

130 - 9



**TECNOLÓGICO  
DE MONTERREY®**



**TECNOLÓGICO  
DE MONTERREY**

**Biblioteca**  
Campus Ciudad de México

**Campus Ciudad de México**

**Escuela de Graduados en Ingeniería y Arquitectura**

Tesis

---

**Asistente robótico para manejo de materiales aplicado a un  
instrumentista quirúrgico**

---

Para la obtención del grado de

**Maestro en Ciencias de la Ingeniería con especialización en Robótica**

Autor:

Ing. Juan Gerardo Martínez Ruiz

Asesor de Tesis:

Dr. Víctor Manuel de la Cueva Hernández

Sinodales:

Dr. Pedro Ponce Cruz

Dr. Ricardo Zavała Yoé

*Diciembre 2010*

LC

b 12924234

R857. R63

M37

. 2010

## Dedicatoria

*A mi esposa e hijo que son mi razón de lucha en esta vida.*

*A mis padres y hermano por su apoyo incondicional y sé que siempre será así.*

*A mi familia que siempre creyó en mí.*

## **Agradecimientos**

*A mis profesores: Víctor de la Cueva, Pedro Ponce, Ricardo Zavala, Patricia Rayón,*

*Alfredo Mantilla, Israel Macías, Víctor Zárate, Alfonso Parra y Marco Paz.*

*Al Tecnológico de Monterrey por brindarme los recursos*

*necesarios para el desarrollo de este proyecto.*

# Índice

<b>Capítulo 1. Introducción</b>	<b>1</b>
1.1 Resumen	1
1.2 Objetivo	4
1.3 Justificación	5
1.4 Trabajos Previos	10
1.5 Metodología	12
<b>Capítulo 2. Antecedentes</b>	<b>15</b>
2.1 Introducción a los robots industriales	15
2.2 Robótica en México y algunas aplicaciones	17
2.3 Clasificación de robots	18
2.4 Cinemática de robots	19
2.5 Robot Mitsubishi MOVEMASTER RV-M1	23
2.5.1 Motores y transmisión	24
2.5.2 Dimensiones y alcance del robot RV-M1	25
2.5.3 Referencias de movimientos del RV-M1	26
2.5.4 Modos de operación del robot RV-M1	29
2.5.5 Programación por medio de <i>teaching box</i>	30
2.5.6 Comunicación	31
2.5.7 Programación por medio de PC	35

2.6 Sistemas de visión y reconocimiento de imágenes	37
2.6.1 Introducción a LabVIEW	40
2.6.2 Módulo para desarrollo de sistemas de visión de LabVIEW	42
2.6.3 Herramientas de procesamiento de imágenes de LabVIEW	43
2.6.3.1 Imágenes en escala de grises y a color	44
2.6.3.2 Histograma	47
2.6.3.3 Igualación del histograma	49
2.6.3.4 Filtrado de imágenes	51
2.6.3.5 Binarización	52
2.6.3.6 Extracción de contornos	53
2.6.3.7 Igualación de patrones	55
2.7 Reconocimiento de voz	59
2.7.1 Generalidades	60
2.7.2 Adquisición y digitalización	66
2.7.3 Procesamiento de la señal de voz	67
2.7.4 Caracterización de la voz	72
2.7.5 Clasificación de la voz	76
2.7.6 Reconocedor de voz de Microsoft Windows	79
<b>Capítulo 3. Diseño de Instrumentista Quirúrgico</b>	<b>81</b>
3.1 Esquema general del sistema	82

3.2 Conexiones y flujo de la información	83
3.3 Soporte de la cámara	84
3.4 Selección de la cámara	86
3.5 Área de trabajo	86
3.6 Sujeción de instrumentos	88
3.7 Diseño de lógica de programación	90
3.8 Diseño de interface de usuario	93
<b>Capítulo 4. Desarrollo e Implementación de Instrumentista Quirúrgico</b>	<b>95</b>
4.1 Selección de robot y PC	95
4.2 Comunicación de robot y PC	97
4.3 Reconocimiento de voz	103
4.4 Reconocimiento de imágenes	104
4.5 Programación de robot	121
4.6 Sujeción	125
4.7 Acceso al sistema	131
4.8 Integración del sistema	131
<b>Capítulo 5. Resultados</b>	<b>137</b>
5.1 Obtención de imágenes plantilla	137
5.2 Compensación de error mecánico	138
5.3 Colocación de instrumentos en un punto fijo	140

5.4 Cambio de cámara y calibración	142
5.5 Iluminación	143
5.6 Mensajes de límites y fallo en reconocimiento	147
5.7 Reconocimiento de Instrumentos en diferentes condiciones	148
5.8 Resultados del reconocedor de voz	153
5.9 Tiempo de ciclo	154
<b>Capítulo 6. Conclusiones</b>	<b>155</b>
6.1 Trabajo futuro	156
<b>Capítulo 7. Referencias Bibliográficas</b>	<b>161</b>
Anexo A. Obtención de la representación de Denavit Hartenberg	167
Anexo B. Funciones de las teclas del teaching box	169
Anexo C. Lista de instrucciones para el RV-M1	171
Anexo D. Tabla de módulos de LabVIEW	175
Anexo E. Fórmulas para calcular el modelo HSL para imágenes a color	177
Anexo F. Ejemplos de Kernels y algunos resultados	179
Anexo G. Programa Escribir2.vi	183
Anexo H. Programas para reconocimiento de voz	185
H.1 Programa SR.vi	185
H.2 Callback_SR.vi	185
Anexo I. Programa del robot	187
Anexo J. Programa main.vi	189

J.1 Caso "Robot Stop"	189
J.2 Caso "Robot Clamps"	190
J.3 Caso "Robot Spatula"	191
J.4 Caso "Robot Senn Miller"	192

## Capítulo 1. Introducción

Esta tesis propone la aplicación e integración de varios elementos para hacer más flexible las aplicaciones de robots industriales ya que la gran mayoría de su mercado gira alrededor de la industria automotriz. Para lograrlo, se propone la integración de un robot industrial y un sistema basado en una computadora personal (PC, por sus siglas en inglés). Este sistema tiene como objetivo, adquirir voz e imágenes, para después procesar la información y mandar instrucciones a un manipulador industrial de cinco grados de libertad, el cual tomará materiales de una zona de trabajo y los colocará en un punto accesible para el usuario.

### 1.1 Resumen

En cuanto a la organización del trabajo, primero se trata de dar una introducción, después se habla acerca de la base teórica de los elementos utilizados a lo largo del proyecto, para posteriormente pasar a una etapa de diseño donde se explica la planeación. Una vez concluida la etapa de diseño, se implementará el sistema en hardware y software, para que por último se muestren los resultados y las conclusiones. En la figura 1 se muestra la organización.



Figura 1.1 Organización del trabajo

## Capítulo 1. Introducción

En el capítulo 1 de éste trabajo se trata de dar un panorama general de lo que son los elementos que se utilizaron en éste trabajo. También se mencionan algunos trabajos previos relacionados con esta tesis y se explica los objetivos, la justificación y metodologías que se seguirán.

En el capítulo 2 se tratará de explicar de manera más detallada tres puntos fundamentales para nuestra propuesta: robots industriales, reconocimiento de imágenes y reconocimiento de voz. Este capítulo empieza hablando de lo que son robots industriales y sus orígenes para poder después mencionar algunas herramientas y consideraciones que se tienen para el cálculo de su cinemática. También se mencionan algunas maneras de clasificar a los robots y sus uniones. Por último en esta sección se da una explicación más a detalle de las características generales del manipulador con el que se trabajó (Mitsubishi RV-M1). En esta parte se explica las capacidades del robot, así como las dimensiones, modos de operación y programación del mismo. También se da una explicación de los tipos de comunicación que utiliza el robot, y se profundiza un poco en la comunicación serial, ya que es la que se utiliza en este proyecto.

La siguiente parte de este capítulo habla del procesamiento de imágenes. Aquí se da una breve introducción y después se describen algunas características fundamentales para el diseño de cualquier sistema de visión. Después se introduce el software LabVIEW y las herramientas con las que se cuenta para el procesamiento de imágenes. También se describen algunos métodos del procesamiento de imágenes como son el histograma, la ecualización, los filtros, la binarización y la detección de contornos. En la siguiente parte de esta sección, se mencionan algunos métodos de reconocimiento de imágenes y se habla de la metodología que utiliza el software LabVIEW en su herramienta de igualación de patrones.

La última parte del Capítulo 2 nos habla del reconocimiento de voz. Se empieza dando una breve descripción de la historia; desde que se empezó a reconocer un pequeño grupo de palabras aisladas, hasta el reconocimiento automático del habla. Después se explica la manera en que se produce la voz y algunas características de las ondas sonoras. Se hace

énfasis entre la diferencia de sonidos sordos y sonoros, ya que es un parámetro muy importante en el reconocimiento del habla. Posteriormente se da una breve descripción de lo que es la adquisición y digitalización de voz para después pasar a la etapa de almacenamiento donde se utilizan diferentes tipos de formatos, ya sean con o sin compresión. Una vez que se han explicado los elementos generales y se tenga almacenada la señal en un archivo digital, lo que sigue es el procesamiento. Para el procesamiento se consideran los siguientes pasos: filtrado, segmentación, ventaneo, caracterización y clasificación. En el filtrado se hace énfasis en los filtros FIR, después se explica lo que es la segmentación y la aplicación de las ventanas a estos segmentos para después caracterizarlos mediante métodos basados en tiempo o basados en frecuencia. Por último se explica algunos tipos de caracterización, pero se hace énfasis en el método de Modelos Ocultos de Markov (HMM por sus siglas en inglés). La última parte de éste capítulo nos da una breve historia del desarrollo del reconocedor de habla de Windows, así como algunas características y trabajo actual.

En el capítulo 3 se habla del diseño y planeación del hardware y software del sistema. Se da una explicación general de cómo se planeó el sistema, el cual se basó en la analogía de un proceso real entre un médico cirujano y un instrumentista, y un modelo de un sistema automático. Aquí se explica el diseño del flujo de información, conexiones, área de trabajado y diseño de la lógica del software.

En el capítulo 4 se detalla la implementación del sistema como es la selección de hardware y software, la comunicación entre dispositivos y la aplicación de herramientas para el reconocimiento de voz e imágenes. En esta sección se menciona cómo fue calibrada la cámara que se utilizó así como la ubicación de los instrumentos en el espacio con respecto al robot y se describen cada una de las rutinas que se programaron en la implementación del sistema y sus diagramas de flujo. Por otro lado se explican algunas consideraciones que se tomaron para compensar errores mecánicos del brazo, y se ahonda en la programación de movimientos del robot. Posteriormente se muestra cómo se resolvió la parte de sujeción de piezas por medio de un electroimán y en base a la herramienta se marcaron límites y áreas de trabajo para evitar colisiones con objetos

## Capítulo 1. Introducción

alrededor. En la última parte de éste capítulo, se habla de como fue la integración de todas las rutinas y la programación de la interface con el usuario.

En el capítulo 5 se explican los resultados obtenidos como son las imágenes plantilla para cada instrumento a reconocer en el sistema. También se muestran resultados de ajustes que se implementaron para disminuir el error mecánico del brazo, y se midió la variabilidad de nuestra posición en la entrega de objetos al usuario. En esta sección se explica cómo es que se resolvieron algunos errores en la colocación de instrumentos y se hace énfasis en la calibración de la cámara en caso de reemplazarla o cambiar de dispositivo. En la parte de reconocimiento de imágenes se muestra la variabilidad que existe en la iluminación bajo las mismas condiciones y se explica cómo es que se logró disminuir ésta variabilidad, al igual que se realizaron pruebas de reconocimiento con uno y varios objetos en la misma imagen tanto separados como encimados. En cuanto al reconocimiento de voz, se hicieron unas pruebas con diferentes locutores para mostrar la efectividad de la herramienta que se está utilizando. Como última parte de este capítulo, se muestran los resultados de tiempo de ciclo del robot, e imágenes de proceso completo.

### 1.2 Objetivo

El objetivo de esta investigación es desarrollar un sistema capaz de reconocer comandos de voz y mediante imágenes digitales reconocer los que se está solicitando para después mandar la información a un robot industrial y ejecutar movimientos que tomen el objeto y lo dejen en un lugar accesible para el usuario.

- Representar el proceso de un instrumentista quirúrgico que manipule instrumentos de cirugía mediante un robot industrial de cinco grados de libertad, una PC con micrófono y una webcam.

- La señal de voz será adquirida por medio de un micrófono y procesada en una PC mediante el programa de reconocimiento de voz de Windows.
- Después de identificar el comando de voz, se pasará la información de lo que se está solicitando al programa LabVIEW 8.5.1 que tomará una imagen de una mesa de trabajo mediante una cámara digital, donde se encontrarán materiales o instrumentos de diferentes formas. Esta imagen será procesada mediante librerías de reconocimiento de LabVIEW y calculará la posición y orientación de la pieza.
- Se establecerá comunicación entre la PC y un robot Mitsubishi RV-M1 el cual recibirá la información de posición y orientación del objeto. Esta información la usará el robot para tomar la pieza solicitada y dejarla al alcance del usuario.

### 1.3 Justificación

Se calcula que para el año 2050, las máquinas y robots, podrían acercarse a los humanos en la manera de razonar las cosas y de percibir el medio ambiente. En la actualidad muchos centros de desarrollo, como los de Intel, están haciendo investigaciones de interfaces hombre-robot, las cuales causarán un gran cambio en las interacciones sociales [1]. Ésta información fue alentadora y contribuyó en la decisión de desarrollar este proyecto.

El desarrollo de esta investigación e integración, demostrará y servirá como ejemplo para futuras aplicaciones con un alto grado de flexibilidad para los robots industriales. El hecho de tener un robot que pueda interpretar palabras de un ser humano y ubicar objetos en un espacio de dos dimensiones, puede servir para ejecutar diferentes tareas a las que normalmente son utilizados. En la tabla 1.1 podemos observar algunas de las posibles aplicaciones futuras.

## Capítulo 1. Introducción

Aplicación	Función
Instrumentista Médico	Facilitar las herramientas al cirujano en una operación.
Ayudante de Chef	Obedecer órdenes para preparar algún tipo de alimento.
Barman Robótico	En un centro de entretenimiento, podrá mezclar y servir bebidas al comando de un cliente.
Manejo de Rutinas en ambiente industrial.	Cambio de programa de ejecución según el producto que identifique por medio de una cámara, y mediante comandos de voz, poder parar la celda de producción y mandar al robot a una posición de servicio o posición de mantenimiento para calibración de ejes o cambio de aceite.
Ayudante para personas discapacitadas	Servir a personas discapacitadas para poder realizar tareas cotidianas como es el ingerir alimentos en una mesa y tener una vida más independiente.

*Tabla 1.1. Posibles aplicaciones futuras*

Como podemos observar la gama de aplicaciones es muy grande, y dependería del diseñador del proyecto la utilidad que le daría al sistema.

Por otro lado, desde la introducción de robots como herramientas en la industria hasta nuestros días, la industria automotriz ha sido la principal promotora y consumidora de éstos mismos, teniendo el 95% de los robots totales en el mundo [2]. Durante el transcurso del año 2009, la crisis de la industria automotriz ha afectado directa o indirectamente a todas las áreas de la economía, ya que las ventas de los “Tres Grandes” (Chrysler, General Motors y Ford) han disminuido drásticamente. En el caso de Ford, durante el primer trimestre del año 2009 tuvo pérdidas netas de \$1,400 millones de dólares [3]. General Motors tuvo pérdidas en el mismo periodo por \$6,000 millones de dólares [4] y Chrysler, por su parte, espera perder \$4.7 billones de dólares durante el 2009 [5]. Ésta caída provocó en paralelo una disminución en ventas y servicios en las empresas dedicadas a los robots industriales. La caída de las órdenes en Norteamérica fue de un 31% durante el primer cuarto del año 2009 [6]. No existiendo más mercado del cual depender, las empresas se ven obligadas a recortar presupuesto y personal. En la figura

1.2 podemos observar las estadísticas de ventas totales de vehículos en Estados Unidos desde Mayo de 1967 hasta Mayo de 2009 [7].

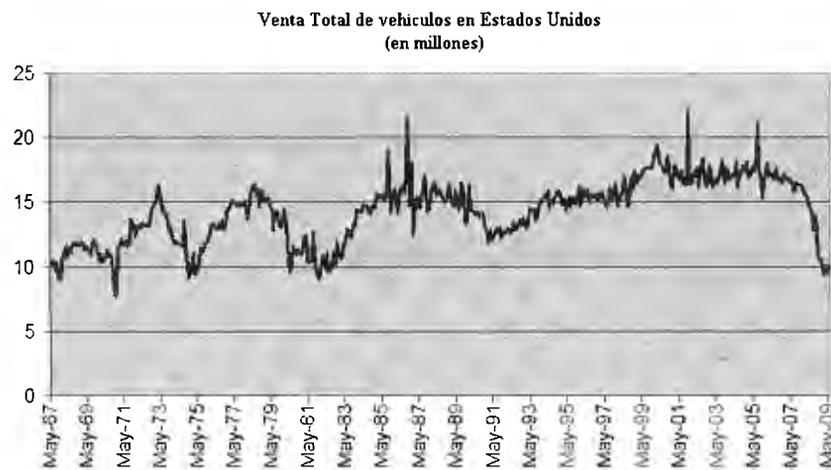


Figura 1.2. Histórico de ventas de vehículos en Estados Unidos.

Esta integración nos dará beneficios y flexibilidad para el uso de robots industriales, ya que la gran mayoría de éstos se utiliza para la industria automotriz y algunos pocos para la industria alimenticia. El aplicar reconocimiento de voz e imágenes, así como otros sistemas inteligentes, podría ampliar las aplicaciones de éstos brazos industriales para abrir el mercado y utilizarlos en actividades cotidianas.

Durante los últimos años los robots han tenido un papel muy importante en el área de la medicina debido a sus múltiples funciones y aplicaciones. Hoy en día se tienen robots aplicados en muchas áreas de la medicina.

En el área de la cirugía, un robot es crítico cuando se necesita precisión y delicadeza extrema. Los robots pueden hacer operaciones complejas con solo pequeñas incisiones, siendo estos procedimientos menos invasivos. El paciente obtiene muchos beneficios con estas prácticas como es el disminuir el tiempo de recuperación, minimizar las

## Capítulo 1. Introducción

posibilidades de contraer una infección y reducir los tiempos de hospitalización que conlleva una disminución de gastos [8].

Existen otras áreas de la medicina donde robots son utilizados como es la educación y la administración. Un ejemplo en el área de la educación son los robots humanoides que sirven para entrenar a estudiantes de medicina simulando alguna complicación en un parto. En cuanto al área de administración, los robots son útiles en la distribución de medicamentos y comida a los pacientes, manejar residuos clínicos, manejar objetos pesados y limpieza de material médico [9]. Algunas de las razones por las cuales emplear robots en la medicina se muestran en la tabla 1.2 [10].

<b>Razón</b>	<b>Áreas de aplicación</b>
Liberar a trabajadores de trabajos simples y muy estructurados	Robots para entrega: correo, comida y suministros
Liberar a trabajadores de ambientes hostiles	Preparación y manejo de material radioactivo
Liberar a trabajadores de posturas físicas demandantes	Cargar y manipular pacientes y equipo pesado
Proveer reportes mas precisos	Cirugía, farmacéutica y laboratorios
Liberar a trabajadores de decisiones complejas	Enfermería / entrega de medicamentos

*Tabla 1.2. Ventajas y áreas de aplicación de robots en la medicina*

En cuanto al aspecto económico, la inversión en el sector salud en Estados Unidos ha ido creciendo, ya que en el año 1984 se destinó 400 billones de dólares y se espera que en el año 2015 la inversión sea de 4 trillones de dólares [11].

Tomando en cuenta éstos gastos, la industria robótica tiene un gran mercado en éste sector. En la figura 1.3 podemos observar una comparativa de la venta de robots de servicio donde se encuentra la industria médica que ocupa el 8% de la venta total [12].

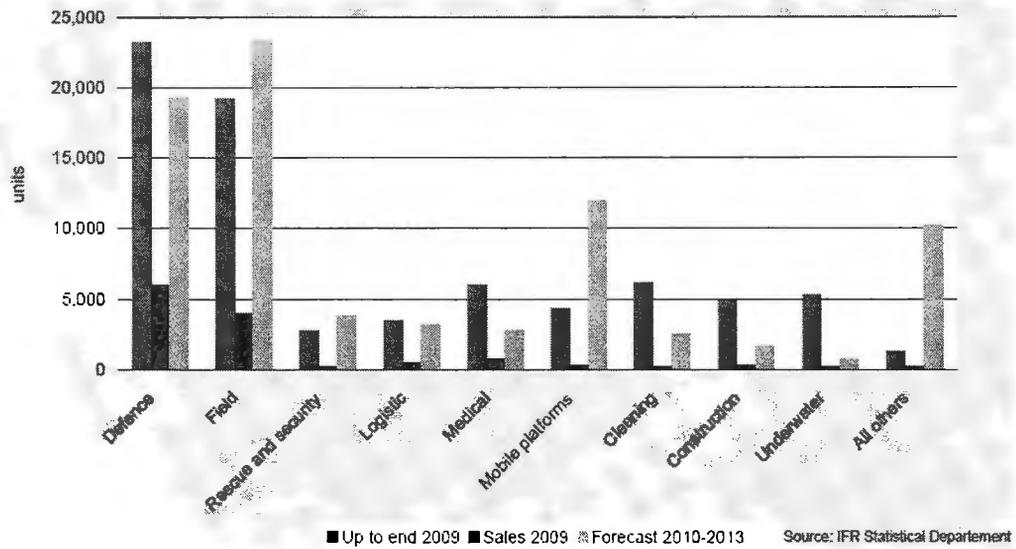


Figura 1.3. Ventas de robots de servicio y pronóstico para los próximos años

Dentro de los robots de servicio los precios son muy diferentes dependiendo de la aplicación. Entre los robots más costosos están los robots para sistemas bajo el agua que tienen un costo de entre \$300,000 dólares hasta más de \$1,000,000 de dólares mientras que los robots para la medicina van desde \$100,000 dólares hasta más de \$1,000,000 de dólares [13]. En la figura 1.4 se muestra una comparación de los costos de robots de servicio.

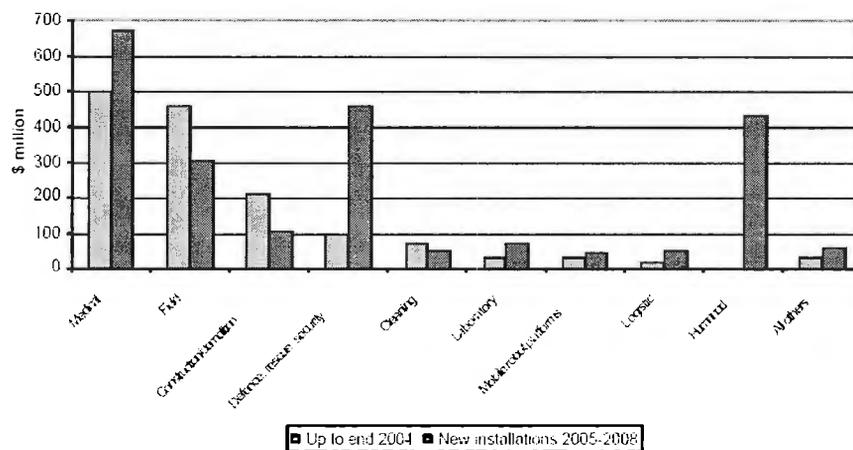


Figura 1.4. Valor de robots instalados en 2005-2008

## Capítulo 1. Introducción

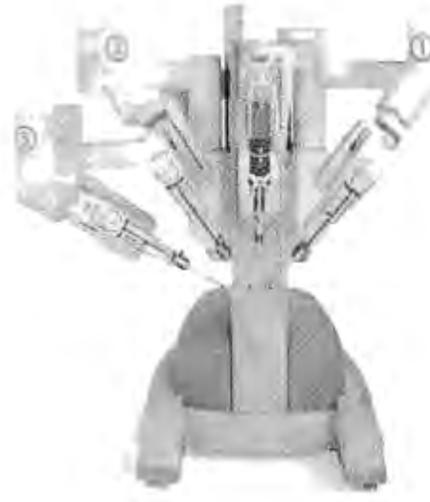
El costo de los robots para fines médicos es muy alto debido a que no existe competencia. Un ejemplo es el sistema Da Vinci desarrollado por la empresa Intuitive Surgical, el cual tiene un costo de \$1.8 millones de dólares [14]. En México solo se cuenta con un robot de éste tipo localizado en el Hospital de San José en Monterrey. Desarrollar más aplicaciones en ésta área podría ocasionar un aumento en la demanda de éste tipo de robots y paralelamente crecer el mercado de los robots industriales.

### 1.4 Trabajos Previos

En la actualidad existen investigaciones como es el proyecto del instrumentista quirúrgico de la universidad de Alcalá en España, liderado por los profesores Rafael Barea y Luciano Boquete [15]. Éste proyecto cuenta con un robot capaz de reconocer comandos vocales y escenas por visión artificial para entregar, recoger y clasificar el instrumental que suele emplearse en determinadas operaciones quirúrgicas. El sistema está formado por un brazo robot industrial, una unidad de proceso para el control del sistema; la unidad de proceso controla los movimientos del brazo robot y además tiene incorporado el hardware para la captura y digitalización de imágenes provenientes de varias cámaras, digitalización de la señal de varios micrófonos comunicados con un sistema reconocedor de comandos vocales y del locutor que los ha pronunciado, control sobre un sistema de iluminación y salida de audio generado en un sintetizador de voz [16]. Una de las ventajas de nuestro proyecto es que se utilizarán herramientas disponibles en el mercado así como software y hardware de fácil implementación.

Otro producto que existe en el mercado es el sistema Da Vinci desarrollado por la empresa Intuitive Surgical localizada en California (figura 1.5). Existen más de 1000 de estos robots en el mundo que se han efectuado cerca de 500,000 operaciones con asistencia robótica, de dichas intervenciones se han registrado cero muertes ocasionadas

por algún error del mecanismo. Estos robots ayudan a reducir el error humano ya que desde una consola, el cirujano maneja los movimientos de los brazos de Da Vinci como si fueran los propios, realizando cirugías más precisas, con poco dolor y una recuperación pronta del paciente [17]. La desventaja de éste sistema es el alto costo debido a que no existen competidores en el ramo.



*Figura 1.5. Robot Da Vinci.*

Juan Humberto Sosa Azuela, Roberto Antonio Vázquez y Ricardo Barrón Fernández del Instituto Politécnico Nacional han desarrollado una enfermera cibernética [18]. El funcionamiento del robot consiste en que el cirujano pida con su voz algún instrumento específico, el sistema basado en PC contará con un programa que reconocerá la voz y la procesará para ejecutar el comando, posteriormente la plataforma robotizada se moverá hasta que dos cámaras montadas sobre el robot localicen el instrumento, lo sujeten y lo entreguen al especialista. La ventaja en nuestro proyecto de utilizar un robot industrial, en vez de uno diseñado recientemente, es que el robot industrial cuenta con características que posiblemente sean difíciles de igualar, y que para nuestra aplicación son muy útiles. Un ejemplo es el robot IRB140 de la compañía sueca ABB. Éste robot tiene una precisión

## Capítulo 1. Introducción

de  $\pm 0.03\text{mm}$  en sus movimientos, cuenta con una capacidad de carga de 6 kg, está certificado por la IPA para generar un mínimo de partículas y se puede montar en cualquier posición como puede ser el techo, la pared o a cualquier ángulo [19].

La empresa *JBT Corporation* ha dedicado recursos en el desarrollo de robots móviles guiados automáticamente para la industria de la medicina. Un producto que maneja es el Transporte Automatizado y Sistemas integrados de logística (ATLIS por sus siglas en inglés) que son robots que ayudan al personal en diferentes tareas. Estos robots realizan labores de limpieza como cambio de ropa, manejo de desechos clínicos y llevan alimentos a los pacientes. El hospital *Forth Valley Royal* en Escocia cuenta con caminos y elevadores especiales para los robots con el fin de evitar el contacto con humanos y así disminuir los riesgos de contagio [9].

Una aplicación más de robótica en el área de la medicina es una silla de ruedas inteligente. Ésta silla recibe señales del movimiento de los ojos, utilizando redes neuro – difusas para diferenciar una de otra. El sistema también incluye control de voz para ambientes con menos de 40dB. El reconocimiento de voz se realiza con el software instalado en el sistema operativo de Microsoft Windows. Estos controles fueron diseñados usando la librería de control Inteligente de LabVIEW [20].

### **1.5 Metodología**

Este trabajo se planteó desde un panorama muy general de un caso en la vida real, hasta su implementación en una aplicación con el objetivo de abrir opciones de mercado para la industria robótica. En la figura 1.6 se muestra un diagrama de la metodología que se utilizó para la implementación del sistema.

Tiempo Invertido

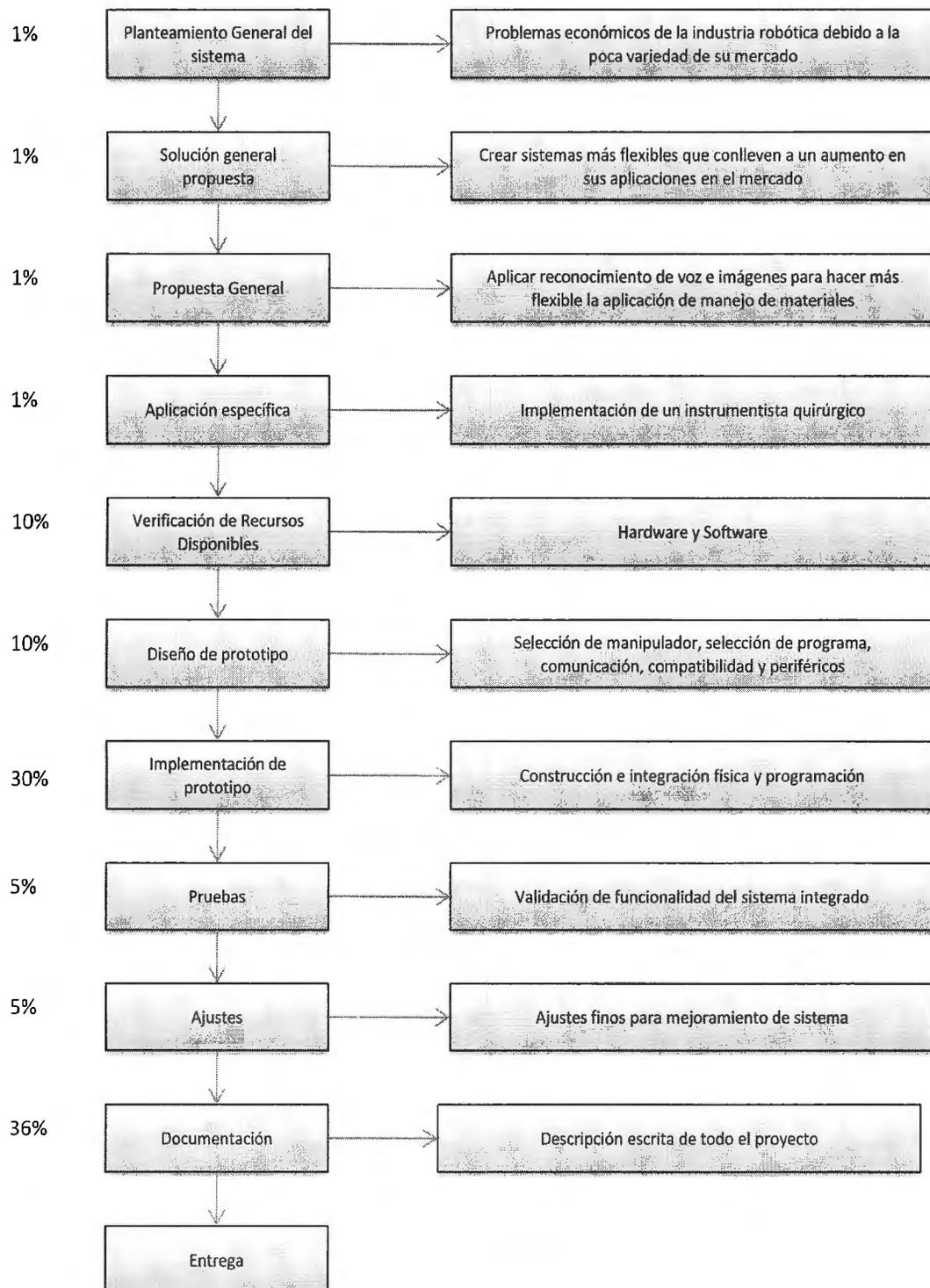


Figura 1.6. Metodología propuesta para desarrollo del proyecto

## Capítulo 1. Introducción

## Capítulo 2. Antecedentes

En este capítulo se hablará sobre algunos antecedentes referentes a los tres elementos fundamentales de este trabajo: robótica industrial, procesamiento de imágenes y procesamiento de voz. Se mencionan algunos acontecimientos históricos, así como herramientas y métodos más utilizados en las tres áreas.

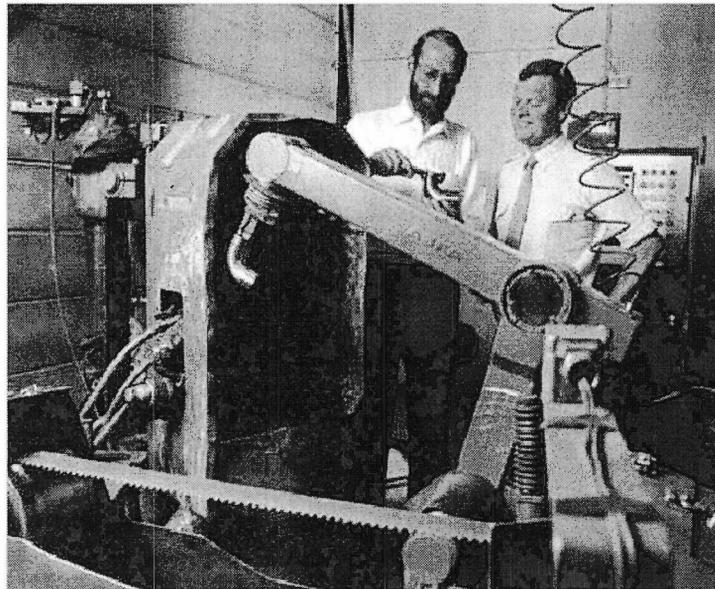
### 2.1 Introducción a los robots industriales

La palabra robot es de origen eslavo ya que en ruso *robota* significa trabajo. En checo significa trabajo forzado y en 1921 Karel Capek escribió una obra teatral que tuvo gran éxito en Broadway llamada *Rossum's Universal Robots*, donde androides trabajaban más del doble que un ser humano. El término *robotics* (robótica) se debe a Isaac Asimov, el famoso escritor de ciencia ficción [21]. Según la Organización Internacional para la Estandarización (ISO por sus siglas en inglés), la definición de robot industrial es: Manipulador multipropósito, reprogramable controlado automáticamente de tres o más grados de libertad [22].

George Devol es considerado el padre de la robótica, ya que en 1946 inventó un aparato mecánico que permitía repetir una secuencia de movimientos [21]. En ese mismo año, Eckert y Mauchly construyeron la primera computadora *Eniac*, que a su vez contribuyó para que en 1957 la corporación *Planet* produjera el primer robot comercial. En 1960 Devol vendió su patente a la corporación *Condec*, y se empezó a producir el robot *Unimate* en su subsidiaria *Unimation*. En 1962 General Motors instaló en su planta de fundición el primer *Unimation*. Ésta tecnología en un principio se basaba en PC, pero con el tiempo se fueron haciendo estudios en diferentes centros de investigación, y en 1978 *Unimation* desarrolló el PUMA (iniciales de Programmable Universal Machine for Assembling) [21].

## Capítulo 2. Antecedentes

A partir de este punto la industria robótica empezó a crecer muy rápidamente en Europa con las empresas ABB robotics y KUKA robotics. En 1973 *ABB robotics* produjo uno de los primeros robots industriales llamado IRB6. Este robot fue vendido en Suecia a la empresa Magnusson para hacer trabajos de esmerilado y pulido de tuberías [23]. Este dispositivo de 6 kg de capacidad era único, no sólo por su sistema de accionamiento, sino también por su configuración antropomorfa y su utilización de un sistema de control por microprocesador (figura 2.1). El siguiente paso de la tecnología de los robots industriales fue reemplazar los motores de corriente continua a corriente alterna. Los motores de corriente alterna entregan un par más alto, son más pequeños que los de corriente continua, no tienen escobillas, por lo que son más sencillos de mantener, y tienen una vida útil más larga, características todas que los usuarios, especialmente los fabricantes de automóviles, demandaban.



*Figura 2.1. Robot IRB6 producido por ABB Robotics en 1973 para la empresa Magnusson. El director Leif Jönsson y Lennart Benz de ASEA vigilan la instalación [23].*

## 2.2 Robótica en México y algunas aplicaciones

Los primeros robots industriales que ingresaron a México, lo hicieron a finales de los 80's, siendo la industria automotriz la principal impulsora de ésta tecnología en sus procesos de manufactura. Para los años 90's, existían entre 200 y 250 robots en producción, pero después del tratado de libre comercio (NAFTA, por sus siglas en inglés), empezó a existir más demanda de sistemas robotizados y para inicios del 2007, se estima que ya había 10,000 robots instalados alrededor del país [24].

Siendo la industria automotriz la principal impulsora de esta tecnología, las aplicaciones más avanzadas se fueron enfocando en procesos de ensamble para automóviles como lo es la soldadura de punto y la soldadura de arco; pero no sólo se dedican a soldar automóviles. Los robots industriales pintan y sueldan refrigeradores y bicicletas; ensamblan circuitos electrónicos; sujetan o separan piezas metálicas en máquinas de control numérico; empaacan, paletizan y transportan productos terminados; cortan y doblan lámina, y realizan un sinnúmero de tareas [25].

Lo que sí es claro, es que la gran mayoría de las aplicaciones están en la industria, pero recientemente se ha intentado penetrar en otras áreas como son las energías renovables, el sector alimenticio y el entretenimiento [26]. Un ejemplo es el RoboBar™, el cual es un sistema que simula ser un barman humano, mezclando y sirviendo bebidas a los clientes de un centro de entretenimiento para adultos [27]. Otro ejemplo utilizado en exposiciones, centros comerciales, parques de diversiones, entre otros, es el Robocoaster™. Esta aplicación se basa en un robot industrial con asientos, parecidos a los de un juego mecánico, sujetos a la muñeca del robot. El robot hace movimientos previamente programados, provocando a los pasajeros una sensación de vértigo y adrenalina [28]. Lamentablemente, estas dos aplicaciones no han sido instaladas en nuestro país; solo en países como Estados Unidos, Alemania y Dinamarca.

## 2.3 Clasificación de robots

Dependiendo del criterio que se utilice, puede haber diferentes tipos de clasificaciones. Por la configuración del brazo, podemos clasificarlos como cartesianos, cilíndricos, esféricos, y articulados. Por su fuente de energía los podemos clasificar en neumáticos, eléctricos e hidráulicos. También podríamos hacer alguna clasificación dependiendo de su complejidad, como por ejemplo entre robots que solo van de un punto pre programado a otro, y los robots que son más adaptables utilizando dispositivos como sensores y cámaras de video. La tabla 2.1 muestra la principal clasificación basada en la configuración del brazo junto con algunas características como son el área de trabajo, y los grados de libertad [29].

En lo que respecta a robots industriales, su gran mayoría son de tipo articulado debido a su alcance, capacidad, versatilidad y alto nivel de libertad, características que son fundamentales para aplicaciones complejas en la industria.

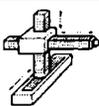
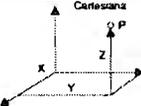
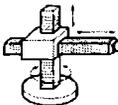
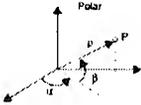
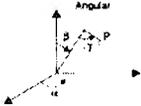
Tipo de robot	Imagen de robot	Zona de trabajo	Grados de libertad
Cartesiano			3 Grados de libertad (GDL) lineales
Cilíndrico			2 GDL lineales y 1 GDL rotatorio.
Polar o esférico			1 GDL lineal y 2 GDL rotatorios
Angular o Articulados			3 o más GDL rotatorios

Tabla 2.1. Clasificación de Robots según la configuración del brazo

Existen más clasificaciones dependiendo de su configuración como son los robots SCARA o los robots paralelos. Los robots tipo SCARA tienen movimiento en dos dimensiones y son útiles para trabajos de toma y puesta e inserciones mientras que los robots paralelos son básicamente una plataforma móvil unida a una base fija mediante varios brazos.

## 2.4 Cinemática de robots

La cinemática es la rama de la mecánica clásica que estudia el movimiento de los cuerpos sin importar la causa, limitándose al estudio de la trayectoria en función del tiempo [30].

Debido a la complejidad de los movimientos en un robot y con respecto a las ramas de la mecánica se tienen dos vertientes las cuales son la cinemática y la dinámica. Dentro de la cinemática se han creado métodos y convenciones para el cálculo de movimientos donde podemos encontrar la cinemática directa y la cinemática inversa.

La cinemática directa de robots nos da la posición y orientación del *gripper* (herramienta que utiliza el robot en la parte final de los eslabones), dadas las variables de cada unión del robot. Por otro lado, la cinemática inversa, nos proporciona la información de cada uno de los eslabones del manipulador dada la posición y orientación final del *gripper*. Los eslabones son elementos rígidos de una cadena los cuales se unen por medio de articulaciones que pueden ser rotacionales o lineales [29]. En la tabla 2.2, podemos observar diferentes tipos de uniones y sus grados de libertad.

Un grado de libertad es un parámetro independiente el cual establece la posición del elemento terminal [29]. Un grado de libertad puede ser un giro o un movimiento lineal.

Capítulo 2. Antecedentes

Tipo	Grados de Libertad (GDL)	Figura
Prismática	1 GDL	
Rotacional	1 GDL	
Cilíndrica	2 GDL	
Planar	2 GDL	
Esférica	3 GDL	

Tabla 2.2. Tipo de uniones y características

Existen varias convenciones para analizar la cinemática directa (CD), pero una de las más usadas es la de Denavit Hartenberg (DH) la cual es de propósito general para cualquier tipo de eslabón y unión [31]. La ecuación 2.1 uno muestra cómo calcular las coordenadas de la posición y orientación de la herramienta. Esta convención es muy aplicada en la robótica por ser simple y práctica donde las matrices A contienen información de las juntas y eslabones de cada uno de los elementos del manipulador, y los cálculos se basan en las matrices homogéneas de rotación y traslación de un punto en tres dimensiones [32]. El desarrollo matemático se puede consultar en el Anexo A de éste documento.

$$T_0^n = A_1 A_2 A_3 \dots A_n \tag{2.1}$$

Para referencia se tienen los parámetros DH de un robot Mitsubishi RV-M1 en la tabla 2.3, basados en la representación de la figura 2.2 [33].

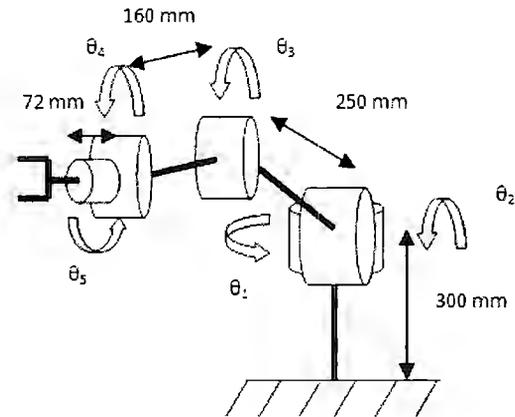


Figura 2.2. Representación de uniones y eslabones para calcular parámetros de DH

Elemento	$\alpha_i$	$a_i$ (mm)	$\theta_i$	$d_i$ (mm)
1	$90^\circ$	0	$\theta_1$	300
2	0	250	$\theta_2$	0
3	0	160	$\theta_3$	0
4	$90^\circ$	0	$\theta_4$	0
5	0	0	$\theta_5$	72 + herramienta

Tabla 2.3. Parámetros de DH

El desplazamiento total del efector debido a rotaciones y traslaciones se calcula mediante la ecuación 2.1:

$$T_5^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 \quad (2.2)$$

Donde cada matriz A equivale a:

$$\begin{aligned}
 A_1^0 &= \begin{pmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} &
 A_2^1 &= \begin{pmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} &
 A_3^2 &= \begin{pmatrix} C_3 & -S_3 & 0 & a_3 C_3 \\ S_3 & C_3 & 0 & a_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 A_4^3 &= \begin{pmatrix} C_4 & 0 & S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} &
 A_5^4 &= \begin{pmatrix} C_5 & -S_5 & 0 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned} \quad (2.3)$$

## Capítulo 2. Antecedentes

En las ecuaciones 2.3,  $C_i$  equivale a  $\cos \theta_i$  y  $S_i$  equivale a  $\sin \theta_i$ .

La cinemática inversa (CI) es primordial en el control de robots ya que la posición final del efector es generalmente conocida por las especificaciones de la tarea a realizar. Lo que necesita hacer es controlador del robot es calcular la posición de cada uno de los eslabones para llegar a la posición deseada [31]. La importancia de la CI se ilustra en la figura 2.3 mediante un diagrama de bloques.

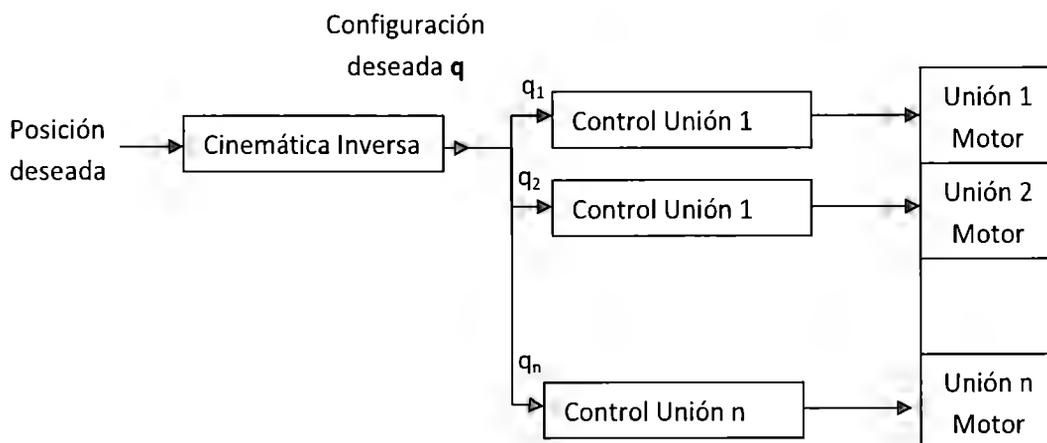


Figura 2.3 Simple control de posición de un brazo robótico

La cinemática inversa y directa se puede aplicar a cualquier tipo de robot. En este trabajo se utilizará el robot Mitsubishi MOVEMASTER RV-M1 el cual utiliza comandos de posición final de efector.

## 2.5 Robot Mitsubishi MOVEMASTER RV-M1

EL robot MOVEMASTER RV-M1 desarrollado por Mitsubishi en los años 80's, es un manipulador de cinco grados de libertad relativamente pequeño comparado con la mayoría de los robots industriales. Este robot cuenta con 3 componentes principales que son: robot, controlador y *teaching box*. La figura 2.4 podemos observar estos componentes. El robot o manipulador es aquel que hace los movimientos y donde se encuentran las partes mecánicas (motores, bandas, interruptores y lazos). El controlador es aquel que hace la parte electrónica y control del sistema. Aquí tenemos las entradas y salidas de señales exteriores, ya sea de seguridad o de algún componente externo como puede ser una computadora personal. Por último tenemos el *teaching box* que es la interface que permite al usuario programar o mover el robot según una secuencia de comandos.

