

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**



PONTIFICIA  
**UNIVERSIDAD**  
**CATÓLICA**  
DEL PERÚ

**CONTROL AUTOMÁTICO DE ADQUISICIÓN DE IMÁGENES EN  
UN MICROSCOPIO ÓPTICO BINOCULAR. SOFTWARE DEL  
PROTOTIPO**

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

**Onego Adrián Aquino Camarena**

ASESORES: Ing. Willy Carrera Soria y Ph. D. Benjamín Castañeda Aphan

**Lima, marzo del 2011**

## RESUMEN

La automatización de procesos, en la actualidad, es una característica inherente a los diversos sistemas modernos. Su desarrollo a través de los años ha permitido diseñar técnicas mucho más precisas y de útil aplicación en diversas áreas. Precisamente en el entorno médico es donde se ha ido realizando bastas innovaciones con el propósito de obtener procedimientos seguros que contribuyan al bienestar del ser humano. El presente trabajo plantea el diseño del software de control de un sistema automático, adaptado a un microscopio, para la captura de imágenes en una muestra médica. Este diseño busca crear un prototipo que sea de utilidad en el área de baciloscopía de los diversos hospitales, ya que es en esa sección donde se realizan procedimientos rutinarios para la obtención de diagnósticos rápidos, que implican una toma de acciones preventivas para salvaguardar la salud del paciente.

El trabajo se desarrolla en cuatro capítulos. El primer capítulo presenta el estado actual de la adquisición automática de imágenes de muestras médicas en microscopios, características de estos sistemas, propiedades y el desarrollo tecnológico que implica su fabricación y constante mejora. El segundo capítulo abarca los conceptos básicos necesarios para la comprensión a fondo del tema tratado. En el tercer capítulo se plantean la problemática y los objetivos que se deben cumplir al término del trabajo. Asimismo, se presenta de manera detallada el desarrollo de la solución al problema: Algoritmo de control del sistema, diagrama de flujo y código de programación; para ello se toman en cuenta las consideraciones necesarias para el desarrollo de la técnica realizada manualmente y la calidad mínima de imagen que solicita el especialista para la generación de un diagnóstico fiable. En el cuarto capítulo se realiza la evaluación del Algoritmo diseñado y la comprobación del cumplimiento de las especificaciones requeridas. Asimismo, se realiza la evaluación del trabajo realizado para ver su funcionamiento integral y los alcances que tiene, sus implicancias dentro de un trabajo aplicativo en laboratorio y los posibles errores en el proceso; obteniéndose resultados satisfactorios pues se verifica el correcto funcionamiento del algoritmo de control del sistema, ello se ilustra mediante imágenes de cada campo analizado y gráficas que describen la claridad de las imágenes dentro de la muestra a partir de un punto de referencia (primera imagen capturada manualmente).

Posteriormente, se citan las conclusiones generadas a partir de las pruebas realizadas y los objetivos planteados. Estas conclusiones indican el cumplimiento de los objetivos generales y específicos así como sus posibles alcances dentro de un laboratorio de baciloscopía y su aplicación en la telemedicina, llamada telemicroscopía.



## TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Título : Control Automático de Adquisición de imágenes en un Microscopio Óptico Binocular. Software del Prototipo  
 Área : Bioingeniería # 822  
 Asesor : Ing. Willy Carrera Soria, PhD. Benjamín Castañeda Aphan.  
 Alumno : Onego Adrián Aquino Camarena  
 Código : 20057236  
 Fecha : 26 de mayo de 2010

## Descripción y Objetivos

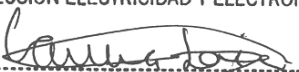
El proceso de análisis de muestras médicas, mediante el uso de un microscopio óptico, realizado en los laboratorios de los hospitales implica una variedad de normas y técnicas. Entre ellas, la baciloscopía destaca por su rapidez de evaluación y la generación de un diagnóstico simple e inmediato, con cierto grado de fiabilidad.

El tiempo de realización de este proceso depende, básicamente, de la cantidad de bacilos que se encuentre en cada campo analizado en la muestra, pero ello no implica modificación alguna en el proceso. Entonces, al seguirse un procedimiento específico, se tiene un plan de acción definido, ello conlleva al planteamiento de la automatización del sistema.

El presente trabajo tiene como objetivo diseñar los algoritmos del software de control para un sistema automático de adquisición de imágenes acoplado a un microscopio óptico binocular. Para ello se toma en cuenta que el proceso automatizado debe cumplir con los requerimientos de la técnica en cuestión, es decir, claridad de la imagen, tiempo de análisis y número de campos analizados por muestra médica.

