limitó a cuatro articulaciones.

- La plataforma implementada en este proyecto presenta aproximadamente 5 grados de diferencia entre la posición del operador y del manipulador, además de una desviación estándar de 2 grados para una misma posición, de acuerdo los resultados mostrados en el Capítulo 5.

Varios son los factores que pudieron afectar los resultados obtenidos para las pruebas de la plataforma tratada en este documento, debido a la técnica utilizada, por ejemplo:

- Se introducen errores de exactitud debido a que no se ha creado un algoritmo de esqueletización que genere un esqueleto totalmente céntrico, por lo cual la inclinación de las líneas puede algunos grados de la posición del eje central del brazo del operador (ver Figura 6.1). Este es el caso de la postura 1, el brazo está muy próximo al torso, creando sombras entre ellos y facilitando la obtención de una línea que no coincide con el eje real del brazo.



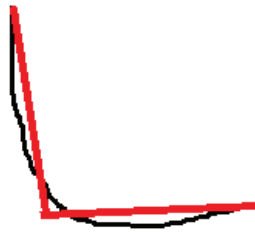
**Figura 6.1 Esqueleto no céntrico**

FUENTE: Elaboración propia

- En cuanto al procedimiento usado por la plataforma, se debe tomar en cuenta el proceso de aproximación al que son sometidos los puntos que forman el esqueleto, pues éste puede omitir puntos fuera del rango de tolerancia que afectan la inclinación final de las líneas, como



se muestra en la Figura 6.2. Este es el caso de los errores en la posición 4, donde el motivo es la diferencia tan sutil entre la posición del antebrazo y la de la palma de la mano, pues al aproximar los puntos la aplicación tomará un punto medio entre la posición de ambos para determinar el ángulo del codo y la inclinación de la porción más alta de la mano (los dedos), para el ángulo de la muñeca.



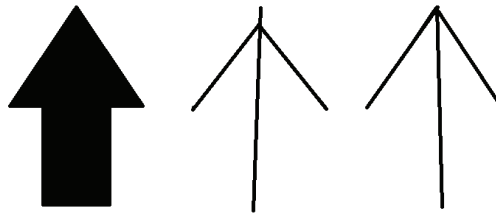
**Figura 6.2 Pérdida en aproximación de líneas**

FUENTE: Elaboración propia

- Los elementos pequeños del esqueleto, como el caso de la muñeca y la referencia para medición del giro de la cintura, son los elementos más vulnerables a errores. Al ser estos segmentos muy cortos y pasar por un redimensionamiento pueden eliminarse píxeles que ayudarían a una mejor determinación de la inclinación de la línea final, lo cual con el proceso de aproximación puede incluso llegar a perderse. Este es el caso de los errores a nivel general de la articulación de la cintura, en especial en la posición 5.

En lo que respecta a la precisión en la reproducción de las diferentes posiciones, se deben añadir otros aspectos como:

- Los algoritmos de esqueletización presentan limitaciones en cuanto a precisión, pues no siempre generan el mismo resultado para una misma imagen, como se muestra en la Figura 6.3.



**Figura 6.3 Resultados aleatorios de esqueletización**

FUENTE: Elaboración propia

- También se debe considerar las restricciones de la plataforma para el envío de datos al manipulador. Como se mencionó en el Capítulo 4.2.3.2.7, para evitar movimientos cortos se asignó una limitación de 4 grados de diferencia mínima entre un movimiento y otro para ser considerado como válido y ser enviado al controlador.

En base a estos factores y a los resultados obtenidos, se puede afirmar que la plataforma implementada en este proyecto facilita el manejo del manipulador CRS-A255, pero su aplicación se limita a operaciones que no requieran de una alta precisión.

## 6.2 RECOMENDACIONES

En base a las conclusiones de este trabajo y para la implementación de trabajos futuros, se sugiere:

- Utilizar una técnica más exacta para la generación del esqueleto, sea ésta un algoritmo de esqueletización más avanzado o mediante segmentación y reconocimiento de puntos clave, como lo realiza el sensor Kinect de Microsoft. Estas técnicas permitirán identificar con mayor exactitud las articulaciones y sus ángulos de movimiento, con lo cual se eliminan los errores provocados por la aproximación de líneas en esta plataforma.

- Probar el funcionamiento de esta plataforma en manipuladores robóticos que posean la opción de movimiento continuo, para lo cual debería modificarse únicamente el envío de los comandos de movimiento, de acuerdo al formato aceptado por el controlador correspondiente.
- Introducir un giroscopio como parte de la plataforma, situado en la muñeca del controlador para detectar su giro axial y de esta manera lograr la totalidad del control del manipulador. También, se sugiere buscar una analogía diferente y más precisa para el movimiento de la cintura, ya que en este movimiento se producen los errores de mayor magnitud.
- Con la implementación de una nueva técnica de esqueletización se espera una reducción drástica de la diferencia entre las posturas del operador y el manipulador. No obstante, si se mantiene la misma técnica, se sugiere trabajar con imágenes de mayor tamaño para la generación del esqueleto, de manera que la apreciación de los segmentos pequeños sea mejor, así como la potenciación del filtro de color de piel, lo cual aportaría mayor precisión en la delimitación de las partes del brazo, previo a la esqueletización.