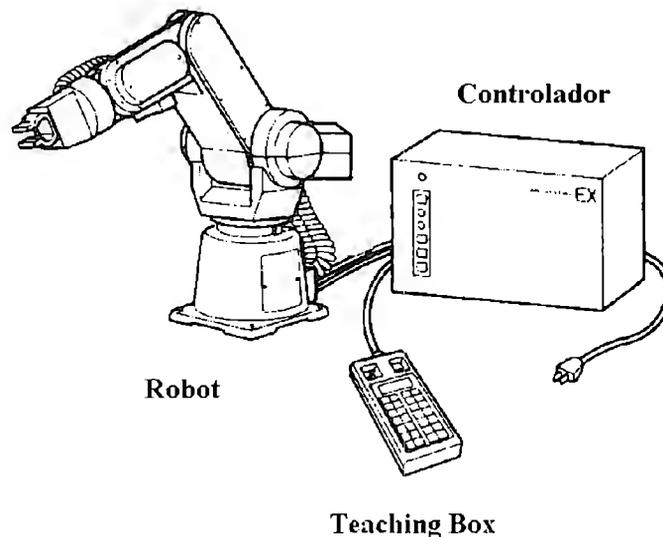


Figura 2.4 Componentes principales de sistema RV-M1

### 2.5.1 Motores y transmisión

El tipo de motores que utiliza el manipulador son de corriente directa tipo servomotor, con una capacidad de 30W para los ejes 1,2 y 3 y de 11W para el eje 4 y 5. El manipulador cuenta con un sistema de frenos y de interruptores de límite los cuales sirven para restringir el movimiento. Cuenta con frenos en los ejes 2 y 3 para poder detener el movimiento a la hora de cortar la alimentación a los motores ya que son los ejes más grandes. En cuanto a los interruptores, cuenta con uno en cada eje para asegurar que los movimientos no excedan un rango de trabajo. En la figura 2.5 se muestra la localización de los elementos antes mencionados, así como su mecanismo para transmisión de torque a los eslabones. La carga máxima del robot es de 1.2 kg, con una velocidad máxima de 1000 mm/seg en la herramienta con una repetibilidad de 0.3 mm. El peso del manipulador es de 19 kg [34].

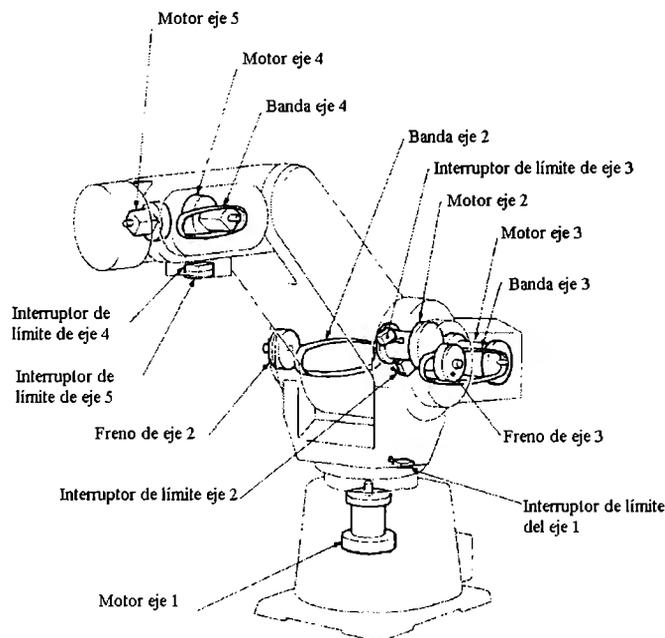


Figura 2.5. Localización de componentes internos

### 2.5.2 Dimensiones y alcance del robot RV-M1

Como ya se mencionó, este robot es pequeño en comparación de los robots industriales. Básicamente está diseñado para realizar trabajos que requieran una capacidad de carga de 1.2kg y un alcance máximo de 470mm de radio con respecto a su base. En la figura 2.6, podemos observar las dimensiones físicas del brazo, y en la figura 2.7 los alcances y área de trabajo del robot [34].

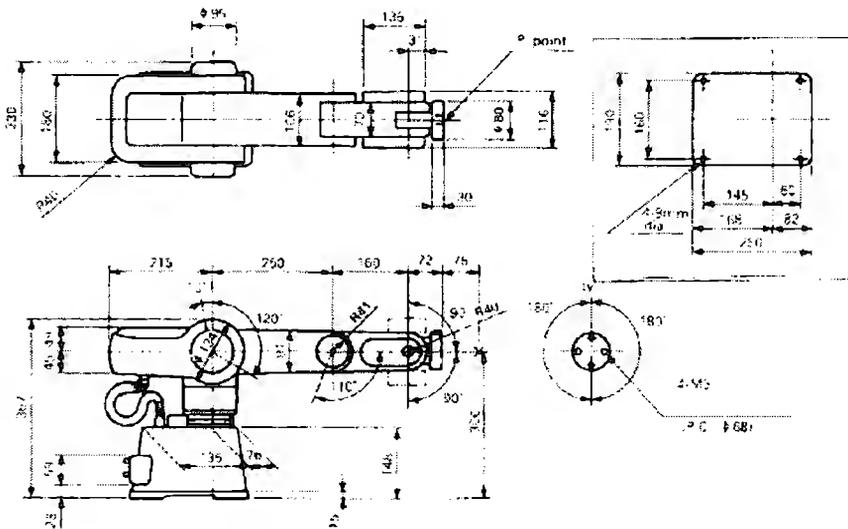


Figura 2.6 Dimensiones físicas del brazo en mm

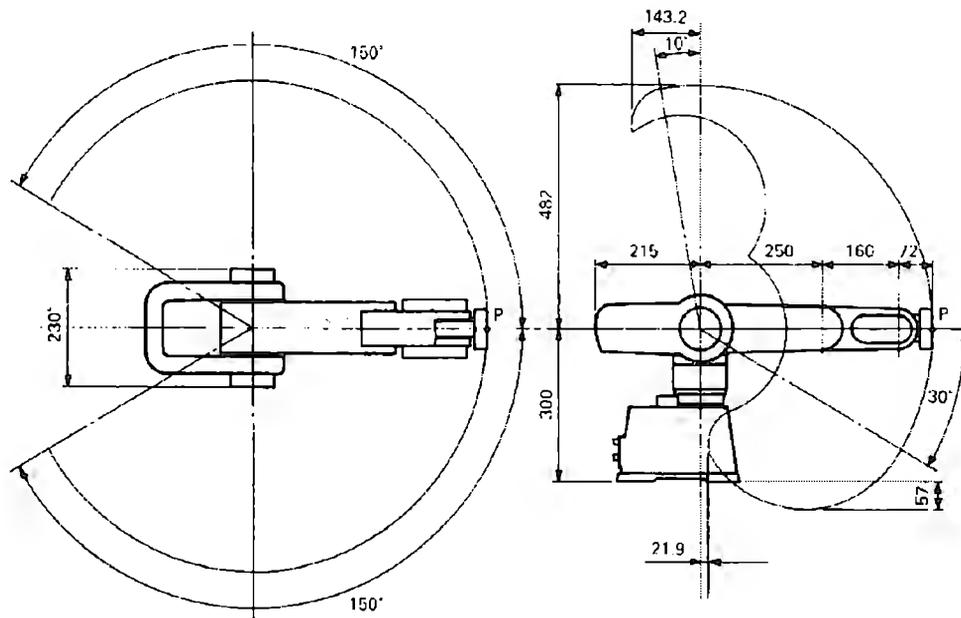


Figura 2.7. Área de trabajo del robot RV-M1

### 2.5.3 Referencias de movimientos del RV-M1

En la mayoría de los robots industriales existen principalmente cuatro tipos de referencias utilizadas para los movimientos del robot: ejes, coordenadas base, coordenadas definidas y coordenadas de la herramienta, y el RV-M1 no es la excepción. En el movimiento referenciado a los ejes, se selecciona el lazo que se moverá y lo único que tiene movimiento es el motor que se seleccionó mientras los demás se mantienen estáticos. Por convención, y para poder identificar más fácilmente los movimientos, se establece una analogía para cada movimiento con el cuerpo humano, la cual se muestra en la figura 2.8.

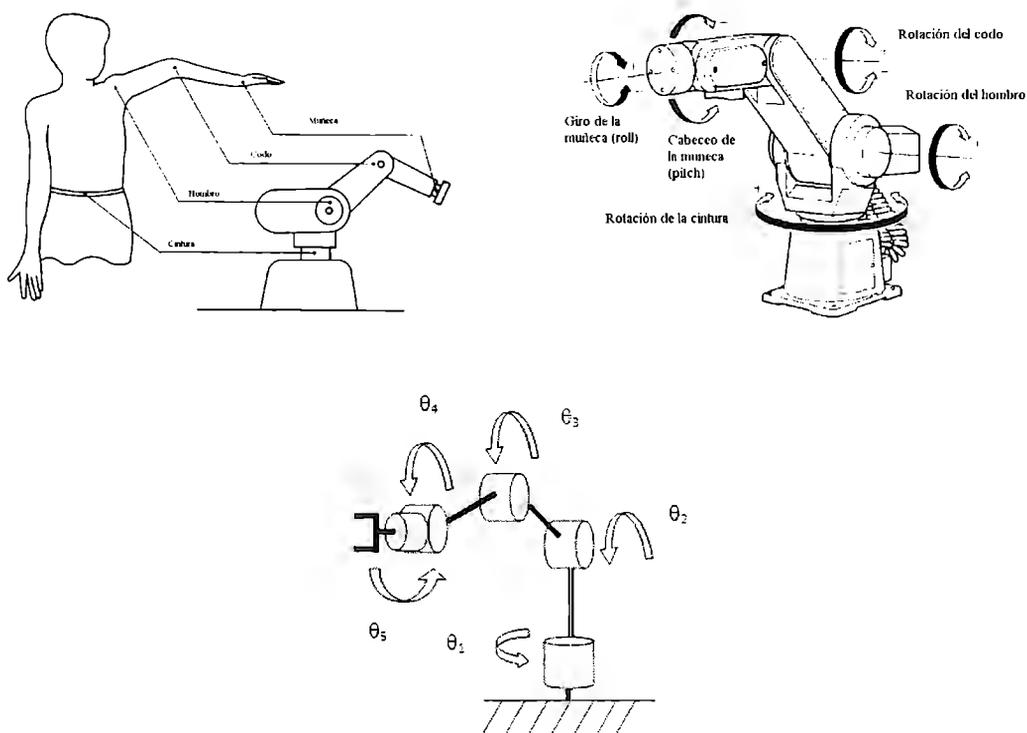


Figura 2.8 Analogía de movimientos por ejes con el cuerpo humano y su diagrama de uniones y eslabones

La segunda referencia para movimientos son las coordenadas base. Existen diferentes localizaciones de coordenadas base dependiendo del modelo, marca y tipo de robot. La mayoría de éstas localizaciones se encuentran en la base en la que está montado el robot. Para el Mitsubishi, el punto de origen de las coordenadas base se encuentra en la parte inferior de la base, justo en medio del robot, y alineado el eje z, al eje 2 (hombro).

Nuestro tercer tipo de referencia es la herramienta o efector final. Esta referencia es muy importante cuando se quiere utilizar nuestro herramienta o efector, ya que es mucho más fácil poder seguir una camino ya estando en el punto de centro de herramienta (TCP por sus siglas en inglés) en la posición inicial. Un ejemplo claro es la soldadura de arco en la cual el robot coloca la punta de la pistola en el inicio del cordón de soldadura y de ahí va siguiendo una trayectoria hasta que el cordón une las partes deseadas. Este movimiento también es muy útil cuando se necesita sacar el robot de alguna posición muy estrecha y se utilizan movimientos relativos a la posición actual de la herramienta.

## Capítulo 2. Antecedentes

Por último tenemos como referencia las coordenadas que el usuario determina. Estas normalmente se utilizan cuando en el área de trabajo se encuentra algún objeto (normalmente simétrico), el cual ayuda a facilitar los movimientos del robot y el proceso en general. Un buen ejemplo es una mesa, en la cual podríamos establecer nuestro origen de coordenadas en alguna esquina. En la tabla 2.4 se muestran ilustraciones donde se visualiza la localización de los ejes coordenados de referencia, así como las convenciones para el robot RV-M1 (en las definidas por el usuario pueden cambiar) [34].

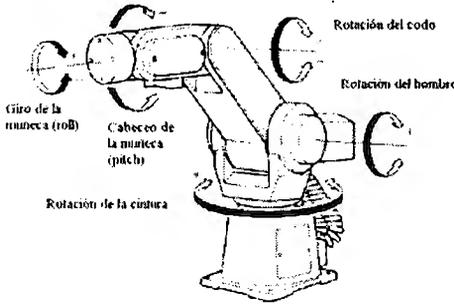
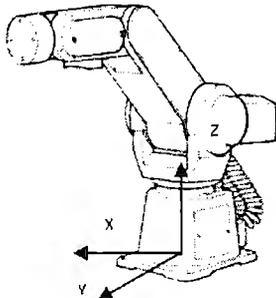
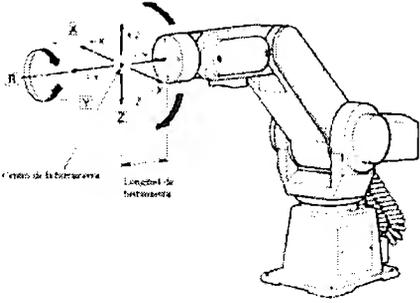
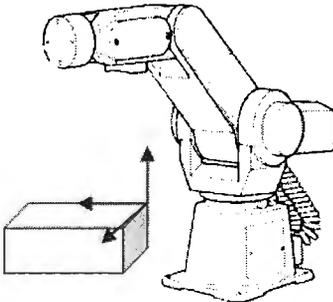
	
Tomando los ejes como referencia	Tomando las coordenadas base como referencia
	
Tomando la herramienta como referencia	Tomando otro punto como referencia

Tabla 2.4 Diferentes tipos de referencias para los movimientos del robot RV-M1

### 2.5.4 Modos de operación de robot RV-M1

El robot Mitsubishi tiene dos modos de operación: mediante una PC, y mediante la unidad de control. Para poder elegir el modo de mando, se selecciona la opción en un interruptor que se encuentra en el costado del controlador. El nombre del interruptor es ST1 y su localización se muestra en la figura 2.9. Cuando este interruptor se encuentra hacia arriba, la operación gira alrededor de la unidad de control. El robot trabaja basado en la información que se encuentra en la memoria EPROM o RAM del controlador, y utiliza los botones de la parte frontal para ejecutar los programas o detenerlos. En este modo de operación cualquier comando enviado por la PC (en caso de existir), no será tomado en cuenta [34].

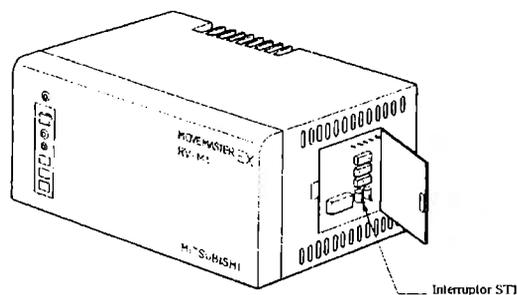


Figura 2.9 Localización del interruptor para modo de operación

El segundo modo de operación es por medio de PC. Para activar este modo, se debe colocar el interruptor ST1 en la posición baja. Cuando se tiene esta selección, la operación del robot gira en torno a una PC. Para que los comandos que se envían tengan efecto, se debe colocar el interruptor del *teaching box* en la posición de apagado. Desde la PC podemos enviar comandos directos, escribir programas completos y correr alguna rutina. Más adelante se dará una explicación de los protocolos de comunicación que utiliza el robot y de la sintaxis que se debe de utilizar para mandar comandos desde una PC [34].

### 2.5.5 Programación por medio de *teaching box*

Programar mediante el *teaching box* es más sencillo y se puede utilizar para tareas que no requieran tanta complejidad. El *teaching box* es el control de mando del usuario, este consta de varios botones e interruptores que nos ayudan a la programación del robot. Con este mando, podemos seleccionar qué tipo de referencia queremos tener y en qué dirección se moverá el robot. También podemos almacenar posiciones, borrar posiciones, escribir un programa, ejecutar un programa, mandar a el origen del robot, y algunas otras tareas que tienen que ver con la programación. La figura 2.10 nos muestra el aspecto físico del *teaching box*, así como sus elementos principales. En el Anexo B, se muestran más a detalle las funciones de cada una de las teclas del *teaching box*.

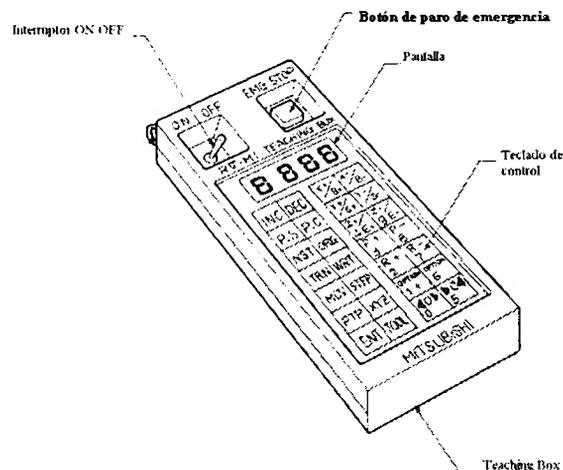


Figura 2.10 Elementos principales del *teaching box*

Un punto muy importante que se debe mencionar es la posición de origen del robot. Cada vez que se enciende el robot, se debe de llevar a su origen para poder igualar el origen mecánico del brazo con el origen del controlador debido a que se trata de un robot relativo. Esta acción es una manera de calibrar los ejes y saber cuál es el cero de los codificadores que tienen los

motores. Para el RV-M1 primero se alinean los ejes 2, 3 y 4 y después los ejes 1 y 5. Cuando el usuario manda al origen al robot, cada eje se mueve hasta tocar los interruptores de límite que se muestran en la figura 3. Para poder tener mejores resultados de repetibilidad y exactitud, se recomienda que esta posición este calibrada cuidadosamente. La posición de origen se muestra en la figura 2.11.

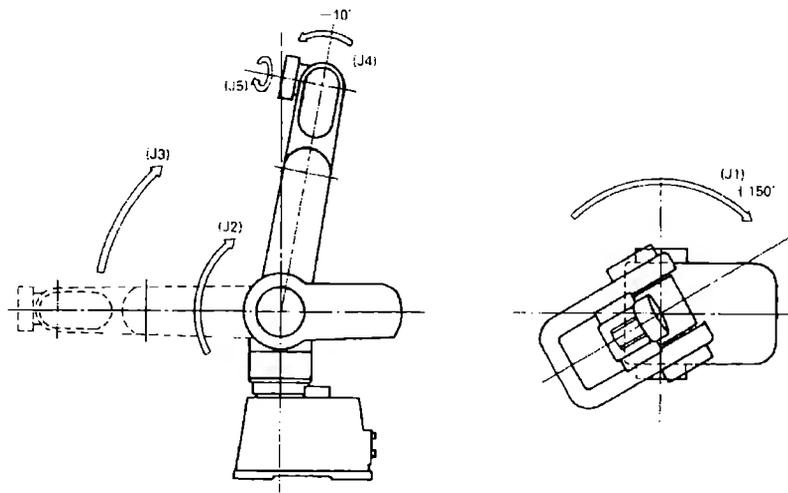


Figura 2.11 Posición de origen del RV-M1

### 2.5.6 Comunicación

Programar el robot mediante una PC siempre nos da más de flexibilidad, ya que con la misma computadora se pueden conectar varios periféricos y combinar distintos tipos de protocolos de comunicación como son el Ethernet, Profibus, serial, paralelo, etc. El RV-M1 cuenta con dos tipos de protocolos de comunicación: el serial y la interface Centronics (paralelo). El puerto paralelo, utiliza 8 bits de transmisión en una sola vía y para asegurar una buena comunicación, el largo del cable debe de ser entre 1 y 2 metros.

## Capítulo 2. Antecedentes

La comunicación serial es un protocolo estándar en casi todas las computadoras. El concepto es simple. El puerto serial manda y recibe bytes de información mediante un bit a la vez. Es más lento que la comunicación en paralelo si nuestra tasa de baudios es baja, pero la ventaja es que podemos enviar información a distancias más largas. Los parámetros que utiliza la comunicación serial son los siguientes [35]:

- Tasa de baudios: Es el número de bits de transferencia por segundo. Cuando nos referimos al ciclo de reloj, estamos hablando de tasa de baudios, por lo que si nuestro protocolo está definido con una tasa de baudios de 4800, nuestro reloj está corriendo a 4800 Hz.
- Bits de datos: Son el número de bits de transmisión. Para puerto serial los bits de transmisión más comunes son 5, 7 u 8.
- Bits de paro: Estos nos dan el final de la comunicación. Los valores típicos son 1, 1.5 y 2.
- Paridad: Es una manera simple de checar errores. Para paridad impar y par, el puerto serial fija el bit de paridad (el último bit después de los bits de datos) a un valor que asegura que la transmisión tenga un número par o impar de bits lógicos. Esto permite que el dispositivo receptor conozca el estado de un bit para así determinar si el ruido está corrompiendo los datos o si los relojes del dispositivo de transmisión y recepción se encuentran fuera de sincronización.

La configuración del robot, tiene que ser la misma que la configuración serial de la computadora. Para establecer la tasa de baudios se utiliza el SW3 y para establecer la configuración de bits de parada, paridad y tamaño de datos, se utiliza el SW2. Estos interruptores están ubicados en la parte lateral del controlador como se muestra en la figura 2.12. La tabla 2.5 muestra las opciones del SW2 y SW3 que tiene nuestro sistema para estas configuraciones. En letras negritas se indican los valores recomendados.

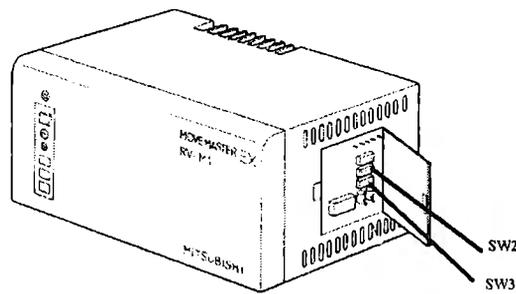


Figura 2.12 Localización de interruptores para configuración del puerto serial

Interruptor SW2					Interruptor SW3				
Factor tasa de Baudios					Tasa de Baudios				
Bit		X1	X16	X64	Bit	X1	X16	X64	
1	0	1	0	1	1	1200	75		
2	0	0	1	1	2	2400	150		
Tamaño de palabra					3	4800	300	75	
		5 bit	6 bit	7 bit	8 bit	4	9600	600	150
3	0	1	0	1	5		1200	300	
4	0	0	1	1	6		2400	600	
Paridad					7		4800	1200	
5	1 activar, 0 desactivar				8		9600	2400	
6	1 Par, 2 Impar								
Bits de paro									
		1	1.5	2					
7	0	1	0	1					
8	0	0	1	1					

Tabla 2.5 Opciones de configuración para SW2 y SW3

RS-232 es la interfaz que normalmente se utiliza para la comunicación serial. Para éste estándar existen 2 tipos de conectores: el DB9 y DB25 los cuales podemos ver en la figura 2.13. Los dos tipos de conectores cuenta con las mismas señales, pero localizadas en diferentes pines. Cada pin tiene una función la cual se explica en la tabla 2.6. Existen los pines para datos

(TX, RX y GND), y los pines para sincronización (RTS, CTS, DSR, DCD y DTR). Cuando tenemos un sistema simple, en cuanto a la comunicación, podemos utilizar la configuración más sencilla que es el bit de transmisión, el de recepción y la tierra. Cuando tenemos sistemas más complejos, los bits de sincronización nos pueden dar más seguridad en el intercambio de información. Las formas más populares de sincronización son: por software, por hardware y por XModem. A continuación se explicará en qué consiste cada una de estas sincronizaciones [36].

- Por Software: En esta sincronización, el control de la información se envía por las líneas de transmisión como si fueran datos. Aquí el emisor envía información y el receptor almacena en un buffer. Cuando el buffer se encuentra lleno o listo para recibir, éste envía dos caracteres (XOFF y XON respectivamente) para controlar el flujo de información.
- Por Hardware: Éste segundo método utiliza líneas de hardware (RTS, CTS, DSR y DTR). En el primer par de líneas, la línea RTS cambia a estado alto cuando el receptor está listo para recibir datos, este valor será leído por el transmisor en la línea CTS, indicando que está libre para enviar datos. El siguiente par de líneas es DTR y DSR se utilizan principalmente para comunicación por modem, permiten al puerto serial y modem indicarse mutuamente su estado. Como regla general, las líneas DTR/DSR se utilizan para indicar que el sistema está listo para la comunicación, mientras que las líneas RTS/CTS se utilizan para paquetes individuales de datos.
- Por XModem: Utiliza un protocolo basado en los siguientes parámetros: *start\_of\_data*, *end\_of\_trans*, *neg\_ack*, *ack*, *wait\_delay*, *start\_delay*, *max\_tries*, *packet\_size*. Estos parámetros deben de ser comunes en ambos lados de la comunicación, y el estándar XModem contiene la definición estándar de éstos. Es importante recordar que no hay que usar XON/XOFF con el protocolo XModem, ya que el número de paquete durante la transferencia por XModem se incrementará conforme se envían los caracteres XON/XOFF, lo que puede causar una falla en la comunicación.

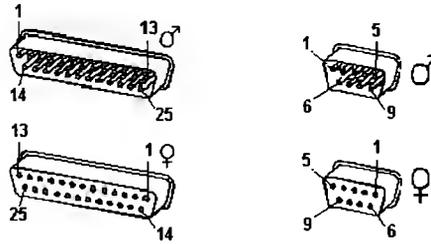


Figura 2.13 A la izquierda el DB25 y a la derecha el DB9

Utilización	B9	DB25	Abreviación	Descripción
<b>Datos</b>	2	3	RXD	Receive data
	3	2	TXD	Transmit data
<b>Sincronización</b>	4	20	DTR	Data terminal ready
	1	8	DCD	Data carrier detect
	6	6	DSR	Data set ready
	7	4	RTS	Request to send
	8	5	CTS	Clear to send
<b>Otros</b>	9	22	RI	Ring indicator
<b>Tierra</b>	5	7	GND	Signal ground

Tabla 2.6. Mapeo de bits en el conector DB9 y DB25

### 2.5.7 Programación por medio de PC

Cuando se trabaja en el modo PC y se tiene establecida la comunicación, lo siguiente es enviar las instrucciones al robot para ejecutar alguna tarea. El lenguaje de este robot es sencillo.

Existen tres maneras de enviar datos al robot. En la primera, se envían datos que se ejecutan directamente. Éste modo nos puede ser útil cuando queremos controlar el robot en tiempo real desde una PC. Aquí en cuanto el robot recibe la instrucción, éste la ejecuta. Tenemos otra manera, en la que la PC envía un programa completo (secuencia de instrucciones), de tal manera que éste queda almacenado en la memoria del robot. El programa no se ejecuta hasta que el usuario manda la instrucción de "RN". La tercera forma de enviar datos, es transfiriendo el programa y ejecutándolo. En la tabla 2.7, podemos ver la diferencia entre la sintaxis de las instrucciones y un ejemplo de cada una. En el Anexo C, se muestra una tabla con todas las instrucciones que utiliza el robot y una breve descripción de su funcionamiento. Como en otros

## Capítulo 2. Antecedentes

programas, se pueden tener ciclos o saltos dependiendo de la lógica que se utilice para programar, además de recibir y controlar señales externas como es la del efector, o la de un controlador lógico programable (PLC por sus siglas en inglés).

Instrucción	Descripción
MO1	Mueve la herramienta del robot a la posición 1. Éste comando se ejecuta directamente, y si es más de una instrucción, ésta se tiene que mandar una por una. El interruptor del <i>teaching box</i> tiene que encontrarse en OFF para que los comandos tengan efecto.
10NT 12SP7 14MO1 16GC 18MO2 20ED	Aquí se escribe, un programa desde la línea 10 hasta la línea 20. En la línea 10 mandamos el robot a su origen, después en la línea 12 establecemos una velocidad de 7 (la velocidad va de 0 a 9, siendo 0 la mínima velocidad y 10 la máxima velocidad). En la línea 14 indicamos que la herramienta se mueva a la posición 1. Después cerramos el <i>gripper</i> en la línea 16 y por último, movemos a la posición 2 en la línea 18 y terminamos el programa en la línea 20. Éste programa se almacena en la memoria del robot, pero no se ejecuta.
10NT 12SP7 14MO1 16GC 18MO2 20ED RN	Aquí se ejecutan exactamente lo mismo, solo que cuando se transfiere, inmediatamente se ejecuta por el comando "RN". El interruptor del <i>teaching box</i> tiene que estar en OFF

Tabla 2.7. Diferencia de sintaxis para comandos de robot

## 2.6 Sistemas de visión y reconocimiento de imágenes

El reconocimiento de imágenes es un tema que se ha ido estudiando desde hace mucho tiempo, pero ha tenido su mayor desarrollo desde finales de los años 70s hasta la fecha [37]. La razón principal es que las computadoras cada día tienen una mayor capacidad de manejo de información como lo son los archivos de imágenes. Podemos tener estos archivos en diferentes tipos de formatos, de los cuales los más populares son: .JPG, .TIFF, .BMP, .PNG, .GIF y .RAW.

El reconocimiento de imágenes tiene muchas aplicaciones como es el análisis de piezas para control de calidad, la ubicación de materiales para su manejo, la identificación del producto para selección de alguna tarea, el mejoramiento de imágenes, reconocimiento de rostros, análisis médico para detección de enfermedades, éstas entre muchas otras más.

Existen algunos parámetros clave en la adquisición y procesamiento de imágenes que no podemos dejar de lado antes de implementar alguna aplicación. A continuación se sugieren algunos de los más importantes en cualquier sistema de visión [38].

**1. Resolución:** Es la cantidad de información mínima que el sistema será capaz de reproducir. Esta resolución depende de la forma más pequeña que deseamos identificar del objeto que estamos viendo.

**2. Resolución del sensor:** Es la cantidad de renglones y columnas que nuestro sensor es capaz de utilizar para reproducir la imagen. Para poder calcular la resolución de nuestro sensor, se propone la ecuación 2.4, en la cual, para aplicarla debemos de conocer el área de vista de nuestra cámara y el tamaño del objeto más pequeño a identificar.

$$\text{Resolución de sensor} = \left( \frac{\text{Tamaño de campo de vista}}{\text{Tamaño del objeto más pequeño}} \right) \times (2) \quad (2.4)$$

## Capítulo 2. Antecedentes

En el mercado existen varios tipos de resoluciones de cámaras como es la 640 x 480 pixeles o 2048 X 2048 pixeles entre muchas más. Si la resolución que se necesita no es justamente las que están en el mercado, debemos de tomar al menos, próxima de mayor resolución. Se tiene que tener cuidado con esto, ya que entre mayor sea la resolución de nuestra cámara, mayor será su costo.

**3. Tamaño de sensor:** Este punto se refiere al tamaño físico del sensor. Éste se mide en diagonal y algunos ejemplos de medidas estándar son mostrados en la figura 2.14. Es importante mencionar que el tamaño de sensor no es precisamente el tamaño de la cámara.

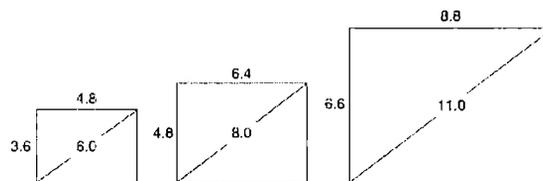


Figura 2.14 Tamaños de sensor 1/3, 1/2 y 2/3 de pulgada respectivamente.

**4. Distancia focal:** Es la distancia entre el sensor y los lentes de la cámara.

**5. Distancia al área de trabajo:** Es la distancia entre los lentes de la cámara y el área de trabajo donde se encuentran los objetos a ubicar.

**6. Distorsión:** Es una aberración causada por errores ópticos de los lentes. Ésta se da cuando la distancia focal es pequeña y la distancia al área de trabajo es grande. Los lentes de una cámara normal darán una distorsión radial, muchas veces llamada efecto de *fishbowl*. Cuando se produce este error en las orillas de de la imagen se puede ver un efecto como de “pecera”. La figura 2.15 muestra un ejemplo de este error.

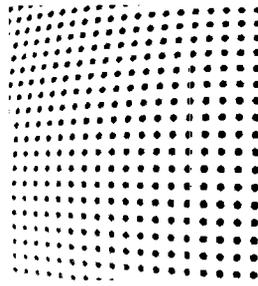


Figura 2.15 Imagen con error de distorsión tipo fishbowl

Para visualizar los puntos del 3 al 6, la figura 2.16 nos ilustra cómo es que determina cada una de estas distancias. Si se conoce el tamaño del sensor, la distancia a la zona de trabajo y el área de vista de la cámara, se puede calcular la distancia focal mediante la ecuación 2.5. Una vez calculado éste parámetro, se recomienda utilizar un cámara con el lente focal más aproximado para tener la imagen enfocada lo mejor posible. En el mercado existen estándares para la distancia focal, los cuales son: 6mm, 8mm, 12.5mm, 25mm y 50mm.

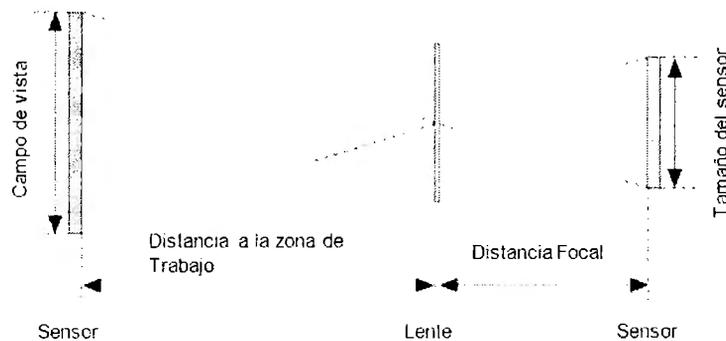


Figura 2.16 Distancias en un sistema de imágenes

$$Distancia\ focal = \frac{(Tamaño\ de\ sensor)(Distancia\ a\ la\ zona\ de\ trabajo)}{Distancia\ de\ vista\ de\ la\ cámara} \quad (2.5)$$

**7. Contraste:** Se define como relación entre la iluminación máxima y mínima de un objeto [39]. Es importante poder tener la imagen lo más contrastada posible para poder identificar los objetos con más facilidad y diferenciarlos del fondo. Esta característica es importante ya que al tener un buen contraste, se puede eliminar ruido del medio ambiente y se puede identificar cada uno de los objetos de manera más rápida.

**8. Iluminación:** En escenarios reales, es muy probable que se tengan cambios de este parámetro el cual puede producir ruido al momento de adquirir la imagen. Este parámetro puede variar dependiendo de las condiciones naturales del día como puede ser la hora o el clima. Si el sistema se encuentra propenso a exteriores, la posición del sol podrá dar sombras en diferentes posiciones dependiendo de la hora, o si el día esta soleado o nublado, se podría tener variación en las intensidades de la imagen. En la industria se ha tratado de resolver éste problema de diferentes maneras. Uno de los más comunes es utilizar filtros para la cámara de diferentes colores y luces del mismo color del filtro, esto para poder eliminar el efecto de la luz natural. Un ejemplo de este método es el utilizado en el sistema eVisionFactory<sup>TM</sup> desarrollado por la empresa canadiense Braintech<sup>TM</sup> [40]. Otro método que se utiliza es el aislamiento del sistema mediante cabinas de inspección, las cuales siempre tienen la misma intensidad de luz. El tipo de iluminación que se utilice, puede determinar el éxito o fracaso de la aplicación.

### 2.6.1 Introducción a LabVIEW

LabVIEW es un entorno de programación gráfica para desarrollar sistemas sofisticados de medida, pruebas y control introducido en 1986 por la empresa *National Instruments*. Este lenguaje utiliza íconos gráficos e intuitivos y cables que le dan un aspecto de diagrama de flujo. LabVIEW ofrece una compatibilidad con miles de dispositivos de hardware y brinda cientos de bibliotecas integradas para análisis avanzado y visualización de datos. La plataforma LabVIEW es escalable a través de múltiples objetivos y sistemas operativos.

LabVIEW tiene herramientas que nos ayudan a poder manejar con más facilidad las interfaces como son: puerto serial, puerto paralelo, TCP/IP, USB, Bluetooth, entre otras. También cuenta con herramientas para poder interactuar con diferente tipo de lenguajes y aplicaciones como es el ActiveX, Matlab, SolidWorks, adquisición y procesamiento de imágenes, etc. LabVIEW cuenta con más de 25 módulos y juegos de herramientas que abarcan una amplia gama de aplicaciones. Entre los más destacados existen los módulos de sistemas embebidos, los cuales ayudan en el desarrollo de aplicaciones con sistemas comerciales en tiempo real y sistemas basados en FPGA. También se incluyen microprocesadores y microcontroladores.

Existe otro tipo de módulos orientados al desarrollo de sistemas de control. Con estos módulos se puede diseñar, simular, visualizar e implementar sistemas reales que requieran la aplicación de éstas herramientas. LabVIEW también cuenta con módulos para procesamiento de señales e imágenes. Con estos módulos se puede adquirir señales, aplicar filtros y procesar datos de manera sencilla ya que existen métodos y algoritmos pre programados que reducen de manera considerable el tiempo de programación.

En la industria es muy común el uso de sistemas distribuidos, donde se tiene una computadora central que monitorea todos los procesos en tiempo real. LabVIEW cuenta con módulos de monitoreo y control industrial compatibles con controladores de automatización programables y controladores de lógica programables (PACs y PLCs por sus siglas en inglés). Existen otros módulos para implementación de software y para generación de reporte y almacenamiento de datos los cuales, junto con los antes mencionados, se detallan en el Anexo D [41] [42].

### 2.6.2 Módulo para desarrollo de sistemas de visión de LabVIEW

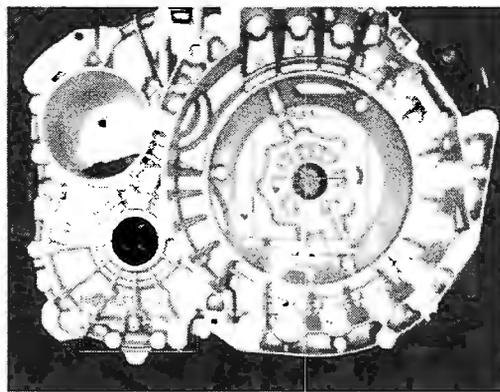
El módulo *Vision* de National Instruments es una colección de funciones de procesamiento de imágenes y visión artificial con las que se puede mejorar imágenes, verificar presencia, ubicar características, identificar objetos y medir partes. El módulo *Vision* ofrece cientos de funciones para procesamiento de imágenes incluyendo igualación de patrones e igualación geométrica, OCR, lectores de código de barras, clasificación de objetos y análisis de partículas. Esta biblioteca de funciones también incluye software controlador para miles de cámaras compatibles con estándares como GigE Vision e IEEE 1394. El software de visión de NI está disponible en dos paquetes: *NI Vision Development Module* (módulo de desarrollo de visión) y el *Vision Builder for Automated Inspection (AI)* de NI. El módulo de desarrollo de visión contiene funciones de visión para usar con NI LabVIEW, NI LabWindows/CVI, C/C++, o Visual Basic. El *Vision Builder* es un ambiente de software interactivo para configurar, evaluar y liberar aplicaciones de visión artificial sin necesidad de programación alguna [43].

Dentro de las funciones más utilizadas se encuentra la mejora de una imagen. Aquí se puede aplicar filtros para afilar bordes, remover ruido, o extraer información de frecuencia. Las herramientas de calibración son útiles para remover errores de perspectiva y no lineales causados por distorsiones del lente y colocación de la cámara. También puede utilizar las herramientas de calibración para aplicar las unidades reales a sus mediciones, de tal manera que las herramientas arrojen valores en milímetros o alguna otra unidad en vez de píxeles.

Otra aplicación muy usada es la revisión de presencia. Ésta es muy sencilla, pero a la vez útil y se aplican herramientas como comparación de color, igualación de patrones, o histogramas. Normalmente esta sólo nos arroja un resultado booleano.

Una de las herramientas más importantes en cuanto a procesamiento de imágenes es la localización de características. Ésta ayuda a encontrar la colocación exacta de los objetos, sirviendo como preámbulo para los análisis subsecuentes. La detección de bordes, la igualación

de patrones en escalas de grises, la igualación de forma, igualación geométrica y la igualación de patrones de color son herramientas que se pueden utilizar. Lo que se obtiene de estos métodos son coordenadas de posición y ángulos de orientación. En la figura 2.17, se observa un ejemplo de un sistema que localiza ciertas características en una pieza automotriz.



*Figura 2.17 Localización de características en una pieza automotriz*

La visión artificial también se utiliza para poder hacer mediciones. Típicamente, se utilizan herramientas de detección de bordes, análisis de partículas, y funciones geométricas para medir distancia, diámetro, cuenta total, ángulos y área. Estas herramientas suelen arrojar un número en vez de una localización [43].

### **2.6.3 Herramientas para el procesamiento de imágenes de LabVIEW**

En éste apartado se hablará de algunas de las herramientas con las que cuenta LabView para el procesamiento de imágenes como son la transformación entre imágenes a color e imágenes en escala de grises, histograma, ecualización, filtros, extracción de contornos y reconocimiento.

### 2.6.3.1 Imágenes en escala de grises y a color

Una imagen digital está formada por un arreglo de valores de  $(m \times n)$  como se muestra en la figura 2.18. La más sencilla de las imágenes es la binaria, la cual se forma por 0 y 1. Estas imágenes son en blanco y negro, donde el 0 es el color negro y el 1 es el color blanco. Por otro lado existen las imágenes en escala de grises donde los valores se extienden en rangos normalmente del 0 al 255, los cuales pueden ser interpretados por las PC. Este tipo de formato de imágenes es el más sencillo a considerar y uno de las más utilizados en la industria.

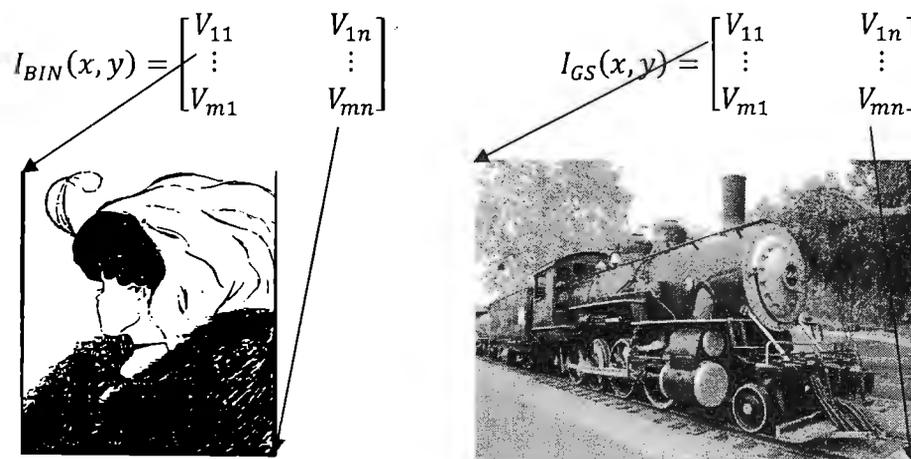


Figura 2.18 Representación de una imagen binaria (izquierda) y otra en escala de grises (derecha)

Las imágenes a color pueden ser representadas por dos modelos: RGB (Rojo, Verde y Azul Por sus siglas en inglés) y HSL (tonalidad, saturación y luminosidad por sus siglas en inglés). En su representación más simple, una imagen a color puede utilizar un byte. Cada color primario consta de 0 a 255 valores los cuales se combinan entre sí para formar otros colores. En el Anexo E, se puede ver más a detalle de cómo obtener los valores del modelo HSL. En la tabla 2.8, se ejemplifican algunas combinaciones de colores que podemos formar con el código RGB. Algunos programas utilizan un sistema conocido como RGB-100, que utiliza números en el rango 0... 100 para indicar un porcentaje de intensidad, esto para poder manejar menos información, pero el RGB-256 es el más popular.

Código Decimal (R,G,B)	Color
(255,0,0)	Rojo
(0,255,0)	Verde
(0,0,255)	Azul
(0,0,0)	Negro
(255,255,255)	Blanco
(192,192,192)	Gris Claro
(128,128,128)	Gris Oscuro
(0,255,255)	Cian Claro
(255,0,255)	Magenta Claro
(255,255,0)	Amarillo Claro

Tabla 2.8 Ejemplos de combinaciones y sus colores correspondientes.

Dentro de las capacidades del módulo de visión de LabVIEW, éste soporta imágenes de color de 32 bits para cualquiera de los modelos RGB o HSL en 4 canales de 8 bits. El primer canal que es de opacidad, tiene algunas aplicaciones, pero no en la industria, así que este canal se ignora por default. Los otros tres canales se refieren a RGB y HSL respectivamente [44].

Existen métodos para transformar imágenes de color a imágenes en escala de grises. Esto se hace para poder simplificar su manejo y reducir el tamaño de los archivos. En la industria es muy común hacerlo, ya que normalmente se trabaja con la parte de formas y ubicación, más que con colores. Existen varios algoritmos, aunque los más comunes se explican a continuación.

- El método de promedio de máximos y mínimos toma el valor máximo de los valores R, G y B al igual que el mínimo, y hace un promedio entre éstos dos valores (ecuación 2.6).

$$V_{m,n} = \frac{\max(R,G,B) + \min(R,G,B)}{2} \quad (2.6)$$

Capítulo 2. Antecedentes

- El método de promedio general, lo que hace es sumar los tres valores de R, G y B y los divide entre tres (ecuación 2.7).

$$V_{m,n} = \frac{R+G+B}{3} \quad (2.7)$$

- El método de luminosidad, asigna pesos a R, G y B basado en la percepción del ojo humano (ecuación 2.8). Es interesante saber que el ojo humano es más sensible al color verde, por lo cual a éste se le asigna un peso mayor [45].

$$V_{m,n} = 0.21R + 0.71G + 0.07B \quad (2.8)$$

En la tabla 2.9 podemos observar las diferencias entre cada uno de éstos métodos.

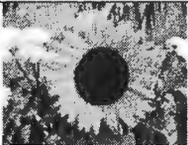
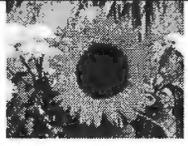
Imagen	Método
	Imagen Original
	Promedio de máximo y mínimo
	Promedio General
	Luminosidad

Tabla 2.9 Diferencia entre diferentes métodos para convertir una imagen a escala de grises

### 2.6.3.2 Histograma

Un histograma puede ser definido como una gráfica de barras de una distribución de frecuencias, en donde los anchos de cada barra son proporcionales a las clases en las que las variables han sido divididas, y las alturas de las barras son proporcionales a la frecuencia de cada clase. LabVIEW cuenta con herramientas para calcular y manejar éste tipo de análisis. El *IMAQ Histogram* y *IMAQ Histogram*. Éstas dos funciones regresan información para poder graficar, y un grupo de datos como es la media y la desviación estándar. Para poder ejemplificar esto y verlo desde un punto de vista más sencillo ver la figura 2.19. Aquí se puede observar que el histograma indica que el valor de los pixeles de la imagen es más frecuente en el rango de los 30 a 50.

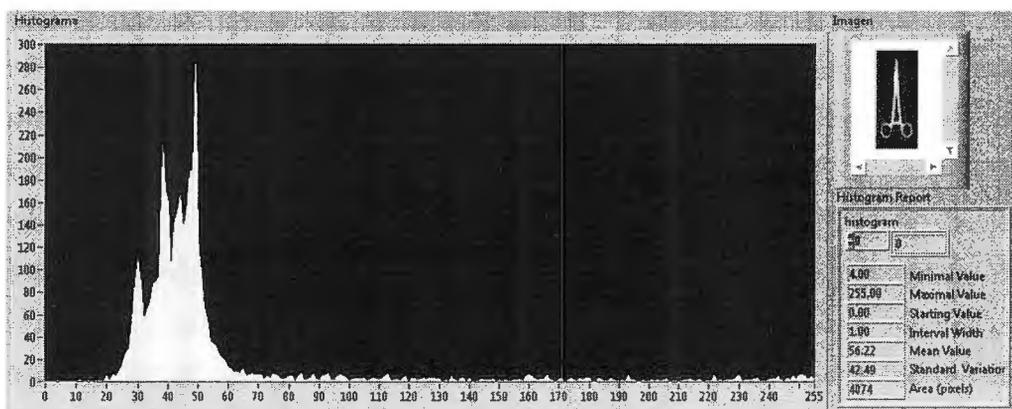


Figura 2.19 Histograma de una imagen

El histograma es de gran utilidad en el tratamiento de imágenes, ya que su manipulación sirve para variar el aspecto de las imágenes y por tanto se puede utilizar para mejorar la calidad de la imagen. Pero también sirve para la detección de objetos mediante el tratamiento de regiones. Algunas de las características de los histogramas se definen a continuación [46]:

- **Probabilidad de ocurrencia:** Es la probabilidad con la que un nivel aparece en la imagen y se define con la ecuación (2.9)

$$P(g) = \frac{N(g)}{M} \quad (2.9)$$

Donde  $M$  es el número de píxeles en la imagen,  $N(g)$  es el número de píxeles en el nivel de intensidad  $g$  y  $0 \leq N(g) \leq M$ .