El diseño del Algoritmo permitirá la comunicación e interrelación entre las subpartes del sistema en general: Cámara Web, Driver de control de Motores y Algoritmo de evaluación de claridad de la imagen. Esta lógica de control implica dos partes: Algoritmo de manejo de los motores y Algoritmo del Sistema Integral. El primero tendrá en cuenta las instrucciones pertinentes para el movimiento de los motores. El segundo algoritmo engloba al sistema en sí: la lógica de evaluación y seguimiento de los campos de la muestra analizada y el envío de instrucciones hacia los motores para los movimientos pertinentes. Posteriormente se realizarán pruebas que verifiquen el correcto funcionamiento de los algoritmos.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
SECCIÓN ELECTRICIDAD Y ELECTRÓNICA

  
Ing. WILLY CARRERA SORIA  
PROFESOR ASOCIADO

MÁXIMO

80

PÁGINAS

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
SECCIÓN ELECTRICIDAD Y ELECTRÓNICA

  
Ing. ANDRÉS FLORES ESPINOZA  
Coordinador de la Especialidad de Ingeniería Electrónica

## TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Título : Control Automático de Adquisición de imágenes en un Microscopio Óptico Binocular. Software del Prototipo

### Índice

#### Introducción

1. Estado Actual de la Adquisición de imágenes de Muestras Médicas en Microscopios, de manera automática
2. Fundamentación Teórica.
3. Diseño de los Algoritmos del Software de Control para la Automatización del proceso de Adquisición de Imágenes de muestras médicas en un Microscopio Óptico Binocular.
4. Evaluación y Resultados

#### Conclusiones

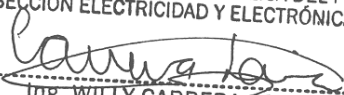
#### Recomendaciones

#### Bibliografía

#### Anexos



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
SECCIÓN ELECTRICIDAD Y ELECTRÓNICA

  
Ing. WILLY CARRERA SORIA  
PROFESOR ASOCIADO

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
SECCIÓN ELECTRICIDAD Y ELECTRÓNICA

  
Ing. ANDRES FLORES ESPINOZA  
Coordinador de la Especialidad de Ingeniería Electrónica



## INDICE GENERAL

<b>INTRODUCCIÓN</b>	4
<b>CAPÍTULO 1: ESTADO ACTUAL DE LA ADQUISICIÓN DE IMÁGENES DE MUESTRAS MÉDICAS EN MICROSCOPIOS, DE MANERA AUTOMÁTICA</b>	6
1.1 Investigaciones académicas realizadas	6
1.1.1 Sistema de Adquisición Automática de Imágenes para Microscopio Óptico	6
1.1.2 Sistema automático para la adquisición y análisis de imágenes de Micobacterias	8
1.2 Equipos comerciales en la Actualidad	10
1.2.1 Microscopio digital LEICA DM6000 B	11
1.2.1.1 Características	11
1.2.2 Microscopio Invertido Motorizado IX81	12
1.2.3 Microscopio Confocal FluoView&Trade 1000	13
1.2.4 Eclipse 90i	15
1.2.4.1 Características	15
<b>CAPÍTULO 2: FUNDAMENTOS TEÓRICOS</b>	18
2.1 Baciloscopía	18
2.1.1 Fundamento Teórico	20
2.1.1.1 Tinción de Ziehl-Neelsen	20
2.1.2 Procedimiento	20
2.1.2.1 Ambiente y Material de trabajo de laboratorio	21
2.1.2.2 Preparación del Extendido	21
2.1.2.3 Coloración (Técnica de Ziehl-Neelsen)	21
2.1.2.4 Observación Microscópica de la muestra coloreada	22
2.1.2.5 Método de Lectura	22

2.1.2.6 Informe de Resultados	22
2.1.3 Aplicaciones	23
2.2 Microscopio Olympus CX31	23
2.2.1 Características y Beneficios	24
2.3 Motor Paso a Paso	25
2.3.1 Principio de funcionamiento	26
2.3.1.1 Motores de Pasos Bipolares	26
2.4 Comunicación Serial	27
2.4.1 RS-232	27
2.5 MATLAB	28
2.5.1 Data Acquisition Toolbox	29
2.5.2 Image Acquisition Toolbox	30
2.6 VMLab	31
<b>CAPÍTULO 3: DISEÑO DE LOS ALGORITMOS DEL SOFTWARE DE CONTROL PARA LA AUTOMATIZACIÓN DEL PROCESO DE ADQUISICIÓN DE IMÁGENES DE MUESTRAS MÉDICAS EN UN MICROSCOPIO ÓPTICO BINOCULAR</b>	<b>32</b>
3.1 Problemática	32
3.2 Objetivos	33
3.2.1 Objetivo General	33
3.2.2 Objetivos Específicos	33
3.3 Planteamiento de la Solución	34
3.3.1 Captura de imagen	36
3.3.2 Desplazamiento a través del Plano XY	37
3.4 Desarrollo de la solución	43
3.4.1 Modo Automático	51
3.4.2 Modo Manual	62
<b>CAPÍTULO 4: EVALUACIONES Y RESULTADOS</b>	<b>66</b>
4.1 Tiempo de Operación del sistema y Cantidad e imágenes fallidas según los parámetros: Tolerancia y Limitador	66
4.2 Valores de Foco promedio por Cantidad de Muestras Totales	70