## REFERENCIAS

- <sup>1</sup> MARR, D. *Vision*. Editorial Freeman. Estados Unidos. 1982
- <sup>2</sup> PALAZZESI, A. (2007). CCD vs. CMOS [Artículo]. Recuperado de: <http://www.neoteo.com/ccd-vs-cmos>
- <sup>3</sup> SILVELA, J. (2000). Sistema eficiente de reconocimiento de gestos de la mano [Disertación de tesis]. Universidad Politécnica de Madrid. España. Recuperado de: <http://www.silvela.org/jaime/PFC.pdf>
- <sup>4</sup> SAEED, K., TABE\_DZKI, M., RYBNIK, M. & ADAMSKI, M. (2010). K3M: A Universal Algorithm for Image Skeletonization and Review of Thinning Techniques. [Artículo]. *Int. J. Appl. Math. Comput. Sci. Volumen 20 (2)* 317-335. DOI: 10.2478/v10006-010-0024-4
- <sup>5</sup> COUPRIE, M. (2006) Note on 2D parallel thinning algorithms. Universite Paris-Est Marne-la-Vallee [Disertación de tesis]. Recuperado de: [http://igm.univ-mlv.fr/LIGM/internal\\_report/pdf/2006\\_01.pdf](http://igm.univ-mlv.fr/LIGM/internal_report/pdf/2006_01.pdf)
- <sup>6</sup> GALLEGOS, G. (2005). Convergencia multi-reticular de propiedades de los esqueletos continuos y discretos generados mediante adelgazamiento [Disertación de tesis]. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. México. Recuperado de: <http://www.ctrl.cinvestav.mx/~biene/ThesisFiles/TesSkelMultiResolGaby.pdf>
- <sup>7</sup>ZHANG, T., SUEN, C. (1984). A fast parallel algorithm for thinning digital patterns [Artículo]. *Communications of the ACM*, volumen 27, número 3. Estados Unidos.

<sup>8</sup> DEL RAYO, M. (2004). Adelgazamiento en imágenes digitales de dimensión dos modeladas por complejos celulares cuadrático [Disertación de tesis]. Centro De Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. México.

<sup>9</sup> ANAK, A. (2005). Enhancement of parallel thinning algorithm for handwritten characters using neural network. Universiti Teknologi Malaysia. [Disertación de tesis] Recuperado de: <http://eprints.utm.my/3796/1/AdelineEngkamatMCD2005TTT.pdf>

<sup>10</sup> JOHNSON, J. (2009). El factor multiplicativo de los sensores. Aprendiendo una pizca de Fotografía Digital [Blog]. Recuperado de: <http://aprendiendofotografiadigital.wordpress.com/2009/03/21/el-factor-multiplicativo-de-los-sensores/>

<sup>11</sup> HORCAJADA, R. (2011). Apuntes generales de anatomía morfológica aplicada. [Apuntes de clase]. Universidad Complutense de Madrid. Recuperado de: [http://eprints.ucm.es/13413/1/C%C3%81NONES\\_ANTROPOM%C3%89TRICOS\\_APLICADOS\\_AL\\_DIBUJO\\_DE\\_FIGURA.pdf](http://eprints.ucm.es/13413/1/C%C3%81NONES_ANTROPOM%C3%89TRICOS_APLICADOS_AL_DIBUJO_DE_FIGURA.pdf)

<sup>12</sup> MELO, J. (2001). Ergonomía aplicada a las herramientas. [Apuntes de clase]. Argentina. Disponible en: <http://www.estrucplan.com.ar/Producciones/imprimir.asp?Id Entrega=66>

<sup>13</sup> VARIOS.(2008) Cintura escapular [Apuntes de clase]. Facultad de Kinesiología. Universidad Nacional del Nordeste. Argentina. Disponible en: <http://www.med.unne.edu.ar/kinesiologia/catedras/biomecanica/clases/escapula.pdf>

<sup>14</sup> PÉREZ, J. (2006). La valoración del daño corporal en la articulación del codo. [Apuntes de clase]. Disponible en: [www.portalciencia.net/vdc/Codo.pdf](http://www.portalciencia.net/vdc/Codo.pdf)

<sup>15</sup> GODÍNEZ, I. (2011) Arcos y ángulos de movimiento. [Apuntes de clase]. Universidad Nacional Autónoma de México. México. Disponible en: <http://es.scribd.com/doc/52756563/Angulos-y-Arcos-de-Movimiento>

<sup>16</sup> Varios. (2000). USB Specification Expanding, Boosting Performance Up to 40 Times Beyond Current Capability. [Página web] Disponible en: <http://www.usb.org/developers/usb20/background/>

<sup>17</sup> PUJIACO, F. (2011). LnxCOMM :: Linux Communication. [Página web] Disponible en: <http://lnxcomm.sourceforge.net/refman/html/index.html>

<sup>18</sup> BRADSKY, G. (2012). OpenCV. Información. [Wiki]. Disponible en: <http://opencv.willowgarage.com/wiki/Welcome>

<sup>19</sup> CORTÉS, X. (2010). jRobcomm – Implementació amb Java d'un controlador per al Braç Robot A255. [Disertación de tesis]. Universitat Rovira i Virgili. España. Disponible en: [https://www.assembla.com/spaces/free\\_robcomm/documents/cEm8eqWBqr36KleJe5cbLA/download?filename=Documentaci%C3%B3.odt](https://www.assembla.com/spaces/free_robcomm/documents/cEm8eqWBqr36KleJe5cbLA/download?filename=Documentaci%C3%B3.odt).

<sup>20</sup> APLITOP (2011). Algoritmo de Douglas-Peucker. Curvado. [Página web]. España. Disponible en: <http://www.aplitop.com/webaplitop/subidas/ayuda/mdt/referencia/curvado/default.htm?url=algoritmodedouglaspeucker.htm>