- **Media:** Es el valor medio de los niveles de gris e informa sobre el brillo general de la imagen. Se define por la ecuación (2.10).

$$\bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_i \sum_j \frac{I(i,j)}{M} \quad (2.10)$$

Donde  $L$  es el número total de niveles de gris. Para una imagen con valores de gris entre 0 y 255;  $L$  sería 256. Una imagen tendrá una media alta y viceversa.

- **Varianza:** Esta mide la dispersión de los alrededores de la media, y está definida por la ecuación (2.11). Una varianza alta corresponde a una imagen con contraste alto y una varianza igual a cero significaría una imagen con un solo tono.

$$\sigma^2 = \sum_{g=0}^{L-1} (g - \bar{g})^2 P(g) \quad (2.11)$$

- **Energía:** Informa sobre la distribución de los niveles de gris. La energía tiene valor máximo a 1 para una imagen con un único nivel de gris y disminuye a medida que aumenta el número de niveles de gris. La ecuación (2.12) nos describe ésta característica.

$$E = \sum_{g=0}^{L-1} P(g)^2 \quad (2.12)$$

- **Entropía:** Esta también informa sobre la distribución de los niveles de grises. Cuanto mayor es el número de niveles, mayor es la entropía. Esta medida tiende a variar inversamente con la energía. La ecuación (2.13) muestra la manera de obtenerla.

$$e = \sum_{g=0}^{L-1} \log_2[P(g)] \quad (2.13)$$

### 2.6.3.3 Igualación del histograma

La igualación del histograma, es una de las técnicas más utilizadas para la mejora del contraste de la imagen original [47]. Se trata de una técnica que realiza la imagen original mediante una determinada transformación del histograma denominada igualación o ecualización. Se trata de encontrar una función  $F(g)$  que realce el contraste general expandiendo la distribución de los niveles de gris. A partir del histograma podemos definir la función de densidad de probabilidad considerando una imagen de  $N \times M$  [48] como se muestra a continuación:

- 1) Suponiendo una imagen de 256 niveles de gris en el rango de 0 a 255, se cumple

$$\sum_{g=0}^{255} N(g) = NM \quad (2.14)$$

- 2) La probabilidad por cada nivel de gris  $g$  viene dada por

$$p(g) = \frac{N(g)}{NM}; g = 0,1,2, \dots, 255 \quad (2.15)$$

Donde se deduce que

$$\sum_{g=0}^{255} p(g) = 1 \quad (2.16)$$

3) La función de densidad de probabilidad resulta ser:

$$P_x \cong \sum_{g=0}^x p(g) \tag{2.17}$$

4) Se trata de realizar una transformación entre funciones de densidad de probabilidad  $P_x(x)$  y  $P_y(y)$ , si se impone la condición de que la función de transformación ser monótona creciente, para cada valor de  $x, y$ , se cumple:

$$\int_0^{F(g)} P_y(y) dy = \int_0^g P_x(x) dx = \sum_{g=0}^x p(g) \tag{2.18}$$

La ecuación (2.15) es la expresión general que se utiliza para generar las ecualizaciones. En la figura 2.20, se muestra un ejemplo hecho con una función pre-programada de LabVIEW.

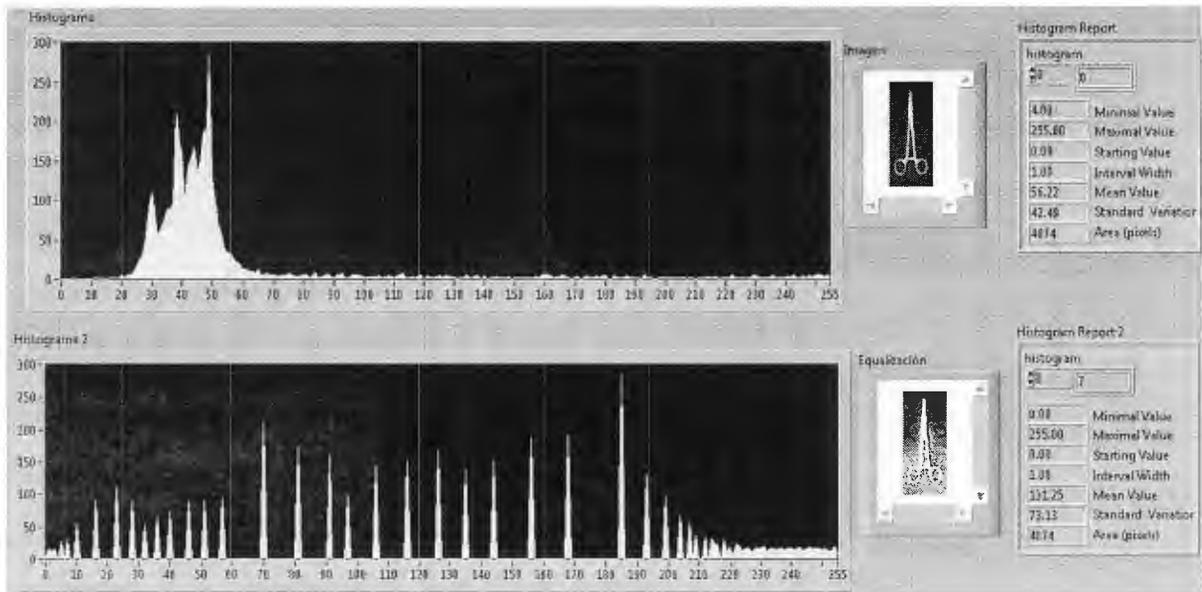


Figura 2.20 Comparación de Imagen consigo misma ecualizada

### 2.6.3.4 Filtrado de Imágenes

Los filtros de imágenes son herramientas que ayudan a atenuar o resaltar datos dependiendo de un criterio específico. Los filtros típicos en las imágenes ayudan a resaltar contraste, detectar bordes y suavizar imágenes. Esto generalmente se realiza utilizando la modificación de píxeles basado en una función de una vecindad local de píxeles. Existen 2 tipos de filtros generalmente: lineales y no lineales. Los lineales son los más utilizados y son en los que nos enfocaremos en éste trabajo.

Los filtros lineales se refieren a una combinación lineal o convolución entre los píxeles vecinos de la imagen y un *kernel* (que significa centro o núcleo). La convolución de una señal en una dimensión se realiza manteniendo una de las dos señales fijas y en la segunda señal se invierte el orden y se desplaza a lo largo del rango de la primera señal. Para cada desplazamiento se multiplican las muestras punto a punto y se suman los productos parciales. El valor de la suma es una de las muestras del vector resultado [49]. La representación matemática de la convolución para una señal continua se muestra en la ecuación (2.19).

$$h(t) = x(t) * y(t) = \int_{-\infty}^{\infty} x(\tau)y(t - \tau)d\tau \quad (2.19)$$

Donde el símbolo “\*” denota convolución. Discretizando la función, obtenemos:

$$h_i = \sum_{k=0}^{N-1} x_k y_{i-k} \quad (2.20)$$

Estas ecuaciones funcionan para señales en 1 dimensión, pero las imágenes son vectores de 2 dimensiones. Para estas señales se aplican las siguientes ecuaciones basadas en las ecuaciones (2.19) y (2.20).

$$h(i, j) = \sum_{m=0}^{M_1-1} \sum_{n=0}^{N_1-1} x(m, n)y(i - m, j - n) \quad (2.21)$$

Esta última ecuación es muy útil para el filtrado de imágenes, ya que la imagen es X y Y es una pequeña matriz llamada Kernel. El tamaño y los valores de éste Kernel nos dan la característica del filtro a utilizar. El Kernel se aplica como se muestra en la imagen 2.21.

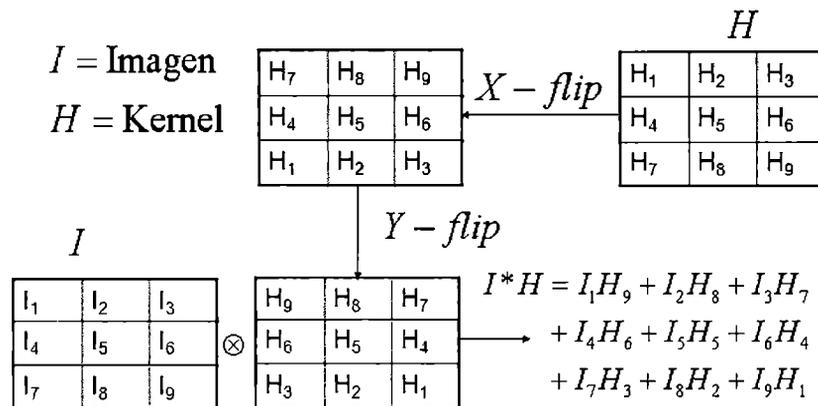


Figura 2.21 Convolución de una imagen con un kernel

LabVIEW cuenta con una herramienta para aplicar la convolución a una imagen llamada *IMAQ Convolute*, y también cuenta con diferentes tipos de kernels sugeridos. En el Anexo F, se pueden observar algunos ejemplos de kernels pre-programados por LabVIEW y sus resultados [50].

### 2.6.3.5 Binarización

La binarización, es uno de los métodos más sencillos, pero a su vez más útiles dentro del procesamiento de imágenes. En este método, lo que se hace es seleccionar un valor de intensidad de píxeles y si el valor es menor o mayor al umbral, se asigna un 1 o un 0. La figura 2.22 muestra un ejemplo de este método.

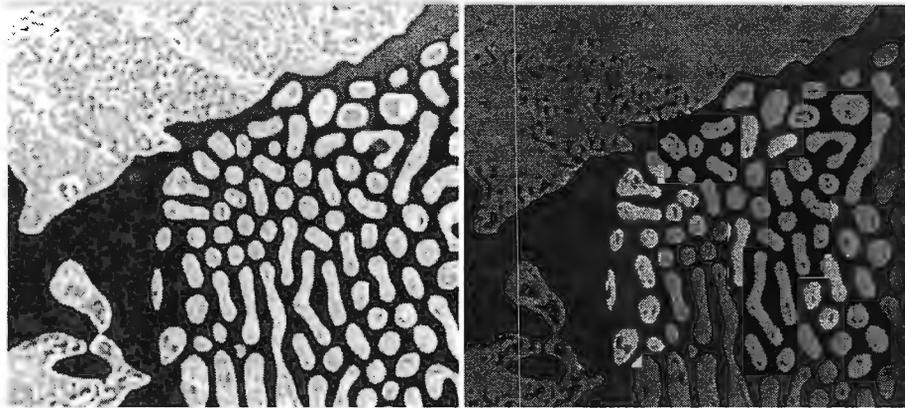


Figura 2.22 Proceso de binarización

### 2.6.3.6 Extracción de contornos

La extracción de contornos es utilizada por muchos métodos de visión. LabVIEW no es la excepción, ya que en algunas de sus herramientas los utiliza como es en la igualación de patrones. Existen varios métodos para encontrar bordes como son los operadores de primera derivada: Sobel, Prewitt y Roberts. Estos se basan en el cálculo del gradiente de una imagen descrito por las ecuaciones 2.22, 2.23 y 2.24 [46].

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.22)$$

$$G[f(x, y)] = G_x^2 + G_y^2 \quad (2.23)$$

$$G[f(x, y)] \approx \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$$

$$\alpha(x, y) = \tan^{-1} \left( \frac{\partial x}{\partial y} \right) \quad (2.24)$$

## Capítulo 2. Antecedentes

Donde  $G$  es un vector bidimensional y perpendicular al borde. Estos valores de  $G$  pueden implementarse por convolución de la imagen con Kernels de  $3 \times 3$ . En la tabla 2.10 se muestra un ejemplo de detección de bordes por método de Sobel con diferentes valores de umbralización.

Imagen Original	Umbral	Imagen Procesada
	20	
	70	

Tabla 2.10 Ejemplo de detección de contornos por Sobel

Dentro de los métodos más utilizados está el detector de contornos de Canny. Éste se basa en la teoría de los operadores de primera derivada, y se desglosa en 3 módulos [46]:

1. Obtención del gradiente (magnitud y ángulo en cada pixel).
2. Adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un pixel de ancho, se conoce como "supresión no máxima".
3. Histéresis de umbral al resultado de la supresión no máxima.

En la figura 2.23, se muestran ejemplos del detector de contorno de Canny.

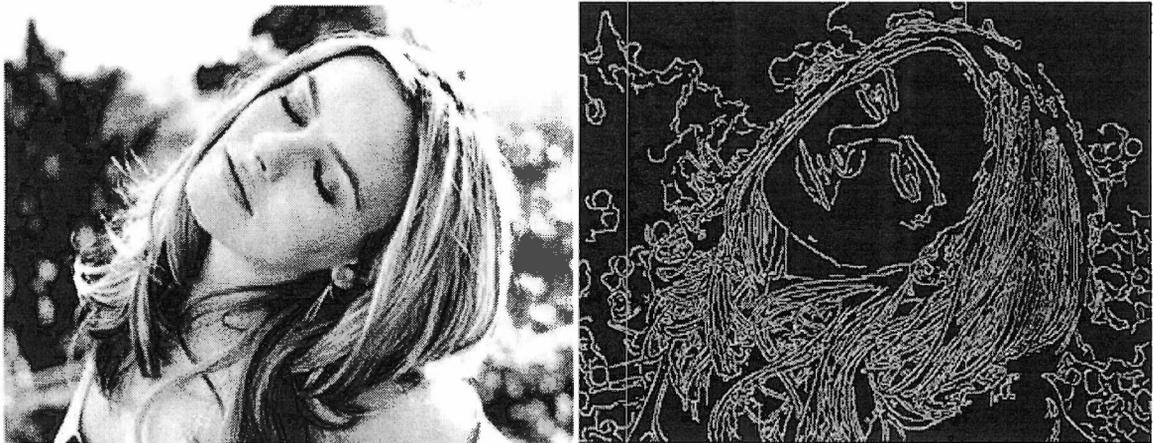


Figura 2.23 Ejemplo de detector de contorno de Canny

### 2.6.3.7 Igualación de patrones

La igualación de patrones es un método para identificar objetos dentro de una imagen basados en una plantilla más pequeña. Existen métodos tradicionales como es la correlación cruzada. Aquí se compara la plantilla con toda la imagen, y cada intensidad de pixel es multiplicada individualmente para al final obtener una suma, la cual es el valor de la correlación. La plantilla se desfasa un pixel y hace el mismo procedimiento hasta que cubre toda la imagen. Matemáticamente se puede definir la correlación cruzada en la ecuación 2.25, y si se normaliza nos queda la ecuación 2.26 [44]. La figura 2.24 nos muestra gráficamente como es que se realiza ésta operación.

$$C_{i,j} = \sum_{x=1}^{L-1} \sum_{y=1}^{K-1} w(x,y)f(x+i,y+j) \quad (2.25)$$

$$C_{i,j} = \frac{\sum_{x=1}^{L-1} \sum_{y=1}^{K-1} w(x,y) f(x+i,y+j)}{\sqrt{\left(\sum_{x=1}^{L-1} \sum_{y=1}^{K-1} w(x,y)^2\right) \left(\sum_{x=1}^{L-1} \sum_{y=1}^{K-1} f(x,y)^2\right)}} \quad (2.26)$$

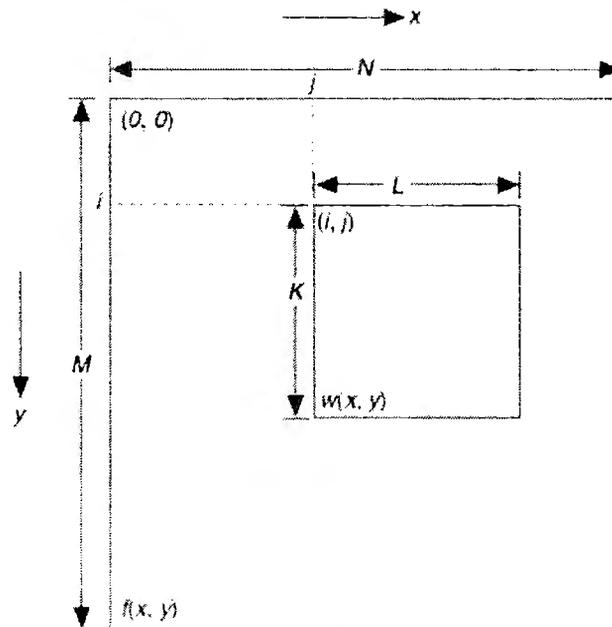


Figura 2.24. Proceso de correlación cruzada

Debido a que el método tiene varias restricciones como es el no poder manejar objetos escalados o que han sido rotados, y computacionalmente es muy demandante, se han buscado otro tipo de técnicas para poder resolver éstos problemas. La localización piramidal es una de ellas y consiste en un método similar al de correlación cruzada, pero divide en partes la imagen y la plantilla y reduce el número de píxeles de información. La rapidez puede ser mejorada hasta en un 75% [44].

National Instruments ha desarrollado algunos algoritmos para mejorar la igualación de patrones, los cuales incluyen una etapa de aprendizaje y una etapa de igualación de patrones. Dentro de la etapa de aprendizaje se pueden mencionar cuatro puntos en los cuales se hace énfasis [51]:

1. **Muestreo pseudo-aleatorio:** Se busca reducir la información que se analiza por medio de muestras aleatorias. En la figura 2.25, se puede observar un ejemplo en donde a la izquierda se tiene la plantilla que se desea buscar, y a la derecha se tiene la imagen tomada junto con un grupo de píxeles aleatorios generados por el algoritmo que utiliza LabVIEW. Los puntos rojos nos dan información acerca de la estructura general de la imagen y los puntos verdes nos dan información más precisa de algunas formas que nos ayudarán a encontrarlo con más precisión.

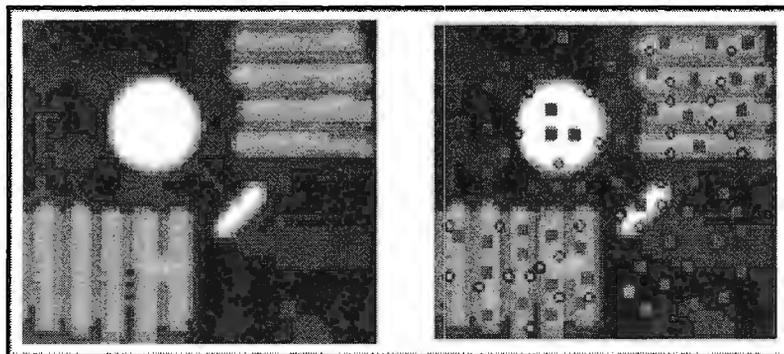


Figura 2.25 Ejemplo de muestreo pseudo-aleatorio

2. **Análisis de estabilidad:** Los píxeles arrojados aleatoriamente se analizan revisando sus píxeles vecindarios. Cada píxel se clasifica dependiendo de qué tan estables son sus alrededores (3x3, 5x5, etc.). Esta información nos ayuda para sólo hacer las comparaciones necesarias en la etapa de igualación de patrones.
3. **Identificación de Formas:** Se realiza una detección de contornos en la plantilla. Los resultados son guardados para después utilizarlos en una búsqueda más fina.
4. **Análisis rotacional invariante:** Se utiliza cuando el usuario requiere encontrar objetos rotados. Aquí se crea un registro circular de intensidades de la plantilla. Una rotación de la plantilla tiene el mismo registro, pero desfasado hacia la izquierda o hacia la derecha un cierto ángulo.

Dentro de la etapa de igualación de patrones, se utiliza la información adquirida en la etapa de aprendizaje para desechar toda la información que no tiene relevancia para el proceso. El algoritmo que se utiliza depende de las opciones que seleccione el usuario. En éste caso, LabVIEW utiliza dos opciones: *Shift – Invariant Matching* y *Rotation – Invariant Matching*.

1. ***Shift – Invariant Matching.*** El primer paso es hacer una correlación utilizando la información arrojada por los puntos del muestreo aleatorio de la plantilla. Después se utiliza la información del análisis de estabilidad para poder ver qué puntos de la imagen a buscar se pueden omitir sin descartar algún punto importante. Por ejemplo: si todos los muestreos arrojan ser estables en una matriz de 3x3, se puede omitir 2 de 3 correlaciones, asegurando que se encontrará el patrón que se busca. Esto puede reducir el tiempo en un factor de 9. El segundo paso es solo trabajar con los candidatos arrojados por el primer paso y por medio de contornos, correlacionar y hacer una búsqueda más fina. Aquí el usuario puede determinar un umbral el cual es la puntuación de ésta correlación para decidir que formas se regresan como identificadas.
2. ***Rotation – Invariant Matching:*** Se utiliza el registro circular que se obtuvo en la etapa de aprendizaje para buscar dentro de las formas encontradas por el *Shift – Invariant Matching*. El usuario puede restringir el radio de giro según su aplicación y reducir la cantidad de información que se maneja. Varios candidatos pueden ser obtenidos en el paso. El segundo paso es utilizar el método de muestreos pseudo-aleatorios para calcular la correlación con los candidatos. Al final tenemos una puntuación que el usuario utiliza para definir si es una igualación de patrón o no.

Para seleccionar las plantillas se recomienda que sean asimétricas, no muy complejas, y con la suficiente información para poder encontrarlas fácilmente en la imagen a buscar sin confundirla con algún otro objeto.

## 2.7 Reconocimiento de voz

La investigación en el campo del reconocimiento de voz se ha venido haciendo desde los años 50's. En la tabla 2.11 se mencionan algunos de los avances más notorios a través de la historia [52].

Década	Acontecimiento
50's	<ul style="list-style-type: none"> <li>• En 1952 en los laboratorios Bell, Davis, Biddulph y Balashek construyeron un sistema para el reconocimiento de un solo dígito con un solo locutor.</li> <li>• En 1956 en los laboratorios RCA, Olson y Belar trataron de reconocer 10 sílabas distintas de un solo locutor.</li> <li>• En 1959 en <i>Univeristy College of England</i>, Fry y Danes trataron de reconocer cuatro vocales y nueve consonantes. En el mismo año, se tuvo investigación en los laboratorios de Lincon MIT para reconocer 10 vocales.</li> </ul>
60's	<ul style="list-style-type: none"> <li>• En los 60's varios laboratorios japoneses se unieron a la investigación e iniciando la década, Suzuki y Nakata del laboratorio de investigación en radio, crearon un sistema de reconocimiento basado en un banco de filtros.</li> <li>• A finales de los 60's en los laboratorios de RCA se investigo sobre los problemas asociados con la no uniformidad de escalas de tiempo en el habla. AL mismo tiempo en la Unión Soviética, Vintsyuk propuso el uso de métodos de programación dinámicos para la alineación del tiempo.</li> <li>• Los últimos avances destacados a finales de ésta década fue el comienzo del programa de investigación de la universidad de Carnegie Mellon por Reddy.</li> </ul>
70's	<ul style="list-style-type: none"> <li>• Los estudios rusos de Velichko y Zagoruyko aportaron ideas de reconocimiento de patrones para el reconocimiento de voz.</li> <li>• Los japoneses Sakoe y Chiba, aportaron estudios sobre métodos de programación dinámica.</li> <li>• En Estados Unidos, Itakura aportó ideas de códigos de predicción lineal (LPC por</li> </ul>

	sus siglas en inglés).
80's	<ul style="list-style-type: none"> <li>• De ser un problema enfocado a reconocer una sola palabra, el problema se extiende a poder reconocer palabras concatenadas.</li> <li>• Cambio de tecnología basada en plantillas a métodos estadísticos, en especial los modelos ocultos de Markov (HMM por sus siglas en inglés).</li> <li>• HMM utilizado por IBM, el instituto de análisis de la defensa (IDA por sus siglas en inglés) y Dragon Systems.</li> <li>• Aplicación de redes neuronales para el reconocimiento de patrones.</li> <li>• Desarrollo de SPHINX por la universidad de Carnegie Mellon.</li> <li>• Desarrollo de BYBLOS por BBN</li> <li>• Investigación de Lincoln Labs</li> <li>• Investigación de MIT</li> <li>• Investigación de AT&amp;T Bell Labs</li> </ul>

*Tabla 2.11 Historia del desarrollo del reconocimiento de voz*

A partir de la década de los 90's Microsoft ha invertido muchos recursos para el desarrollo de esta tecnología. El CEO de Microsoft, Bill Gates, ha apoyado el desarrollo de ésta tecnología ya que el mismo ha dicho que "Un gran, gran progreso. El poder dictar un correo electrónico en vez de escribirlo" [53].

### 2.7.1 Generalidades

Los sonidos de voz en el ser humano son producidos por un conjunto de elementos como son los pulmones, las cuerdas vocales, la lengua, la posición de los labios, el tracto vocal, etc. Cada idioma tiene su fonética y características. La anatomía del aparato fonador del ser humano se puede dividir en las siguientes cavidades, las cuales se muestran los principales elementos en la figura 2.26:

- **Cavidades infraglólicas:** Está conformada por el diafragma, los pulmones y la tráquea. Tienen como misión proporcionar la corriente de aire respirada, necesaria para producir el sonido.
- **Cavidad glótica:** Está formada por la laringe donde se encuentran las cuerdas vocales, responsables de producir la vibración básica para la generación de voz.
- **Cavidades supraglólicas:** Ésta se conforma por cuatro cavidades, la faríngea, nasal, bucal y labial.

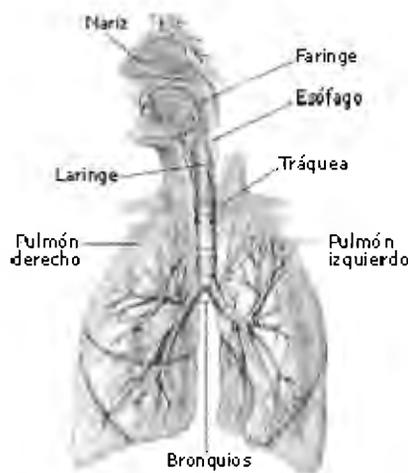


Figura 2.26 Aparato fonador humano

Para producir el habla, se requiere de una fuente de energía que produzca la presión del aire el cual es expulsado en la exhalación. Después se requiere de un órgano vibratorio el cual son las cuerdas vocales. Después las fosas nasales, la cavidad bucal y la faringe funge como una caja de resonancia para que al final el paladar, la lengua, los dientes, labios y mandíbulas actúen como un sistema de articulación.

Existen parámetros que se pueden analizar como son el tono, la intensidad y el timbre. El tono se refiere a la frecuencia con la que vibran las cuerdas vocales. La intensidad se refiere a la amplitud de la onda, lo cual nos da un parámetro muy útil que es la energía. El timbre es la

## Capítulo 2. Antecedentes

diferencia entre dos sonidos con la misma intensidad y el mismo tono, pero con diferentes componentes armónicos.

La voz es un conjunto de ondas de presión acústica producidas por el tracto vocal y cuyo arreglo está gobernado por las reglas del lenguaje. La generación de voz es un equivalente a un proceso de filtrado, en el cual el tracto vocal funciona como un filtro desde la apertura de las cuerdas vocales, hasta los labios. Cuando la señal de voz se propaga a través del tracto vocal, el espectro de esta señal queda determinado por la selectividad del tracto para dejar pasar determinadas frecuencias. Para la voz humana tenemos un rango de frecuencias entre 100Hz. y 10kHz. Existen diferencias en cuanto a las frecuencias dependiendo el género. La figura 2.27 nos muestra la distribución de las frecuencias fundamentales para hombres y mujeres. Los hombres tienen una media de 125 Hz con una desviación estándar de 20.5 Hz, mientras que las mujeres tienen una media de 250 Hz y una desviación de 41 Hz. Un dato interesante, es que éste rango es reducido por la telefonía para poder manejar menos información y hacer más rápida la transmisión y procesamiento de datos. La frecuencia significativa de la voz humana está por debajo de los 3.5kHz.

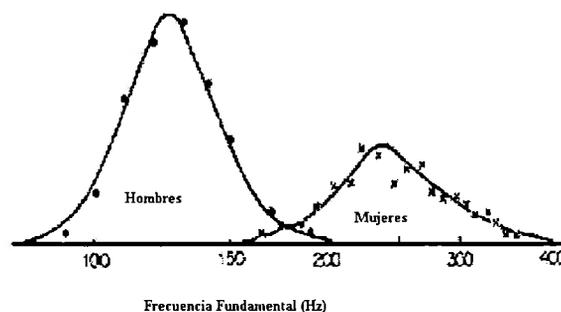


Figura 2.27 Distribución de frecuencias fundamentales de hombres y mujeres

Una de las características más importantes de la voz es el tono. Éste es el número de veces por segundo que las cuerdas vocales se unen en la fonación. El tono está relacionado con la frecuencia fundamental. Cuando el tono es grave, la frecuencia es baja y cuando el tono es agudo, la frecuencia es alta. La intensidad de voz es otra característica, la cual se refiere a la

amplitud de la señal de voz y depende de que tanto estén abiertas las cuerdas vocales y la cantidad de presión debajo de la laringe. Existe un tercer elemento importante el cual es el timbre. Éste nos permite identificar entre dos sonidos con el mismo tono y la misma intensidad mediante el contenido armónico y las características dinámicas del sonido.

Los formantes dentro de la voz son frecuencias de resonancia del tracto vocal, es decir, frecuencias donde las armónicas tienen mayores amplitudes. La cantidad de formantes y su ubicación son características distintivas de cada sonido. Las vocales tienen estructuras formantes bien definidas, las cuales se muestran en la tabla 2.12. Estos formantes son útiles para caracterizar la voz.

	<b>Primer Formante (Hz)</b>	<b>Segundo Formante (Hz)</b>	<b>Tercer Formante (Hz)</b>
<b>A</b>	861	1464	2584
<b>E</b>	431	2239	3600
<b>I</b>	258	2412	3531
<b>O</b>	474	861	3900
<b>U</b>	258	689	

*Tabla 2.12 Formantes de las vocales*

La frecuencia es un parámetro muy importante ya que los sonidos sonoros son producidos por la vibración de las cuerdas vocales y por consiguiente, las ondas sonoras cuentan con componentes frecuenciales que podríamos analizar en el dominio de la frecuencia, como por ejemplo las vocales. Esos sonidos tienen altos contenidos energéticos y pueden ser modelables matemáticamente con un tren de pulsos. Los sonidos sordos no siguen un patrón que se repita. Estos sonidos son producidos cuando el aparato fonador solo deja pasar aire de los pulmones a la cavidad nasal sin hacer vibrar las cuerdas vocales. Los sonidos sordos son ondas con poco contenido energético y pueden ser modelables matemáticamente como ruido blanco. Un ejemplo de estos sonidos es la letra "s".

## Capítulo 2. Antecedentes

La energía es un parámetro que podemos considerar, puesto que los sonidos sordos tienen poca energía y los sonidos sonoros cuentan con un alto contenido energético. Esta característica puede servirnos para clasificar la onda en el dominio del tiempo, ya que a simple vista podríamos notar cuando se presenta un sonido sordo o un sonoro [52]. Para ejemplificar y visualizar la diferencia entre un sonido sonoro y uno sordo, la figura 2.28 nos muestra la gráfica de una grabación de la palabra “abeja” con una frecuencia de muestreo de 8000 Hz. Los valores del sonido están normalizados de -1 a 1. La figura 2.29 y 2.30 nos muestran una ampliación de la letra “a” y de la letra “j” respectivamente. En estas ampliaciones podemos ver como en la letra “a”, que es un sonido sonoro, tenemos un cierto patrón que se repite, a diferencia de la letra “j” que no sigue ningún patrón.

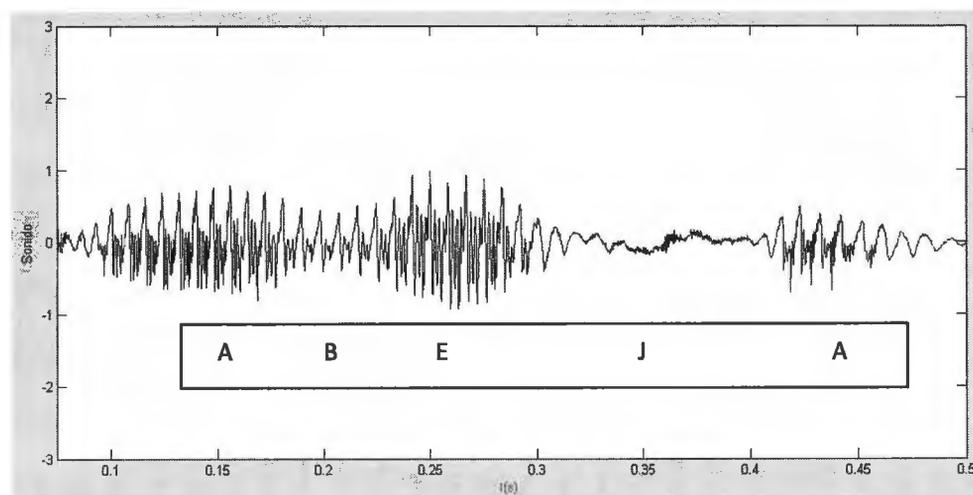


Figura 2.28 Gráfica de la palabra "abeja"

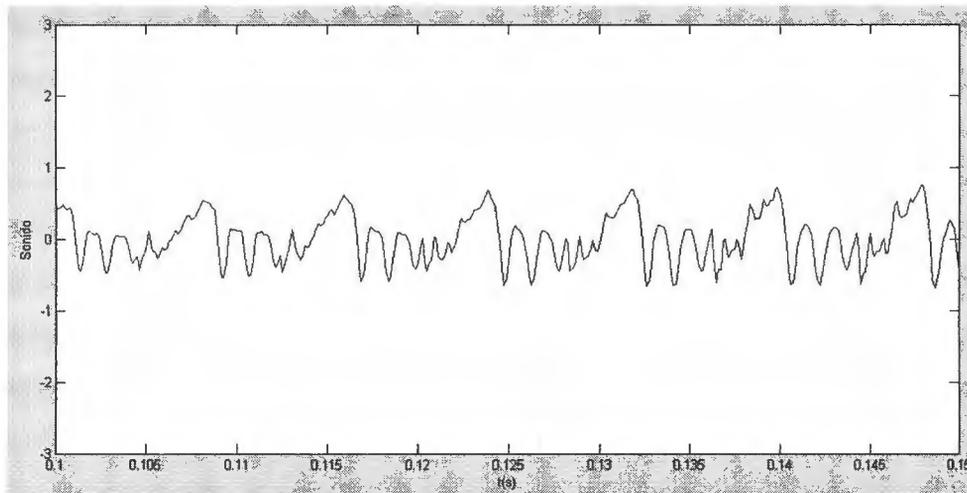


Figura 2.29 Ampliación de la letra "a"

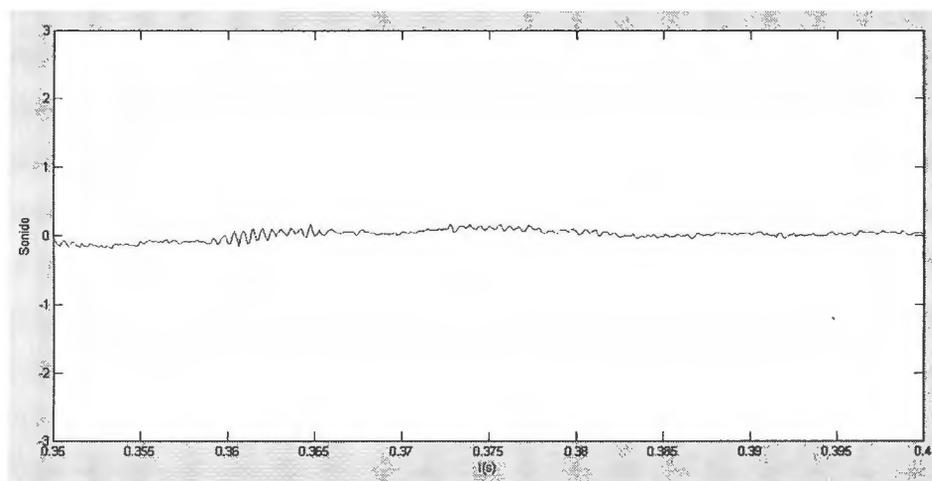


Figura 2.30 Ampliación de la letra "j"

Todas estas características se utilizan para poder caracterizar la voz. Antes de pasar a la caracterización, hablaremos un poco de la adquisición de la señal y el acondicionamiento de la misma.

### 2.7.2 Adquisición y digitalización

En un sistema la etapa de adquisición y acondicionamiento de la señal es muy importante en el procesamiento de voz. Esta etapa puede ser la diferencia entre tener éxito en un sistema o no. Este paso se refiere al cómo se adquiere la señal para después ser procesada y caracterizada. La digitalización de la voz es el proceso mediante el cual un sistema adquiere una señal analógica mediante un transductor, y mediante un convertidor análogo digital, se convierte en información que puede ser analizada con una PC. La figura 2.31 nos muestra el proceso de digitalización en un diagrama de bloques. Entre la adquisición y la conversión puede o no haber un pre-procesamiento aplicando algún tipo de filtro.



Figura 2.31 Proceso de digitalización de la voz

Para poder convertir la voz humana en una señal digital, se necesita establecer una frecuencia de muestreo. El teorema de muestreo de Nyquist, mostrado en la ecuación 2.27, dice que para poder replicar con exactitud la forma de una onda es necesario que la frecuencia de muestreo sea superior al doble de la máxima frecuencia a muestrear.

$$f_s \geq 2f_{max} \quad (2.27)$$

Una vez muestreada la información, se tiene que determinar un valor para cuantificar la señal. En los sistemas digitales se asignan números de bits por lo que asignar  $n$  bits nos dará  $2^n$  niveles de cuantificación.

Los tipos de archivos de audio se pueden dividir en dos grupos. Los archivos con pérdida y los archivos sin pérdida. Los archivos de sonido con pérdida son aquellos que usan un algoritmo de compresión con pérdida, es decir un tipo de compresión que representa la información (por ejemplo una canción), pero intentando utilizar para ello una cantidad menor de información. Esto hace que sea imposible reconstruir exactamente la información original del archivo. Se podrá reconstruir tan sólo una aproximación a la información original que contenía el archivo. El empleo de estos métodos de compresión con pérdida suele usarse en información analógica que quiere digitalizarse, como por ejemplo imágenes, audio, vídeo etc. Además tiene la gran ventaja de que obtendremos datos digitalizados que ocupen menos espacio en disco. Un ejemplo son los MP3, ACC, OGG, RM, MP4, WMA, etc.

Los archivos de sonido sin pérdida son aquellos que usando o no métodos de compresión, representan la información sin intentar utilizar menor cantidad de la información original. Hacen posible una reconstrucción exacta de la información original. Ejemplos son los archivos con extensiones AIF, FLAC, WAV, MIDI, etc.

### **2.7.3 Procesamiento de la señal de voz**

El siguiente paso, una vez almacenada la información en algún formato es el pre-procesamiento. Normalmente, para poder analizar la voz tenemos que filtrar la onda para poder eliminar el ruido del medio ambiente y enfocarnos en la información que realmente representa la voz humana. Una analogía podría ejemplificarse si tomamos en cuenta el fondo de una imagen digital y el sonido del medio ambiente de una señal acústica. Ésta es información

## Capítulo 2. Antecedentes

que está presente, pero no necesariamente es deseada en el sistema. Existen diferentes clasificaciones de filtros como se muestra en la tabla 2.13.

Filtro	Tipo de Filtro	Descripción
Analogico	Pasivos	Basado en resistores y capacitores
	Activos	Basado en resistores, capacitores y amplificadores operacionales
Digital	FIR (Respuesta Finita al Impulso)	No recursivos
	IIR (Respuesta Infinita al Impulso)	Recursivos

Tabla 2.13 Clasificación de Filtros

Los filtros típicamente usados para audio son de tipo FIR. En este tipo de filtrado no existe retroalimentación. Además, la respuesta al impulso  $H(w)$ , es de duración finita ya que si la entrada se mantiene en cero durante  $M$  periodos consecutivos la salida también será cero. Algunas de las ventajas de este tipo de filtros son las siguientes [54]:

- Un filtro FIR puede ser diseñado para tener fase lineal.
- Siempre son estables porque son hechos únicamente con ceros en el plano complejo.
- Los errores por desbordamiento no son problemáticos porque la suma de productos en un filtro FIR es desempeñada por un conjunto finito de datos.
- Un filtro FIR es fácil de comprender e implementar.

Uno de los métodos más usados para el diseño de filtros FIR en el procesamiento de voz es la segmentación y ventaneo. Para la segmentación de la señal de voz, normalmente se utilizan tramas de 20 ó 30 ms, con un desplazamiento típico de 10 ms. En la figura 2.32 se puede visualizar el procedimiento de segmentación.

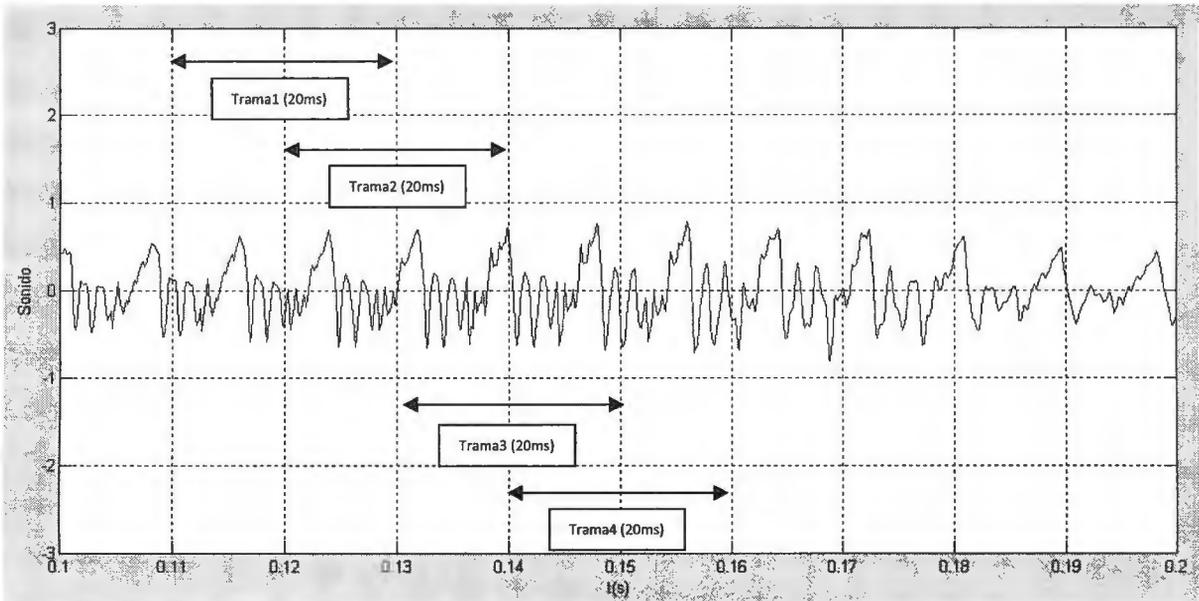


Figura 2.32 Ilustración de método de segmentación

Por último a cada trama generada se le aplica una ventana la cual es una función matemática que elimina los problemas causados por los cambios abruptos de la señal y elimina discontinuidades. Se aplica una convolución entre las ventanas y cada segmento obtenido. Existen diferentes tipos de ventanas, y las más utilizadas se resumen en la tabla 2.14 [55].

Ventana	$w(n)$	Imagen
Rectangular	$w(n) = 1$ para $n = 0, 1, 2, \dots, N - 1$	
Hamming	$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N}$	

Capítulo 2. Antecedentes

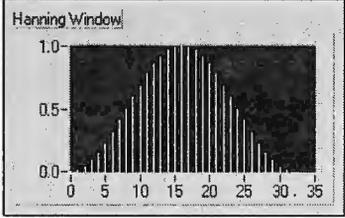
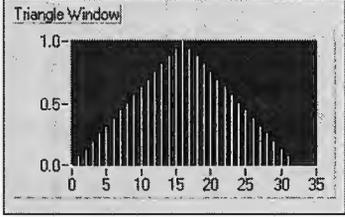
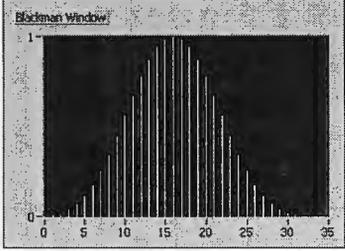
Hanning	$w(n) = 0.5 - 0.5\cos\frac{2\pi n}{N}$	
Triangular	$w(n) = 1 - \left \frac{2n-N}{N}\right $ para $n = 0, 1, 2, \dots, N - 1$	
Blackman	$w(n) = 0.42 - 0.5\cos\frac{2\pi n}{N} + 0.08\cos\frac{4\pi n}{N}$	

Tabla 2.14 Ventanas más comunes

Una vez seleccionado el tipo de segmentación y ventaneo, se puede aplicar el filtro. En la figura 2.33, 2.34 y 2.35 se puede observar la diferencia entre palabra “abeja” sin filtrar, y la misma palabra aplicando un filtro FIR pasabajas con una frecuencia de corte de 1000 Hz aplicando una ventana de Hamming, y también aplicando un filtro pasabandas con una frecuencia de corte mínima de 1000 Hz y una frecuencia de corte máxima de 2500 Hz.

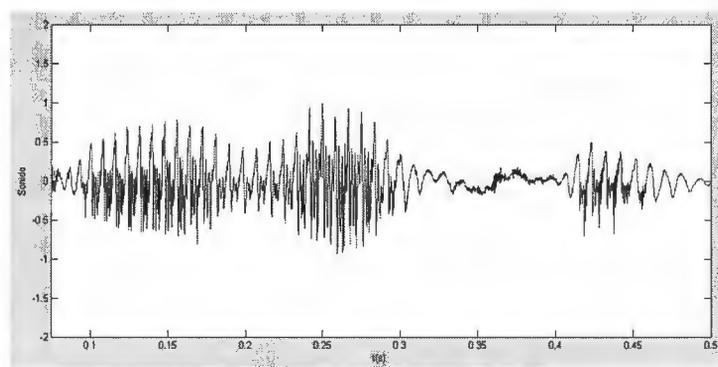


Figura 2.33 Palabra “abeja” sin filtrar

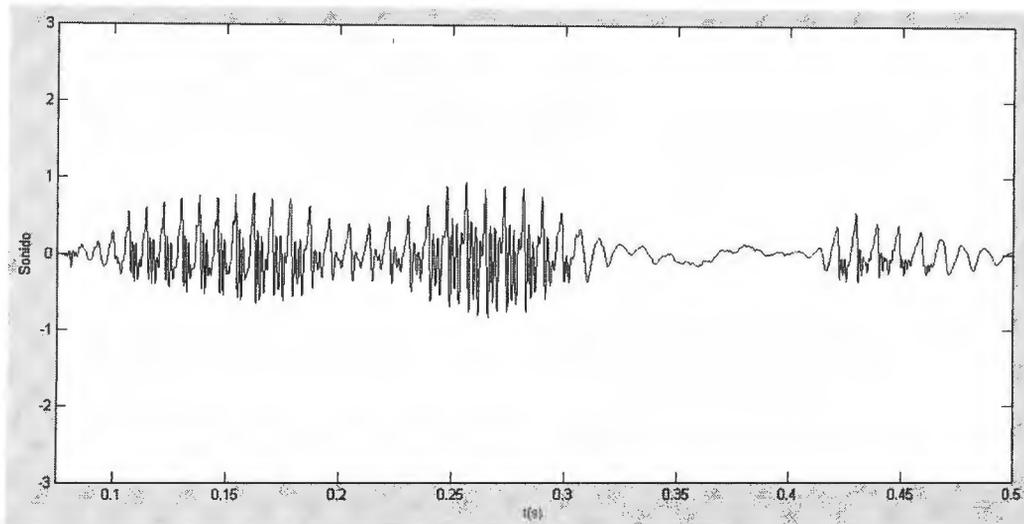


Figura 2.34 Palabra "abeja" con un filtro pasabajas con una frecuencia de corte de 1000 Hz

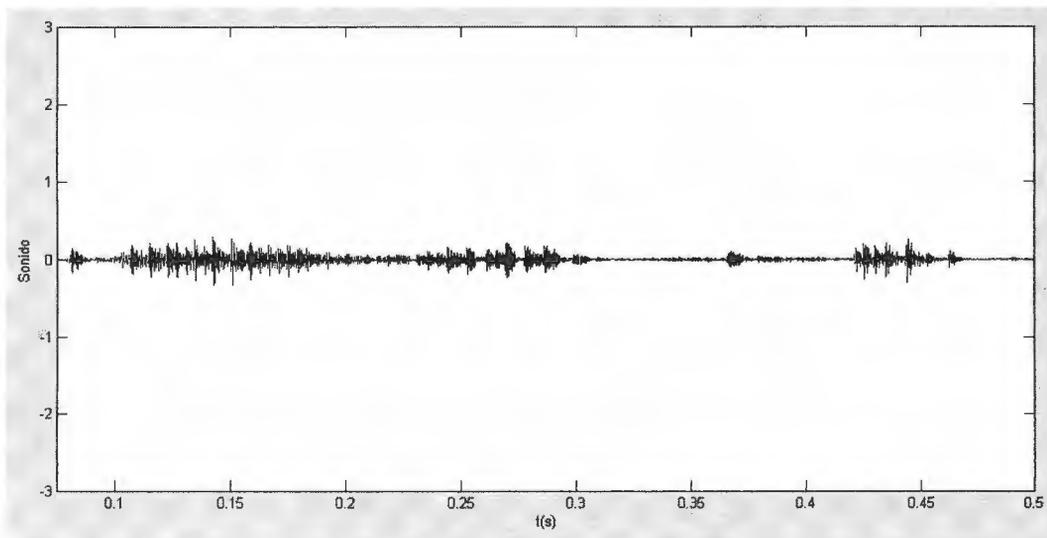


Figura 2.35 Palabra "abeja" con un filtro pasabandas con una frecuencia de corte mínima de 1000 Hz y una frecuencia de corte máxima de 2500Hz

El filtro tiene como función hacer un preénfasis para suavizar el espectro y reducir las inestabilidades del cálculo asociadas con las operaciones aritméticas de precisión finita (filtros FIR).