<b>CONCLUSIONES</b>	74
<b>RECOMENDACIONES</b>	76
<b>BIBLIOGRAFÍA</b>	77



## INTRODUCCIÓN

La constante necesidad de incrementar la productividad en los procesos tanto industriales como de menor envergadura ocasionan que la automatización se convierta en una de las opciones más importantes para la elaboración de un producto o el incremento de la eficiencia de una labor en particular. El desarrollo, mejora y simplificación de los procesos automáticos a lo largo de la historia ha permitido el constante avance en cuanto a calidad de producto y servicio, es por eso que los fundamentos de la automatización, actualmente, son la base de todo proceso, su aplicación en los diversos campos permite una mejora sustancial de la calidad de trabajo y la disminución considerable del tiempo utilizado.

Bajo la premisa de ahorro de tiempo y mejora de la calidad de los procesos es que se elabora este proyecto, que pretende realizar la labor de captura automática de imágenes de una muestra médica para su posterior análisis y conteo de bacilos, para la elaboración de un diagnóstico. Este trabajo consta de cuatro capítulos. El primero comprende los trabajos realizados en otras instituciones educativas y los equipos que existen en el mercado actual, para de esa forma tener un punto de comparación y proyección a posibles alcances. El segundo capítulo define los conceptos y la terminología básica, necesarios para el entendimiento del desarrollo del proyecto en capítulos posteriores. El tercer capítulo abarca el planteo de la problemática y los objetivos a cumplir durante el desarrollo del trabajo. Asimismo, se describe el modo y desarrollo de la solución, es decir, la explicación detallada de los recursos usados para la elaboración del algoritmo de control de Adquisición de imágenes en una muestra



médica, de manera automática. El cuarto capítulo comprende las pruebas realizadas para la verificación del cumplimiento de los objetivos, tanto generales como específicos. Cabe resaltar que las pruebas realizadas verifican el correcto funcionamiento del sistema y la eficiencia de operación del mismo, esto se describe mediante fotografías tomadas a todos los campos de cada muestra y gráficos que detallan, cuantitativamente, la obtención de imágenes correctamente enfocadas.

Finalmente se presentan las conclusiones y posibles alcances del sistema automatizado. En otras palabras, se explica de manera detallada el cumplimiento de los objetivos y se muestran sus posibles alcances.



## CAPÍTULO 1

### ESTADO ACTUAL DE LA ADQUISICIÓN DE IMÁGENES DE MUESTRAS MÉDICAS EN MICROSCOPIOS, DE MANERA AUTOMÁTICA

Este capítulo busca describir los trabajos realizados a lo largo del mundo en busca de la automatización de los microscopios convencionales, para darles un valor agregado y la versatilidad de realizar labores complejas. Las diversas investigaciones realizadas permitieron elaborar productos que actualmente se encuentran en el mercado y que son de vital ayuda en los laboratorios de hospitales internacionales y de gran envergadura. Con esa premisa se parte el desarrollo de este capítulo, pues es necesario conocer las tecnologías actuales para tener un punto de referencia que nos permita evaluar el trabajo que se realiza.

#### 1.1 Investigaciones académicas realizadas:

En esta sección se muestran los trabajos académicos realizados por alumnos y profesores de prestigiosos Centros de Estudio.

##### 1.1.1 Sistema de Adquisición Automática de Imágenes para Microscopio Óptico

El trabajo realizado en la universidad de Santander [1], busca la creación de funciones para la estimación del grado de enfoque de la imagen de un campo en una muestra médica. Asimismo, se realiza el desarrollo de un algoritmo de autoenfoco para una platina motorizada en el eje Z, y otro algoritmo que realice el barrido automático de la

muestra en el plano X-Y. La resolución mecánica a la que se trabaja se muestra en tabla1.1.

Tabla1.1 Cuadro de resolución por grados de los motores de pasos usados [1]

EJE	RESOLUCIÓN MOTOR (GRADOS/PASO)	RESOLUCIÓN DESPLAZAMIENTO ( $\mu\text{M}$ /PASO)
X	1.8	23
Y	3.6	52
Z	1.8	0.4

La parte mecánica del sistema está basada en motores que transmiten el movimiento a través de poleas reductoras y el uso de bandas rígidas acopladas a las perillas del microscopio. El control de los motores se hace a través del puerto paralelo de la PC y la adquisición de imágenes se hace mediante el uso de una cámara web USB de 640 x 480 píxeles [1].

Para el análisis de enfoque de la imagen capturada, se toma en cuenta la información de los píxeles de la imagen y a través de diversos algoritmos predefinidos se evalúa cual es el que mejor analiza la imagen. Se plantean algoritmos como: Conteo de gradiente Absoluto (CGA), Entropía (ENT), Transformada de Fourier (TFU), F5 de Vollat (F5V), Laplaciano (LAP), etc. [2] (Ver Tabla 1.2) arreglos matemáticos que permiten obtener un valor dependiendo de la claridad y nitidez de la imagen.

Tabla 1.2 Algoritmos para la estimación del grado de enfoque de una imagen [1]

ALGORITMO	TÉCNICA
Conteo del gradiente absoluto (CGA)	Diferenciación de imágenes
Entropía (ENT)	Histograma
Transformada de Fourier (TFU)	Representación en el dominio de la frecuencia
F5 de Vollat (F5V)	Medida de autocorrelación
Laplaciano (LAP)	Detección de bordes
Suma de la magnitud de la diferencia (SMD)	Diferenciación de imágenes
Gradiente al cuadrado (SG)	Diferenciación de imágenes
Gradiente absoluto con umbral (TAG)	Diferenciación de imágenes
Tenengrad (TEN)	Detección de bordes
Tenengrad modificado (TEN8)	Detección de bordes
Varianza del gradiente modificada (VDG8)	Detección de bordes
Varianza (VAR)	Contraste de la imagen

### 1.1.2 Sistema automático para la adquisición y análisis de imágenes de Micobacterias

Este trabajo realizado en la Universidad Autónoma de México [2], presenta un algoritmo desarrollado en lenguaje Java, que permite la automatización funcional de un microscopio de fluorescencia (Ver figura 1.1) y el procesamiento de las imágenes adquiridas campo por campo. El diseño del sistema es modular, es decir, formado por partes definidas que trabajan en conjunto, cada una con cierto trabajo en específico, ello permite el desarrollo de cada parte y el trabajo cooperativo en grupo para el desarrollo de un mejor sistema.



Fig. 1.1 Vista general de un microscopio de fluorescencia [2]

El tipo de Microscopio que se desea automatizar es el de fluorescencia. Este tipo de microscopio es muy utilizado en el estudio de organismos orgánicos e inorgánicos, se basa en la radiación electromagnética que emiten ciertas moléculas al ser excitadas. La excitación del objeto de estudio se logra mediante la aplicación de una molécula llamada fluoróforo que absorbe energía con una longitud de onda determinada y emite dicha energía con otra longitud de onda que depende del ambiente en el que se encuentra la molécula y las características propias de la molécula. El problema radica en capturar las imágenes a una velocidad relativamente alta, ya que el tiempo de “brillo” es muy pequeño, es decir, conforme pasa el tiempo la fluorescencia se va degradando (*photobleaching*) [2] (Ver figura 1.2). Otro problema que se presenta es la búsqueda de un plano focal óptimo, debido a que hay algunos organismos que se desplazan en distintos planos focales, haciendo difícil su ubicación y enfoque.

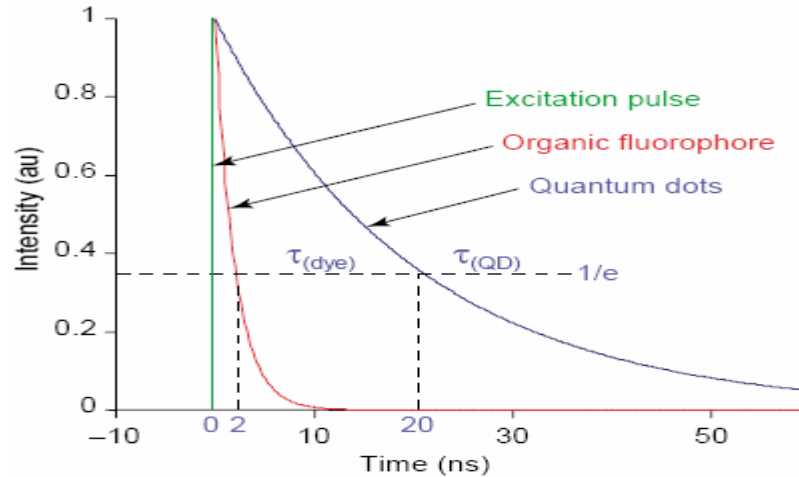


Fig. 1.2 Diagrama intensidad vs. Tiempo en donde se muestra como la fluorescencia se degrada conforme el tiempo avanza.