### 2.7.4 Caracterización de la voz

La caracterización de la voz se puede dividir en dos categorías: en dominio del tiempo y en dominio de la frecuencia. Existen diferentes tipos de métodos para cada uno pero en éste trabajo sólo se mencionará un ejemplo para cada uno.

En la caracterización basada en el dominio del tiempo, existe un método muy utilizado que es el análisis por energías. Este análisis utiliza el método de correlación donde se ve la similitud entre las energías de un par de señales. Cuando se utiliza una auto-correlación, se puede calcular la potencia promedio de la señal. En otras palabras, una auto-correlación es una correlación cruzada de la señal consigo misma. En la figura 3.36 se puede observar los segmentos de la palabra “abeja” después de aplicarles una ventana Hamming. En la figura 3.37 se observan los promedios de los valores absolutos de la auto-correlación de cada segmento. Los segmentos son de 30ms cada uno.

Este método es muy útil cuando se quiere diferenciar entre sonidos sonoros de sonidos sordos ya que en la gráfica un sonido sonoro es aquel que contiene más energía.

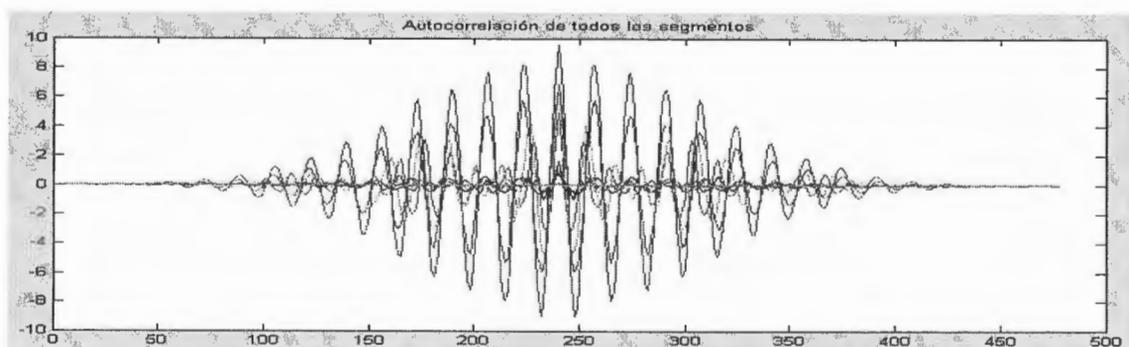


Figura 2.36 Segmentos con aplicación de ventana de Hamming



Figura 2.37 Promedio de los valores absolutos de la autocorrelación

Para entender más los métodos basados en frecuencia, primero es importante entender el modelo clásico de reproducción de voz. Este modelo se compone de un filtro variable en el tiempo, un generador de ruido aleatorio y de un generador de impulsos (figura 3.38). Los parámetros del filtro varían en función de la acción consciente que se realiza al pronunciar una palabra. El modelo tiene dos entradas, que dependen del tipo de señal. Para señales sonoras (vocales) la excitación es un tren de impulsos de frecuencia controlada, mientras que para las señales no sonoras (consonantes) la excitación es ruido aleatorio. La combinación de estas dos señales modela el funcionamiento de la glotis [56].

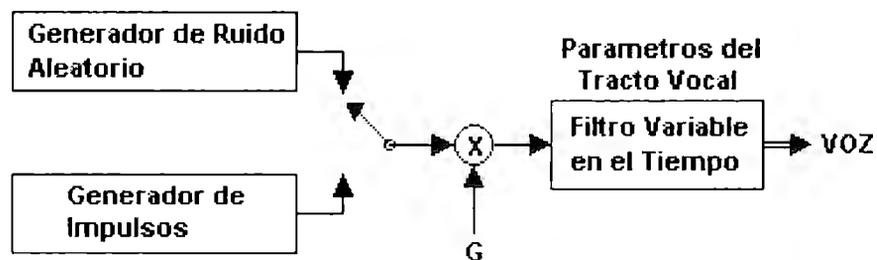


Figura 2.38 Modelo de producción de voz

## Capítulo 2. Antecedentes

El método de predicción lineal caracteriza la forma del espectro de un segmento de voz con un número reducido de parámetros. La codificación lineal predictiva, predice una señal en el dominio del tiempo con base en una combinación de muestras previas linealmente distribuidas (ecuación 2.28).

$$\tilde{s}(n) = \sum_{k=1}^p a_k s(n - k) \quad (2.28)$$

Donde  $\tilde{s}(n)$  es la señal a predecir,  $s(n - k)$  es la señal en tiempos anteriores,  $a_k$  ( $1 < k < p$ ) es un conjunto de constantes reales conocidas como coeficientes de predicción, que necesitan ser calculados y  $p$  es el orden del predictor. Una de las ventajas de este método es que además de ser muy preciso, es muy adecuado para la implementación computacional, pues es sencillo y de rápida ejecución. El error entre el valor real de la función y la función aproximada esta dado por la ecuación 2.29.

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^p a_k s(n - k) \quad (2.29)$$

Aplicando la transformada Z a la ecuación se tiene:

$$\begin{aligned} e(z) &= s(z) - \sum_{k=1}^p a_k s(z) z^{-k} \\ e(z) &= s(z) \left( 1 - \sum_{k=1}^p a_k z^{-k} \right) \end{aligned} \quad (2.30)$$

Sacando la función de transferencia:

$$A(z) = \frac{e(z)}{s(z)} = 1 - \sum_{k=1}^p a_k z^{-k} \quad (2.31)$$

Es decir que  $A(z)$ , a la que se denomina filtro inverso, será la función de transferencia de un sistema donde la entrada será la señal y la salida será el error. En la figura 2.39. se muestra el sistema y su inverso.

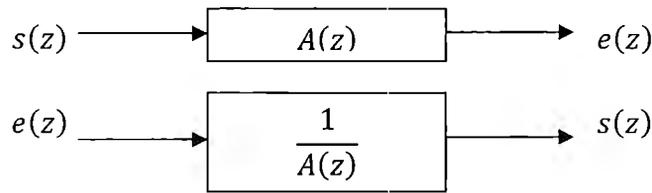


Figura 2.39 Función de transferencia del sistema y su inverso

El método de predicción lineal toma como base un modelo del tracto vocal representado como un filtro (usualmente lineal) variable en el tiempo (ecuación 2.32) donde  $G$  es la ganancia del filtro, que depende de la naturaleza de la señal (sonora o no sonora) [52]. Entonces, dada la señal  $s(n)$ , el problema consistirá en determinar los coeficientes de predicción y la ganancia.

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (2.32)$$

El error puede minimizarse obteniendo para los parámetros la relación matricial de la ecuación 2.33. Para resolver ésta matriz el método más utilizado es el de autocorrelación [56].

$$\begin{bmatrix} r_1 \\ r_2 \\ \dots \\ r_p \end{bmatrix} = \begin{bmatrix} r_0 & r_1 & \dots & r_{p-1} \\ r_1 & r_0 & \dots & r_{p-2} \\ \dots & \dots & \dots & \dots \\ r_{p-1} & r_{p-2} & \dots & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_p \end{bmatrix} \quad (2.33)$$

El vector  $[r_1, r_2 \dots r_p]$  corresponde a los  $p$  primeros coeficientes de autocorrelación de un segmento de señal. A la matriz que multiplica al vector de coeficientes se le denomina Matriz de Toeplitz. Por lo tanto, calculando los primeros  $p$  coeficientes de autocorrelación de un segmento de señal podemos generar la matriz de correspondiente. Con esto, sólo queda calcular su inversa y multiplicarla por el vector de autocorrelación para obtener los coeficientes

## Capítulo 2. Antecedentes

del filtro en la ventana de análisis. Esta matriz con la estructura Toeplitz permite la aplicación de la recursión de Levinson-Durbin para resolver el sistema.

Existen varios métodos para obtener el orden de los coeficientes predictores, uno de ellos consiste en sustituir el valor de la frecuencia de muestreo ( $f_s$ ) en la ecuación 2.34.

$$p = 4 + \frac{f_s}{1000} \quad (2.34)$$

Una vez caracterizado la señal, el siguiente paso es la clasificación. Aquí es cuando tomamos estos coeficientes y los procesamos para identificar que sonido es.

### 2.7.5 Clasificación de voz

Existen diferentes tipos de métodos de clasificación de voz, pero las principales se pueden agrupar en cuatro grandes técnicas [57]:

1. Técnicas topológicas basadas en el cálculo y comparación de distancias como el *"Dynamic Time Warping o DTW"*.
2. Redes Neuronales
3. Técnicas estadísticas o probabilísticas como los Modelos Ocultos de Markov (HMM)
4. Sistemas basados en el conocimiento como los sistemas expertos o algoritmos genéticos.

Para efectos prácticos se mencionarán los elementos básicos del método HMM, ya que es una de las técnicas más utilizadas. Un HMM es básicamente una cadena de Markov en la que cada observación de salida es una variable aleatoria  $X$  generada de acuerdo a una función de

probabilidad asociada a cada estado, formalmente hablando un modelo oculto de Markov está definido por [56]:

- $O = \{O_1, O_2, \dots, O_M\}$  Observaciones de salida, que pueden ser discretas o continuas.
- $\Omega = \{1, 2, \dots, N\}$  Conjunto de estados que representa el espacio de estados.
- $A = \{a_{ij}\}$  Matriz de transición de probabilidades, donde  $a_{ij}$  es la probabilidad de que se haga la transición desde el estado  $i$  al estado  $j$ .
- $B = \{b_i(k)\}$  Matriz de probabilidad de salida, donde  $b_i(k)$  es la probabilidad de emitir el símbolo  $k$  o en el estado  $i$ .
- $\pi = \{\pi_i\}$  Distribución de estados inicial donde  $\pi_i$  es la probabilidad de empezar en el estado  $i$ .

Para ejemplificar el método HMM, consideremos el sistema de la figura 2.40 [58]:

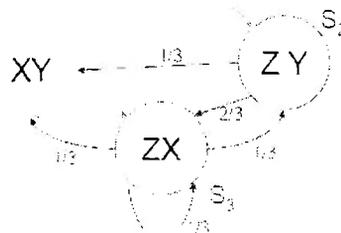


Figura 2.40 Sistema con tres estados

En este sistema, se presentan tres estados con las probabilidades de cada una de las transiciones. Por cada estado ( $S$ ), tenemos 2 posibles salidas, y sabemos que el sistema puede empezar aleatoriamente en el estado  $S_1$  o  $S_2$ . Las observaciones son  $X, X, Z$ . Teniendo esta información, podemos obtener los parámetros completos del modelo  $\lambda = (A, B, \pi)$ .

1. Número de observaciones:  $M = 3$

2. Número de estados:  $M = 3$

3. Matriz de probabilidades de transición:

$$A = \begin{bmatrix} 0 & 1/3 & 2/3 \\ 1/3 & 0 & 2/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \quad (2.35)$$

4. Matriz de probabilidades por salida de estado:

$B_1(X) = \frac{1}{2}$	$B_1(Y) = \frac{1}{2}$	$B_1(Z) = 0$
$B_2(X) = 0$	$B_2(Y) = \frac{1}{2}$	$B_2(Z) = \frac{1}{2}$
$B_3(X) = \frac{1}{2}$	$B_3(Y) = 0$	$B_3(Z) = \frac{1}{2}$

5. Distribución de estados inicial:

$$\pi_1 = \frac{1}{2} \quad \pi_2 = \frac{1}{2} \quad \pi_3 = 0$$

Una vez obtenidos estos parámetros, existen técnicas de solución que han sido estudiadas y comprobadas como el algoritmo de Viterbi para encontrar la secuencia óptima de estados asociados con las observaciones, y el algoritmo de Baum-Welch para ajustar los parámetros del modelo y maximizar la probabilidad de las secuencias en las observaciones [59].

### **2.7.6 Reconocedor de voz de Microsoft Windows**

Microsoft se ha involucrado en la investigación y desarrollo del reconocimiento de voz por más de 15 años, y ha reconocido que esta tecnología será muy importante en las interfaces de hombre-máquina. En 1993, los fundadores del sistema de reconocimiento de voz de Microsoft, fueron reclutados de la universidad de Carnegie Mellon. Este grupo, con la licencia de la universidad y el sistema Sphinx II, que en aquel momento llevaba la vanguardia, formaron el punto inicial del desarrollo de Microsoft en esta área. Los integrantes de dicho grupo eran Xuedong Huang, Fil Alleva, and Mei-Yuh Hwang. La adquisición de Entropic en 1999 hacía ver el interés de Microsoft por ésta tecnología. Con Entropic vino HTK, una librería de herramientas para la utilización de los Modelos Ocultos de Markov (HMM) desarrollado por el grupo de reconocimiento de voz de la Universidad de Cambridge. Esta librería es gratuita en la actualidad bajo la licencia de Microsoft.

En 2007, la adquisición de Tellme Networks reafirmó el interés de Microsoft por la tecnología para el reconocimiento de voz como media de acceso a la información en cualquier momento desde cualquier dispositivo. Durante el transcurso de los años, las personas que han estado trabajando en este proyecto, pasó de ser un pequeño grupo a cientos de personas en tres continentes diferentes. En 1998, con la tecnología madurando, Microsoft decidió incluir esta herramienta en su plataforma de Windows. En el año 2001, Windows XP se lanzó con la aplicación de SAPI 5.01 que incluía herramientas para reconocimiento y síntesis de voz. Microsoft Office XP fue una de las aplicaciones que utilizó esta plataforma para crear una interface donde el usuario podía dar comandos, controlar y dictar palabras. Cuando Windows Vista fue lanzado, el reconocimiento de voz fue integrado en su plataforma de accesibilidad, y esto se integró a todos los aspectos del sistema. Hoy en día la tecnología de reconocimiento de voz de Microsoft soporta varios idiomas como es el inglés (E.U y Reino Unido), japonés, chino (simplificado y tradicional), alemán, francés, y español.

## Capítulo 2. Antecedentes

Microsoft tiene seis laboratorios alrededor del mundo que operan de manera similar a un departamento de ciencias computacionales de una universidad donde los investigadores publican documentos, colaborando con la comunidad de investigación y dentro de sus actividades se dedican a dar conferencias. El grupo de investigación se enfoca más en construir prototipos que en escribir código. Las contribuciones de Microsoft han hecho más robusto el reconocimiento de voz en ambientes con ruido, los algoritmos mejoran la tasa de reconocimiento a través del entrenamiento de las palabras [53]. El grupo de Microsoft, tiene varios proyectos en curso como son los siguientes [60]:

1. Procesamiento de Audio: Captura del sonido, mejora del habla, cancelación del eco, arreglo de micrófonos, arreglo de bocinas y sonido espacial.
2. Robustez del ruido: Investigación sobre cómo mejorar el reconocimiento con ruido de fondo presente.
3. Modelado Acústico: Modelado de micrófonos y variaciones acústicas.
4. Modelado de Lenguaje: Diferenciar entre palabras
5. Comprensión: Con éste proyecto se pretende no solo reconocer las palabras que el usuario dice, sino también entender que es lo que significan
6. Búsqueda por voz: Aquí se enfoca en la búsqueda de información como negocios desde un teléfono.

## Capítulo 3. Diseño de Instrumentista Quirúrgico

La instrumentación quirúrgica se define como "el arte y la capacidad de participar en el acto quirúrgico mediante el manejo del instrumental y accesorios, y la aplicación de técnicas de esterilización" [61]. Para poder diseñar un sistema que simule a un instrumentista quirúrgico, se debe comprender el proceso general y después desglosarlo en partes para poder modelarlo.

La analogía entre un sistema real y un sistema simulado que sirve como base para nuestro diseño se muestra en la figura 3.1.

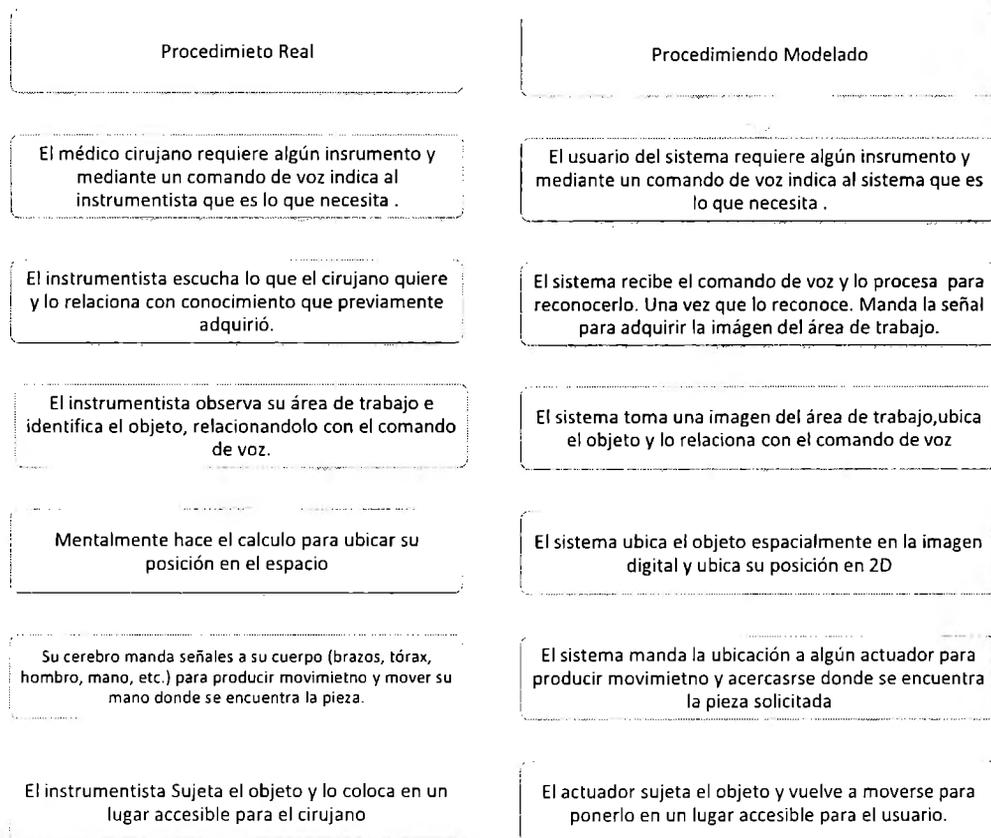


Figura 3.1. Procedimiento general de un médico cirujano y un instrumentista y su modelado con un sistema automático.

### 3.1 Esquema general del sistema

Debido a su carácter académico, el proyecto será desarrollado con recursos que se tengan disponibles en la institución.

Para poder empezar a implementar un sistema que modele el proceso antes descrito, se requiere establecer el ambiente y el área de trabajo.

Existen elementos fundamentales como son: el usuario, un robot industrial (que funge como actuador en nuestro sistema), Unidad de procesamiento de datos (PC) y los instrumentos a utilizar. La figura 3.2 muestra un posible arreglo del sistema. El manipulador cuenta con un área de toma, donde se colocan aleatoriamente los instrumentos solicitados por el usuario. Esta área puede ser restringida dependiendo de las capacidades del manipulador. Dentro de la zona de trabajo tiene que existir un área de entrega de instrumentos que se encuentre al alcance del usuario. Para evitar problemas de ruido con el cableado, se buscó colocar la PC y el controlador, cerca del manipulador.

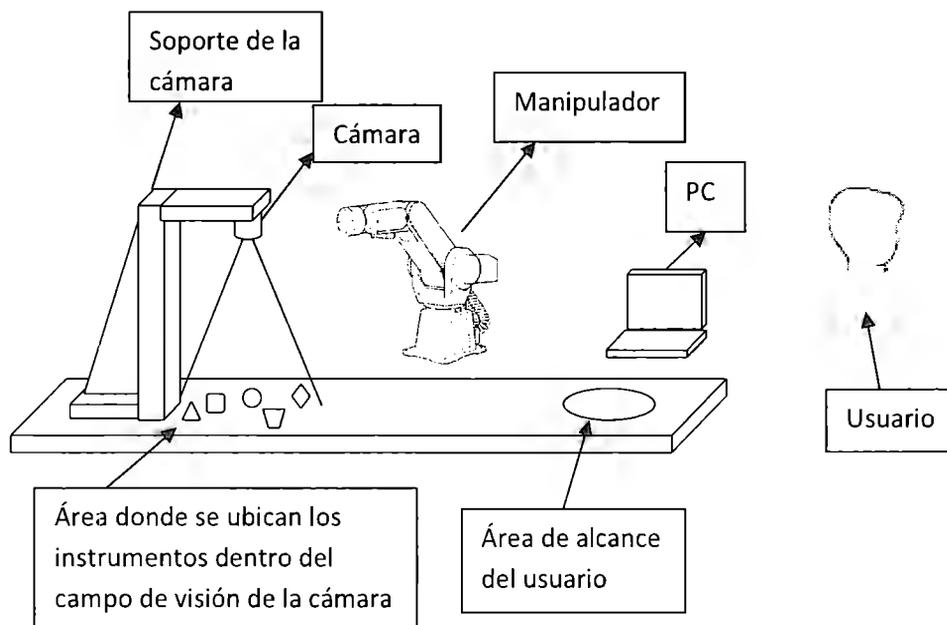


Figura 3.2. Arreglo del sistema

### 3.2 Conexiones y flujo de la información

En este esquema el usuario emite un comando de voz que viaja por el aire y es recibido por un transductor, que en este caso es un micrófono integrado en la PC. La PC analiza la señal adquirida para reconocer lo que el usuario está solicitando. Una vez reconocido el comando, se manda una señal a la cámara por medio de un puerto USB para tomar la imagen y almacenarla en la PC. La imagen es analizada y comparada con una imagen patrón previamente cargada por el usuario. El programa localiza en el espacio el instrumento y calcula las coordenadas y orientación. La PC manda esta información al controlador del robot por medio de otro puerto USB (En este caso al puerto USB se conecta un adaptador serial) y éste ejecutará una rutina previamente programada para tomar el objeto y dejarlo en un espacio al alcance del usuario. La figura 3.3 muestra las conexiones del sistema.

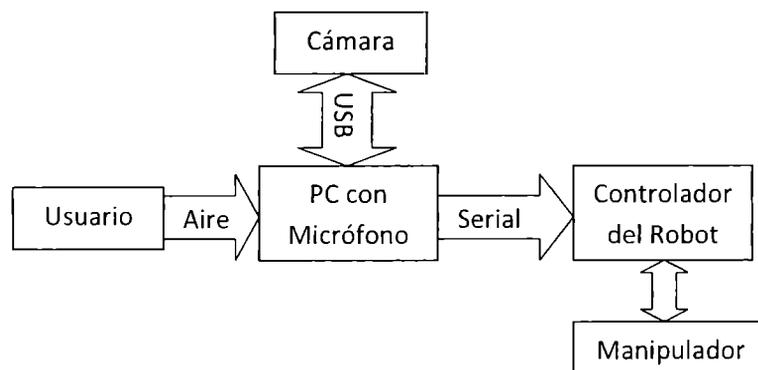


Figura 3.3. Conexiones del sistema

### 3.3 Soporte de la cámara

La cámara del sistema es fijada en un punto encima del área de trabajo. El lente de la cámara tiene que estar colocado lo más paralelamente a la superficie donde se ubican los objetos para evitar distorsión y errores de perspectiva. Éste soporte tiene que ser lo más robusto posible para evitar vibraciones y soportar alguna colisión del manipulador. El área de trabajo del robot y el campo de visión de la cámara son las que determinan la altura del soporte. Debido a que la cámara tiene que estar encima de los objetos a identificar, y el robot tiene que entrar entre la cámara y los objetos, se diseñó un soporte en forma de “S” con ángulos rectos en sus dobleces como se muestra en la figura 3.4.

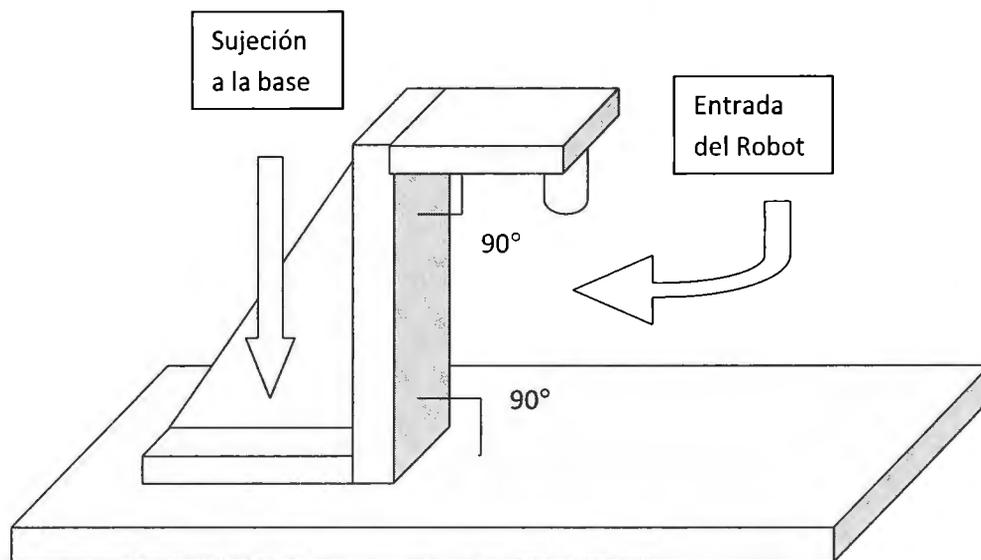


Figura 3.4. Diseño de soporte de cámara

El diseño se modela en el software CATIA V. En la figura 3.5 y 3.6 se muestran imágenes de la pieza con dimensiones propuestas. La altura de la base se determinó conforme a la extensión máxima del robot en su posición vertical. Esta distancia se obtiene de la hoja de especificaciones y es de aproximadamente 800mm. El material con lo que se construye la

base es acrílico, ya que es un material relativamente ligero, estético y manejable. El calibre tiene que ser lo suficientemente grueso para poder evitar vibraciones y soportar colisiones del manipulador. El soporte cuenta con un refuerzo en forma triangular localizado en la parte posterior para evitar que la pieza se flexione por la carga.

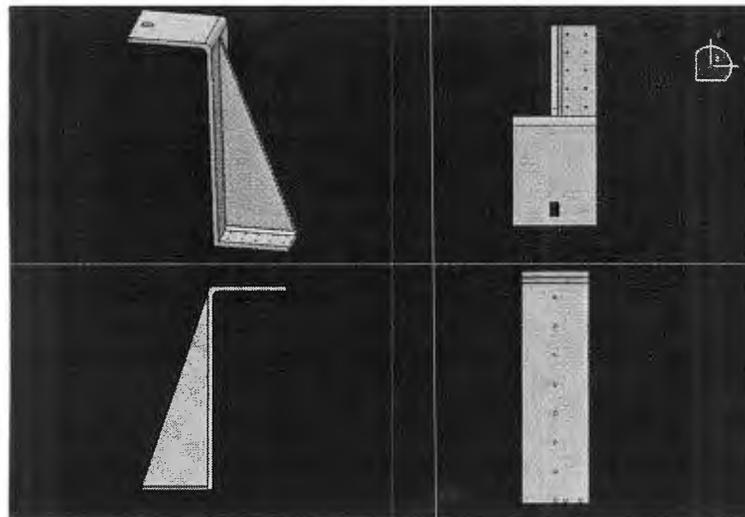


Figura 3.5. Diferentes vistas del modelo del soporte de la cámara

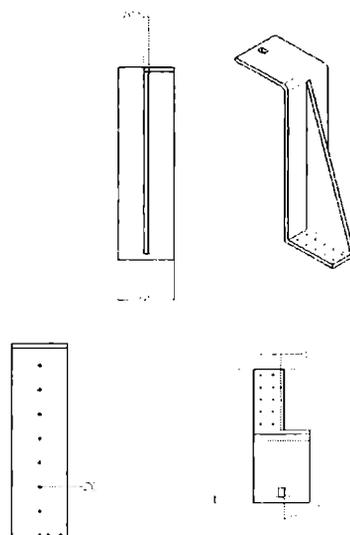


Figura 3.6. Diseño y medidas propuestas del soporte de la cámara

### 3.4 Selección de la cámara

En cuanto a la cámara se refiere, existe una amplia gama de opciones, pero para este proyecto se consideró una cámara del tipo *Webcam* con conexión USB, ya que de lo contrario, sería muy costoso el adquirir otro tipo de cámara y posiblemente, si utilizáramos una cámara análoga, se tendría que comprar una alguna tarjeta capturadora de cuadros (*Frame Grabber*). Hoy en día podemos conseguir *Webcams* con resoluciones suficientemente grandes para este tipo de trabajos como pueden ser 2 mega píxeles.

El software LabVIEW es compatible con este tipo de cámaras ya que permite configurar cualquier dispositivo de imagen que use *DirectShow* [62]. Ésta es una arquitectura que permite manipular archivos multimedia en un ambiente Windows. La librería que contiene estas funciones se llama NI-IMAQ para cámaras USB, y contiene funciones para adquirir una sola imagen o imágenes continuas, seleccionar modos de video e incluye asistentes para crear de manera rápida algún prototipo. Los requerimientos mínimos para poder utilizar ésta librería son los siguientes [63]:

1. Microsoft Windows Vista/Vista x64/XP/2000
2. LabVIEW 7.0 o más reciente.
3. Módulo *Vision Development* 7.1 o más reciente.
4. National Instruments *Vision Acquisition Software 7.1* o más reciente.

### 3.5 Área de trabajo

Para este trabajo se utiliza un robot articulado de cinco grados de libertad. Los robots de cinco grados de libertad, cuentan con un volumen de área de trabajo en forma de casi una esfera completa. Considerando que el robot se ubica encima de una mesa, se puede considerar que el área de trabajo para colocar los instrumentos quirúrgicos tiene forma de media luna. Las dimensiones de la media luna dependerán de la altura a la que se

encuentre el robot y del gripper que se usa. En la figura 3.7 se puede ver, en las partes sombreadas, un aproximado del área de trabajo en milímetros de una vista superior al robot y de una vista lateral.

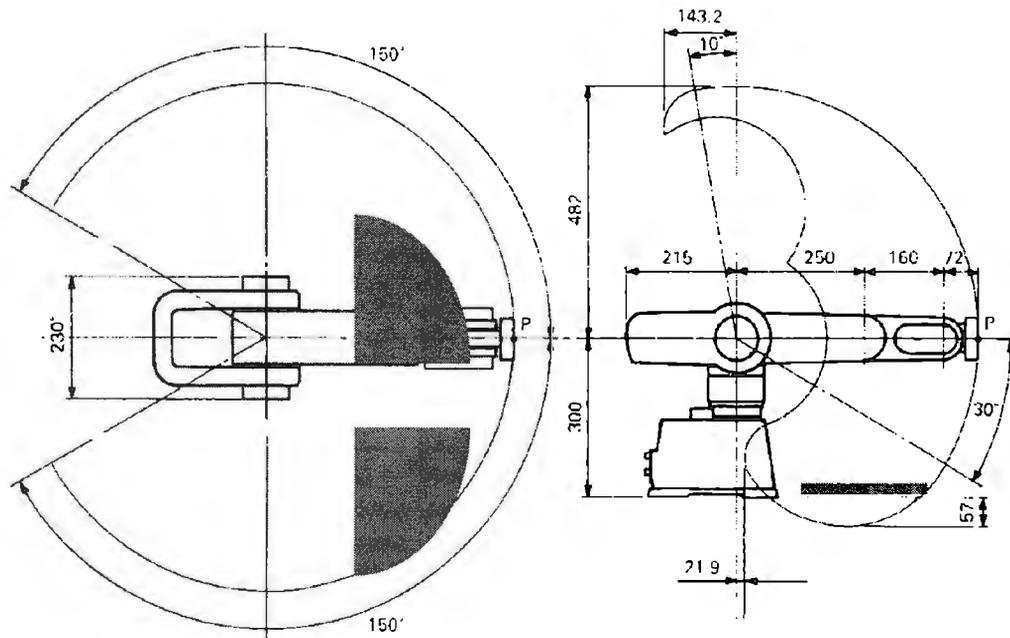


Figura 3.7. Área de trabajo aproximada considerada para la colocación de instrumentos

Esta misma área de trabajo es en la que se enfocará para el análisis de imágenes por lo que busca que esta superficie tenga un color contrastante a las piezas para poder ayudar en el reconocimiento. Considerando que los materiales de los instrumentos quirúrgicos son hechos a base de aceros inoxidable, según la norma mexicana de la secretaría de salud NOM-068-SSA1-1993, y que a su vez concuerda con la norma internacional ISO 7153-1:1991, se utilizará un material oscuro. Otra característica que se busca en el material que sirve como superficie donde se coloquen los instrumentos, es que sea de un material acolchonado. Esto se hace con la intención de que los instrumentos no se maltraten y el robot tenga un poco de tolerancia en sus movimientos al momento de tomar las piezas.

En cuanto a las condiciones de ruido en las imágenes y en la señal de voz, se determinarán parámetros basados en la experimentación que se realice durante la implementación.

### 3.6 Sujeción de Instrumentos

Debido a la no uniformidad de las piezas, es muy complicado diseñar un gripper que pueda sujetar diferentes instrumentos con diferentes formas y tamaños. En la figura 3.8 podemos ver algunos de los instrumentos quirúrgicos que se utilizan [64]. Para poder diseñar una herramienta que cumpla con la tarea, lo que se buscó es un parámetro que sea uniforme para la mayoría de los instrumentos el cual no se encontró en su forma, por lo que la opción fue considerar el material.

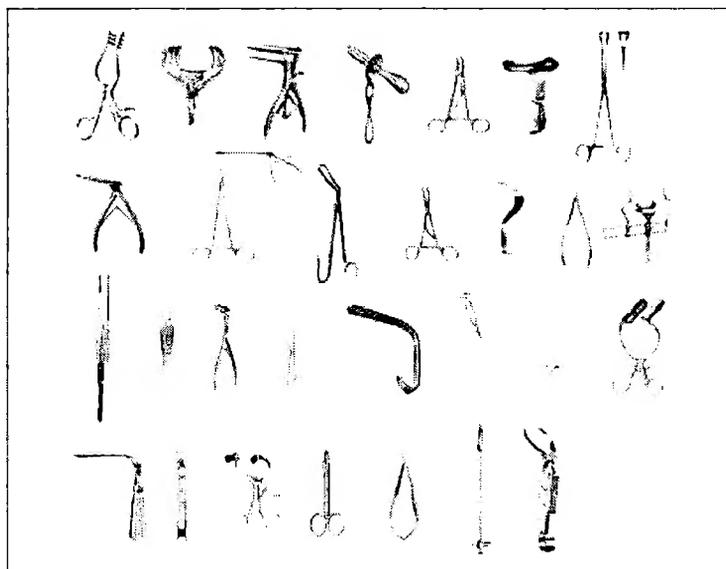


Figura 3.8. Ejemplo de diferentes tipos de instrumentos quirúrgicos

Considerando que el material es acero inoxidable, depende de los componentes de la aleación sus propiedades magnéticas. Una aleación común en instrumentos quirúrgicos es hierro, cromo y níquel. Un acero inoxidable tiene como estructura principal el hierro, el cual es un material magnético. La aleación con el cromo nos ayuda a evitar la corrosión y el desgaste, y la aleación con el níquel nos da un buen acabado. La presencia del níquel en la aleación es lo que contrarresta las propiedades magnéticas.

Se propuso una solución basada en un electroimán que se active cuando la herramienta del manipulador este en una posición sobre el instrumento, de esta manera se creará un campo magnético y sujetará la pieza. En el punto fijo donde se deja la pieza al alcance del usuario, se desactiva el electroimán para dejar la pieza. Actualmente en la industria existen electroimanes de carga que pueden soportar hasta 3 toneladas [65]. El principio que se utiliza es el mostrado en la figura 3.9 donde una corriente es inducida a través de una bobina con un núcleo de hierro (no mostrado en la figura), y por consecuente se crea un campo magnético que genera una fuerza de atracción determinada por la ecuación 3.1.

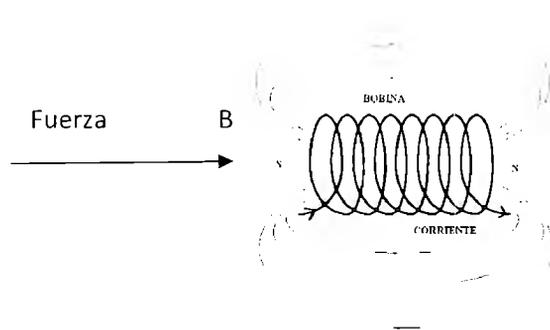


Figura 3.9. Principio de un electroimán.

$$F = IxBxl \quad (3.1)$$

Donde  $F$  es la fuerza,  $I$  es la corriente,  $B$  es el campo magnético y  $l$  es la longitud del alambre [66].

### 3.7 Diseño de lógica de programación

Para la programación del sistema se utiliza el software LabVIEW 8.5 con licencia académica la cual cuenta con las librerías de visión las cuales contienen herramientas que se utilizan para el reconocimiento de imágenes. Esta herramienta es accesible, y por otro lado, la programación basada en bloques de LabVIEW es de fácil implementación. Otra de las ventajas de LabVIEW es su fácil integración con aplicaciones .NET como es el *Windows Speech Recognizer 8.0 (WSR)* que se utiliza para el reconocimiento de voz. Se decidió utilizar este sistema ya que es un recurso el cual está incluido en el paquete de Windows Vista y ha sido desarrollado y probado por varios años. Por la parte de comunicación, LabVIEW cuenta con herramientas para configurar diferentes tipos de puertos, entre ellos el serial, que se usa para la comunicación con el robot. El esquema general de la lógica del programa se muestra en la figura 3.10.

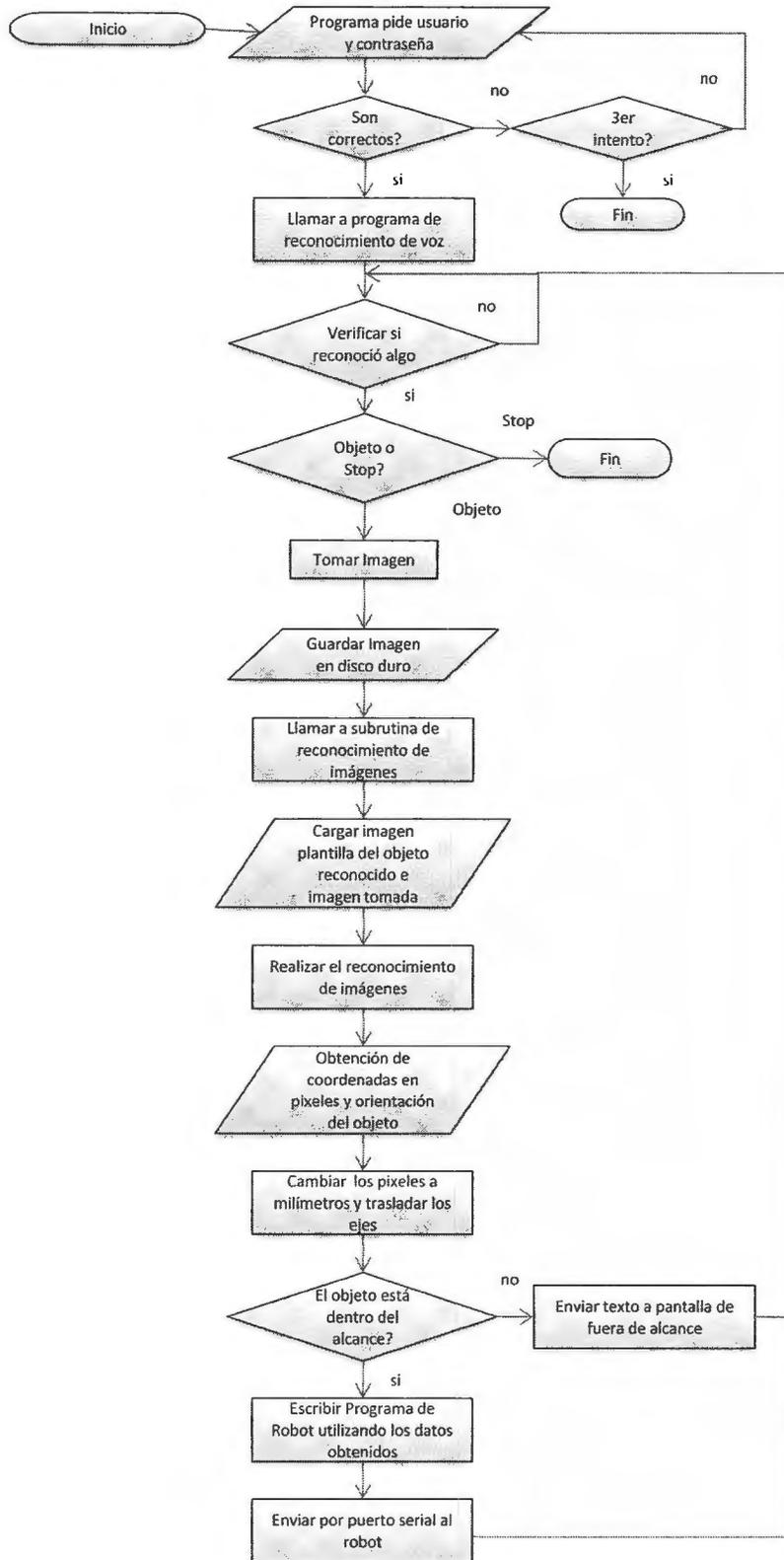


Figura 3.10. Diagrama de flujo general del programa

### Capítulo 3. Diseño de Instrumentista Quirúrgico

El programa cuenta con una etapa de seguridad al principio, la cual pregunta un usuario y contraseña para poder utilizar el sistema. Esto se pensó, para que sólo personas autorizadas puedan ejecutar el programa. Una vez que se tenga acceso al programa, lo primero que trata sistema es de identificar un comando de voz el cual tenga en una pequeña base de datos previamente creada. Existen dos tipos de comandos: solicitud de un objeto o finalización del sistema. En caso de que el comando haya sido de finalización de sistema, el programa parará su ejecución. En caso de que el sistema reciba la solicitud de un objeto, lo primero que se hace es tomar una imagen del área de trabajo y se guarda en la memoria de la PC. Luego se llama a una subrutina de reconocimiento de imágenes a la que se carga la imagen previamente tomada y la plantilla del objeto a buscar. Las plantillas de los objetos se guardaran en el disco duro. Con ésta información se aplica reconocimiento de imágenes y se tiene como salida unas coordenadas en dos dimensiones con unidades en pixeles y también la orientación del objeto. Esta información tiene que estar en unidades que tengan sentido para el robot, que en este caso son milímetros.

Una vez cambiadas las unidades se trasladan los ejes coordenados, ya que en las imágenes tenemos un origen diferente al origen del robot. El programa verifica que estas coordenadas estén dentro de unos límites previamente establecidos por el usuario con la finalidad de evitar colisiones del manipulador con algún objeto dentro del área de trabajo.

En caso de que la pieza no esté dentro de los límites establecidos, el programa envía un mensaje de “objeto fuera de alcance”, y regresa al punto donde empieza a reconocer comandos de voz. Si la pieza a encontrar esta dentro de los límites, se utilizan los datos para construir un programa que se envía al robot para que ejecute un ciclo completo. El programa es enviado mediante una comunicación serial. Por último, el robot recibe el programa y lo ejecuta inmediatamente haciendo movimientos de aproximación hasta llegar a un punto cerca al área de trabajo. Una vez que el robot se posiciona sobre el objeto, realiza movimientos más finos para tomar la pieza y dejarla en un lugar accesible para el usuario.

### 3.8 Diseño de interface de usuario

En la interface del programa con el usuario se trata de mostrar información relevante al usuario, así como proporcionar campos para entrada de datos. Se consideran datos importantes a mostrar los siguientes: comando de voz reconocido, imagen tomada, ubicación de la pieza en la imagen, coordenadas calculadas y programa que se envía al robot. Como campos de entrada, lo único que se tienen son los campos de seguridad (usuario y contraseña). Un bosquejo de la interface se muestra en la figura 3.11.

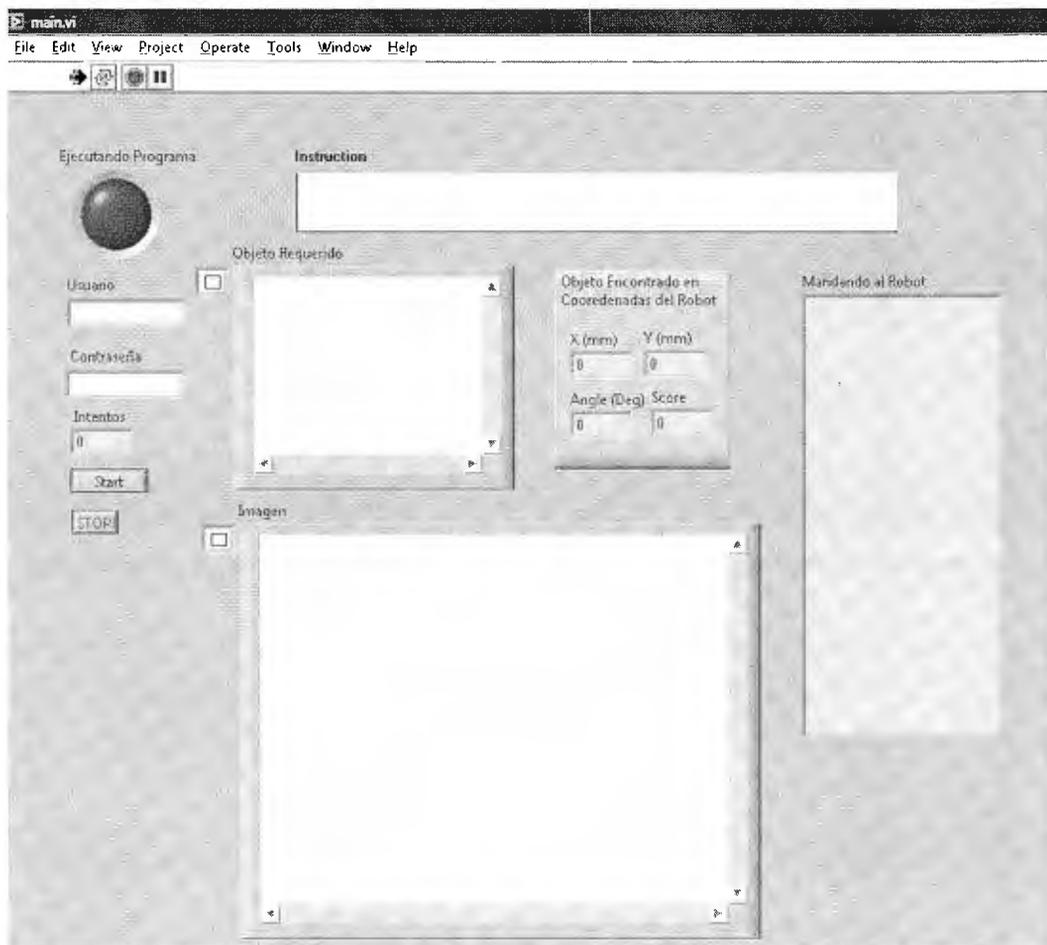


Figura 3.11. Interface con el usuario.

### Capítulo 3. Diseño de Instrumentista Quirúrgico

## Capítulo 4. Implementación de Instrumentista Quirúrgico

En este capítulo se explican los pasos que se siguieron para la implementación de un Instrumentista quirúrgico basado en reconocimiento de voz, reconocimiento de imágenes y un robot industrial. La implementación es dividida en partes muy específicas, que después se integran formando un sistema completo. En esta sección se detalla que fue lo que se utilizó así como algunos problemas que se fueron encontrando durante su desarrollo y las soluciones a los mismos.

### 4.1 Selección del robot y PC

El primer paso de este proyecto fue elegir el robot y la PC a utilizar. La elección se realizó en base a las características de cada equipo y a los recursos con que se contaban. En la tabla 4.1 se muestra un cuadro comparativo de las características que se tomaron en cuenta para la elección del robot, y sombreadas están las características críticas que se consideraron para la elección del robot.

	FANUC	PUMA	MITSUBISHI	CRS	ABB
Disponibilidad	Nula	Alta	Alta	Alta	Alta
Condiciones físicas	Muy Buenas	Malas	Buenas	Muy Malas	Muy Buenas
Alcance	Bueno	Bueno	Regular	Regular	Regular
Comunicación con PC	Factible	Factible	Factible	Factible	Factible
Precio	Sin Costo	Sin Costo	Sin Costo	Sin Costo	Muy Alto
Grados de libertad	6	6	5	6	6

Tabla 4.1 Tabla de comparación de robots

#### Capítulo 4. Diseño de Instrumentista Quirúrgico

La decisión de utilizar el robot Mitsubishi RV-M1 se tomó ya que cuenta con las características suficientes para realizar este proyecto. Una de las ventajas que tiene el utilizar este robot es que ya cuenta con una mesa diseñada para colocarlo y tiene el suficiente espacio para colocar los instrumentos y la base de la cámara. La figura 4.1 muestra una imagen de la base donde es montado el robot el área de trabajo.

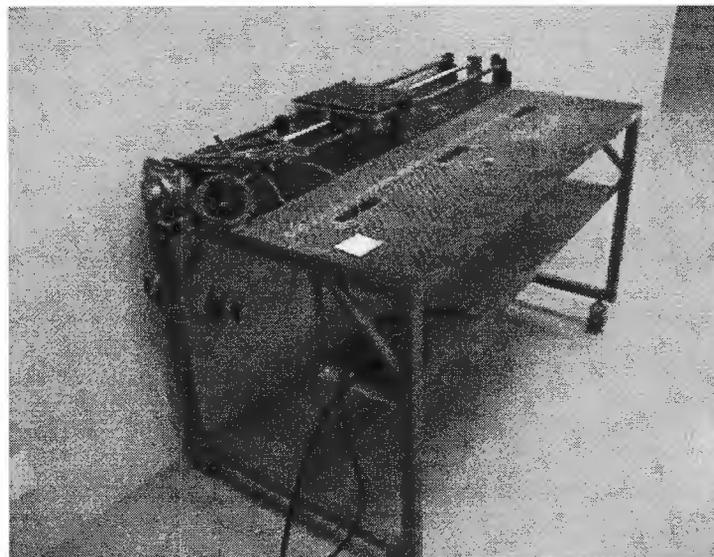


Figura 4.1 Mesa que se utilizará para montar el robot y será nuestra área de trabajo

En cuanto a la selección de la PC, se utiliza una computadora portátil tipo *laptop* que cuenta con Windows Vista y un puerto USB marca Toshiba modelo Satellite E105 con las características que se muestran en la tabla 4.2. Se eligió este equipo ya que es un recurso accesible y cuenta con las características necesarias para el proyecto.

Procesador	Intel® Core™2 Duo
Sistema Operativo	Windows Vista® Home Premium (SP1, 64-bit)
Memoria RAM	4GB PC6400 DDR2 800MHz SDRAM
Memoria Disco Duro	320GB HDD (5400rpm)
Periféricos Integrados	Webcam y Micrófono integrados
Puertos USB	2-USB (2.0)

Tabla 4.2 Especificaciones de computadora personal.

Este equipo cuenta con el reconocedor de voz de Microsoft para Windows versión 8.0 en inglés, por lo que los comandos que reconoce el sistema son en idioma inglés. El equipo cuenta con LabVIEW 8.5 que se tiene accesible en laboratorio del ITESM CCM con licencias académicas y las librerías de visión.

## 4.2 Comunicación del robot y PC

El siguiente paso en la implementación fue lograr establecer comunicación entre el robot y la PC. El primer problema que se encontró fue con la conexión física ya que el tipo de puertos con los que cuenta el robot y los puertos de la PC son diferentes. El robot utiliza conector RS232C (DB25), y la PC sólo cuenta con puertos USB. En el mercado existen adaptadores USB-Serial, pero los más comunes son los USB –DB9, así que la solución fue adquirir un cable USB-DB9, después conectar un convertidor DB9-DB25, y por último construir un cable DB25 tipo *null modem* con las conexiones de transmisión de información cruzadas como se muestran en la figura 4.2 [67].

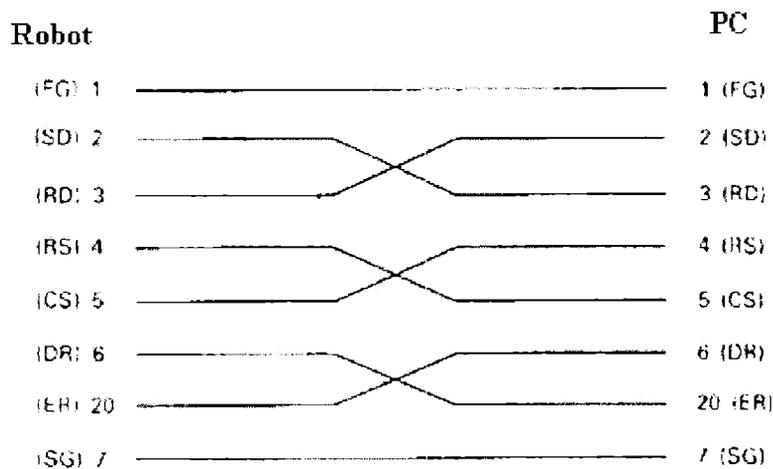


Figura 4.2 Conexiones del conector RS232C

Una vez construido el cable, la conexión queda como se muestra en la figura 4.3.



Figura 4.3 Cable de conexión PC-Robot

Una vez solucionado la conexión física del robot con la PC, se configuraron los puertos tanto en el robot como en la PC. Las dos configuraciones deben de tener los mismos valores para que puedan establecer comunicación. El Robot cuenta con dos interruptores en la parte lateral de la unidad de control llamados SW2 y SW3. Con el SW2 podemos configurar los parámetros del puerto y el SW3 establece la tasa de baudios a la que trabajará la conexión. Los valores utilizados son los estándares del robot y se muestran en la tabla 4.3.

Tasa de Baudios	9600
Bits de parada	2
Paridad	Par
Tamaño de palabra	7 Bits
Factor de tasa de baudios	X16

Tabla 4.3 Configuración puerto serial del robot.

Por parte de la configuración en la PC, LabVIEW cuenta con una herramienta para configurar el puerto llamada *VISA Configure Serial Port*. En la figura 4.4 se muestra la configuración que se asignó al puerto COM9 de la PC establecido por Windows cuando se conectó el adaptador USB-DB9. Esta configuración se guardó en un *subVI* (nombre que se dan a las subrutinas de LabVIEW) llamado **SCROBOT.vi** que da como salida la configuración del puerto, y un *cluster* (grupo de datos de diferente tipo) de datos de error establecidos por LabVIEW como son estatus de error, código y origen de error.

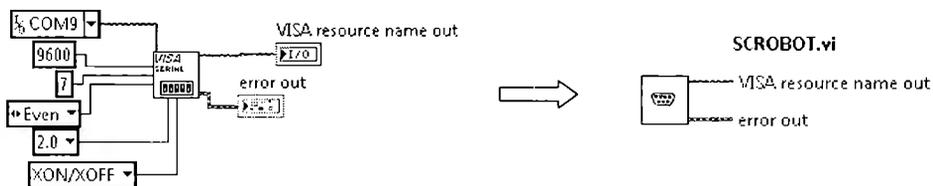


Figura 4.4 Configuración de puerto para PC utilizando herramienta *VISA Configure Serial Port* de LabVIEW

En la figura 4.5 podemos observar un simple programa para enviar una instrucción al robot utilizando el subVI que se acaba de crear y en la figura 4.6 se observa la posición del robot que se obtendría al ejecutar la instrucción. La instrucción del robot, define la posición número 30 con las coordenadas (0,200,0) de la herramientas con un *pitch* (cabeceo) de -93 grados y un *roll* (guiñada) de la muñeca de -10 grados. Los iconos de *VISA Write* y *VISA Close* (segundo y tercero), son las funciones que mandan la información al puerto serial y cierran la comunicación. El último ícono *Simple Error Handler.vi* es una función para manejo de errores. En caso de haber un error, muestra un cuadro de diálogo si se desea. Por último se tiene que a la instrucción se concatena un *Carriage Return* (salto de línea) para identificar que la línea de programa ha acabado. Este detalle es importante ya que sin él, el robot no ejecutará el comando.

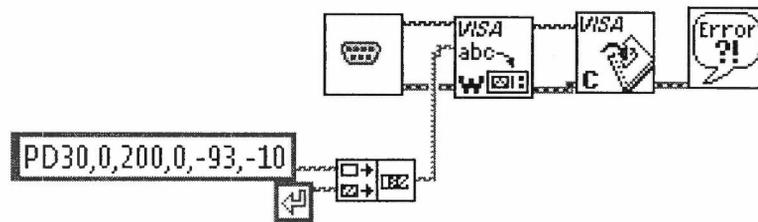


Figura 4.5 Programa que manda a definir una posición al robot

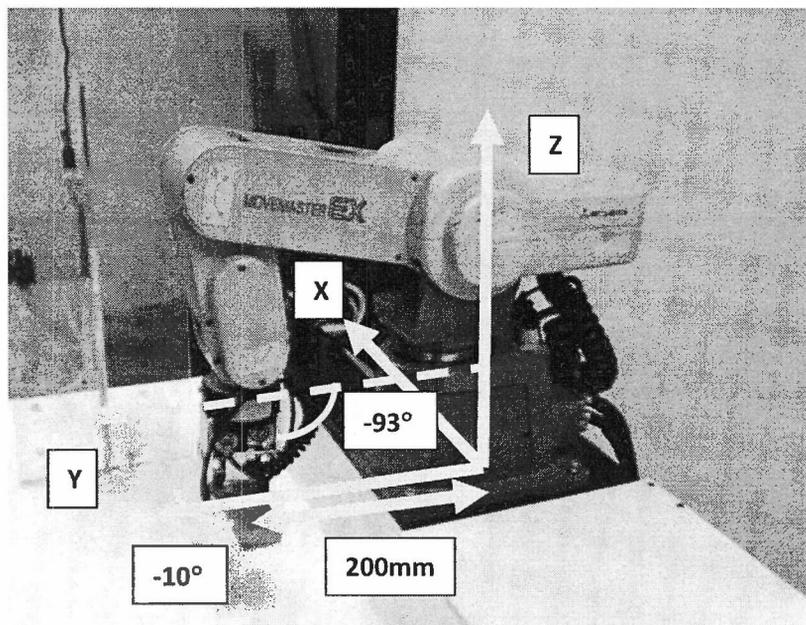


Figura 4.6 Posición física que se obtiene con la instrucción de la figura 4.5

Con este pequeño programa se pudo verificar que la conexión es correcta y que las instrucciones mandadas desde LabVIEW son ejecutadas por el robot. Para poder facilitar el manejo del robot y los movimientos se creó un programa el cual nos permitiera manipular el robot mediante algunas funciones. Con el programa creado se puede realizar las siguientes funciones:

- **Definir una posición del robot y guardarla en un punto determinado:** Para lograr esto el usuario sólo tiene que seleccionar en qué punto va a grabar la posición del robot y determinar las coordenadas en milímetros de la herramienta (x,y,z) así como su *pitch* y *roll*. El programa muestra en una gráfica de 3D el punto en el espacio que estamos definiendo. Esto sirve como referencia para visualizar la posición y evitar alguna colisión o enviar una posición fuera de alcance.
- **Mover el robot a una posición determinada:** Sólo seleccionamos el número de posición y el robot ejecuta el movimiento.
- **Mover el robot a su posición de origen:** Se manda el robot a su posición origen con respecto a los ejes coordenados de la base
- **Mover el robot a su posición nido:** El robot se mueve al origen de los ejes con respecto a los interruptores de límite y los *encoders* de los motores.
- **Borrar Programa:** Se borra lo que se tenga en el programa guardado en memoria desde la línea 10 hasta la línea 150.
- **Cargar Programa:** Se carga un programa al robot creado manualmente en un campo de texto.

En la figura 4.7 se observa la interface con el usuario, en la figura 4.8 se puede ver el diagrama de flujo del programa y en el Anexo G se muestra el código completo del programa *Escribir2.vi*.

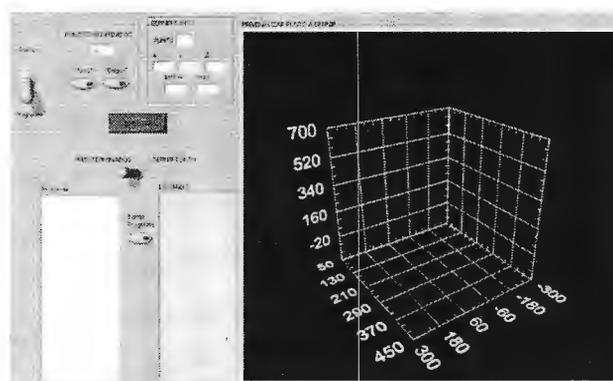


Figura 4.7 Interface del programa para la manipulación del robot

Capítulo 4. Diseño de Instrumentista Quirúrgico

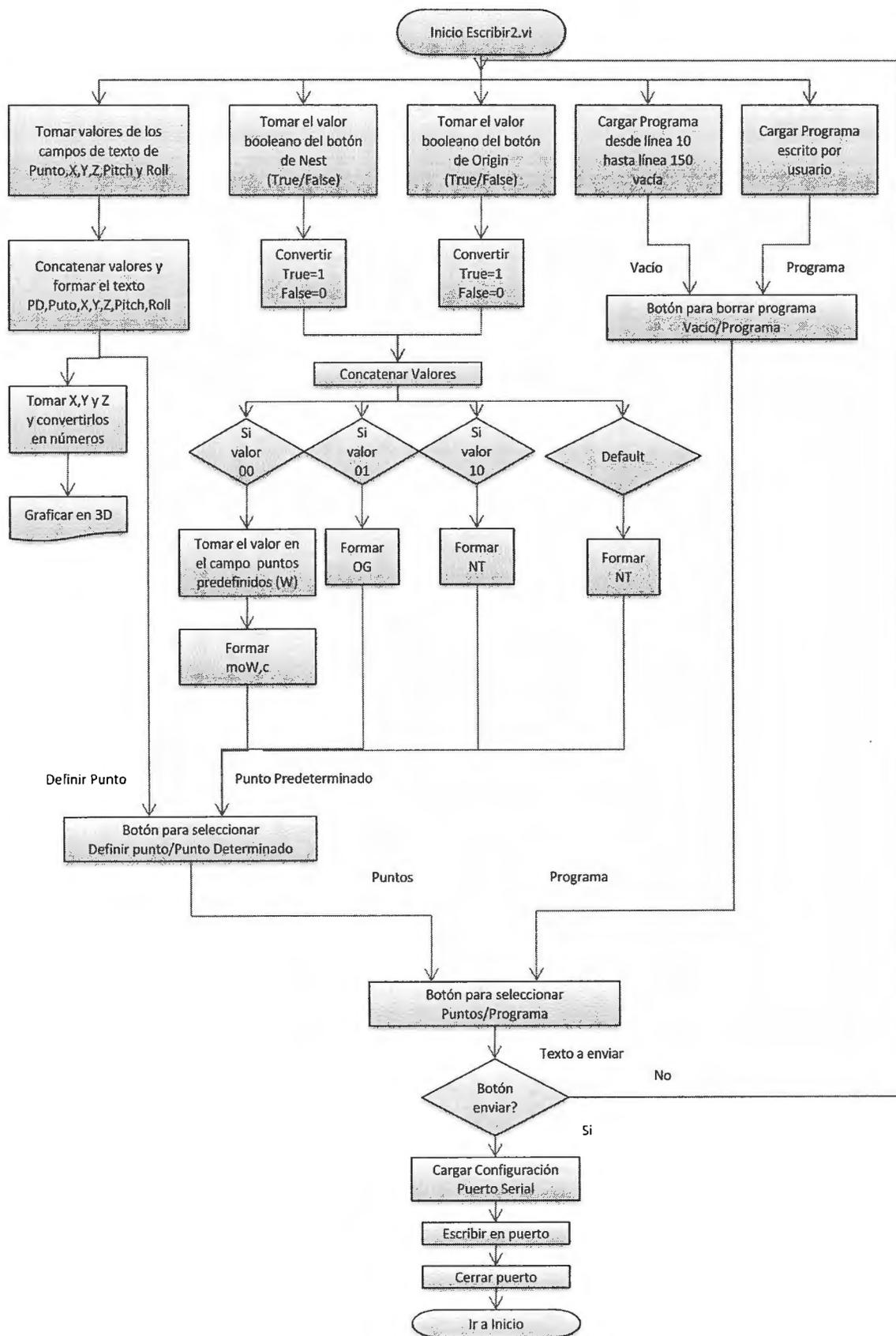


Figura 4.8 Diagrama de flujo del programa para manipular el robot

Hasta aquí se termina la parte de comunicación y el siguiente paso es el reconocimiento de voz.

### 4.3 Reconocimiento de voz

Para el reconocimiento de voz se usó el programa *Windows Speech Recognizer 8.0 (WSR)* que viene incluido en el sistema operativo de la PC. Se decidió utilizar ésta herramienta ya que es accesible y compatible con LabVIEW. WSR es una aplicación .NET, y esta característica permite llamar a sus clases y métodos con herramientas de LabVIEW. Su integración está basada en un programa publicado en el sitio web de National Instruments dentro de un foro de discusión [68]. Esta función recibe una lista de comandos de voz la cual se agrega a una lista de posibles opciones. Después se llama al programa WSR el cual regresa un texto según el comando que haya reconocido. El texto que se tiene como salida, puede ser llamado por medio de una referencia, la cual se utiliza para integrarla al sistema. En la figura 4.9 se puede ver la función **SR.vi**.

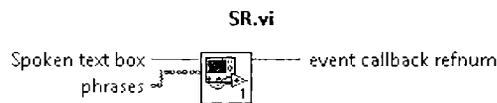


Figura 4.9 Herramienta que se utilizará para el reconocimiento de voz SR.vi

En el Anexo H se observa el programa *SR.vi*, así como las subrutinas que éste utiliza. En cuanto al reconocimiento de voz, esta función cumple con las características necesarias para el objetivo del trabajo; introducimos la lista de comandos que queremos que reconozca y tenemos a la salida el texto del comando de voz.

## 4.4 Reconocimiento de imágenes

Esta etapa del desarrollo del proyecto empieza con la selección de la cámara que se utiliza para la adquisición de imágenes. En el mercado existen muchos tipos de cámaras, pero desde el principio se necesitaba alguna que fuera compatible con LabVIEW para evitar gastar tiempo y recursos en la configuración de la misma. La primera opción que se buscó fue adquirir una de la misma marca. Dentro de la gama de productos se encontró la *Smart Camera NI 1722*. La cual cuenta con accesorios como iluminación y soportes. Este tipo de cámaras están diseñadas para la industria y por lo tanto son cámaras muy robustas y confiables. La desventaja es el costo por lo que se buscó otra opción.

Se decidió utilizar una cámara USB tipo webcam, ya que son compatibles con la librería de visión de LabVIEW. Esta librería puede manejar dispositivos que estén basados en la arquitectura DirectShow por lo cual se seleccionó la cámara *LifeCam NX 3000* de Microsoft con una resolución de 2 mega píxeles.

Una vez escogida la cámara, el siguiente paso es configurarla con LabVIEW para poder controlarla y poder adquirir imágenes. La librería de visión contiene herramientas llamadas IMAQ USB. Estas herramientas sirven para poder controlar cámaras USB tipo webcam que sean compatibles con la tecnología *DirectShow* de Windows. En la figura 4.10 se muestra un programa para tomar imágenes donde primero se crea una sesión de IMAQ USB de la cámara que se está utilizando, después se crea un espacio temporal en memoria para almacenar la imagen que se tomará y se le asigna un nombre a ésta variable. El siguiente ícono, realiza la toma de una sola imagen, que el ícono posterior guarda en una dirección especificada por el usuario con un formato establecido. En caso de existir el archivo lo sobrescribe y éste se almacena en el disco duro de la PC. Por último el programa cierra la sesión dejando libre el dispositivo. Este subVI lo nombramos *RobotImage.vi* para después utilizarlo en la integración.

#### Capítulo 4. Implementación de Instrumentista Quirúrgico

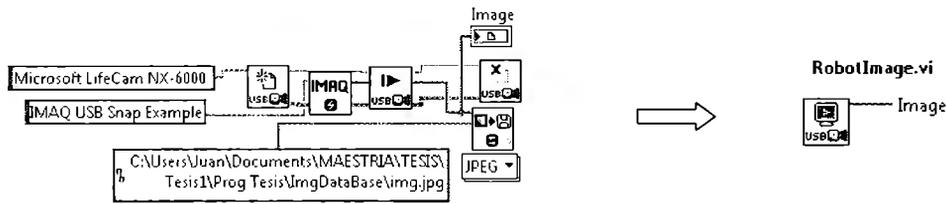


Figura 4.10 SubVI RobotImage.vi

Una vez comprobado el funcionamiento de la cámara, se colocó en una posición fija sobre el área de trabajo. Para esto se construyó un soporte el cual permite sujetar la cámara en una posición fija y a su vez da espacio para la entrada del robot. Otra ventaja del soporte es que deja el lente de la cámara paralelo al plano donde se encontrarán los objetos ya que de lo contrario la imagen tendría efectos de perspectiva. En figura 4.11 se muestra el soporte y la colocación de la cámara.

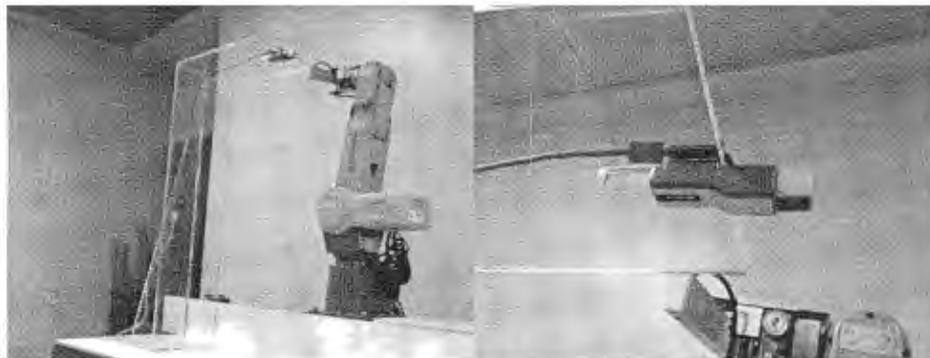


Figura 4.11 Soporte y colocación de la cámara

El soporte fue fabricado de acrílico de 9mm de espesor y una altura de 800mm. Se eligió acrílico ya que es un material estético, rígido y fácil de maquinar. Para corroborar el campo visual de la cámara y la resolución, se tomó una imagen ya colocada en su posición y se muestra en la figura 4.12. La imagen que captura es de 352 x 288 píxeles, y el campo de vista es de aproximadamente 900mm, por lo que el tamaño de la forma más pequeña a reconocer se determina por la ecuación 2.2.

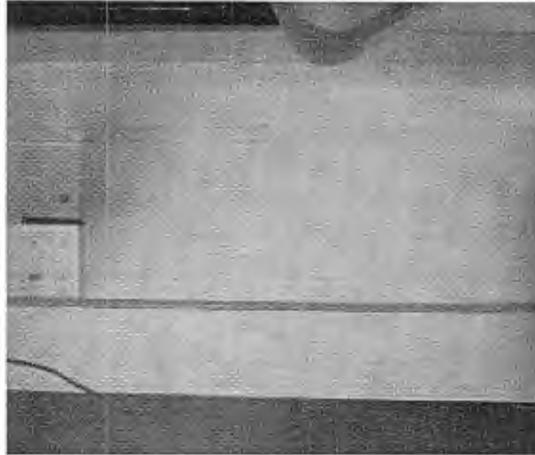


Figura 4.12 Imagen tomada desde la posición en el soporte.

Despejando el tamaño del objeto más pequeño de la ecuación 2.2 se tiene:

$$\text{Tamaño del objeto más pequeño} = \left( \frac{\text{Tamaño de Campo de vista}}{\text{Resolución de sensor}} \right) x(2), \quad (4.1)$$

$$\left( \frac{900mm}{288pix} \right) x(2) = 6.25mm$$

Este parámetro servirá para poder aproximar el tamaño mínimo de los objetos a reconocer, por lo cual se tiene que definir qué instrumentos se van a utilizar para ver si la cámara cuenta con la suficiente resolución para poder identificarlo. Para este proyecto se considerarán 3 instrumentos quirúrgicos los cuales se muestran en la figura 4.13

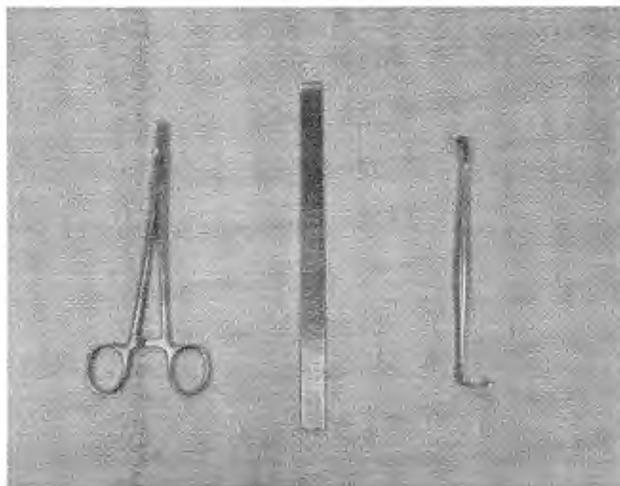


Figura 4.13 Imagen de instrumentos para reconocer (de izquierda a derecha; pinzas, espátula y Senn Miller)

Se puede decir que la cámara cuenta con la resolución necesaria para éste trabajo ya que el objeto más pequeño que se quiere reconocer es el Senn Miller, el cual mide 160mm x 10mm, y es mayor en ambos sentidos al parámetro de referencia calculado de 6.25mm.

La etapa de reconocimiento se implementará con la herramienta *IMAQ Match Pattern 2*. Esta herramienta toma una plantilla previamente definida por el usuario y la busca en una imagen que deseamos inspeccionar. La función cuenta con los siguientes valores de entrada:

- **Imagen:** Imagen de referencia en la que se busca la localización de la plantilla.
- **Imagen plantilla:** Imagen que se desea buscar y es una salida de la función *IMAQ Learn Pattern 2*.
- **Datos de configuración para la búsqueda:** Se obtienen de la función *IMAQ Setup Match Pattern 2*, e indican si buscamos el objeto con una rotación mayor a  $\pm 4^\circ$  o no.
- **Número de objetos que deseamos encontrar en la imagen:** Indica la cantidad máxima de objetos que se quieren encontrar en la imagen.

#### Capítulo 4. Diseño de Instrumentista Quirúrgico

- **Puntuación de reconocimiento:** Indica que tanto se parece la plantilla al objeto que se reconoció en la imagen, y sirve como umbral para establecer que tan estricto se quiere ser en la búsqueda.

Para poder hacer un programa sencillo de reconocimiento de imágenes, se debe tener al menos una imagen patrón y una imagen en donde buscar el patrón. Se utilizó el programa *RobotImage.vi* para tomar una foto de la plantilla y otra foto del área de trabajo y se guardaron en el disco duro. En la figura 4.14 se observa la foto del patrón y la del área de trabajo. Como ejemplo se utilizaron las pinzas.

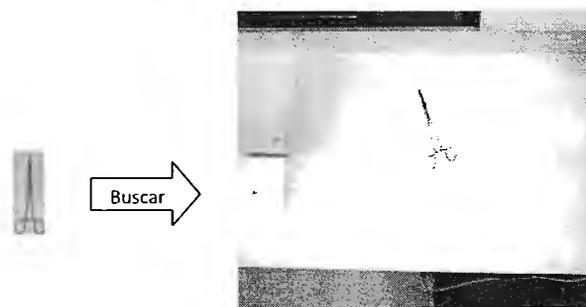
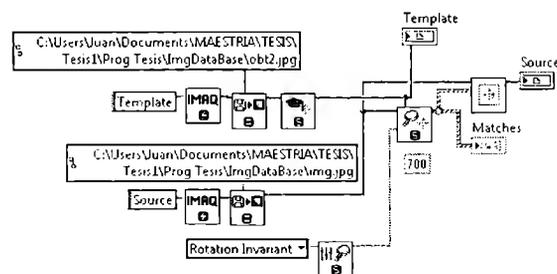


Figura 4.14 Ejemplo de un patrón y de una imagen para buscar.

Una vez obtenidas las imágenes en el disco duro, se realizó un programa sencillo de reconocimiento llamado *Imagenes.vi* el cual se muestra en la figura 4.15.



4.15 Programa Imagenes.vi

El programa crea espacio en memoria tanto para la imagen patrón como para la imagen a analizar. Después carga dos imágenes de una dirección específica de disco duro, siendo estas las que utilizaremos para el reconocimiento. Llamamos a la función *IMAQ Match Pattern 2* donde las entradas serán: las dos imágenes previamente cargadas, el modo en que se busca (*Rotation Invariant*) y el umbral que se considera para la búsqueda.

Este programa servirá como base para la integración final que se presenta más adelante. La salida que se obtiene con *Imágenes.vi* es un grupo de datos de localización de la imagen, pero los datos que interesan son las coordenadas en X y Y en pixeles, el ángulo y la puntuación. En la figura 4.16 se observa el resultado del programa que el usuario visualiza.

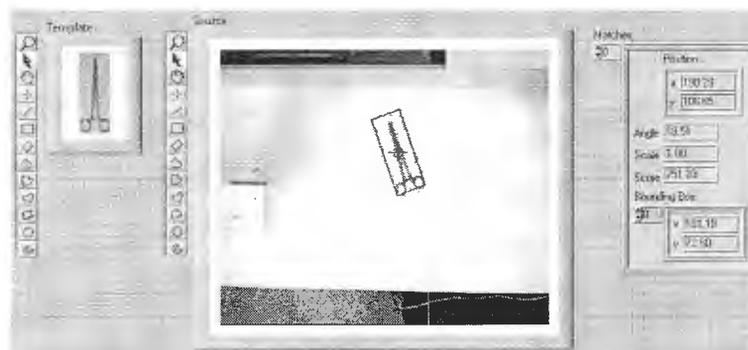


Figura 4.16 Resultados del programa de reconocimiento de imágenes

Las coordenadas que se obtienen están en pixeles y con referencia a la imagen. Estas coordenadas tienen como origen el primer pixel que se encuentra arriba a la izquierda de la imagen. Estas unidades y referencias no son las mismas a las del robot, así lo siguiente es hacer una conversión de unidades, una translación de ejes y un cálculo de ángulo.

Para la conversión de unidades se tiene que saber a cuantos milímetros equivale un pixel, para lograrlo se imprimió un patrón de círculos de 40mm de diámetro y 60mm de

#### Capítulo 4. Diseño de Instrumentista Quirúrgico

separación entre sus centros. El patrón utilizado es de 10 x 8 círculos y se muestra en la figura 4.17.

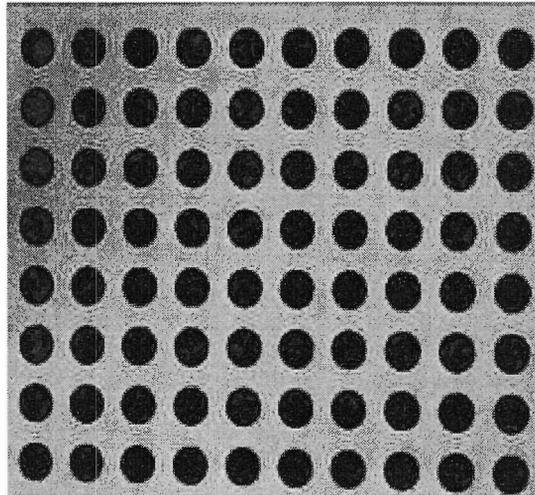


Figura 4.17 Patrón de calibración

Primero se pensó en hacer una calibración utilizando funciones de LabVIEW, pero al ver que la calibración era casi lineal se realizó haciendo una simple interpolación lineal. La manera en que se calculó la equivalencia de un pixel a milímetros, se muestra en la figura 4.18.

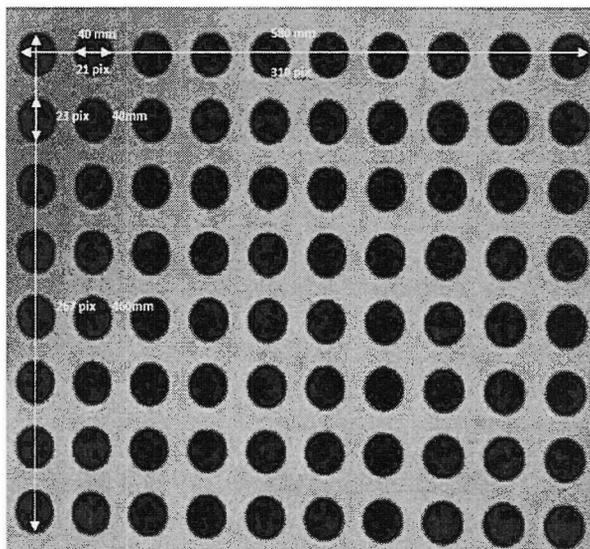


Figura 4.18 Medidas en pixeles y milímetros del patrón de calibración

Para el eje X consideramos el valor de un solo círculo y la distancia de lado a lado del patrón. Para un círculo tenemos:

$$\frac{40mm}{21pix} = 1.9 \frac{mm}{pix}$$

Considerando de lado a lado en sentido horizontal:

$$\frac{580mm}{310pix} = 1.87 \frac{mm}{pix}$$

Por lo que

$$1.90 \frac{mm}{pix} \approx 1.87 \frac{mm}{pix}$$

La relación es casi lineal por lo que tomaremos un valor intermedio de 1.88 milímetro por pixel para los cálculos. Esta es nuestra resolución para nuestros resultados. Haciendo el mismo procedimiento para el eje Y:

$$\frac{40mm}{25pix} = 1.60 \frac{mm}{pix}$$

Considerando de lado a lado en sentido vertical:

$$\frac{460mm}{285pix} = 1.61 \frac{mm}{pix}$$

Por lo que

$$1.60 \frac{mm}{pix} \approx 1.61 \frac{mm}{pix}$$

En el caso del eje Y se toma un factor de conversión de 1.60 milímetros por pixel. Una vez que se obtuvo la conversión se tiene que trasladar la referencia, que actualmente está en la imagen, a la referencia localizada en las coordenadas base del robot. En la figura 4.19 se muestra cómo están definidas las coordenadas para el robot y para la imagen.

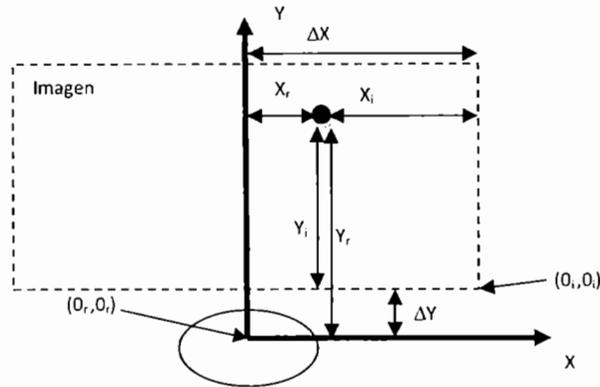


Figura 4.19 Sistema de coordenadas de la imagen y del robot

Con base en la figura anterior tenemos las ecuaciones 4.2 y 4.3

$$Y_r = Y_i + \Delta Y \quad (4.2)$$

$$X_r = \Delta X - X_i \quad (4.3)$$

Donde los valores  $(X_i, Y_i)$  son las coordenadas en pixeles que arroja la función *IMAQ Match Pattern 2*. Para calcular  $\Delta X$  se lleva al robot a una posición  $(0, Y_r)$  y se pone una marca. Después se toma una imagen y se mide en que pixel se encuentra el centro de la marca y resulta que  $\Delta X = 185$  pixeles. Para calcular  $\Delta Y$  se lleva al robot a una posición  $(0, 310\text{mm})$  y se pone una marca, después se cuentan 159 pixeles del inicio de la imagen a la marca. Los pixeles se transforman a milímetros por la ecuación 4.4 y tenemos

$$(\text{pixeles})(\text{Factor de conversión en } Y) = \text{Medida en mm} \quad (4.4)$$

$$(159\text{pixeles})(1.60) = 254.4\text{mm}$$

Si se resta 310mm del robot se tiene que  $\Delta Y = 310\text{mm} - 254.4\text{mm} = 55.6\text{mm}$ , y convirtiendo a pixeles para estandarizar:

$$\Delta Y = \frac{55.6\text{mm}}{1.60} = 34.75\text{pixeles} \quad (4.5)$$

**Nota:** No existen fracciones de pixeles, pero para nuestros cálculos se consideran estos decimales y al final se redondean las cantidades.

Como resultado tenemos la posición del objeto en las coordenadas del robot y con medidas reales utilizando traslación de ejes y el producto con el factor de conversión:

$$Y_r = (Y_i + 34.75)(1.6) \quad (4.6)$$

$$X_r = (185 - X_i)(1.88) \quad (4.7)$$

Para finalizar con la implementación del procesamiento de imágenes, se tiene que calcular la orientación de la pieza con respecto a la herramienta del robot. La función de LabVIEW *IMAQ Match Pattern 2* arroja una orientación con respecto al eje Y de la imagen con coordenadas que van desde 0° a 359° (figura 4.20). En la tabla 4.4 podemos observar las orientaciones encontradas del mismo objeto en una imagen rotada en pasos de 90°.

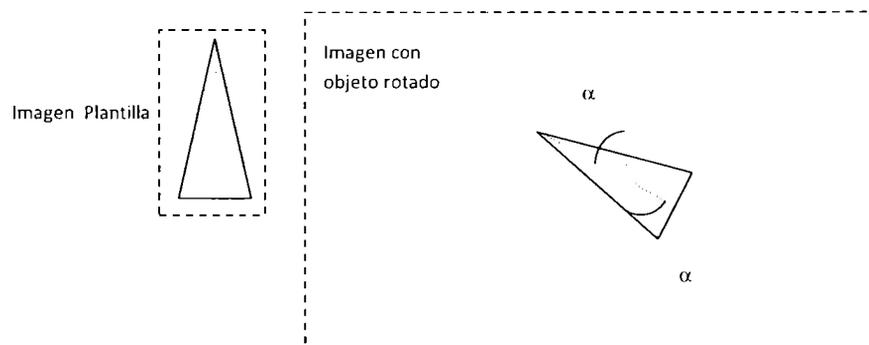


Figura 4.20 Ángulo calculado por la función IMAQ Match Pattern 2

Capítulo 4. Diseño de Instrumentista Quirúrgico

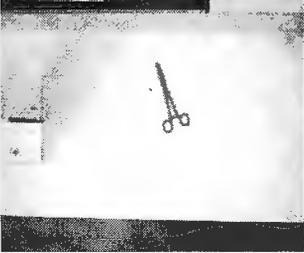
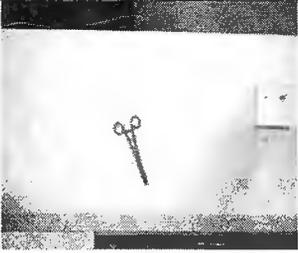
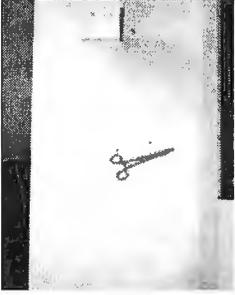
Objeto a encontrar	Imagen a analizar	Ángulo encontrado
		0°
		19.51°
		108.99°
		108.99°
		198.24°

Tabla 4.4 Cálculo de ángulo con imágenes rotadas 90° en sentido contrario a las manecillas del reloj.

Una vez identificado cual es el ángulo  $\alpha$  que arroja la imagen, se establece cuál va a ser la orientación de la sujeción del robot para tomarla como referencia y considerar el ángulo cero en la herramienta del robot. Para el caso de los tres objetos utilizados,

consideraremos el punto de sujeción colocando el lado más largo de la herramienta del robot, perpendicular al lado más largo de los instrumentos. En la figura 4.21 se muestra en cuadros punteados el área donde se colocará la herramienta del robot para tomar cada una de las piezas.

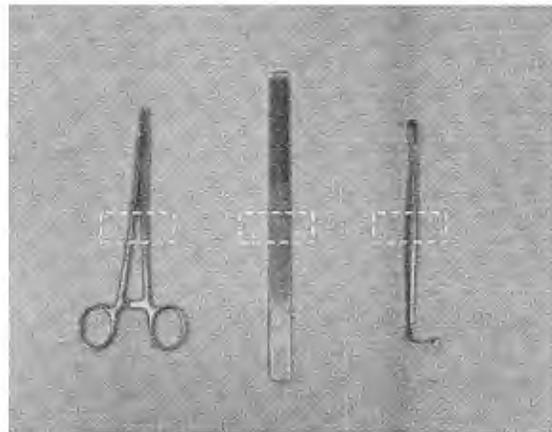


Figura 4.21 Áreas donde se ubicará la herramienta del robot para la sujeción de cada instrumento.

Es importante hacer énfasis en que el ángulo del *roll* que se introduce en la instrucción **PDw,X,Y,Z,pitch,roll** toma como referencia el brazo del robot donde está montada la herramienta y no la base del robot. Para poder visualizar el punto de toma y el ángulo *roll*, se toma un instrumento y se coloca en las coordenadas (0,300) con respecto al robot con un ángulo de 0°, y después se lleva al robot a un punto encima del objeto. Aquí se puede observar que el ángulo de la herramienta del robot con respecto al brazo no es cero exactamente y existe un error de aproximadamente -10°. En la figura 4.22 se puede observar el desfase de -10° con respecto al brazo. Este error es a causa de la calibración del robot debido a la mala colocación del sensor de proximidad del eje 5. Para poder corregir el error se tiene que ajustar el sensor de tal manera que registre su origen de manera correcta, pero para efectos prácticos, se compensa este desfase mediante software considerando en nuestros cálculos -10° en el giro de la herramienta.

#### Capítulo 4. Diseño de Instrumentista Quirúrgico

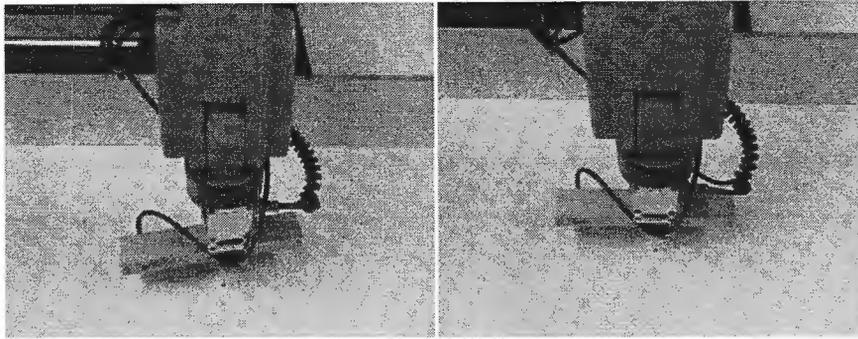


Figura 4.22 Robot con error mecánico (izquierda) y robot con error mecánico compensando (derecha)

Para resolver el problema de orientación, se considera un punto en el plano  $(X_r, Y_r)$  con referencia a las coordenadas del robot. Al momento de mover el robot a este punto, se tiene que compensar el error mecánico de  $-10^\circ$ , el giro del brazo y por último el giro de la pieza con respecto a su plantilla. El ángulo del giro del brazo  $\beta$ , se calcula basándose en la figura 4.23, los  $-10^\circ$  es una constante y el giro de la pieza nos los da la función *IMAQ Match Pattern 2*. Lo que resta es calcular  $\beta$ , y se tiene que :

$$\beta = \arctan\left(\frac{X_r}{Y_r}\right) \quad (4.8)$$

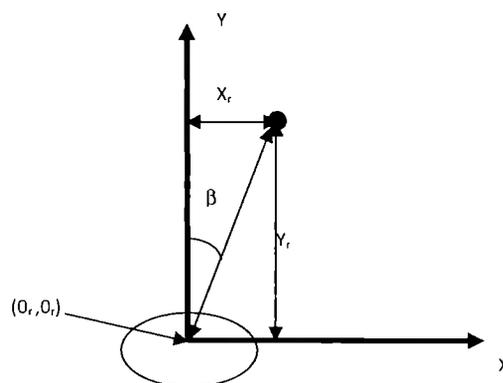


Figura 4.23 Ángulo de giro del robot.

Obteniendo éstos valores, se tiene que el ángulo total  $\omega$  mandado al robot para hacer el movimiento de *roll* es:

$$\omega = e_r + \beta + \alpha \quad (4.9)$$

El *roll* del robot se da en ángulos de  $-180^\circ$  hasta  $180^\circ$  por lo que después de calcular  $\omega$ , si nuestro valor es mayor de  $180^\circ$ , se resta  $360^\circ$  para obtener el rango que se busca. Por lo anterior, se tiene que:

$$\omega = \begin{cases} \omega - 360^\circ & \text{si } \omega \geq 180^\circ \\ \omega & \text{para lo demás} \end{cases} \quad (4.10)$$

En la figura 4.24 se puede ver, en secuencia de imágenes, como son los movimientos del robot para calcular la orientación de la herramienta. Con éste cálculo se asegura que el robot encontrará el objeto en cualquier punto del área de trabajo ajustando la orientación de la herramienta para que lo tome siempre de una misma manera.

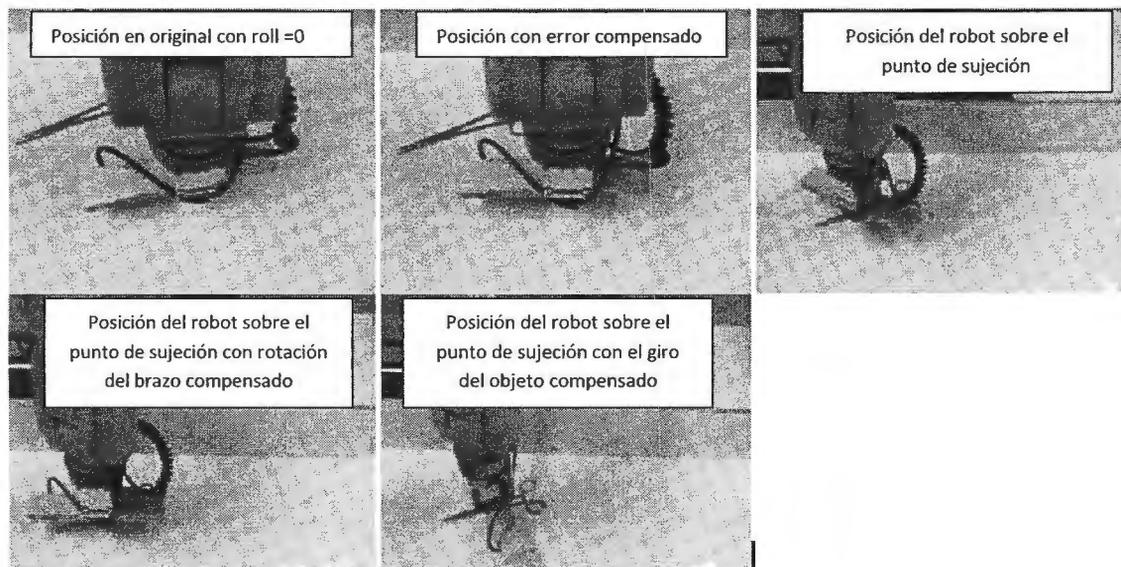


Figura 4.24 Secuencia de imágenes para visualizar la orientación de la herramienta del robot.

En cuanto al *pitch* o cabeceo del robot, para todos los casos será de  $-90^\circ$  con respecto a los ejes X y Y, ya que siempre va a llegar con la herramienta paralela a éste plano. Cuando se hicieron pruebas, se observó que también existe un error mecánico en los movimientos

#### Capítulo 4. Diseño de Instrumentista Quirúrgico

del pitch de  $-5^\circ$ , pero al igual que el *roll*, se puede compensar por medio de software. Por lo que:

$$pitch = -90^\circ + e_r = -90^\circ + (-5^\circ) = -95^\circ \quad (4.11)$$

El *pitch* es una constante de  $-95^\circ$  para los puntos de sujeción. En la figura 4.25 se puede visualizar el error del *pitch* y su corrección por medio de software.