Fuente: <http://bme240.eng.uci.edu/students/07s/yokabe/advant3.gif>

El modo de solución que se plantea se divide en dos secciones consecutivas, el primero es el de motorizar las perillas de desplazamiento en eje X-Y-Z para un movimiento preciso; la segunda parte consta en el desarrollo de un software que controle todas las partes motorizadas. Asimismo, para la captura de imágenes se usó una cámara de alta resolución.

El control de los motores se hace a través del puerto RS232C (serial) y el manejo de la cámara de 3.3MP es a través de conexión IEEE 1394 (Firewire) [2]. El control total del sistema se basa principalmente en la construcción de plugins bajo lenguaje de programación Java. Un plugin es un pequeño pedazo de código que contribuye a la codificación de un algoritmo mayor; de esta forma el sistema se hace modular ya que se pueden ir agregando más plugins para mejorar la eficiencia del sistema y así aumentar su complejidad. (Ver figura 1.3)

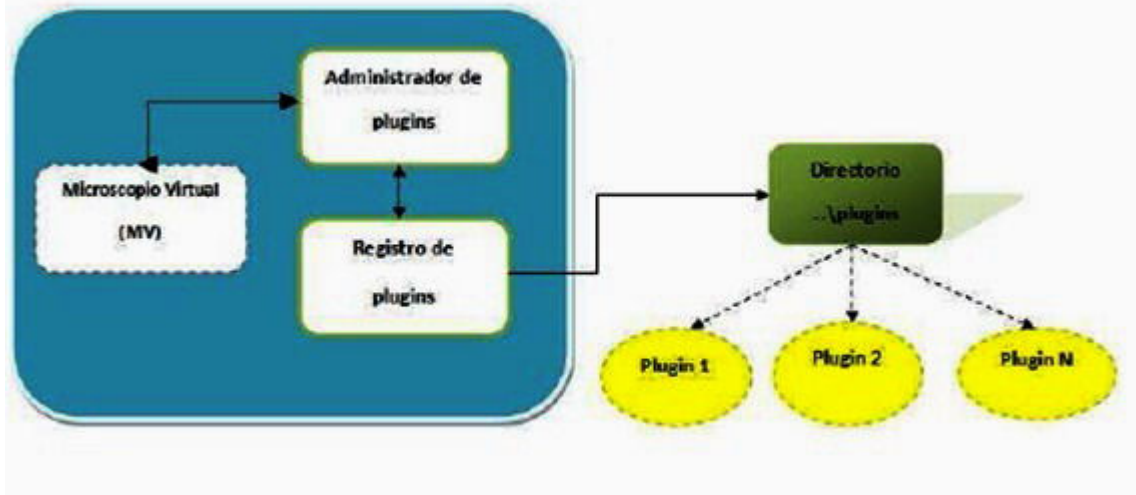


Fig. 1.3 Diagrama de bloques del algoritmo de programación basado en plugins [2]

## 1.2 Equipos comerciales en la Actualidad:

A lo largo de los últimos años, se han realizado bastas innovaciones en el campo de la microscopía, ello ha conllevado a tener sistemas cada vez más complejos para el análisis de organismos orgánicos e inorgánicos y un desarrollo de alta tecnología que nos permite tener diagnósticos rápidos con una alta calidad de nitidez y resolución.

A continuación se muestran los equipos comerciales más resaltantes en el mercado actual, sus características y alcances.



### 1.2.1 MICROSCOPIO DIGITAL LEICA DM6000 B



Fig. 1.4 Microscopio Leica DM6000 B

Fuente:

[http://www.microscopy.uk.com/microscopes/leica\\_microscopes/leica\\_DM6000B\\_microscope.html](http://www.microscopy.uk.com/microscopes/leica_microscopes/leica_DM6000B_microscope.html)

#### 1.2.1.1 Características:

- Automatización Vertical (eje Z)
- Cámaras fotográficas *fireware*.
- Software especial para el análisis y archivo de la imagen capturada.
- Ajuste automático de la transmisión de luz, de manera que se puede trabajar en ambientes donde haya variaciones repentinas de luz o en cualquier ambiente oscuro.
- Eje de fluorescencia automatizado.
- Platina motorizada.
- Control del sistema a través del tacto de la pantalla “touch” LeicaScreen control de desplazamiento suave “SmartMove”.

### 1.2.2 Microscopio Invertido Motorizado IX81

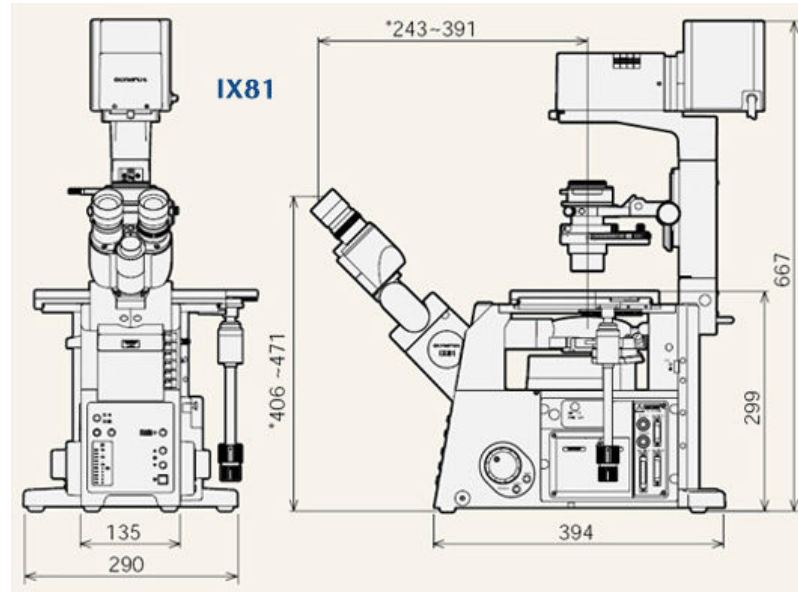


Fig. 1.5 Dimensiones del Microscopio Motorizado IX81 de la Compañía Olympus

Fuente:

[http://www.olympuslatinoamerica.com/spanish/seg/seg\\_product\\_detail\\_esp.asp?q=840&d=2&s=14&c=55](http://www.olympuslatinoamerica.com/spanish/seg/seg_product_detail_esp.asp?q=840&d=2&s=14&c=55)

El modelo IX81 de la compañía Olympus (Ver figura 1.5), es lo más avanzado actualmente en microscopía invertida motorizada, el estativo (soporte general del microscopio) proporciona acceso hasta a 9 puertos de entrada o salida para diversos dispositivos, por ejemplo, se puede capturar una foto y varios puertos pueden tener acceso a la imagen obtenida.

Posee un eje Z motorizado, revólver de 6 posiciones y un selector de eje de luz, con el cual se puede trabajar bajo cualquier condición sin mostrar la menor variación en la nitidez de la imagen obtenida; se puede trabajar bajo múltiples longitudes de onda, fluorescencia o deconvolución (distorsión de la imagen). Asimismo permite un ajuste automático de la lente del eje z a través del motor de alta resolución, con pasos de hasta  $0.01\mu\text{m}$ . Por otro lado, posee un iluminador recto mejorado con un 20% más de brillantez. Asimismo, el estativo permite la adaptación del bloque de barrido

automático del modelo FluoView™500, de esta manera se realiza el desplazamiento inteligente a través de la platina que sujeta a la muestra a analizar.

### 1.2.3 Microscopio Confocal FluoView&Trade 1000



Fig. 1.6 Microscopio Confocal FlouView&TRade 1000 de la Compañía Olympus

Fuente: <http://www.olympusfluoview.com/>

Este equipo de generación de imágenes (Ver figura 1.6), está diseñado para la observación confocal de células vivas, ello implica el uso de fluorescencia y barrido láser para la excitación de los organismos vivos y su observación mediante el grado de luminosidad que se obtenga. Este modelo presenta mejoras sustanciales con respecto al desarrollo del sistema confocal y la velocidad de procesamiento de la imagen, así como la sensibilidad óptima para captar imágenes de células vivas sin dañar la estabilidad de la muestra.

Además, este sistema presenta varias innovaciones como:

#### **Sistema SIM SCANNER:**

Esta tecnología combina el trabajo de dos láseres, el primero realiza la excitación de la muestra para obtener la fluorescencia necesaria y el segundo realiza la emisión de la

imagen. El barrido del primer láser genera una alta resolución de la imagen y el segundo simplemente la muestra. De manera que se obtiene una imagen clara tomando en cuenta que la fluorescencia decae con el paso del tiempo, entonces si se tiene una acción de emisión de imagen casi simultánea a la acción de excitación de la muestra, se logra una calidad óptima de la imagen.

### **Sistema de Barrido Espectral:**

El diseño del sistema consiste en el uso de dos tubos de detección espectral, ambos independientes, cada uno con una red de difracción y una ranura variable que permite la división de longitudes de onda de alta resolución y una selección del ancho de banda de alta velocidad. Estas características permiten la división, con un alto grado de precisión, de las señales superpuestas de una emisión fluorescente.

### **Alta sensibilidad**

Permite la detección inmediata de fluorescencia en la muestra, incluso con láseres de baja intensidad, lo que previene el daño de las células vivas analizadas.

### **Gran Precisión y Velocidad:**

- Control ultra-preciso de la intensidad del láser para evitar daños en las células vivas.
- El sistema espectral proporciona una resolución de longitud de onda de 2nm.
- Alta velocidad de Generación de imágenes: 16 cuadros por segundo.
- Velocidad de Espectroscopia: 1ms/100nm.
- Motor de pasos interno, totalmente automatizado que permite un desplazamiento ultrasensible de hasta 0.01 micrones.