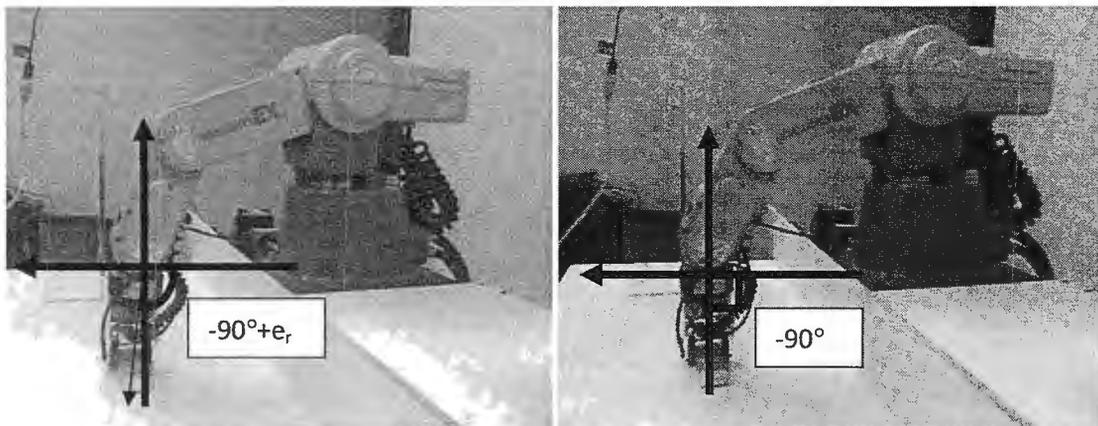


Figura 4.25 Pitch y su corrección por medio de software.

Debido a que este sistema solo trabaja en dos dimensiones, se tiene que determinar las alturas en las que debe trabajar el robot para manejarlas como constantes. La superficie donde se ubican los objetos se encuentra a  $-125\text{mm}$  en el eje Z. La figura 4.26 indica cómo determinar la altura a la que opera el robot.

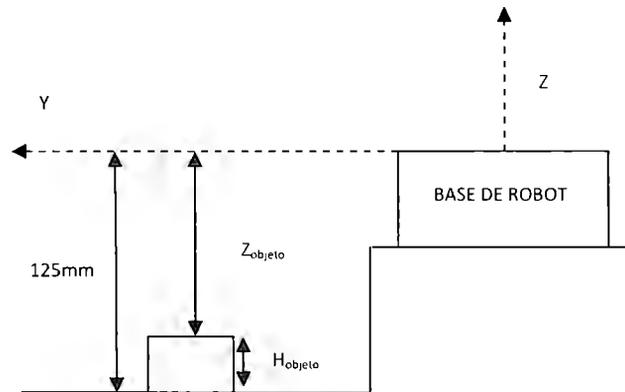


Figura 4.26 Cálculo de altura para cada objeto

Tomando en cuenta la figura 4.26 se calculan las alturas para los tres objetos que se utilizan en el proyecto restando los espesores de cada herramienta a la altura total desde el eje de referencia a la superficie donde se encuentran los objetos.

$$Z_{pinzas} = 125mm - 5mm = 120mm$$

$$Z_{espátula} = 125mm - 3mm = 122mm$$

$$Z_{Senn Miller} = 125mm - 7mm = 118mm$$

Después de hacer la conversión de unidades y traslación de ejes, se adjuntaron estas modificaciones al programa *Imagenes.vi* mostrado en la figura 4.27. Este programa recibe las direcciones en disco duro de donde se encuentran ubicadas la imagen plantilla y la imagen donde se quiere ubicar la plantilla. El programa toma la plantilla y la empieza a buscar en la imagen a analizar. Una vez encontrado el objeto, el programa regresa la ubicación en dos dimensiones y su orientación, todo con respecto a la imagen. Después, con los valores obtenidos anteriormente, el programa hace la conversión de unidades, la traslación de ejes y calcula la orientación con respecto al robot. Al final se tienen valores que pasan a la siguiente etapa que es la programación del robot. El diagrama de flujo se muestra en la figura 4.28.

## Capítulo 4. Diseño de Instrumentista Quirúrgico

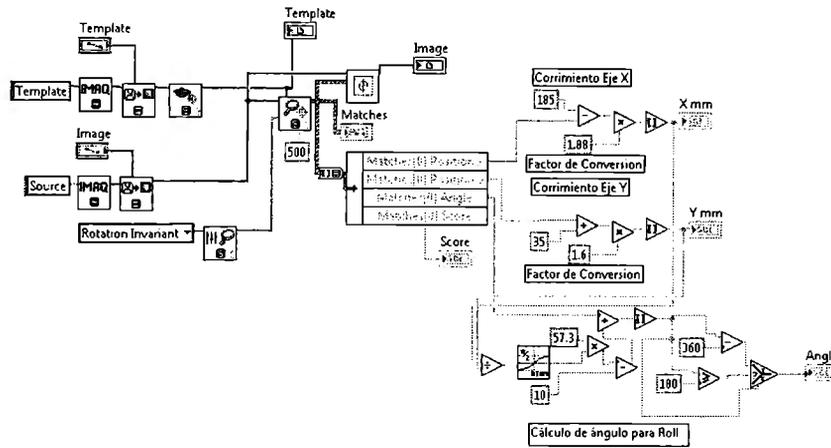


Figura 4.27 Programa Imagenes.vi

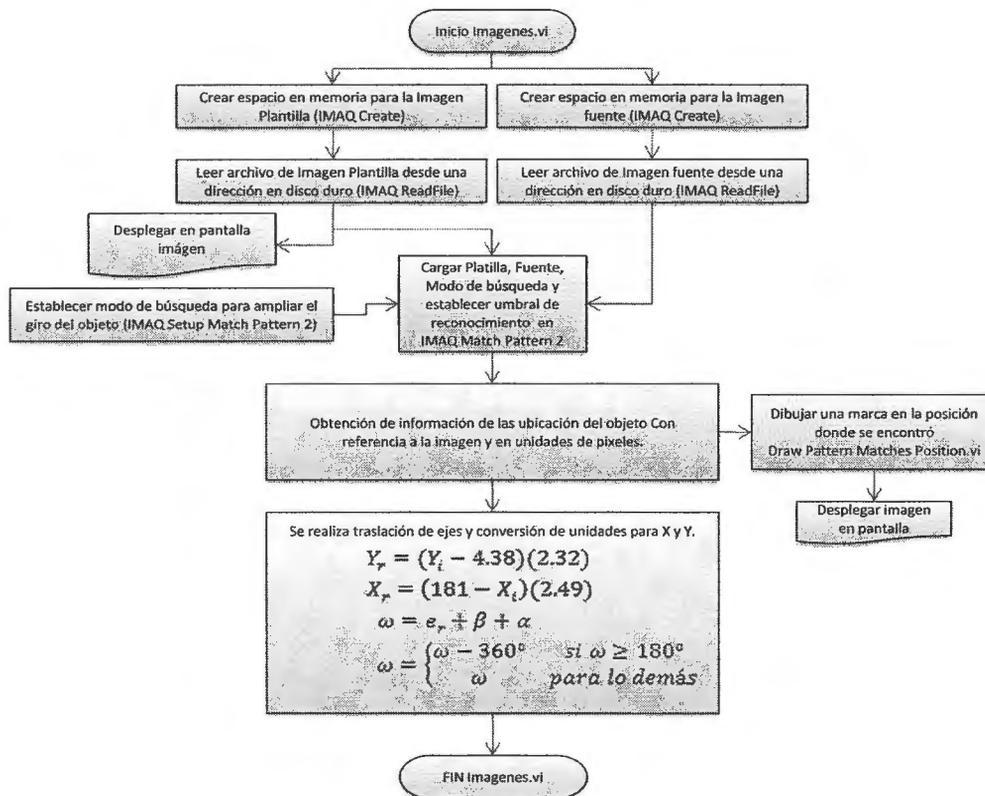


Figura 4.28 Diagrama de flujo de Imagenes.vi

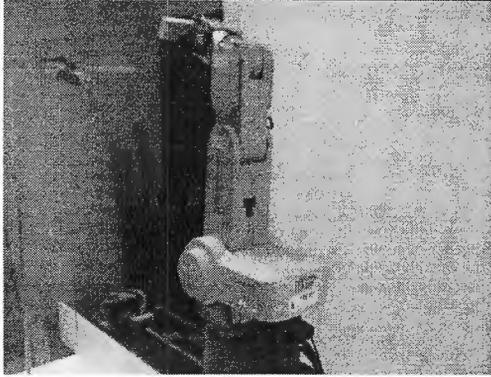
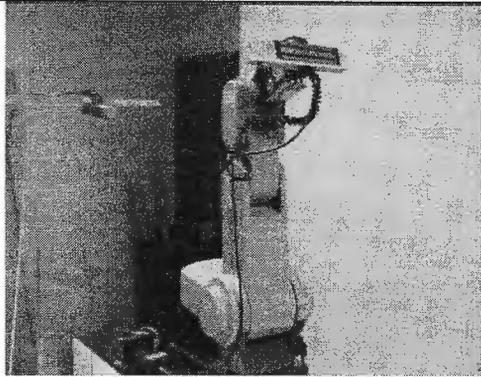
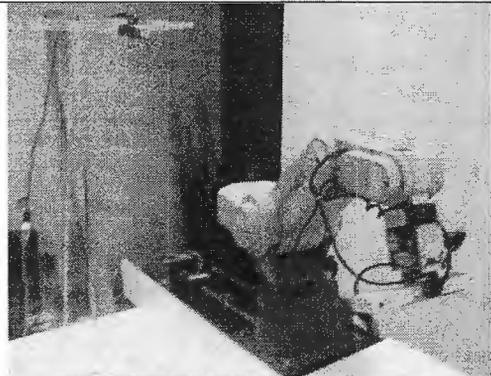
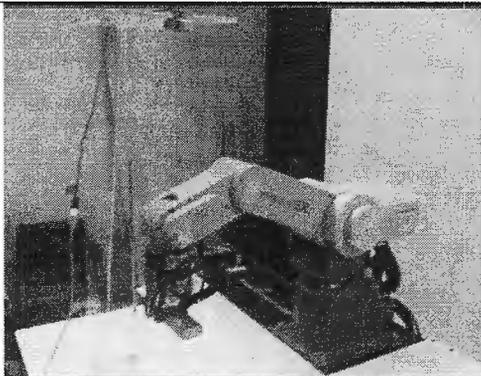
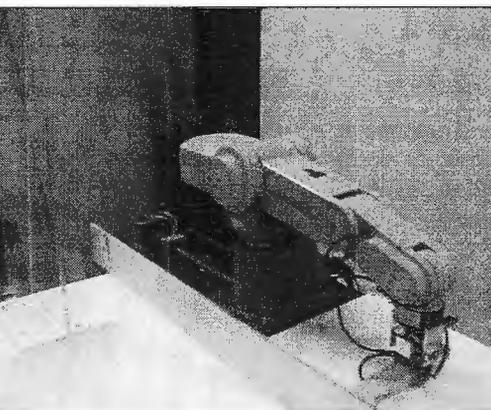
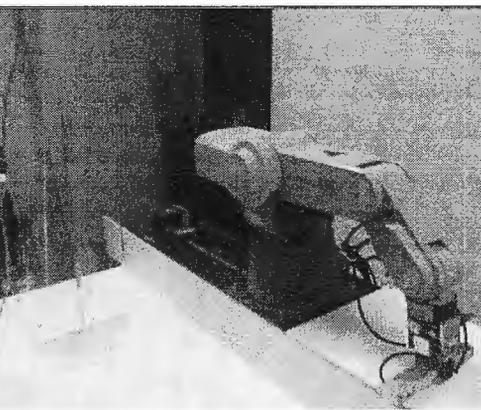
## 4.5 Programación de robot

Una vez calculado los parámetros de ubicación del objeto, la siguiente etapa es programar el robot, para lo cual se utilizan dos métodos para definir los puntos: de manera manual con el *teaching box*, y por la instrucción "*Position Define*". Las posiciones usadas se describen en la tabla 4.5.

Posición	Método de definición	Descripción
5	Teaching box	Posición segura del robot
10	Teaching box	Posición girada para librar el soporte de la cámara
20	Teaching box	Primera posición de aproximación al área de trabajo
30	Teaching box	Posición encima del área de trabajo
40	Instrucción PD	Posición arriba del objeto
50	Instrucción PD	Posición de toma del objeto
60	Teaching box	Posición arriba del punto donde dejará el objeto
70	Teaching box	Posición donde dejará el objeto

Tabla 4.5 Puntos utilizados en la rutina del robot.

Las posiciones definidas por el *teaching box* sólo se enseñan una vez y se guardan en memoria. Estas posiciones se realizaron de manera visual ya que son puntos en el aire que no requieren mucha precisión. Para generar las instrucciones, se necesita prender el *teaching box*, mover el robot al punto deseado y después oprimir las teclas de "*Position Set*" (P.S) en el control de mando y asignar el número de posición deseado. En la tabla 4.6 se muestran las posiciones que se enseñaron manualmente. Este proceso se tiene que realizar antes de correr el programa, ya que de lo contrario, al momento de enviar la posición, el robot mandará un error debido a que no encontrará ningún dato en ese punto.

<p><b><u>Posición 5</u></b></p>	<p><b><u>Posición 10</u></b></p>
	
<p><b><u>Posición 20</u></b></p>	<p><b><u>Posición 30</u></b></p>
	
<p><b><u>Posición 60</u></b></p>	<p><b><u>Posición 70</u></b></p>
	

*Tabla 4.6 Posiciones definidas manualmente.*

Para las posiciones creadas mediante software con la instrucción *Position Define*, se programó la rutina **RobotProgram.vi** la cual utiliza los valores calculados por **Imágenes.vi** para definir puntos y después envía una rutina completa al robot. La sintaxis de esta instrucción es “*PDpunto,X,Y,Z,pitch,roll*”, donde “*PD*” indica el nombre de la instrucción, “*punto*” es el número de punto al que se le asigna la posición, “*X,Y,Z*” son las coordenadas del punto, “*pitch*” es el cabeceo o giro del eje 4 y “*roll*” es el giro de la herramienta o del eje 5. Entre instrucción e instrucción se manda a llamar a una subrutina llamada **tiempo.vi** (figura 4.30) la cual establece un retraso entre envíos de información de 100 ms ya que la velocidad del procesador de la PC es mayor a las del controlador del robot y en caso de omitir ése retraso, se tienen problemas en el envío de las instrucciones. Después de enviar las instrucciones para los puntos a definir, el mismo programa **RobotProgram.vi**, manda una secuencia de movimientos utilizando todos los puntos. El programa se muestra en la figura 4.29 y su diagrama de flujo se muestra en la figura 4.31

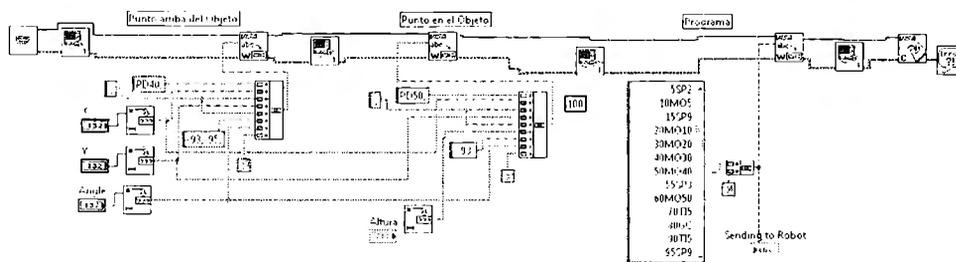


Figura 4.29 Programa RobotProgram.vi.

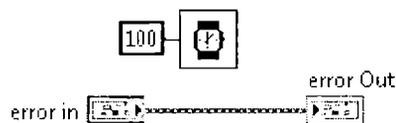


Figura 4.30 Programa tiempo.vi.

#### Capítulo 4. Diseño de Instrumentista Quirúrgico

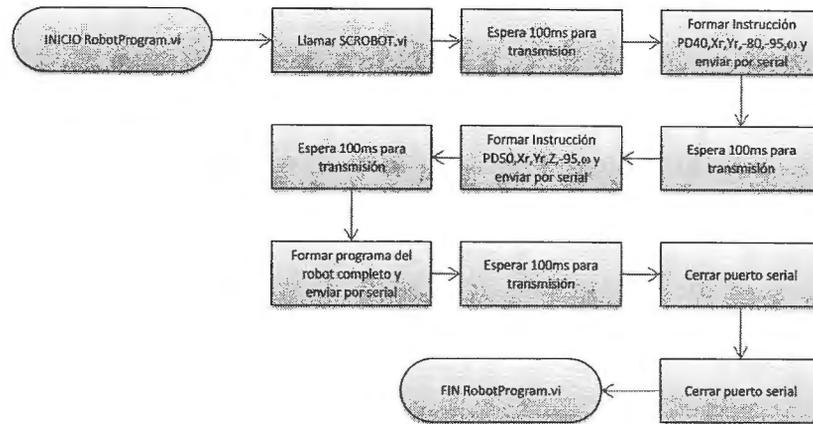


Figura 4.31 Diagrama de flujo de RobotProgram.vi

El programa que se envía al robot, se muestra en el Anexo I. Este programa empieza estableciendo una velocidad lenta (máxima de 200 mm/s en el efector) y después manda al robot a una posición segura. Si el robot no se encuentra en la posición segura, se empieza a mover y da tiempo para pararlo si en su camino tiene algún obstáculo con el que pueda colisionar. Si el robot ya se encuentra en posición segura, avanza a la siguiente instrucción que es establecer una velocidad rápida (máxima de 1000 mm/s en el efector). Aquí el robot empieza a moverse en el espacio a los puntos de aproximación definidos manualmente (puntos 10, 20 y 30) hasta llegar a los puntos finos definidos por la instrucción *Position Define* (puntos 40 y 50). Aquí disminuye la velocidad y el robot se mueve al punto de toma. Una vez ubicado el robot en este punto, toma el objeto y se vuelve a establecer una velocidad rápida para movimientos de salida. Por último lleva el objeto a una posición fija donde lo deposita y regresa a la posición segura. La lógica de movimientos del robot se muestra en la figura 4.32.

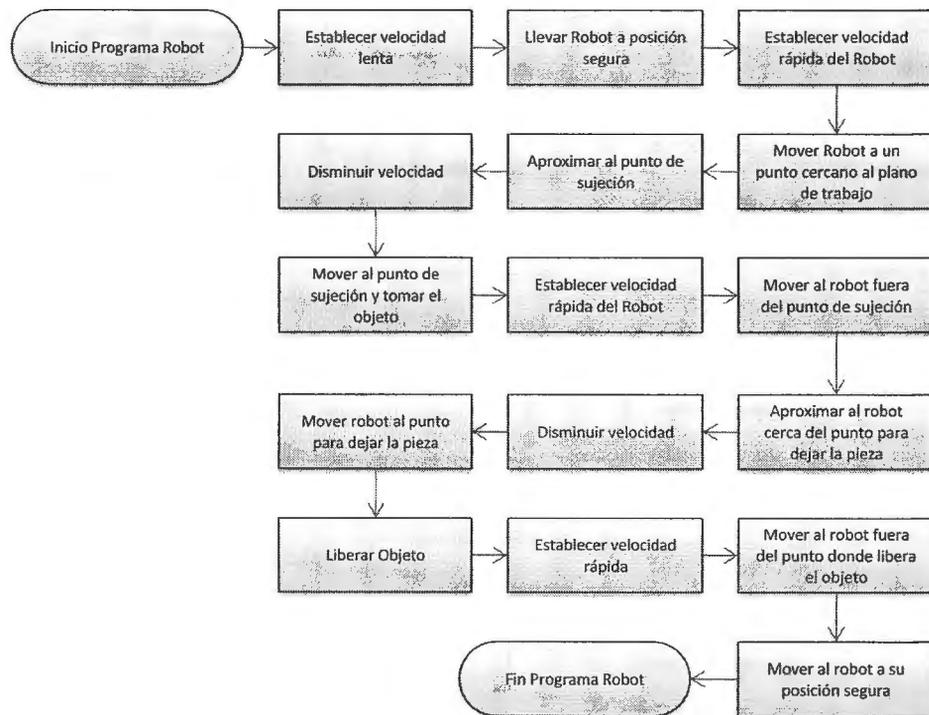
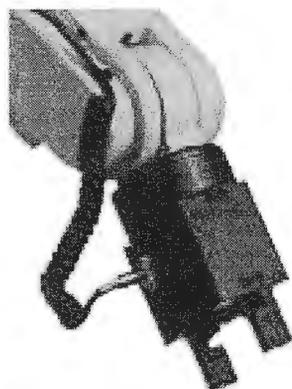


Figura 4.32 Lógica de movimientos del robot.

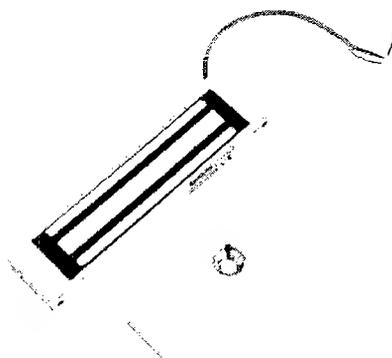
## 4.6 Sujeción

El robot RV-M1 cuenta por default con un sistema de sujeción que consta de un efector en forma de “tenaza” el cual abre y cierra. Esta herramienta es útil cuando se tienen objetos con formas regulares, pero en el caso de éste proyecto es diferente. En la figura 4.33 se observa la forma de la herramienta, la cual cuenta con las conexiones y el control pre-programado en el controlador. Como se mencionó en el capítulo de diseño de éste trabajo, se buscó una característica que compartieran los objetos para pensar en el tipo de herramienta que se tiene que montar al robot para la sujeción de las piezas. La característica que se identifica es el tipo de materia el cual es uniforme para los instrumentos quirúrgicos que se utilizan. Éste es un tipo de material metálico que cuenta con propiedades magnéticas las cuales se pueden aprovechar con un electroimán.



*Figura 4.33 Herramienta mecánica incluida con el robot RV-M1.*

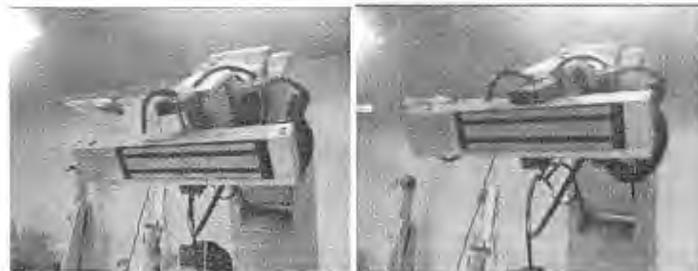
La solución que se dio al electroimán fue utilizar uno diseñado para control de accesos que cuenta con una capacidad de carga de 130kg. Estos electroimanes se utilizan comúnmente en puertas que se abren y cierran electrónicamente. La manera en que funcionan es muy sencilla; se coloca el electroimán en el marco de una puerta con una conexión a un sistema de control que solo activa o desactiva la corriente que pasa por el dispositivo. Después a la puerta se le fija una contraparte metálica la cual al momento de cerrar, el electroimán la atraerá impidiendo que se abra. En la figura 4.34 se muestran los dos componentes.



*Figura 4.34 Componentes de un sistema de acceso basado en un electroimán.*

En este trabajo se utiliza sólo la parte del electroimán, la cual se montó en la herramienta del robot de tal manera que el centro del electroimán se ubica en el centro de la herramienta tipo tenaza.

El siguiente problema es la activación del dispositivo el cual se resolvió aprovechando el movimiento mecánico de la herramienta tipo tenaza para oprimir un interruptor de límite. Cuando la tenaza cierra, oprime el interruptor dejando pasar la corriente a la bobina del electroimán y generando campo magnético. En la figura 4.35 se muestra la instalación del electroimán y el interruptor de proximidad, así como su posición en abierto (sin tomar objeto) y cerrado (para tomar objeto). De esta manera se respetan las instrucciones del robot "GO" y "GC" para dejar y tomar piezas respectivamente.



*Figura 4.35 Instalación de electroimán y su posición para dejar (izquierda) y tomar (derecha).*

Otra de las ventajas que se tiene utilizando el electroimán es que dará más tolerancia para la toma de piezas que si se implementara una sujeción a base de presión. Una vez montado el electroimán se cableó y el circuito quedó como se muestra en la figura 4.36. Al circuito se le agregó un diodo para evitar corriente en sentido contrario.

Capítulo 4. Diseño de Instrumentista Quirúrgico

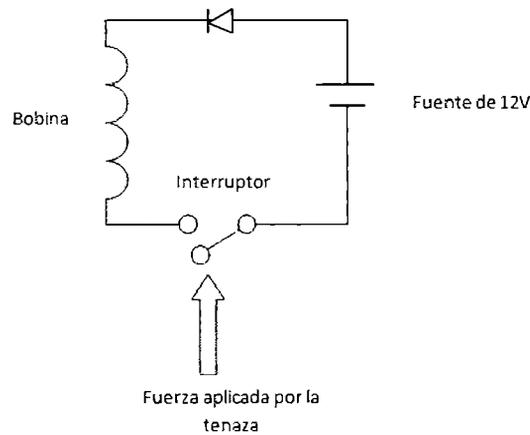


Figura 4.36 Circuito de sistema de sujeción por electroimán.

Una de las desventajas de utilizar el electroimán como herramienta es que su tamaño no es tan práctico. Para probar que las herramientas tuvieran las propiedades suficientes para ser sujetadas magnéticamente se realizaron pruebas y se encontró que las piezas son atraídas en un volumen como se muestra en la figura 4.37 utilizando un voltaje de 12V y una corriente de 200mA.

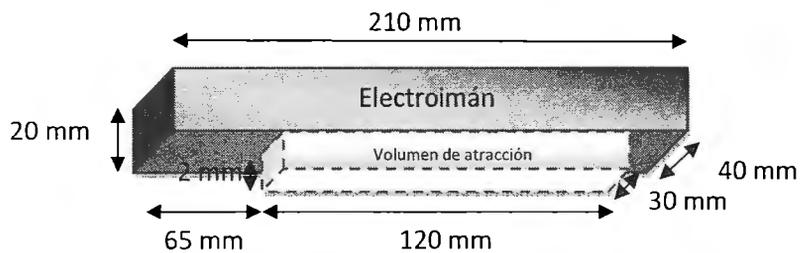


Figura 4.37 Volumen de atracción del electroimán.

Las dimensiones de la herramienta y su forma, hacen reducir el área de trabajo ya que de lo contrario, se pueden tener colisiones con los objetos que se encuentran alrededor de ella. Por éste motivo, el programa establece unos límites de trabajo los cuales son basados en la herramienta y los objetos que están alrededor. En la figura 4.38 se observa cómo es que se establecieron estos límites donde:

- A: Área de trabajo.
- R: Radio de operación del robot en una altura de -120 mm con respecto al eje Z.
- H: Área que ocupa la herramienta.
- L1: Límite establecido por el soporte de la cámara.
- L2: Límite establecido por la mesa de trabajo.
- X: Eje X con respecto al robot.
- Y: Eje Y con respecto al robot.

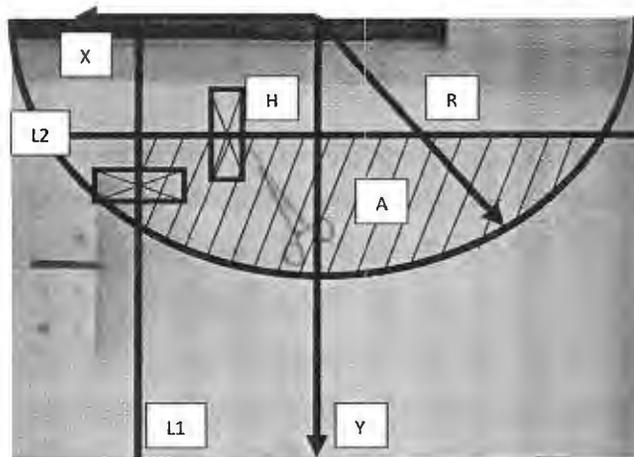


Figura 4.38 Consideraciones para límites de operación.

Considerando la figura anterior, se observa que se tienen 3 límites de operación: L1, L2 y R. En el caso de L1 se midió a qué distancia colisionaba la herramienta con la base de la

Capítulo 4. Diseño de Instrumentista Quirúrgico

cámara, para L2 se midió a qué distancia colisionaba la herramienta con la mesa de trabajo y R se calculó tomando en cuenta la distancia del punto encontrado considerando los valores  $X_r$  y  $Y_r$ , por lo que los límites quedaron como se muestra a continuación:

$$X_r < 190mm \quad (4.12)$$

$$Y_r > 200mm \quad (4.13)$$

$$R = \sqrt{X_r^2 + Y_r^2} \quad (4.14)$$

$$R < 300mm$$

Se tiene una función llamada **limites.vi** la cual contiene estas condiciones y en caso de cumplirse las tres, manda una salida booleana verdadera, de lo contrario manda una salida falsa. El programa se muestra en la figura 4.39.

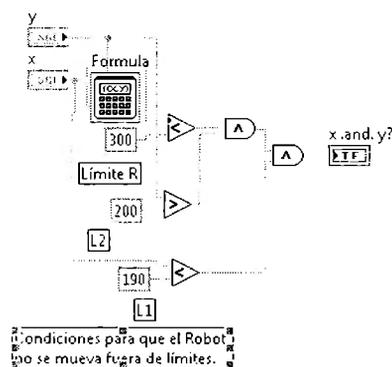


Figura 4.39 Programa *limites.vi*.

## 4.7 Acceso al sistema

Una vez concluidas todas las etapas de la implementación, se desarrollo una pequeña fase de seguridad la cual permitirá utilizar el programa solo a personas que cuenten con un nombre de usuario y contraseña. Esto se hizo para restringir el uso del sistema a personas no indicadas o evitar su mal uso. El programa es muy sencillo y lo que recibe son dos campos de texto y los compara con otros dos para verificar si son iguales. Si son iguales, manda un indicador booleano verdadero y si alguna es incorrecta manda un indicador falso. La función **Login.vi** se muestra en la figura 4.40 y más adelante se utiliza en la integración del programa principal.

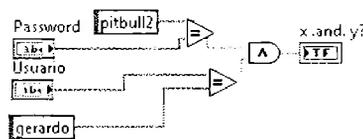


Figura 4.40 Programa Login.vi.

## 4.8 Integración del sistema

Para la integración del sistema se creó una función llamada **main.vi**. La cual contiene la lógica completa del sistema y manda a llamar a todas las funciones antes descritas.

En general se puede dividir el funcionamiento del programa en tres grandes secciones como se muestra en la figura 4.41.

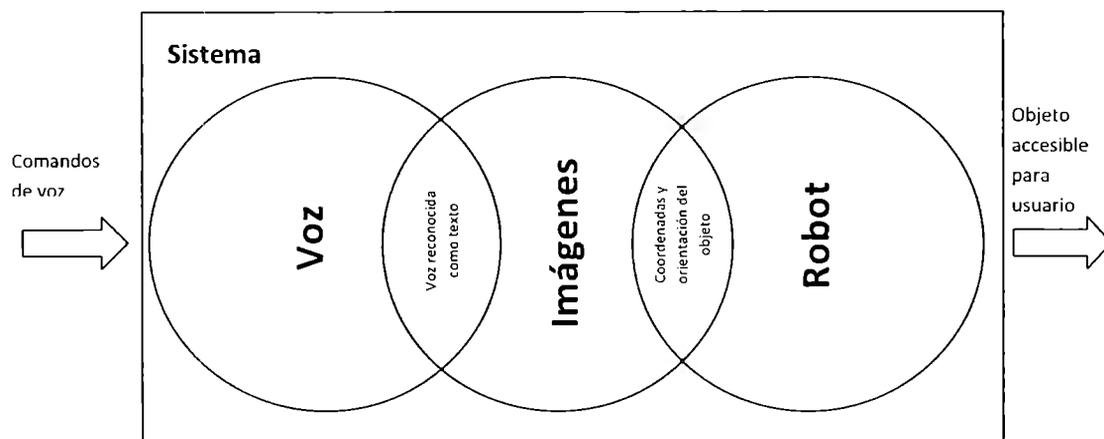
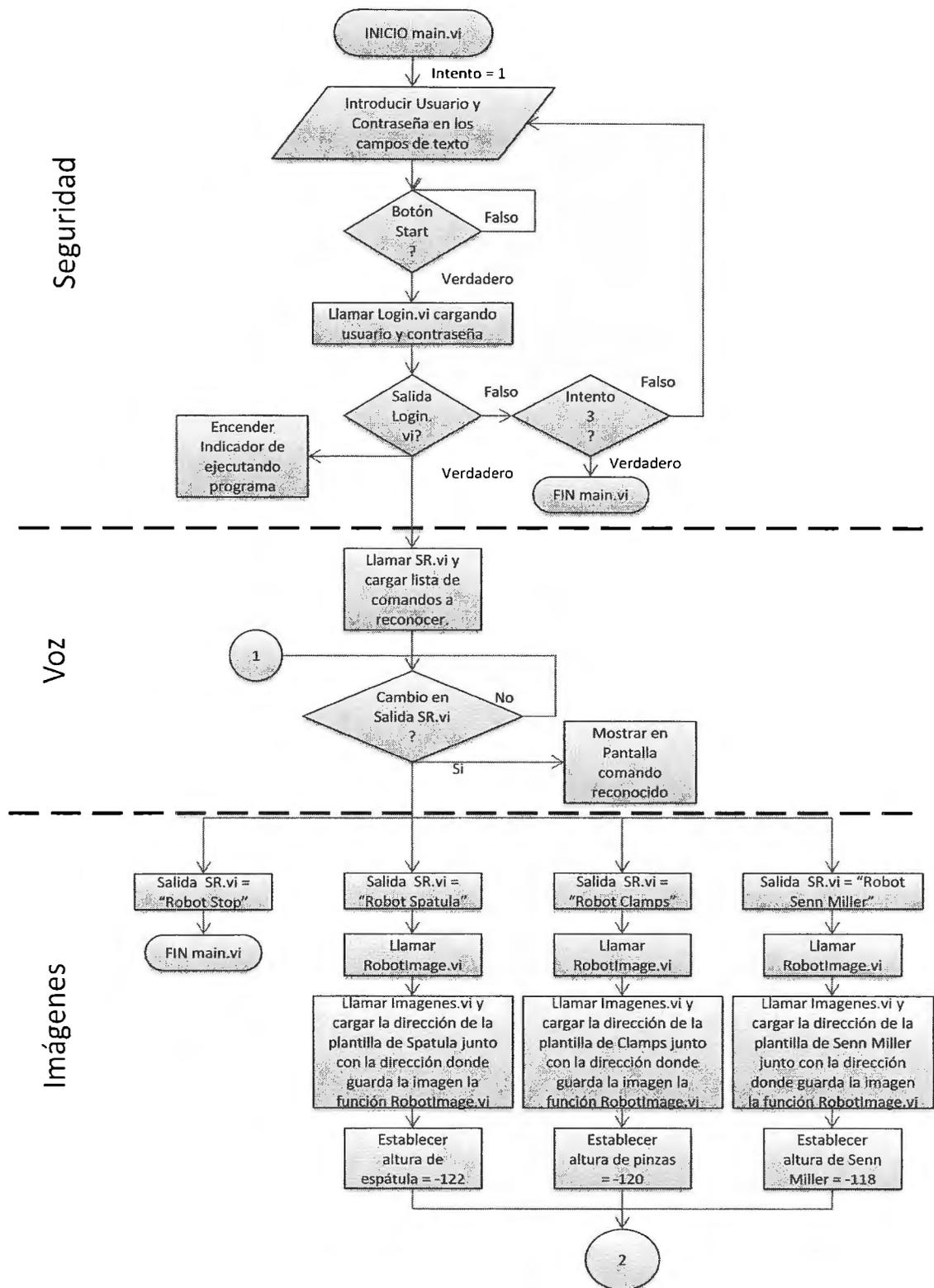


Figura 4.41 Funcionamiento general del sistema.

El programa **main.vi** empieza con una etapa de seguridad la cual valida un usuario y una contraseña por medio de la función **Login.vi**. Una vez que el usuario sea validado, pasa a una etapa de reconocimiento de voz, en donde se llama a la función **SR.vi** la cual ejecuta el programa WSR de Windows y se mantiene funcionando. Dentro de la función **main.vi** verificamos cuando ocurre un evento en la salida de **SR.vi**, lo que significa que el programa WSR ha reconocido algo. El texto que se tiene en la salida lo guarda en una variable llamada "instruccion" y se muestra en pantalla. Esta variable la se introduce a una estructura de casos, y dependiendo del texto reconocido son las tareas que se ejecutan. En el sistema existen solo 4 casos: "Robot Stop", "Robot Spatula", "Robot Calmps" o "Robot Senn Miller". En el caso de "Robot Stop", el programa para su funcionamiento; en los casos de "Robot Spatula", "Robot Clamps" y "Robot Senn Miller", que son muy parecidos, lo primero que se hace es llamar a la función **RobotImage.vi** la cual toma una imagen del área de trabajo y la guarda en una dirección en disco duro. En caso de que ya exista la imagen con el mismo nombre, la sobrescribe. La siguiente acción es llamar a la función **Imagenes.vi** indicándole la dirección de la imagen guardada por **RobotImage.vi** y la dirección de la plantilla a buscar. Esta función nos regresa las coordenadas (X,Y) y la orientación, con respecto al robot, del objeto a reconocer. Los datos obtenidos se muestran en pantalla junto con la distancia Z calculada para cada instrumento. Al igual

que estos datos, se muestra la imagen de la plantilla y la imagen tomada en tiempo real junto con una marca de donde se reconoció el objeto. En caso de no reconocer nada, la salida de puntuación será igual a cero y manda un mensaje al usuario para después regresar a verificar si se ha reconocido algún otro comando de voz. En caso de que haya detectado algo, se verifica con la función **limites.vi** si las coordenadas donde se encuentran los objetos están dentro del rango de operación del robot. En caso de no estarlo, el sistema manda un mensaje al usuario de que el objeto esta fuera de límites y regresa a verificar si WSR ha reconocido algún otro comando. En caso de estar dentro de límites, llama a la función **RobotProgram.vi** y se cargan las coordenadas (X,Y,Z) y el ángulo para definir las posiciones de localización del objeto y después se envía una rutina completa al robot para ejecutar movimientos, tomar la pieza y dejarla al alcance del usuario. En la figura 4.42 se muestra el diagrama de flujo del sistema completo.

Capítulo 4. Diseño de Instrumentista Quirúrgico



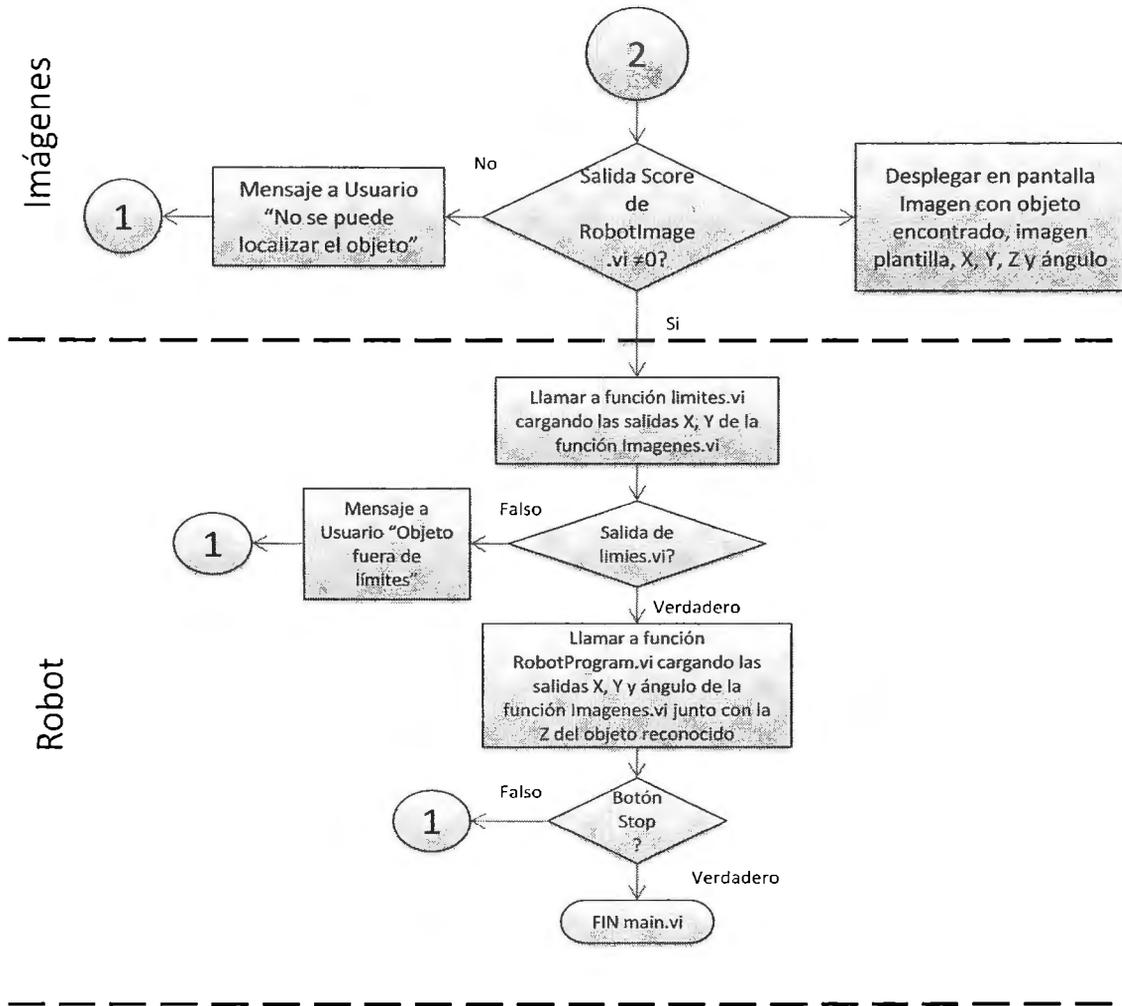


Figura 4.42 Diagrama de flujo de programa main.vi.

La interface de este programa se muestra en la figura 4.43 donde se observan: campos para introducir un usuario y contraseña, el texto reconocido, la imagen plantilla, la imagen en tiempo real, los datos en el espacio calculados, el programa que se está enviando al robot y algunos botones de control. Aquí termina la integración y en el Anexo J se muestra el programa **main.vi** completo con sus distintos casos. Para la utilización del programa en cualquier PC sin necesidad de contar con el programa LabView, se generó un archivo ejecutable llamado **Asistente Robotico.exe**.

## Capítulo 4. Diseño de Instrumentista Quirúrgico

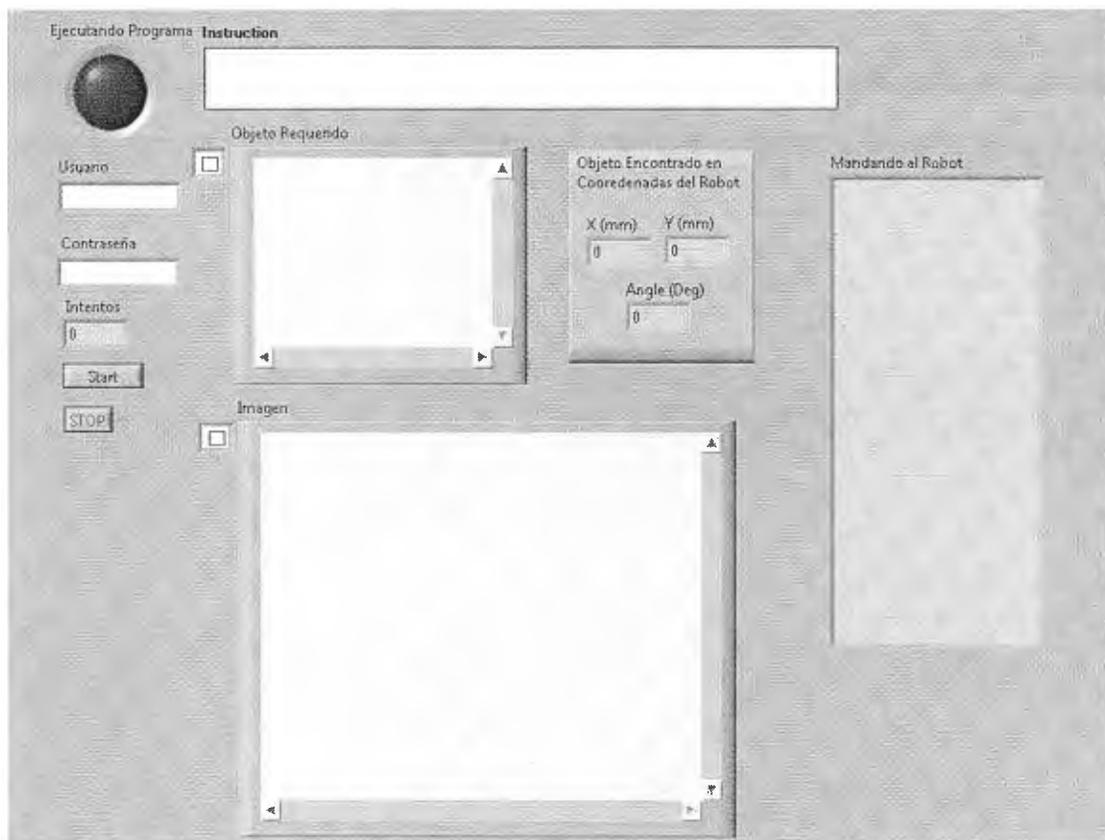


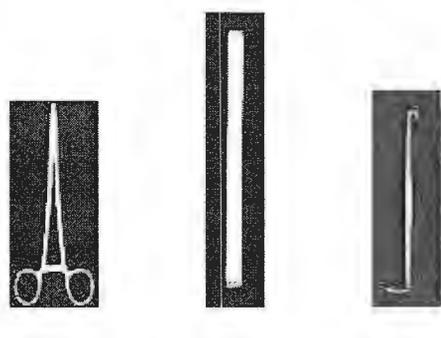
Figura 4.43 Interface de main.vi.

## Capítulo 5. Resultados

En este capítulo se explican los resultados obtenidos, así como algunos cambios que se hicieron para mejorar el funcionamiento del sistema. Se muestran algunos resultados tanto de la salida del sistema como del procesamiento de datos.

### 5.1 Obtención de imágenes plantilla.

Las imágenes de los objetos que sirven como plantillas se muestran en la figura 5.1. Estas plantillas se tomaron de acuerdo a como se desea que sea la orientación de entrega para el usuario. El software calcula la orientación basándose en la imagen plantilla. En este caso se toma la parte de sujeción en la zona inferior de la imagen ya que éste será el punto más cercano al usuario.



*Figura 5.1 Imágenes plantillas (de izquierda a derecha: pinzas, espátula y Senn Miller)*

## 5.2 Compensación de error mecánico

Debido a las condiciones físicas del robot, se tienen errores mecánicos, los cuales se reflejan al momento de tomar los instrumentos. Para corregir estos errores se necesita ajustar los interruptores de localización de origen del robot y hacer una calibración nueva. Para efectos prácticos del proyecto el error se compensa por medio de software. El robot muestra un error más grande mientras se aleja del centro del área de trabajo. En especial el eje X, es el que presenta mayor error ya que es el sentido que el robot recorre más distancia. Para corregir el error, se lleva al robot a la coordenada (0,250), donde 250mm es un punto a la mitad del área de trabajo del robot en sentido de la coordenada Y. Una vez ubicado el robot en esta posición, se mueve desde un extremo del eje X hasta el otro en pasos de 20mm, es decir, desde -200mm hasta 200mm. En cada uno de los pasos se mide el desplazamiento real del centro de herramienta del robot, y se grafica desplazamiento real contra desplazamiento del robot. En la figura 5.2 se puede ver la gráfica que se obtuvo.