**Barrido Automático en tres dimensiones**, de tal manera que se analiza la muestra en su totalidad, obteniendo imágenes ultra-nítidas. Mediante el uso de un software propio de la compañía Olympus llamado **FluoView** se puede tener el completo control del sistema desde la PC con una pre-visualización de la imagen enfocada, que permite observar en tiempo real el desarrollo del barrido de la muestra.

## 1.2.4 Eclipse 90i



Fig. 1.7 Entorno de trabajo Eclipse 90i de la Compañía Nikon

Fuente: <http://www.tissuegnostics.com>

El microscopio **ECLIPSE 90i** de la compañía Nikon (Ver figura 1.7), es un sistema completamente equipado para cualquier prestación dentro de la microscopía digital. Este sistema está apto para trabajar bajo cualquier condición, ante las aplicaciones de mayor demanda.

### 1.2.4.1 Características:

**Lente Exclusivo “ojo de mosca”:** innovador arreglo de lentes para lograr una iluminación uniforme a través del campo de visión. (Ver figura 1.8)



Fig. 1.8 Lentes “Ojo de mosca” aplicado en el modelo Eclipse 90i

Fuente: [http://tecnicenlaboratorios.com/Nikon/Eclipse\\_90i\\_Clave.htm](http://tecnicenlaboratorios.com/Nikon/Eclipse_90i_Clave.htm)



**Objetivos Plan Apo VC:** estos lentes corrigen la distorsión cromática que se presente en un enfoque mal realizado, de tal manera que siempre se obtengan imágenes con excelente resolución. (Ver Figura 1.9)



Fig. 1.9 Objetivos Apo VC del modelo Eclipse 90i (Nikon)

Fuente: [http://tecnicenlaboratorios.com/Nikon/Eclipse\\_90i\\_Clave.htm](http://tecnicenlaboratorios.com/Nikon/Eclipse_90i_Clave.htm)

**Iluminador Universal de Epí-fluorescencia:** diseñado por la compañía Nikon, este eliminador de ruido permite que los especímenes de poca fluorescencia sean captados nítidamente. (Ver figura 1.10)

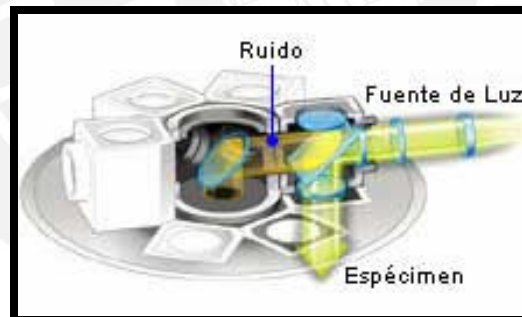


Fig. 1.10 Diagrama explicativo del iluminador universal del modelo Eclipse 90i (Nikon)

Fuente: [http://tecnicenlaboratorios.com/Nikon/Eclipse\\_90i\\_Clave.htm](http://tecnicenlaboratorios.com/Nikon/Eclipse_90i_Clave.htm)

**Enfoque motorizado de Alta Precisión:** sistema con retroalimentación que monitorea constantemente el punto de ubicación de la platina en el eje Z. Su movimiento es muy fino: 50 nanómetros por paso del motor que transmite el movimiento a la platina.

**Control Motorizado en confort:** el programa desarrollado por la misma compañía Nikon: **iControl**, permite manipular el eclipse 90i a través de un PC, mediante un



interfaz gráfica que muestra la navegación a través de la muestra y los comandos específicos para los movimientos que se deseen hacer en 3 dimensiones.

**Auto-ajuste:** Según el objetivo que se utilice, se ajustan automáticamente los diafragmas de apertura y campo, los filtros, ubicación de la platina motorizada, velocidad de enfoque, sensibilidad, etc.



## CAPÍTULO 2

### FUNDAMENTOS TEÓRICOS

En este capítulo se describen las bases conceptuales que permitirán un mejor entendimiento de los capítulos siguientes, pues se explican detalladamente los conceptos que implican el desarrollo de la solución a la problemática planteada al inicio de este trabajo: La adquisición automática de imágenes en una muestra médica, a través de un microscopio óptico binocular.

Es necesario, en un principio, explicar la técnica que se utiliza para llevar a cabo el proceso de conteo y reconocimiento de los bacilos. Esta labor se realiza convencionalmente en un laboratorio de baciloscopía por lo que se sigue un procedimiento muy detallado.