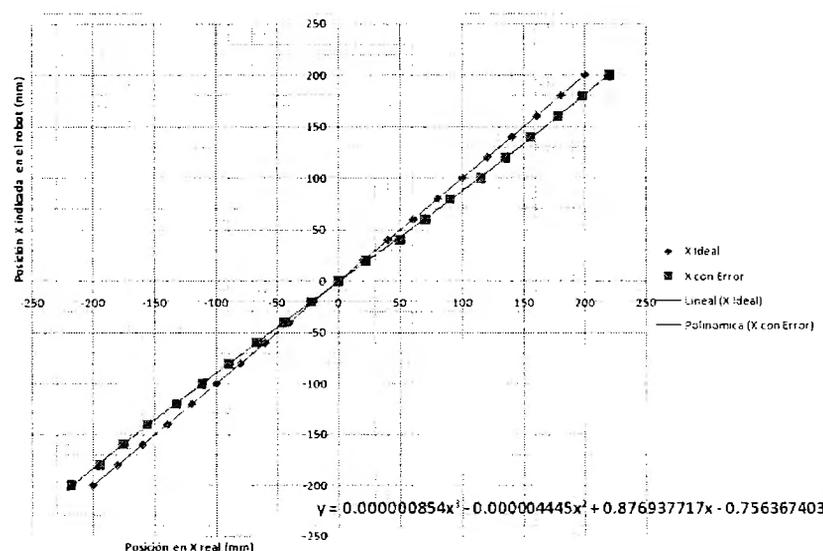


Figura 5.2 Curva para compensar error mecánico

Después de observar la gráfica se aproxima una curva de tercer orden, que se utilizará como función para convertir el punto real en X, a un punto equivalente al robot. Este ajuste se incluye a la salida de la función *imagenes.vi* para poder compensar el error mecánico. En la figura 5.3 se puede ver en que parte del código se anexa ésta compensación.

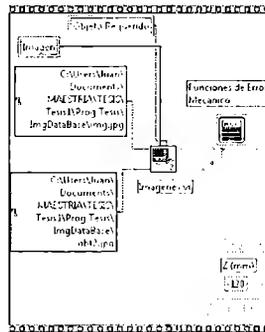


Figura 5.3 Programación de corrección de error en el código.

Para mostrar los resultados se colocó un instrumento en un extremos del área de trabajo y se ejecutaron dos ciclos completos; con función de compensación y sin función de compensación. En la figura 5.4 se muestra la imagen del área de trabajo y la ubicación del objeto. En la figura 5.5 se muestran las imágenes de la posición del robot en el momento de tomar la pieza. Como se observa en la imagen, la precisión del punto de toma mejoró en aproximadamente 15mm.

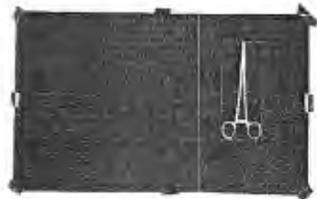


Figura 5.4 Imagen de prueba para mostrar los resultados de la función de compensación.

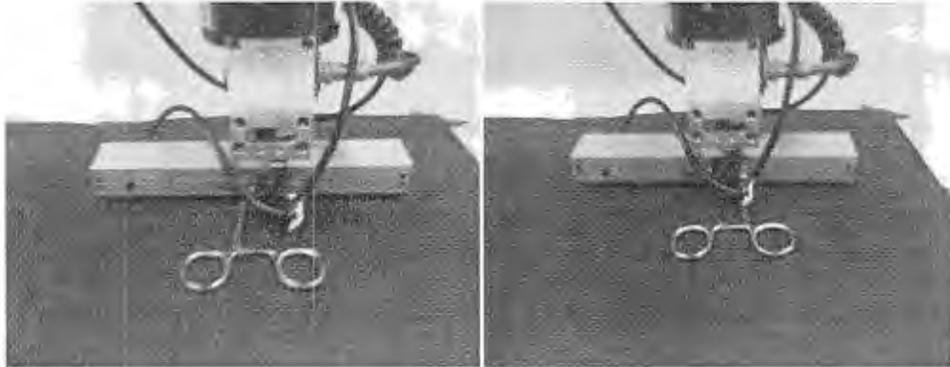


Figura 5.5 A la izquierda punto de toma sin la función de compensación, y a la derecha punto de toma con compensación.

### 5.3 Colocación de instrumentos en un punto fijo

Se realizaron pruebas para dejar los instrumentos en un punto accesible para el usuario. Lo primero que se verifica es la variabilidad en la colocación del instrumento. Para poder medir ésta variabilidad se colocan las pinzas en el punto ideal de sujeción del electroimán y después se lleva al robot al punto de colocación. Aquí se marca en la superficie la posición de un punto fijo en la parte de sujeción de las pinzas como se muestra en la figura 5.6. Después de ejecutar 30 ciclos se midió su variabilidad con respecto a los ejes X y Y.



Figura 5.6 Ubicación de punto de referencia para medición de variabilidad.

La variabilidad máxima en el eje X va de -20mm hasta 20mm, y la variabilidad en el eje Y va de -11mm hasta 20mm lo cual podemos considerar aceptable para un ajuste fino por parte del usuario. En la imagen 5.7 podemos ver los resultados obtenidos con respecto a la variabilidad.

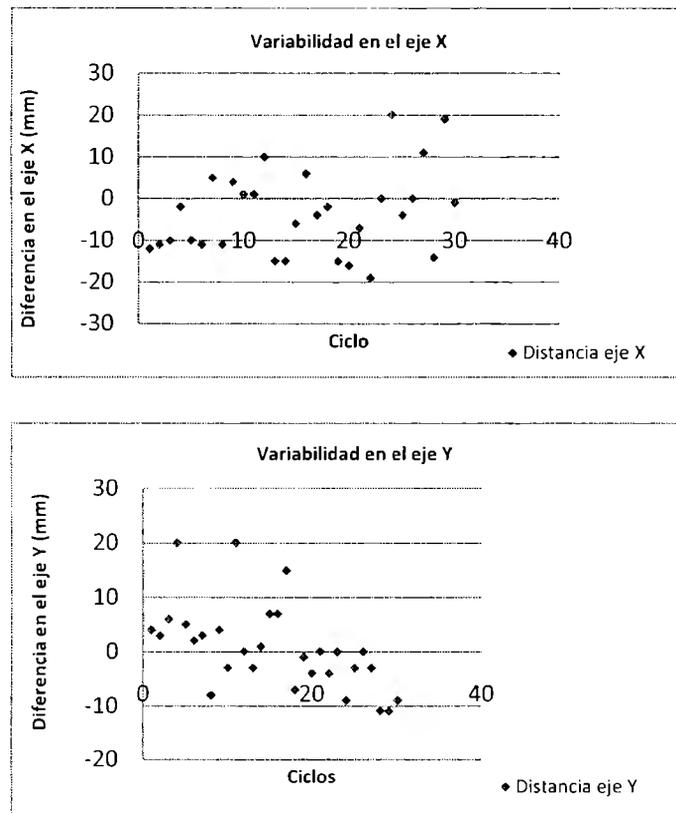


Figura 5.7 Variabilidad del punto de colocación en el eje X y Y.

Durante las pruebas se puede observar que al desactivar el electroimán, las piezas no se quedan en el lugar deseado, sino que se quedan adheridas en la herramienta del robot. Esto se debe a que aún desconectando el electroimán, la bobina se queda con un campo magnético remanente el cual se podría eliminar aplicando una campo coercitivo en sentido contrario para contrarrestar el efecto.

## Capítulo 5. Resultados

Para solucionar este problema de manera sencilla y evitar tiempo en diseñar un circuito se optó por hacer un movimiento de *pitch* de  $10^\circ$  al momento de dejar los instrumentos para ejercer una palanca con un extremo de la pieza. En la figura 5.8 se muestra la solución.



*Figura 5.8 Solución al problema de corriente remanente.*

Después de 20 pruebas se verificó que no se quedaran adheridos los instrumentos en el electroimán.

### **5.4 Cambio de cámara y calibración.**

Es importante mencionar que los valores de calibración de la cámara son específicos para la que es utilizada. En caso de reemplazar la cámara, se tiene que hacer el mismo procedimiento mencionado en la sección 4.4 de éste documento para ajustar la relación de píxeles y milímetros y calcular la traslación de referencia de la imagen para el robot. Durante la elaboración de éste trabajo se probaron 2 cámaras: NX6000 y NX3000. En la figura 5.9 se muestra el cambio en la relación de milímetros y píxeles en las dos cámaras.

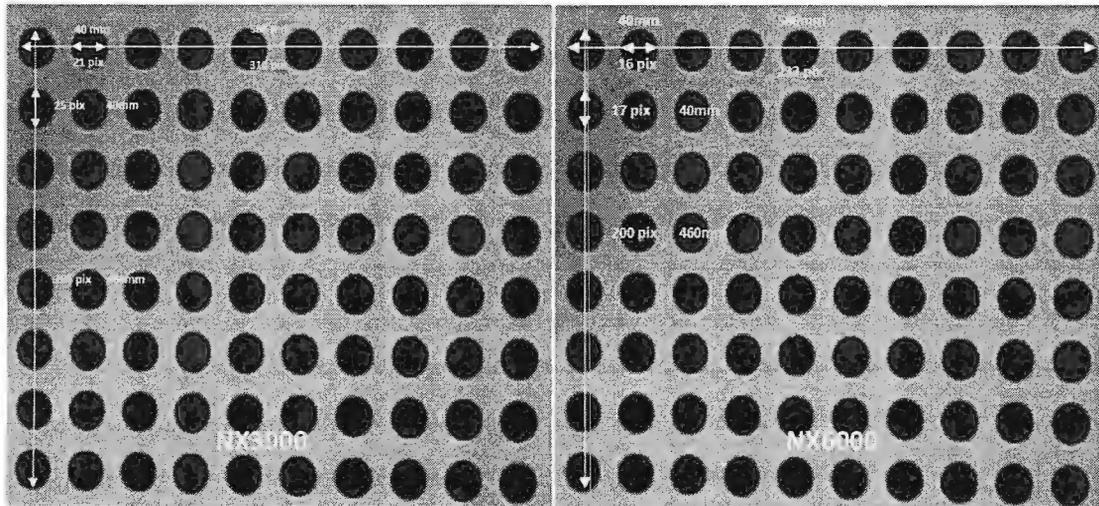


Figura 5.9 Comparación entre relación de pixeles de 2 cámaras distintas.

## 5.5 Iluminación

La iluminación varía de una imagen a otra, y dependiendo la variación era la efectividad del reconocimiento. En la figura 5.10 se muestran dos ejemplos de una misma imagen con diferentes niveles de iluminación. Estas dos imágenes se tomaron bajo las mismas condiciones.

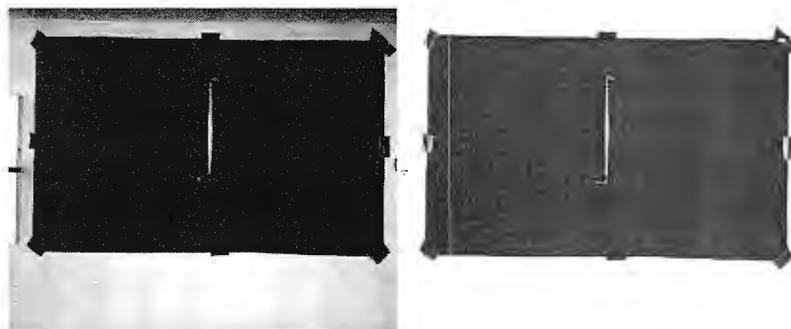


Figura 5.10 Diferencia en iluminación de imágenes bajo las mismas condiciones.

## Capítulo 5. Resultados

Para poder entender más la diferencia en los valores de iluminación, se tomaron 100 imágenes y se calculó el promedio del valor de todos los píxeles de cada una. En la figura 5.11 se muestra el histograma que se obtuvo de las 100 pruebas y se puede observar que existe un comportamiento uniforme entre los cambios de iluminación. El 98% de las imágenes tienen un valor promedio de píxeles que cae en 75 o 120. Cuando se tienen problemas con el reconocimiento de imágenes, es cuando la plantilla se toma en alguna de las dos condiciones típicas de iluminación antes mencionadas, y la imagen a reconocer se toma en la otra condición de iluminación.

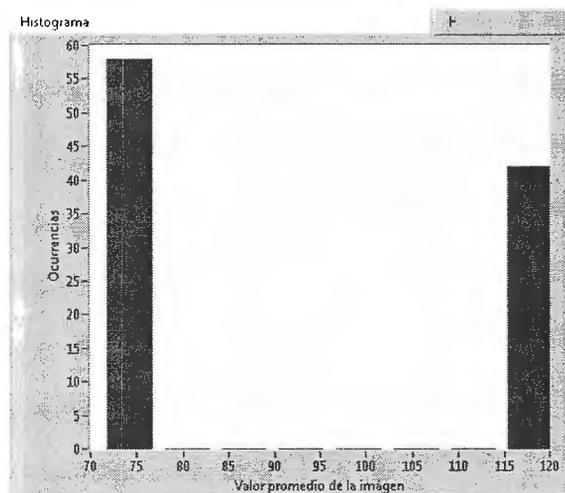


Figura 5.11 Histograma del valor promedio de píxeles en 100 imágenes tomadas.

Este histograma tiene una desviación estándar de 24.52 y una media de 92. Para tratar de reducir la desviación estándar y por consiguiente la variabilidad de la iluminación en las imágenes a reconocer, lo que se hace es retomar la imagen hasta que el promedio de valores de los píxeles sobrepasara un umbral. El valor del umbral que se toma es de 120, y se basó en la figura 5.11. Se decidió tomar éste valor en vez de 75 ya que lo que se busca es tener más iluminación para poder visualizar las formas en los instrumentos. El programa que se modificó para tomar en cuenta el umbral fue **RobotImage.vi**, y se muestra en la figura 5.12. En la figura 5.13 se muestra el diagrama de flujo del mismo.

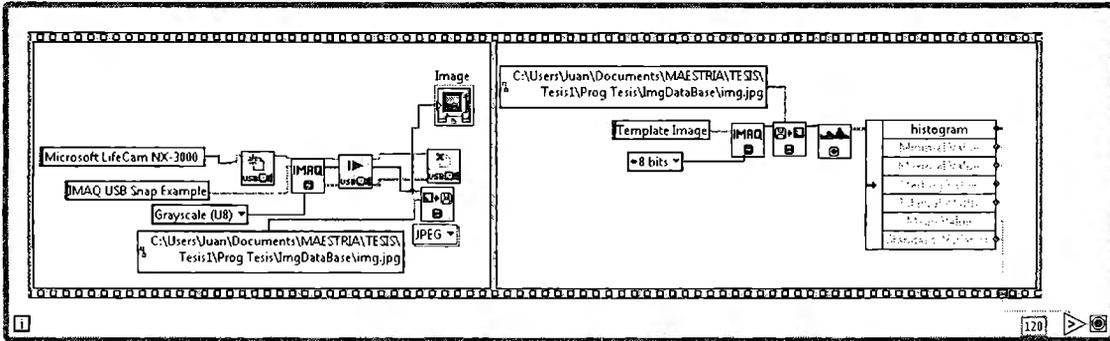


Figura 5.12 Programa modificado RobotImage.vi

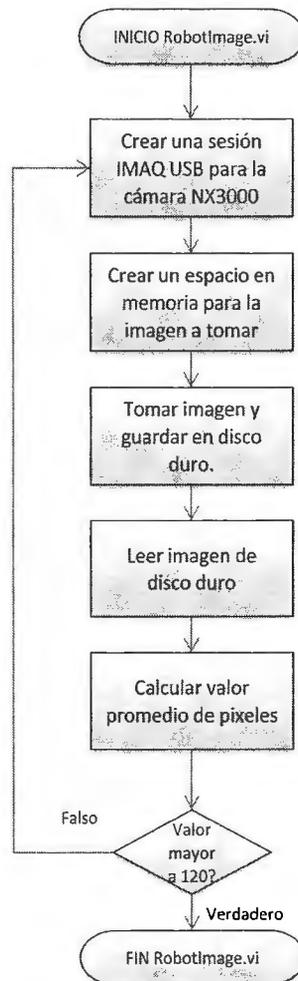


Figura 5.13 Diagrama de flujo de RobotImage.vi

## Capítulo 5. Resultados

Haciendo las modificaciones del programa se retomaron 100 imágenes, y se volvió a calcular el histograma de sus promedios. Ahora se tiene una media de 120.70 con una desviación estándar de 0.14, lo cual fue una mejora notable para disminuir la variabilidad de iluminación en las imágenes a reconocer. En la figura 5.14 se muestra el histograma de las 100 pruebas.

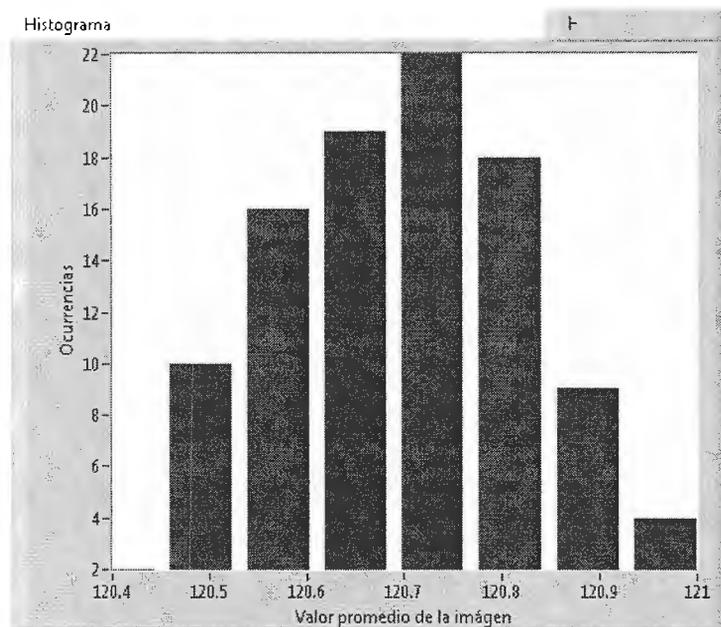


Figura 5.14 Histograma de 100 pruebas con umbral de 120

Después de 20 ciclos con un solo instrumento en el área de trabajo en posiciones aleatorias, se verifica la efectividad del reconocimiento en un 100%. También se puede observar que la variación en la iluminación de las imágenes a analizar era casi imperceptible a simple vista. Por lo tanto se puede definir que las condiciones de iluminación del sistema bajo el arreglo que se tiene, se puede cuantificar en que el nivel promedio de los pixeles sea mayor a 120 para poder obtener resultados confiables.

## 5.6 Mensajes de límites y de falla en reconocimiento

Para comprobar la funcionalidad de los límites de seguridad desarrollados en la sección 4.6 de éste documento, se colocan las pinzas en dos extremos diferentes de la mesa de trabajo y se intenta correr el programa. La figura 5.15 nos muestra como en las dos ocasiones el programa detectó el objeto fuera de alcance y mostró en pantalla un mensaje indicando el estado del instrumento junto con un sonido de error. Esta limitante nos ayuda a evitar colisiones de la herramienta del robot con objetos alrededor de nuestra área de trabajo.

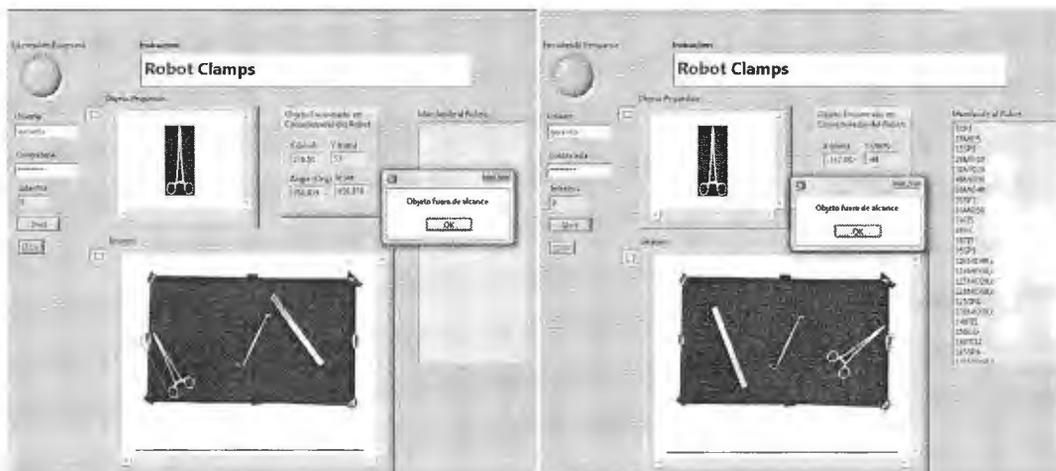


Figura 5.15 Muestra de mensaje "Objeto fuera de alcance".

Otro indicador que se programó fue el mensaje y el sonido de error cuando el sistema no reconoce el objeto, éste mensaje indica que el objeto no se localiza y sugiere que el usuario diga la palabra "cancel" para poder intentar de nuevo. En la figura 5.16 se muestra una pantalla cuando no se encuentra el objeto solicitado.

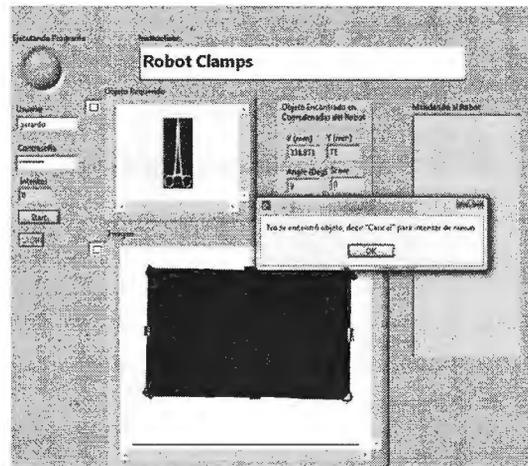
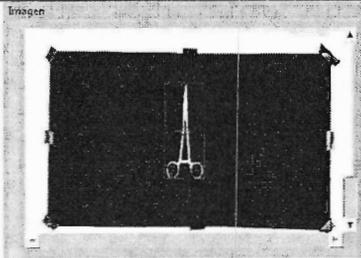
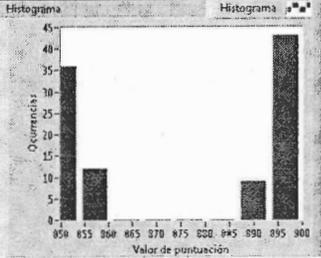
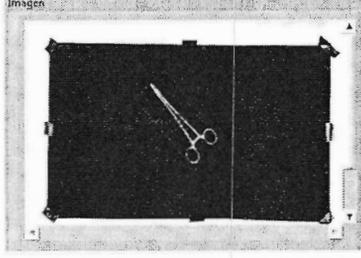
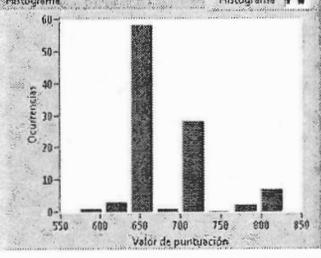
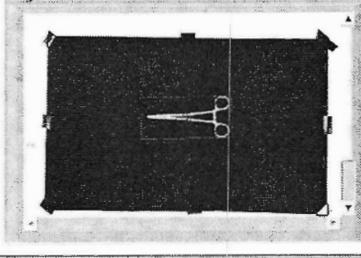
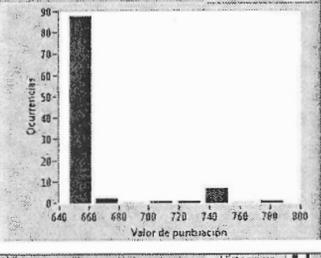
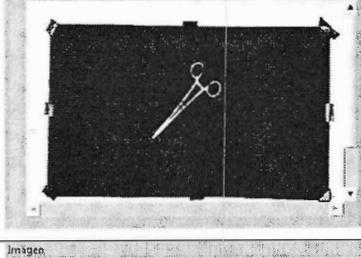
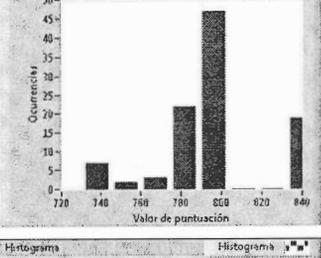
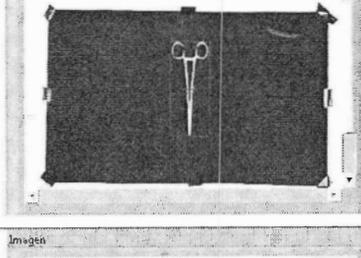
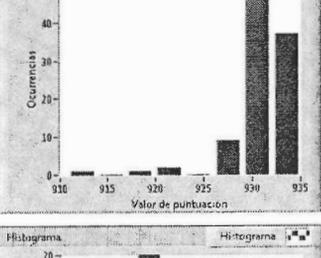
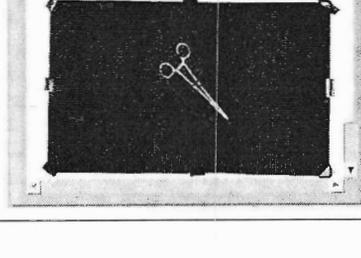
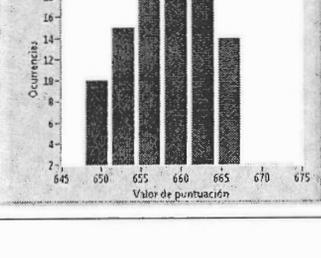


Figura 5.16 Muestra de mensaje "No se encontró objeto, decir "Cancel" para intentar de nuevo".

## 5.7 Reconocimiento de instrumentos en diferentes condiciones

La función *IMAQ Match Pattern 2* cuenta con el parámetro *Match Score* que determina la sensibilidad de reconocimiento. Para poder ajustar la sensibilidad, se tomaron 800 imágenes de las pinzas en diferentes posiciones con un umbral de reconocimiento muy bajo para poder observar cual es la distribución de la puntuación en cada caso. Con ésta distribución se pudo ajustar el valor de sensibilidad a 600 puntos de un total de 1000, de tal manera que no quedara muy estricto para el reconocimiento, ni muy sensible que fuera a reconocer objetos encimados. La tabla 5.1 muestra las pruebas que se hicieron y las distribuciones que se obtuvieron. Para verificar los resultados con la espátula y el instrumento Senn Miller, se corrieron 100 imágenes de cada uno de los últimos. En la figura 5.17 se muestran los resultados de las 100 pruebas que se realizaron a cada instrumento.

Orientación	Media	Desviación estándar	Distribución
0°	874.74	21.78	 
45°	680.85	53.37	 
90°	662.81	27.60	 
135°	795.07	27.55	 
180°	930.77	3.26	 
225°	658.55	5.57	 

Capítulo 5. Resultados

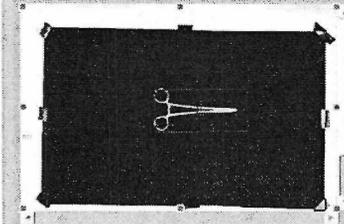
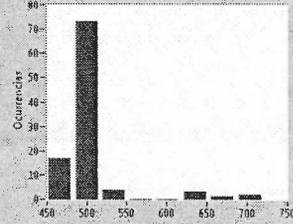
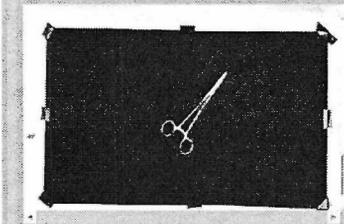
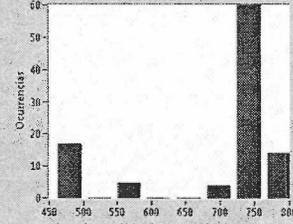
270°	505.78	45.33		
315°	685.43	109.60		

Tabla 5.1 Resultados de puntuación para pinzas en diferentes orientaciones

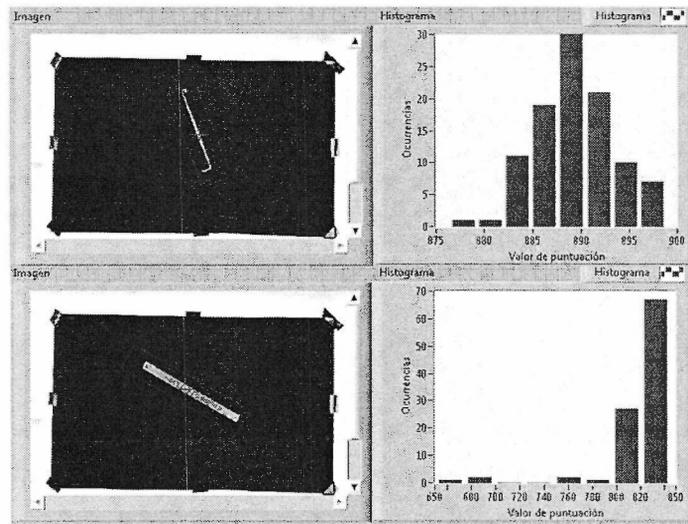


Figura 5.17 Resultados de puntuación para Senn Miller (arriba) y espátula (abajo)

Para probar el umbral que se está usando, se colocan las 3 piezas encimadas, y se ejecutan 100 reconocimientos con un umbral de 500, y 100 reconocimientos con un umbral de 600. En la figura 5.18 se observa que utilizando un umbral de 500 se tiene ocasionalmente algún reconocimiento erróneo, pero utilizando un umbral de 600 en ninguno de los casos

se reconoce la pieza, lo cual es conveniente para evitar tomar una pieza equivocada, o colisionar la herramienta debido a la diferencia de alturas.

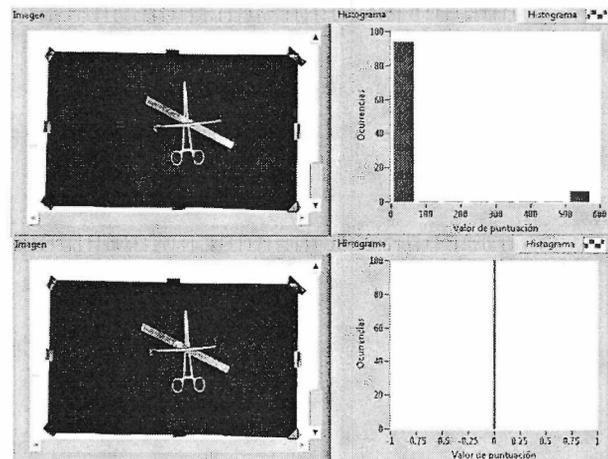


Figura 5.18 Resultados del reconocimiento con un umbral de 500 (arriba) y un umbral de 600 (abajo)

La última prueba es reconocer un instrumento cuando en la imagen aparecen 3 objetos diferentes sin encimar. Los resultados fueron buenos ya que en las 100 imágenes se reconoce el objeto y casi el 100% de las veces la puntuación es superior a 900. En la figura 5.19 se muestran los resultados del reconocimiento de un instrumento cuando existen más instrumentos en el área de trabajo.

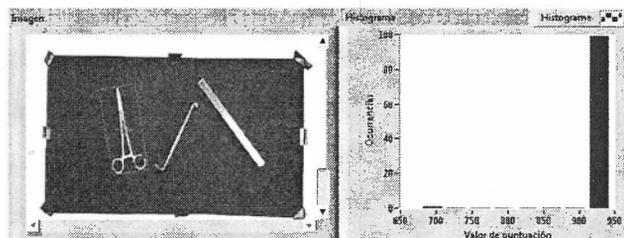


Figura 5.19 Resultados del reconocimiento de un objeto cuando existen 3 instrumentos en la misma imagen

## Capítulo 5. Resultados

En el caso del reconocimiento cuando existen 3 objetos en el área y 2 de ellos están muy pegados, el sistema es capaz de reconocer el objeto pero por razones del tamaño del electroimán, el robot puede llegar a tomar 2 que estén demasiado cerca como se muestra en la figura 5.20.

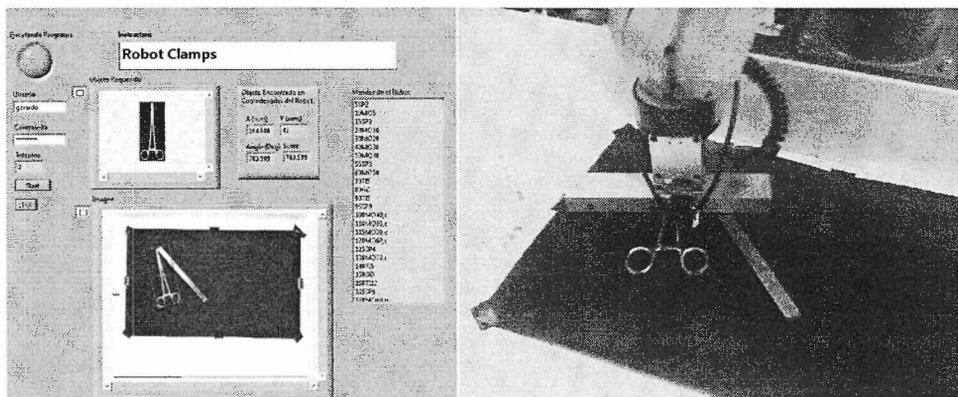


Figura 5.20 Ejemplo de caso cuando existen 2 objetos muy cerca

Al probar diferentes “sentidos” de las piezas se observaron diferentes resultados. Para piezas simétricas como las pinzas y la espátula, no se tiene ningún problema ya que cualquier lado de la pieza que se tenga, es la misma forma. Para herramientas irregulares, se llega a tener problemas ya que la forma de la pieza cambia según el lado en que se ve. La figura 5.21 muestra el ejemplo del instrumento Senn Miller, el cual es irregular, pero debido al tamaño de las formas características de la pieza, en nuestro sistema llega a hacer imperceptible cuando se detecta la forma. En el caso de la Senn Miller, se tiene reconocimiento ya que las partes que son diferentes son muy pequeñas como para ser detectadas por el programa y su forma es parecida. En este caso el sistema no es capaz de asegurar una orientación de entrega de la pieza, pero se decidió permitir esto ya que el instrumento puede ser sujetado en los dos sentidos. Para el caso de otro instrumento como un bisturí, se tiene que ajustar el programa para asegurar de alguna manera la

orientación de entrega del mismo ya que este instrumento puede provocar una herida si se sujeta por el lado afilado.

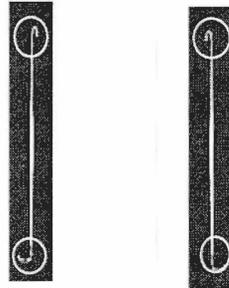


Figura 5.21 Ejemplo de imagen de Senn Miller y su espejo. Se indican las formas que pueden caracterizar al instrumento

## 5.8 Resultados del reconocedor de voz

La herramienta que se utilizó para el reconocimiento de voz fue el *Windows Speech Recognizer* (WSR). Esta herramienta cuando se utiliza por primera vez se tiene que “entrenar” con la persona que utilizará el sistema. Con el fin de mostrar la efectividad de la herramienta que se utilizó, se realizaron 2 pruebas de reconocimiento con diferentes locutores. En la primera prueba se dijeron 30 palabras con un ruido del medio ambiente de 49 dB que fue medido con un sonómetro. La segunda prueba se realizó con un ruido de 72 dB en el medio ambiente generado por música donde se dijeron las mismas 30 palabras (figura 5.22). Los locutores que realizaron la prueba fueron la persona que entrenó el sistema (hombre), otro hombre y una mujer. En la tabla 5.2 se observan los resultados obtenidos.

Locutor	Porcentaje efectividad sin ruido	Porcentaje efectividad con ruido
Hombre (entrenó el sistema)	93%	60%
Hombre	60%	50%
Mujer	17%	7%

Tabla 5.2 Resultado de WSR en diferentes condiciones

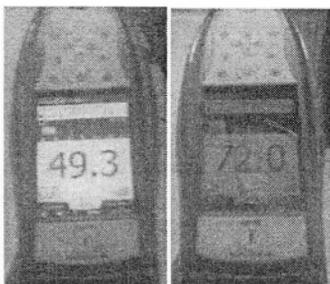


Figura 5.22 Nivel de ruido para pruebas de reconocimiento de voz

Del experimento anterior nos asegura tener un reconocimiento del 93% de efectividad cuando el locutor es la persona que entrena el sistema y el ruido no rebase los 49 dB.

### 5.9 Tiempo de ciclo

El tiempo de ciclo que se obtuvo en el sistema es de 22.5 segundos desde que el usuario dice un comando hasta que el robot regresa a la posición inicial. En la tabla 5.3 se muestran los tiempos en cada punto del programa. Este tiempo comparado con la reacción de un instrumentista puede llegar a ser alto. Se hicieron 30 pruebas con una persona y el tiempo de ciclo es de aproximadamente 3 segundos. Éste tiempo puede cambiar dependiendo de las habilidades de la persona y su experiencia.

Tiempo	Punto	Comentario
00:00:00	5	Inicio
00:02:00	10	
00:03:30	20	
00:04:00	30	
00:06:00	40	
00:08:00	50	Tomar Objeto
00:09:30	40	
00:11:00	30	
00:12:30	20	
00:13:00	60	
00:16:00	70	Dejar Objeto
00:18:00	60	
00:19:30	20	
00:20:30	10	
00:22:30	5	Inicio

Tabla 5.3 Tiempos de ejecución del sistema.

## Capítulo 6. Conclusiones

Según los resultados obtenidos se puede concluir que la posición en la que el sistema entrega los instrumentos es aceptable en cuando a las variables de posición y orientación considerando que el instrumento Senn Miller puede tomarse de cualquiera de sus dos extremos. Las pinzas y la espátula son instrumentos simétricos donde el robot entregará las herramientas en una misma orientación. La variabilidad del sistema propuesto es aceptable ya que el error a la salida es considerablemente ajustable para el usuario.

En cuanto al funcionamiento del robot se tienen algunos errores mecánicos debido a su estado físico, pero las compensaciones que se hicieron fueron suficientes para poder tener una salida satisfactoria.

El utilizar un electroimán como herramienta, solucionó el problema para sujetar los objetos utilizados para este trabajo, pero dependiendo de la forma y los objetos a usar, el diseño de un herramental es fundamental para poder tener una salida aceptable.

Con respecto a las imágenes se puede decir que la luz es un factor elemental en el reconocimiento de imágenes utilizando la herramienta IMAQ Match Pattern de LabVIEW, y para poder tener resultados confiables se requiere tener controlado este factor. El reconocimiento obtenido es notablemente mejor si se tiene menos variabilidad de iluminación y una media en los niveles de pixeles de 120.7.

Para efectos de prototipo, el utilizar una cámara web dio buenos resultados, pero si se quiere tener un sistema más robusto, una cámara con mayor definición nos daría mejores resultados ya que se tendría mayor información y por consiguiente podríamos detectar formas más pequeñas y así caracterizar mejor nuestro instrumento. También se observa que entre menor resolución del objeto a reconocer, menor será la precisión de la posición del reconocimiento.

## Capítulo 6. Conclusiones

En cuanto al reconocimiento de voz, la herramienta que se usa (WSR), da un buen porcentaje de reconocimiento siempre y cuando el locutor sea el que entrene el sistema y los niveles de ruido no sobrepasen 49dB. Para otros locutores, y especialmente para personas de género distinto, y con ruido del ambiente mayor a 49dB el porcentaje de reconocimiento puede llegar a ser bajo.

El tiempo de ciclo del sistema es alto comparado con la reacción de un ser humano ya que el sistema tarda 22.5 segundos en completar el ciclo y una persona puede tardar 3 segundos en hacer un ciclo completo.

En general se puede concluir que la integración de reconocimiento de voz, reconocimiento de imágenes y un robot industrial puede dar como resultado una aplicación más flexible como es un instrumentista quirúrgico. El poder integrar herramientas accesibles en el mercado facilita mucho el trabajo para poder desarrollar sistemas más complejos e inteligentes y por consecuencia se podría abrir más el mercado de los robots industriales. El sistema implementado puede servir como base para trabajos futuros más especializados dependiendo de la aplicación que se le dé.

### 6.1 Trabajo Futuro

Una buena calibración en el robot actual daría una mejor salida en cuanto a la variabilidad. Un mantenimiento completo aseguraría la precisión del robot reportada por el fabricante.

El poder hacer más grande el área de trabajo del robot sería muy útil ya que permitiría manejar más objetos o tener otras aplicaciones. Existen varias maneras de ampliar el área de trabajo como el agregar una extensión a la herramienta o montar el robot sobre una plataforma móvil la cual mueva el robot completo. Otra opción es sustituir el robot por uno de mayor capacidad.

La manera en que se comunica la PC con el robot es en un sentido mediante un archivo de texto que envía la PC. En cuanto llega el robot empieza a ejecutar las instrucciones. Una manera de tener más seguridad en el sistema es ir monitoreando los movimientos del robot paso a paso y tener una comunicación en ambos sentidos.

En cuanto a la calibración de la cámara, se puede hacer no lineal aplicando otro método y lograr una mayor precisión en los extremos de la imagen. LabVIEW cuenta con herramientas que se pueden utilizar para poder hacer calibraciones las cuales pueden ser aplicadas y tener un sistema más robusto. También se puede integrar una cámara con mayor resolución para obtener más información y así aumentar el porcentaje de reconocimiento.

Con lo que respecta a la iluminación se puede disminuir la variabilidad de la intensidad de luz en la imagen aplicando un filtro que compense estos cambios de luz. Podría aplicarse un filtro dinámico el cual varíe dependiendo del cambio de luz.

Hoy en día muchos sistemas operan de manera inalámbrica, así que se podría aplicar este principio para poder operar el robot de manera remota. Teniendo 2 PCs, una conectada al robot y otra en la ubicación del usuario, se podrían mandar instrucciones por medio de alguna red de comunicación como puede ser Ethernet y controlar el robot estando en otra ubicación. También se puede simplemente instalar un micrófono inalámbrico a la PC y poder decir comandos desde una distancia mayor.

El diseño mecánico de un herramental para el robot es una parte que se puede estudiar dependiendo la aplicación que tenga el sistema.

Se podría crear un algoritmo para poder identificar objetos que se encuentren encimados o muy juntos y poder programar una rutina al robot que le permita separarlos y tomar el objeto solicitado por el usuario. Esto evitaría colisionar con objetos alrededor y aseguraría la sujeción del instrumento.

Tener otra cámara en la zona de entrega aseguraría la colocación del instrumento verificando que no exista un objeto presente aplicando el mismo algoritmo de

## Capítulo 6. Conclusiones

reconocimiento de imágenes. En caso de que existiera un objeto en la zona de entrega se podría tomar la pieza y colocarla en una zona de esterilización. En otra aplicación se podrían poner los objetos de vuelta en la zona de trabajo para poder tenerlos accesibles en caso de ocuparlos por segunda vez.

Para poder asegurar un mejor reconocimiento de voz se puede pensar en asociar mediante el usuario y contraseña, un perfil de entrenamiento de voz; esto quiere decir que varias personas entrenen al sistema y buscar la manera que las caracterizaciones se guarde en diferentes bases de datos y así crear perfiles. Cuando se ingrese el nombre y contraseña del programa, en automático utilice las características específicas de voz de ese locutor para el reconocimiento de voz.

En la parte de calibración de la cámara, se podría desarrollar un sistema automático colocando un patrón en alguna parte del área de trabajo, y ejecutando una rutina que pueda calcular la relación de pixeles y milímetros y a su vez la translación de ejes.

Otro proyecto a futuro podría ser el corregir el error mecánico por medio de un sistema de visión que determine el ángulo de error que se tenga entre un eslabón y otro tomando como referencia alguna posición previamente establecida en una imagen patrón.

En cuanto a seguridad del sistema, se podría evitar alguna colisión con objetos dentro de la trayectoria del robot, si se utilizan imágenes en tiempo real que detecten objetos en el camino, y mediante alguna planificación de movimientos evite dichos obstáculos recalculando rutas.

Se podría diseñar un ciclo para que disminuya el tiempo de ejecución y tratar de igualar a la reacción de un ser humano.

En general las ventajas, límites y posibles áreas a desarrollar se muestran en la tabla 6.1

<b><u>Ventajas</u></b>	<b><u>Límites</u></b>	<b><u>Temas a desarrollar</u></b>
Sistema modular de fácil implementación por productos comerciales	Restringido a cambios e innovaciones del fabricante	Desarrollar programas de reconocimiento (visión y voz) en lenguajes básicos (Ej. C o Java)
Sistema expandible a N número de objetos debido a su estructura	Piezas tengan formas características que pueda detectar fácilmente	Obtener más resolución de la imagen diseñando un soporte que se acerque más al área de trabajo
Información de PC-Robot por medio de un archivo .txt	Comunicación mediante puerto serial para el robot en uso	Desarrollar alguna comunicación de un protocolo más actual como puede ser Ethernet.
Uso del electroimán permite tener un rango de error en la posición del robot en el momento de sujetar las piezas.	Debido a la forma y características del electroimán solo puede sujetar piezas "planas" y no hay un control cuando las piezas están muy juntas o encimadas	Desarrollar un efector que pueda sujetar la mayoría de las piezas y algún algoritmo que si encuentra piezas muy encimadas o juntas el robot sea capaz de separarlas para tomar el objeto
Proceso de calibración es igual para cualquier cámara	Tiempo de calibración relativamente alto	Crear un sistema automático para calibrar la cámara sin tener que emplear mucho tiempo en el proceso.
Costo bajo del robot	Poco alcance y %de error de $\pm 2$ cm (relativamente alto)	Adquisición de nuevo robot con más capacidades o mantenimiento mecánico del existente.
Costo bajo de la cámara	Resolución limitada	Adquisición de una cámara con más capacidades para poder detectar formas más pequeñas
Discriminación de Imágenes con intensidad de Iluminación baja (promedio menor a un umbral establecido)	No se asegura funcionamiento si la intensidad no llega al umbral establecido o si la intensidad de luz cambia en el ambiente	Crear una rutina que calcule automáticamente este valor de umbral para establecer las condiciones de luz
La colocación de los objetos es satisfactoria en el lugar previamente programado	No se asegura que éste lugar este libre y si existe otro objeto en la zona de entrega, podría colisionar el robot	Instalar una cámara en la zona de entrega que asegure que no exista algo más y en caso de encontrarse ser capaz de tomarlo y devolverlo a alguna zona determinada.
Alto reconocimiento de voz para la persona que entrena al sistema	Bajo reconocimiento para alguna otra persona	Crear perfiles que permitan asociar el usuario con algún perfil de entrenamiento de voz.
Capaz de manejar bases de datos de instrumentos muy grandes con solo programarlo una vez. La base se puede transferir a otro sistema.	La velocidad de ejecución es relativamente lenta comparado con la reacción de un instrumentista humano entrenado	Modificar el ciclo de programa para reducir tiempos y tratar de igualar el tiempo de ciclo de una persona.

Tabla 6.1 Ventajas, límites y áreas a desarrollar

## Capítulo 6. Conclusiones

## Capítulo 7. Referencias bibliográficas

- [1] **Rattner, Justin.** Electronics Design, Strategy, News. *Technology will bring man, machine closer by 2050.* [Online] Agosto 22, 2008. [Cited: Mayo 29, 2009.]  
<http://www.edn.com/article/CA6589612.html?industryid=47037>.
- [2] **Robotic Industries Association.** Robotics Online. [Online] Enero 18, 2010. [Cited: Junio 3, 2010.] [http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Feature-Article/Robotics-Market-Cautiously-Optimistic-for-2010/content\\_id/1936](http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Feature-Article/Robotics-Market-Cautiously-Optimistic-for-2010/content_id/1936).
- [3] **McClure, Steve.** Robotics Business Review. *Food Processing Industry Looks Appealing to Robotics Vendors.* [Online] Abril 8, 2009. [Cited: Mayo 35, 2009.]  
<http://www.roboticsbusinessreview.com/industry/view/food-processing-industry-looks-appealing-to-robotics-vendors/>.
- [4] **Nación.com.** GM registra pérdidas de \$6.000 millones en el primer trimestre . [Online] Mayo 7, 2009. [Cited: Mayo 25, 2009.]  
[http://www.nacion.com/ln\\_ee/2009/mayo/07/economia1957335.html](http://www.nacion.com/ln_ee/2009/mayo/07/economia1957335.html).
- [5] **Bennett, Jeff.** THE WALL STREET JOURNAL. *Ford registra pérdidas por US\$1.400 millones, pero dice tener dinero para el resto del año.* [Online] Abril 24, 2009. [Cited: Mayo 23, 2009.]  
[http://online.wsj.com/public/article/SB124060427850653947.html#mod=2\\_1362\\_leftbox](http://online.wsj.com/public/article/SB124060427850653947.html#mod=2_1362_leftbox).
- [6] **Isidore, Chris.** CNNMoney.com. *Chrysler to lose \$4.7B this year.* [Online] Mayo 4, 2009. [Cited: Mayo 27, 2009.] [http://money.cnn.com/2009/05/04/news/companies/chrysler\\_losses/](http://money.cnn.com/2009/05/04/news/companies/chrysler_losses/).
- [7] **Charting the Economy.** Auto Market. [Online] Julio 15, 2009. [Cited: Octubre 4, 2010.]  
<http://chartingtheeconomy.com/?cat=15>.
- [8] **RobotWorx.** Robots in Medicine. [Online] [Cited: Noviembre 30, 2010.] <http://www.used-robots.com/robot-education.php?page=robots+in+medicine>.
- [9] **BBC News.** Forth Valley Royal Hospital to use robot workers. June 17, 2010.
- [10] *Survey of robotics in the health care industry.* **Minifie, J.R.** 1989. Engineering Costs and Production.
- [11] **A. Gandsas, M. Parekh, M. M. Bleech and D. A. Tong.** Robotic Telepresence: Profit Analysis in Reducing Length of stay after Laparoscopic Gastric Bypass. 2007.
- [12] *Service Robots.* **The International Federation of Robotics and United Nations.** Geneva : s.n., 2010. World Robotics survey 2010.

## Capítulo 7. Referencias bibliográficas

- [13] *Issues its 2005 World Robotics survey*. **UNECE/IFR**. Geneva : s.n., 2005.
- [14] *Robotics in Surgery*. **Tsoi, Gennifer**. 2010, Yale Scientific Magazine.
- [15] **Universidad de Alcalá**. Diario de Noticias. *Investigadores de la UAH desarrollan un robot que entrega, recoge y cuenta el instrumental utilizado en determinadas operaciones de cirugía*. [Online] Julio 10, 2007. [Cited: Mayo 29, 2009.] <http://www.uah.es/noticias/noticia.asp?Id=1381>.
- [16] **Boquete Vázquez, Luciano, et al.** 200202553 España, 2006.
- [17] **Moctezuma, Regina**. Da Vinci, el cirujano robot. *CNN Expansión*. Octubre 1, 2007.
- [18] **Muñoz, Raquel Vargas**. Presidencia de la República. *El IPN crea una enfermera robot para evitar infecciones*. [Online] Marzo 7, 2006. [Cited: Mayo 29, 2009.] <http://fox.presidencia.gob.mx/buenasnoticias/ciencia/?contenido=23877&pagina=10>.
- [19] **ABB Robotics**. Products & Services. *IRB140*. [Online] 2009. [Cited: Mayo 29, 2009.] <http://www.abb.com.mx/product/seitp327/b0168cc2e1ea9fd5c125717900223403.aspx>.
- [20] **Ponce, Pedro**. *Intelligent Wheelchair and Virtual Training by LabVIEW*. NewYork : SpringerLink, 2010.
- [21]. **Madrigal, Rafael Iñigo**. *Robots industriales manipuladores*. México : Alfaomega, 2005.
- [22]. **International Organization of Standardization**. Manipulationg Industrial Robots - Vocabulary. [Online] 1994. ISO 8373:1994.
- [23]. *Historia de un éxito, una mirada retrospectiva a la aportación de ABB a la robótica industrial*. **Marshall, David and Bredin, Christina**. s.l. : Revista ABB, Febrero 2008.
- [24]. **Fanuc Robotics**. México y su desarrollo educativo - Robots para la educación. [Online] 2009. [Cited: Abril 4, 2009.] [www.fanucrobotics.com.mx](http://www.fanucrobotics.com.mx).
- [25.] **CNN Expansión**. Robots: Soluciones a la mano. [Online] Febrero 5, 2008. [Cited: Mayo 27, 2009.] <http://www.cnnexpansion.com/manufactura/articulos-de-interes/robots-soluciones-a-la-mano>.
- [26]. **Robotic Industries Association**. Robotics Online. [Online] Enero 18, 2010. [Cited: Junio 3, 2010.] [http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Feature-Article/Robotics-Market-Cautiously-Optimistic-for-2010/content\\_id/1936](http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Feature-Article/Robotics-Market-Cautiously-Optimistic-for-2010/content_id/1936).
- [27]. **MOTOMAN**. MOTOMAN Products. *Robobar 2009*. [Online] 2009. [Cited: Mayo 2009, 22.] <http://www.motoman.com/products/nonindustrial/robobar.php>.
- [28]. **KUKA Robotics**. KUKA Entertainment. *Robocoaster*. [Online] 2009. [Cited: Abril 4, 2009.] <http://www.kuka-entertainment.com/en/products/robocoaster/>.

- [29]. **Ollero Baturone, Aníbal**. *Robótica, Manipuladores y robots móviles*. Barcelona : Alfaomega, 2007.
- [30]. **Resnick, Robert**. *Física*. 4ta. México : CECSA, 2004. Vol. I.
- [31]. **Manseur, Rachid**. *Robot Modeling and Kinematics*. Boston : Da Vinci Engineering Press, 2006.
- [32]. **Spong, Mark W**. *Robot Dynamics and Control*. New York : Wiley, 1989.
- [33]. **Venturello, Juan Miguel**. *Herramienta para la programación y simulación de movimientos del manipulador robótico Mitsubishi Movemaster RV-M1*. Santiago de Cali : s.n., 2006.
- [34]. **Mitsubishi Electric Corporation**. *Industrial Micro-Robot System model RV-M1 Manual*. Tokyo : s.n., 1989.
- [35]. **National Instruments**. NI Developer Zone. [Online] Octubre 2009, 2009. [Cited: Julio 29, 2010.] <http://zone.ni.com/devzone/cda/tut/p/id/2895>.
- [36]. —. Comunicación serial:conceptos generales. [Online] Junio 6, 2006. [Cited: Julio 1, 2010.] <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>.
- [37]. **Zhao, Wenyi**. *Face Processing, Advanced Modeling and Methods*. s.l. : Elsevier, 2008.
- [38]. **National Instruments**. *LabView Machine Vision and Image Processing Course Manual*. 2006.
- [39]. **Real Academia Española**. Diccionario de la lengua española - Vigésima segunda edición. [Online] Octubre 18, 2010.
- [40]. **Braintech**. Products. *eVisionFactory*. [Online] 2009. [Cited: Junio 2, 2009.] <http://www.braintech.com/products-evf.php>.
- [41]. **National Instruments**. Familia de Productos NI LabVIEW. [Online] Julio 19, 2010. [Cited: Julio 19, 2010.] <http://www.ni.com/labview/family/esa/>.
- [42]. —. Módulo NI Vision Development. [Online] Julio 19, 2010. [Cited: Julio 19, 2010.] <http://www.ni.com/vision/esa/vdm.htm>.
- [43]. —. 10 Aspectos a Considerar al Elegir Software de Visión. [Online] Julio 28, 2010. [Cited: Julio 28, 2010.] <http://zone.ni.com/devzone/cda/tut/p/id/5603>.
- [44]. **Relf, Christopher G**. *Image Aquisition and Processing with LabVIEW*. s.l. : CRC PRESS, 2004.
- [45]. **GIMP Software**. Color Tools Documentation. [Online] Agosto 2010. [Cited: Agosto 4, 2010.] <http://docs.gimp.org/2.6/en/gimp-tool-desaturate.html>.
- [46]. **Martinsanz, Gonzalo Pajares**. *Visión por Computador, Imágenes digitales y aplicaciones*. Mexico : Alfaomega, 2009.

## Capítulo 7. Referencias bibliográficas

- [47]. **Zhu, Chan.** *Computer Vision Image Understanding*. 1999.
- [48]. **M. Mazo, L. Boquete y R. Barea.** *Visión Artificial*. Alcalá de Henares, Madrid : Servicio de Publicaciones de la Universidad de Alcalá, 1996.
- [49]. **National Instruments.** Convolution. *Product Reference*. [Online] Agosto 9, 2010. [Cited: Agosto 9, 2010.] <http://zone.ni.com/reference/en-XX/help/371361D-01/lvanls/convolution/>.
- [50]. —. *NI Vision for LabVIEW help*. Septiembre 22, 2010.
- [51]. —. Pattern Matching Strategies. [Online] Agosto 16, 2010. [Cited: Agosto 16, 2010.] <http://zone.ni.com/devzone/cda/tut/p/id/3763>.
- [52]. **Rabiner, Laurence.** *Fundamentals of Speech Recognition*. s.l. : Prentice Hall, 1993.
- [53]. **Microsoft.** Microsoft Speech Technologies. [Online] Septiembre 6, 2010. [Cited: Septiembre 6, 2010.] <http://www.microsoft.com/speech/developers.aspx>.
- [54]. **Jorge, Reynaldo Cárdenas.** *Diseño e implementación de un software para la decodificación de voz utilizando Matlab*. Pamplona-Colombia : s.n., 2006.
- [55]. **National Instruments.** Characteristics of Different Smoothing Windows. [Online] Agosto 2006. [Cited: Septiembre 2010, 13.] [http://zone.ni.com/reference/en-XX/help/371361B-01/lvanlsconcepts/char\\_smoothing\\_windows/](http://zone.ni.com/reference/en-XX/help/371361B-01/lvanlsconcepts/char_smoothing_windows/).
- [56]. *IMPLEMENTACIÓN DE UN RECONOCEDOR DE PALABRAS AISLADAS DEPENDIENTE DE UN LOCUTOR*. César San Martín S., Roberto Carrillo A. 1, Chile : Revista Facultad de Ingeniería, U.T.A. , Mayo 3, 2004, Vol. 12.
- [57]. **Tecnológico de Monterrey.** *Clase de Procesamiento de Señales*. Ciudad de México : s.n., 2008.
- [58]. **Moore, Andrew W.** Hidden Markov Models. *School of Computer Science, Carnegie Mellon University*. [Online] 2010. [Cited: Septiembre 19, 2010.] <http://www.cs.cmu.edu/~awm/tutorials>.
- [59]. **Rabiner, Laurence.** *A Tutorial of Hidden Markov Models and Selected Applications in Speech Recognition*. 1989.
- [60]. **Microsoft Research.** Speech Technology (Redmond). [Online] 21 Septiembre, 2010. [Cited: Septiembre 2010, 21.] <http://research.microsoft.com/en-us/groups/srg/>.
- [61]. **Fuller, Joana R.** *Instrumentación Quirúrgica: Principios y Práctica*. Buenos Aires : Medica Panamericana, 1995.
- [62]. **Microsoft.** DirectShow System Overview. [Online] Agosto 31, 2010. [Cited: Septiembre 2010, 2010.] [http://msdn.microsoft.com/en-us/library/dd375470\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd375470(v=VS.85).aspx).

- [63]. **National Instruments**. NI-IMAQ for USB Cameras. [Online] Agosto 19, 2009. [Cited: Septiembre 2010, 27.] <http://zone.ni.com/devzone/cda/epd/p/id/5030>.
- [64]. **Bhasin Sons Private Limited**. [Online] 2010. [Cited: Septiembre 28, 2010.] <http://bhasinsons.tradeindia.com/>.
- [65]. **Truninger**. Electroimanes. [Online] 2010. [Cited: Septiembre 28, 2010.] <http://www.truninger.com>.
- [66]. **Murphy, Roberto**. *Teoría electromagnética*. México : Trillas, 2001.
- [67]. **Mitsubishi Electric Corporation**. *Industrial Micro-Robot System model RV-M1 Manual*. Tokyo : s.n., 1989.
- [68]. **National Instruments**. Voice Recognition with LabView and Windows Vista. [Online] Septiembre 20, 2007. [Cited: Octubre 7, 2010.] <http://forums.ni.com/t5/LabVIEW/Voice-Recognition-with-LabVIEW-and-Windows-Vista/td-p/581702/page/2>.

## Capítulo 7. Referencias bibliográficas

## Anexo A

### Obtención de la representación de Denavit Hartenberg

Para obtener las ecuaciones de Denavit Hartenberg (D-H), partimos de las ecuaciones homogéneas de traslación y rotación de un punto en 3 dimensiones. Las ecuaciones homogéneas son un sistema genérico de coordenadas para expresar cambios de coordenadas entre marcos de referencia tridimensionales. La matriz de traslación es la siguiente:

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde  $\Delta x, \Delta y, \Delta z$  son los desplazamientos en cada eje respectivamente. Por otro lado las matrices de rotación según el eje son:

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C & -S & 0 \\ 0 & S & C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_y = \begin{bmatrix} C & 0 & S & 0 \\ 0 & 1 & 0 & 0 \\ -S & 0 & C & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_z = \begin{bmatrix} C & -S & 0 & 0 \\ S & C & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde  $C = \cos\theta$ ,  $S = \sin\theta$ .

## Anexo A. Obtención de la representación de Denavit Hartenberg

La convención de D-H, nos propone utilizar el producto de transformaciones homogéneas, utilizando 4 matrices básicas para representar el cambio de ejes de referencia en aplicaciones robóticas:

$$A_i^{i-1} = R_{z,\theta_i} T_{z,d_i} T_{x,a_i} R_{x,\alpha_i}$$

$$A_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

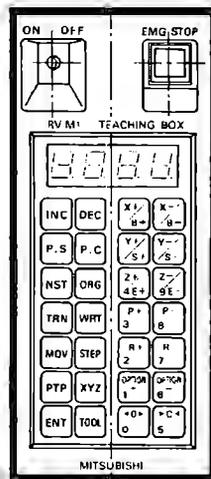
Donde  $a_i$ ,  $\alpha_i$ ,  $d_i$  y  $\theta_i$  son parámetros relacionados con el enlace  $i$  y la junta  $i$ . Estos parámetros son longitud del enlace, giro del enlace, desplazamiento del enlace y ángulo de la unión respectivamente. Por último para obtener las coordenadas de la herramienta y la orientación hacemos el producto de cada uno de los enlaces:

$$T_0^n = A_1 A_2 A_3 \dots A_n$$

## Anexo B

### Funciones de las teclas del *teaching box*

En la tabla A1, podemos ver todas las funciones que se pueden realizar con el *teaching box*



	Elemento	Función	Descripción
1		Interruptor de encendido/apagado	Éste habilita o deshabilita el control de mando. Cuando se utiliza, el <i>teaching box</i> , éste debe de estar encendido, y cuando se trabaja en modo PC, se tiene que apagar.
2		Paro de emergencia	Paro inmediato del robot y se activa el indicador de error ubicado en el controlador.
3		Incremento de posición	Mueve el robot a una posición predefinida, con un número de posición mayor a la que se encuentre el robot.
4		Decremento de posición	Mueve el robot a una posición predefinida, con un número de posición menor a la que se encuentre el robot.
5		Establecer posición	Establece la posición actual a un número indicado.
6		Borrar posición	Borra la posición de algún número previamente establecido.
7		Regresar a origen	Regresa al robot a su posición de origen.
8		Regresa a origen cartesiano	Regresa al robot a su origen cartesiano.

Anexo B. Funciones de las teclas del *teaching box*

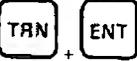
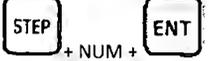
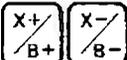
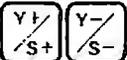
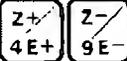
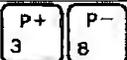
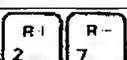
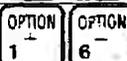
9		Transferir	Transfiere el contenido de la memoria EPROM a la memoria RAM del controlador.
10		Escribir	Escribe lo que tiene la memoria RAM del controlador a la memoria EPROM.
11		Mover herramienta	Mueve la herramienta del robot a un punto predefinido.
12		Correr programa por pasos	Ejecuta el programa paso a paso empezando en una posición indicada.
13		Mover el robot por ejes	Cuando se selecciona ésta opción, se está indicando que el movimiento será por ejes. Cuando se enciende el <i>teaching box</i> , esta opción se habilita automáticamente.
14		Mover el robot por coordenadas base	Ésta opción sirve para mover el robot con referencia a sus coordenadas base.
15		Mover el robot por coordenadas de herramienta	Ésta opción sirve para mover el robot con referencia a sus coordenadas de herramienta.
16		Completar funciones	Completa las funciones de la 1 a la 12.
17		Mover en dirección de coordenada "X" o cintura	Con movimientos en coordenadas se mueve en dirección de "X" y con movimientos por ejes se mueve la cintura.
18		Mover en dirección de coordenada "Y" u hombro	Con movimientos en coordenadas se mueve en dirección de "Y" y con movimientos por ejes se mueve el hombro.
19		Mover en dirección de coordenada "Z" o codo, y sirve como tecla numérica	Con movimientos en coordenadas se mueve en dirección de "Z" y con movimientos por ejes se mueve el codo. Sirven como teclas 4 y 9.
20		Mover la muñeca (cabeceo) y sirve como tecla numérica.	Con movimientos por ejes se mueve la muñeca (cabeceo). Sirven como teclas 3 y 8.
21		Mover la muñeca (giro) y sirve como tecla numérica.	Con movimientos por ejes se mueve la muñeca (giro). Sirven como teclas 2 y 7.
22		Mover ejes externos y sirve como tecla numérica.	Mueve ejes externos en caso de haber y sirve como teclas 1 y 6.
23		Control de mano y sirve como tecla numérica.	Cierra o abre la mano y sirve como tecla 0 y 5.

Tabla A.1 Funciones de las teclas del *teaching box*

## Anexo C

### Lista de instrucciones para el RV-M1

En éste apartado podremos ver todas las instrucciones que tiene nuestro sistema, así como una descripción de la misma y su sintaxis. Las tablas A1-A6, están divididas según su objetivo.

	Nombre	Formato	Descripción	Programa	Observaciones
1	Decrement Position	DP	El robot se mueve a una posición predefinida con un número de posición más pequeño que el actual.	Si	
2	Draw	DW X, Y, Z	Mueve la herramienta a un lugar alejado de la actual y cubre la distancia especificada en los ejes X, Y, y Z.	Si	
3	Here	HE a	Define las coordenadas de la posición actual mediante la asignación del número (a).		$1 \leq a \leq 629$
4	Home	PO	Establece la posición de referencia en el sistema de coordenadas cartesianas.	Si	
5	Increment position	IP	El robot se mueve a una posición predefinida con un número de posición superior a la actual.	Si	
6	Move Approach	MA $a_1, a_2$ [, O/C]	Mueve la herramienta desde la posición actual a una posición alejada de la posición $a_1$ , en incrementos según se especifica en la posición " $a_2$ ".	Si	$1 \leq a_1, a_2 \leq 629$ O: mano abierta; C: mano cerrada
7	Move continuous	MC $a_1, a_2$	El robot se mueve continuamente a través de puntos intermedios predefinidos entre la posición $a_1$ y $a_2$	Si	$1 \leq a_1, a_2 \leq 629$
8	Move Joint	MJ w, s, e, p, r	Convierte cada articulación del ángulo especificado desde la posición actual.	No	
9	Move	MO a [, O/C]	Mueve la herramienta a la posición a.	Si	$1 \leq a \leq 629$ O: mano abierta; C: mano cerrada
10	Move Position	MP x, y, z, p, r	Mueve la herramienta a una posición cuyas coordenadas (posición y ángulos) se especifican como x, y, z, p, r.	No	
11	Move Straight	MS a, n [, O/C]	El robot se mueve a una posición inicial a través de "n" puntos intermedios en línea recta.	Si	$1 \leq a \leq 629$ $1 \leq n \leq 99$ O: mano abierta; C: mano cerrada

Anexo C. Lista de instrucciones para el RV-M1

12	Move Tool	MT a, b[, O/C]	La herramienta se mueve de la posición actual a una posición "a" en pasos "b" en dirección de las coordenadas de la herramienta	Si	$1 \leq a \leq 629$ O: mano abierta; C: mano cerrada
13	Nest	NT	Devuelve al robot al origen mecánico.	Si	
14	Origin	OG	El robot se mueve a la posición de referencia en el sistema de coordenadas cartesianas.	Si	
15	Pallet Assign	PA i, j, k	Define el número de puntos (j, k) de la matriz (columna y fila) del pallet {i}.	Si	$1 \leq i \leq 9$ $1 \leq j, k \leq 255$
16	Position Clear	PC a <sub>1</sub> [, a <sub>2</sub> ]	Borrar todos los datos de las posiciones a <sub>1</sub> a a <sub>2</sub>	No	$A_1 \leq a_2$ $1 \leq a_1, a_2 \leq 629$ (ó a <sub>1</sub> = 0)
17	Position Define	PD a, x, y, z, p, r	Define las coordenadas (x, y, z, p, r) de la posición "a".	No	$1 \leq a \leq 629$
18	Position Load	PL a <sub>1</sub> , a <sub>2</sub>	Asigna las coordenadas de la posición a <sub>1</sub> a la a <sub>2</sub>	Si	$1 \leq a_1, a_2 \leq 629$
19	Pallet	PT a	Calcula las coordenadas de un punto de la matriz del pallet e identifica las coordenadas.	Si	$1 \leq a \leq 9$
20	Position Exchange	PX a <sub>1</sub> , a <sub>2</sub>	Intercambio de las coordenadas de la posición a <sub>1</sub> para las de posición a <sub>2</sub>	Si	$1 \leq a_1, a_2 \leq 629$
21	Shift	SF a <sub>1</sub> , a <sub>2</sub>	Cambia las coordenadas a <sub>1</sub> en incrementos que representan las coordenadas de la posición de a <sub>2</sub> y redefine las nuevas coordenadas.	Si	$1 \leq a_1, a_2 \leq 629$
22	Speed	SP a [,H/L]	Establece la velocidad de operación y tiempo de aceleración / deceleración para el robot. 0: la velocidad mínima; 9: velocidad máxima	Si	$0 \leq a \leq 9$ H: tiempo de aceleración/deceleración alta, L: aceleración/deceleración baja
23	Timer	TI a	Detiene el movimiento en un tiempo a. (unidad: 0.1 segundos)	Si	$0 \leq a \leq 32767$
24	Tool	TL a	Establece la distancia entre la base de la mano y el punto final de la herramienta.	Si	$0 \leq a \leq +300.0$ Unidad: mm

Tabla A2. Instrucciones de Posición y Movimiento

	Nombre	Formato	Descripción	Programa	Observaciones
25	Compare counter	CP a	Carga el valor del contador en el registro interno	Si	$1 \leq a \leq 99$
26	Disable act	DA a	Desactiva la interrupción por medio del bit a de una entrada externa.	Si	$0 \leq a \leq 7$ (15)
27	Decrement counter	DC a	Decrementos de "a" por uno	Si	$1 \leq a \leq 99$
28	Delete line	DL a <sub>1</sub> , [, a <sub>2</sub> ]	Borra el contenido del programa desde las líneas a <sub>1</sub> a a <sub>2</sub> .	No	$A_1 \leq a_2$ $1 \leq a_1, a_2 \leq 2048$
29	Enable act	EA a <sub>1</sub> , a <sub>2</sub>	Habilita las interrupciones de	Si	(-15) (+15)

Anexo C. Lista de instrucciones para el RV-M1

			un bit $a_1$ de una terminal externa y especifica el número de la línea $a_2$ en que el programa salta cuando se produce la interrupción.		$-7 \leq a_1 \leq +7$ +: ON; -: OFF $1 \leq a_2 \leq 2048$
30	End	ED	Termina el programa	Si	
31	If equal	EQ $a_1$ (or &b), $a_2$	Se produzca un salto a la línea $a_2$ si la entrada externa o los datos del contador es igual a $a_1$ (or &b)	Si	$[-32767] (32767)$ $0 \leq a_1 \leq 255$ {decimal} $0 \leq b \leq \&FF$ {hex} $[\&8001] (\&7fff)$ $1 \leq a_2 \leq 2048$
32	Go sub	GS $a$	Permite la secuencia de instrucciones para saltar a subrutina que empieza con un número de línea	Si	$1 \leq a \leq 2048$
33	Go to	GT $a$	Permisos de la secuencia del programa para saltar a un número de línea sin condiciones	Si	$1 \leq a \leq 2048$
34	Increment counter	IC $a$	Contador "a" se incrementa por 1	Si	$1 \leq a \leq 99$
35	If larger	LG $a_1$ (or &b), $a_2$	Hace saltar a ocurrir a la línea $a_2$ , si los datos externos de entrada o los datos del contador es mayor que $a_1$ (o &b)	Si	$(-32767) (32767)$ $0 \leq a_1 \leq 255$ {decimal} $0 \leq b \leq \&FF$ {hex} $(\&8001) (\&7FFF)$ $1 \leq a_2 \leq 2048$
36	If not equal	NE $a_1$ (or &b), $a_2$	Se produce un salto a la línea $a_2$ si la entrada o el contador no es igual a $a_1$ (o &b)	Si	$(-32767) (32767)$ $0 \leq a_1 \leq 255$ {decimal} $0 \leq b \leq \&FF$ {hex} $(\&8001) (\&7FFF)$ $1 \leq a_2 \leq 2048$
37	New	NW	Se eliminan todos los programas y los datos de la memoria RAM	No	
38	Next	NX	Especifica el intervalo de un bucle en un programa ejecutado por orden de RC	Si	
39	Repeat cycle	RC $a$	Repite el ciclo especificado en el comando NX (a veces)	Si	$1 \leq a \leq 32767$
40	Run	RN $a_1$ [, $a_2$ ]	Ejecuta un número de línea de $a_1$ a $a_2$ , no incluyendo $a_2$	No	$1 \leq a_1, a_2 \leq 2048$
41	Return	RT	Completa la subrutina activada por la GS y vuelve al programa principal.	Si	
42	Set counter	SC $a_1$ , [ $a_2$ ]	Carga $a_2$ al contador $a_1$	Si	$1 \leq a_1 \leq 99$ $-32767 \leq a_2 \leq 32767$
43	If smaller	SM $a_1$ (or &b), $a_2$	Causas que se produzca un salto a la línea $a_2$ si las entradas o los datos del contador es menor que $a_1$ (o &b)	Si	$(-32767) (32767)$ $0 \leq a_1 \leq 255$ {decimal} $0 \leq b \leq \&FF$ {hex} $(\&8001) (\&7FFF)$ $1 \leq a_2 \leq 2048$

Tabla A2. Instrucciones de Control de programa

	Nombre	Formato	Descripción	Programa	Observaciones
44	Grip close	GC	Cierra herramienta	Si	
45	Grip flag	GF $a$	Define el estado cerrado/abierto de la herramienta, se utiliza junto con el comando PD	Si	$a = 0$ (abierto), $=1$ (cerrado)

Anexo C. Lista de instrucciones para el RV-M1

46	Grip open	GO	Abre herramienta	Si	
47	Grip pressure	GP a <sub>1</sub> , a <sub>2</sub> , a <sub>3</sub>	Define la fuerza de agarre y el tiempo de retención	Si	0 ≤ a <sub>1</sub> , a <sub>2</sub> ≤ 15 0 ≤ a <sub>3</sub> ≤ 99 (unidad: 0.1 segundo)

Tabla A3. Instrucciones de Control de herramienta

	Nombre	Formato	Descripción	Programa	Observaciones
48	Input direct	ID	Obtiene señal externa incondicionalmente	Si	
49	input	IN	Obtiene señal externa y la sincroniza desde un puerto de entrada	Si	
50	Output bit	OB a	Establece el estado del bit a de una salida externa	Si	-7 ≤ a ≤ +7 (-15) (+15) +: encendido: -: apagado
51	Output direct	OD a (or &b)	Establece el bit de salida a incondicionalmente.	Si	(-32767) (32767) 0 ≤ a ≤ 255 (decimal) 00 ≤ b ≤ &FF (hex) (&8001) (&7FFF)
52	Output	OT	Establece el bit de salida a de forma sincronizada a través del puerto de salida	Si	(-32767) (32767) 0 ≤ a ≤ 255 (decimal) 00 ≤ b ≤ &FF (hex) (&2701) (&7 FFF)
53	Test bit	TB a <sub>1</sub> , a <sub>2</sub>	Casos que se produzca un salto a la línea a <sub>2</sub> a través de la señal a <sub>1</sub> en la terminal de entrada externa	Si	-7 ≤ a <sub>1</sub> ≤ +7 (-15) (+15) +: encendido: -: apagado 1 ≤ a <sub>2</sub> ≤ 2048

Tabla A4. Instrucciones de Control de entradas y salidas

	Nombre	Formato	Descripción	Programa	Observaciones
54	Counter read	CR a	Lee el contenido del contador a	Si	1 ≤ a ≤ 99
55	Data read	DR	Lee los datos en la terminal externa, usada junto con los comandos ID e IN	si	
56	Error read	ER	Lee el estado de error	No	
57	Line read	LR a	Lee el contenido del número de línea a	No	1 ≤ a ≤ 2048
58	Position read	PR a	Lee las coordenadas de la posición a	Si	1 ≤ a ≤ 629
59	Where	WH	Lee las coordenadas de la posición actual	Si	

Tabla A5. Instrucciones para leer por el puerto RS232C

	Nombre	Formato	Descripción	Programa	Observaciones
60	Reset	RS	Restablece el modo de error II	No	
61	Transfer	TR	Transferencias de contenidos EPROM a RAM	No	
62	Write	WR	Transferencias de contenidos RAM a EPROM	No	
63	Comment		Permite al programador a hacer un comentario después del comando '.	Si	

Tabla A6. Instrucciones varias

## Anexo D

### Tabla de módulos de LabVIEW

En la tabla D1, se muestran los diferentes módulos y herramientas con las que cuenta LabVIEW.

	<b>Módulos y Herramientas</b>
<b>Diseño Embebido</b>	<ul style="list-style-type: none"> <li>- LabVIEW Real-Time</li> <li>- NI Real-Time Execution and Trace Toolkit</li> <li>- LabVIEW FPGA</li> <li>- LabVIEW Microprocessor SDK</li> <li>- LabVIEW Statechart</li> <li>- LabVIEW Mobile</li> <li>- LabVIEW DSP</li> <li>- LabVIEW Embedded para Microcontroladores ARM</li> </ul>
<b>Diseño de control y simulación</b>	<ul style="list-style-type: none"> <li>- LabVIEW Control Design and Simulation</li> <li>- LabVIEW PID Control Toolkit</li> <li>- LabVIEW Real-Time</li> <li>- NI Real-Time Execution and Trace Toolkit</li> <li>- LabVIEW FPGA</li> <li>- LabVIEW Statechart</li> <li>- LabVIEW Simulation Interface Toolkit</li> <li>- LabVIEW System Identification Toolkit</li> </ul>
<b>Procesamiento de imágenes y señales</b>	<ul style="list-style-type: none"> <li>- LabVIEW Vision Development</li> <li>- LabVIEW MathScript RT</li> <li>- LabVIEW Advanced signal Processing Toolkit</li> <li>- LabVIEW Digital Filter Design Toolkit</li> <li>- LabVIEW Adaptive Filter Toolkit</li> <li>- Sound and Vibration Toolkit</li> <li>- Spectral Measurements Toolkit</li> <li>- LabVIEW Modulation Toolkit</li> <li>- Vision Builder for Automated Inspection</li> <li>- LabVIEW Math Interface Toolkit</li> </ul>
<b>Monitoreo y control industrial</b>	<ul style="list-style-type: none"> <li>- LabVIEW Real – Time</li> <li>- NI Real-Time Execution and Trace Toolkit</li> <li>- LabVIEW FPGA</li> <li>- LabVIEW Datalogging and Supervisory Control</li> <li>- LabVIEW Touch Panel</li> <li>- LabVIEW Statechart</li> <li>- NI Motion Assistant</li> <li>- LabVIEW NI SoftMotion</li> </ul>

Anexo D. Tabla de módulos de LabVIEW

<p><b>Desarrollo e implementación de software</b></p>	<ul style="list-style-type: none"> <li>- LabVIEW Application Builder para Windows</li> <li>- LabVIEW VI Analyzer Toolkit</li> <li>- LabVIEW Statechart</li> <li>- LabVIEW Desktop Execution Trace Toolkit</li> <li>- LabVIEW Paneles Remotos</li> <li>- NI Requirements Gateway</li> <li>- LabVIEW Test Framework Toolkit</li> </ul>
<p><b>Generación de reportes y almacenamiento de datos</b></p>	<ul style="list-style-type: none"> <li>- LabVIEW Signal Express</li> <li>- Report Generation Toolkit para Microsoft Office</li> <li>- LabVIEW Database Connectivity Toolkit</li> <li>- LabVIEW DataFinder Toolkit</li> <li>- LabVIEW Internet Toolkit</li> </ul>

**Tabla D1. Módulos y herramientas de LabVIEW**

## Anexo E

### Fórmulas para calcular el modelo HSL para imágenes a color

Las ecuaciones E1 nos muestran cómo es que se calculan los parámetros para el modelo de tono, saturación y luminosidad para imágenes a color. Los valores (R, G y B) tienen que ser expresados en números del 0 al 1. MAX equivale al máximo de éstos valores y MIN al mínimo.

$$H = \begin{cases} 0, & \text{si } MAX = MIN \\ \left(60 \times \frac{G - B}{MAX - MIN} + 360\right) \bmod 360, & \text{si } MAX = R \\ 60 \times \frac{B - R}{MAX - MIN} + 120, & \text{si } MAX = G \\ 60 \times \frac{R - G}{MAX - MIN} + 240, & \text{si } MAX = B \end{cases}$$

$$L = \frac{1}{2}(MAX + MIN)$$

$$S = \begin{cases} 0, & \text{si } MAX = MIN \\ \frac{MAX - MIN}{MAX + MIN} = \frac{MAX - MIN}{2L}, & \text{si } L \leq \frac{1}{2} \\ \frac{MAX - MIN}{2 - (MAX + MIN)} = \frac{MAX - MIN}{2(1 - L)}, & \text{si } L > \frac{1}{2} \end{cases}$$

Ecuación E1. Formulas para calcular el modelo HSL.

## Anexo E. Fórmulas para calcular el modelo HSL para imágenes a color

## Anexo F

### Ejemplos de Kernels y algunos resultados

La tabla F1 nos muestra algunos ejemplos de kernels para filtros Prewitt y Sobel pre-programados. La tabla F2 nos muestra ejemplos con filtros gradiente y Laplacianos y la tabla F3 muestra imágenes filtradas con kernels para suavizar y Gaussianos.

Filtros Prewitt: Estos filtros nos ayudan a resaltar bordes en cierta dirección		
Kernel	Imagen Original	Imagen Procesada
$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$		
Kernel	Imagen Original	Imagen Procesada
$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -1 \end{bmatrix}$		
Filtros Sobel: Son muy parecidos a los Prewitt, con la diferencia que resaltan la variación de intensidad de luz a lo largo de un eje		
Kernel	Imagen Original	Imagen Procesada
$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$		
Kernel	Imagen Original	Imagen Procesada
$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 1 & -1 \\ 2 & 1 & 0 \end{bmatrix}$		

Tabla F1. Filtros Prewitt y Sobel

**Filtros Gradiente:** Estos filtros son buenos cuando se desea mapear variaciones de intensidades a lo largo de un eje específico. También nos ayuda a hacer ver más gruesa las líneas de contornos detectadas. Mientras más incrementes el tamaño de tu Kernel, más se incrementará el grosor de tu línea.

Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>0</td><td>-2</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>-2</td><td>-2</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>-1</td><td>-1</td><td>-1</td></tr> </table>	1	1	1	0	0	1	2	2	0	0	1	2	0	-2	-1	0	0	-2	-2	-1	0	0	-1	-1	-1		
1	1	1	0	0																							
1	2	2	0	0																							
1	2	0	-2	-1																							
0	0	-2	-2	-1																							
0	0	-1	-1	-1																							
Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>0</td><td>-1</td><td>-1</td><td>-1</td><td>0</td></tr> <tr><td>-1</td><td>-2</td><td>-2</td><td>-2</td><td>-1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>2</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	0	-1	-1	-1	0	-1	-2	-2	-2	-1	0	0	1	0	0	1	2	2	2	1	0	1	1	1	0		
0	-1	-1	-1	0																							
-1	-2	-2	-2	-1																							
0	0	1	0	0																							
1	2	2	2	1																							
0	1	1	1	0																							

**Filtros Laplacianos:** La manera más fácil de entender éstos filtros es como un filtro gradiente en todas las direcciones. Al igual que los filtros gradiente, al aumentar el tamaño del kernel, aumentará el grosor de nuestros bordes.

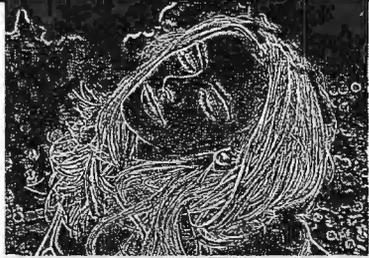
Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>24</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> </table>	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	24	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
-1	-1	-1	-1	-1																							
-1	-1	-1	-1	-1																							
-1	-1	24	-1	-1																							
-1	-1	-1	-1	-1																							
-1	-1	-1	-1	-1																							
Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>25</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> <tr><td>-1</td><td>-1</td><td>-1</td><td>-1</td><td>-1</td></tr> </table>	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	25	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1		
-1	-1	-1	-1	-1																							
-1	-1	-1	-1	-1																							
-1	-1	25	-1	-1																							
-1	-1	-1	-1	-1																							
-1	-1	-1	-1	-1																							

Tabla F2. Filtros Gradiente y Laplacianos

**Filtros de Suavizamiento:** Estos filtros se utilizan para promediar y el efecto que crean es como hacer una imagen borrosa. Normalmente se utilizan para eliminar ruido, pero a la vez nos pueden borrar detalles que tal vez nos sirvan en un procesamiento.

Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	1	1	1																							
1	1	1	1	1																							
1	1	1	1	1																							
1	1	1	1	1																							
1	1	1	1	1																							

Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	1	1	1																							
1	1	1	1	1																							
1	1	0	1	1																							
1	1	1	1	1																							
1	1	1	1	1																							

**Filtros Gaussianos:** Éste filtro es utilizado como una técnica efectiva para la disminución de ruido. Aquí disminuimos el hacer borrosa las imágenes

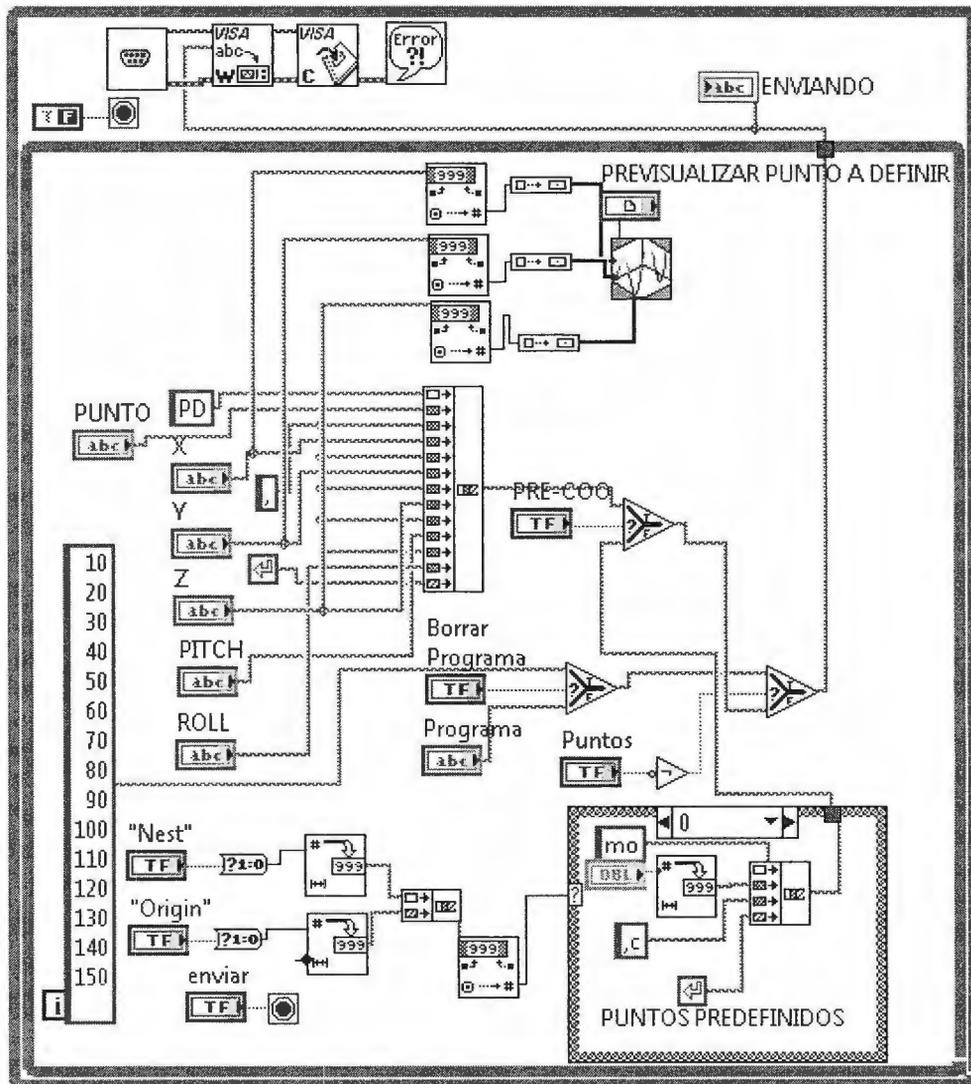
Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>1</td><td>2</td><td>4</td><td>2</td><td>1</td></tr> <tr><td>2</td><td>4</td><td>8</td><td>4</td><td>2</td></tr> <tr><td>4</td><td>8</td><td>16</td><td>8</td><td>4</td></tr> <tr><td>2</td><td>4</td><td>8</td><td>4</td><td>2</td></tr> <tr><td>1</td><td>2</td><td>4</td><td>2</td><td>1</td></tr> </table>	1	2	4	2	1	2	4	8	4	2	4	8	16	8	4	2	4	8	4	2	1	2	4	2	1		
1	2	4	2	1																							
2	4	8	4	2																							
4	8	16	8	4																							
2	4	8	4	2																							
1	2	4	2	1																							

Kernel	Imagen Original	Imagen Procesada																									
<table border="1"> <tr><td>16</td><td>8</td><td>4</td><td>8</td><td>16</td></tr> <tr><td>8</td><td>4</td><td>2</td><td>4</td><td>8</td></tr> <tr><td>4</td><td>2</td><td>1</td><td>2</td><td>4</td></tr> <tr><td>8</td><td>4</td><td>2</td><td>4</td><td>8</td></tr> <tr><td>16</td><td>8</td><td>4</td><td>8</td><td>16</td></tr> </table>	16	8	4	8	16	8	4	2	4	8	4	2	1	2	4	8	4	2	4	8	16	8	4	8	16		
16	8	4	8	16																							
8	4	2	4	8																							
4	2	1	2	4																							
8	4	2	4	8																							
16	8	4	8	16																							

Tabla F3. Filtros de Suavizamiento y Gaussianos

# Anexo G

## Programa Escribir2.vi

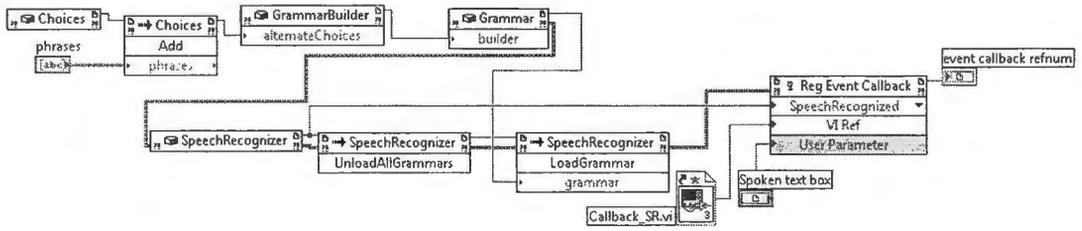


Anexo G. Programa Escribir2.vi

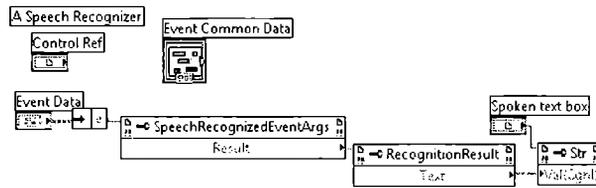
## Anexo H

### Programas para reconocimiento de voz

#### H.1 Programa SR.vi



#### H.2 Programa Callback\_SR.vi



## Anexo H. Programas para reconocimiento de voz

## Anexo I

### Programa del Robot

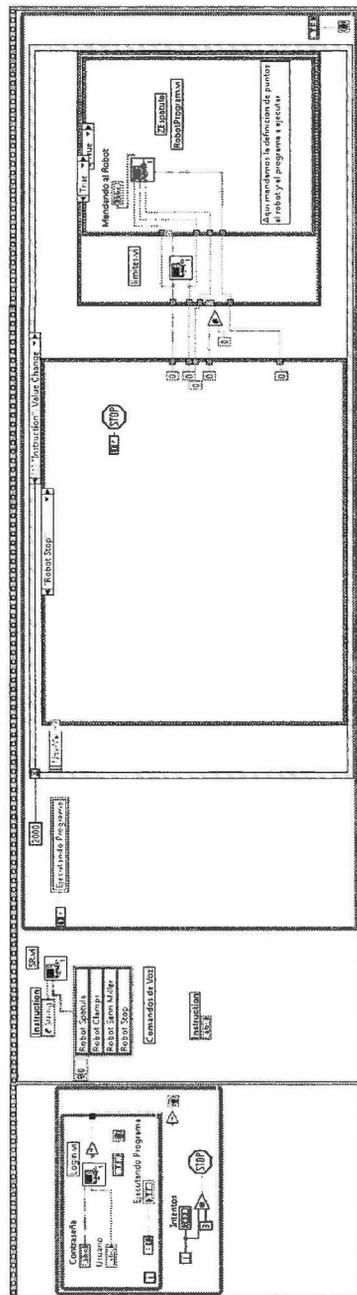
5SP2	'Establece velocidad lenta (Aire)
10MO5	'Mueve robot a posición 5 (seguridad)
15SP9	'Establece velocidad rápida
20MO10	'Mueve Robot a posición 10 (Aire)
30MO20	'Mueve Robot a posición 20 (Aire)
40MO30	'Mueve Robot a posición 30 (0, 300,-120,-95,-10)
50MO40	'Mueve Robot a posición 40
55SP3	'Establece velocidad lenta
60MO50	'Mueve Robot a posición 50
70TI5	'Espera 0.5 segundos
80GC	'Cierra Gripper
90TI5	'Espera 0.5 segundos
95SP9	'Establece velocidad rápida
100MO40,c	'Mueve Robot a posición 40 asegurando el gripper cerrado
110MO30,c	'Mueve Robot a posición 30 asegurando el gripper cerrado
115MO20,c	'Mueve Robot a posición 20 asegurando el gripper cerrado
120MO60,c	'Mueve Robot a posición 60 asegurando el gripper cerrado (-390,0,50,-90,-10)
125SP2	'Establece velocidad lenta
130MO70,c	'Mueve Robot a posición 70 asegurando el gripper cerrado (-390,0,10,-80,-10)
140TI5	'Espera 0.5 segundos
150GO	'Abre Gripper
160TI12	'Espera 1.2 segundos
165SP9	'Establece velocidad rápida
170MO60,o	'Mueve Robot a posición 60 asegurando gripper abierto
180MO20,o	'Mueve Robot a posición 20 asegurando gripper abierto
190MO10,o	'Mueve Robot a posición 10 asegurando gripper abierto
200MO5,o	'Mueve robot a posición 5 (seguridad) asegurando gripper abierto
ED	'Acaba Programa
RN	'Ejecutar Programa

## Anexo I. Programa del robot

## Anexo J

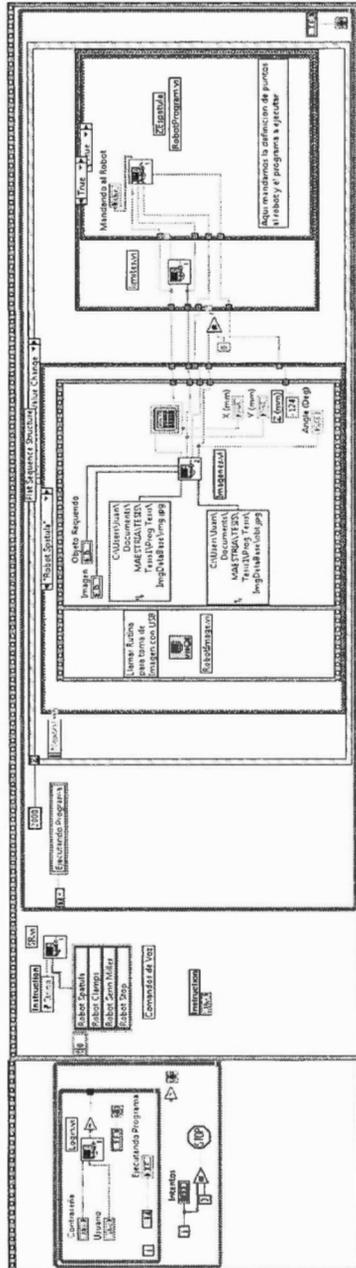
### Programa main.vi

#### J.1 Caso "Robot Stop"





### J.3 Caso "Robot Spatula"



J4. Caso "Robot Senn Miller"