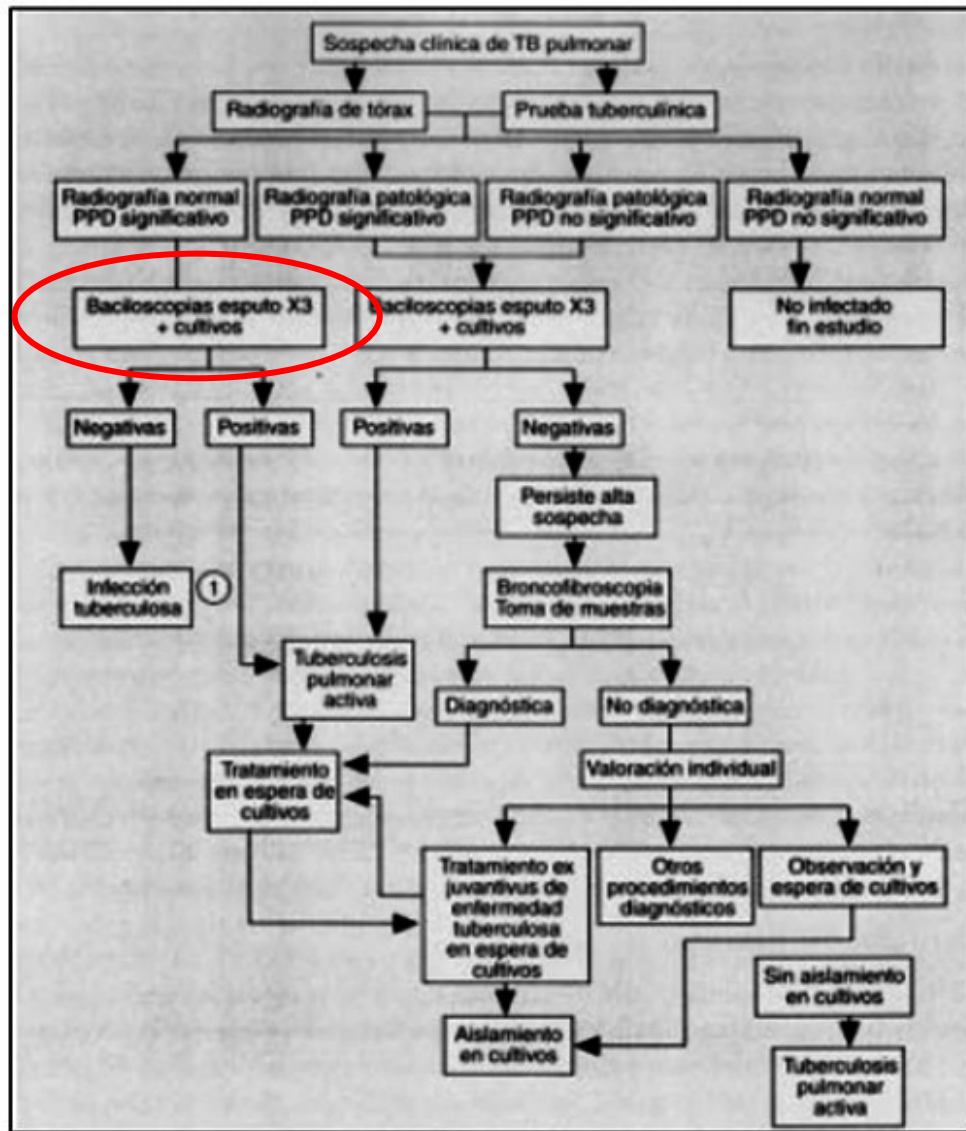
#### 2.1 Baciloscopía

La baciloscopía es una técnica fundamental en toda investigación bacteriológica ya que posee un alto grado predictivo para el diagnóstico. Consiste básicamente en la observación microscópica de una muestra médica para el conteo de bacilos tratados previamente mediante la tinción de Ziehl-Neelsen, técnica que permite la identificación de los bacilos mediante la coloración que adquieren al ser expuestos a un determinado tipo de alcohol. La observación se realiza campo a campo; entiéndase por **campo** a cada cuadro en la muestra, enfocado por la lente del microscopio, que va a ser analizado. Esta técnica es recomendada por la Organización Mundial de la Salud

(OMS) y la Unión Internacional Contra la Tuberculosis y Enfermedades Respiratorias (UICTER) debido a que asegura resultados fiables, con un entrenamiento sencillo y de manera económica.

La baciloscopía, al ser una técnica con un alto grado de credibilidad, ocupa un lugar primordial entre diversas técnicas de detección de enfermedades como la tuberculosis, como se observa en la figura 2.1. Su rapidez de operación permite tomar medidas preventivas y de acción para la recuperación del paciente en el caso de una posible infección.

Fig. 2.1 Diagrama de Evaluaciones Secuenciales para el diagnóstico de TBC pulmonar



## 2.1.1 Fundamento Teórico

### 2.1.1.1 Tinción de Ziehl-Neelsen

Las micobacterias como *M. Tuberculosis* y *M. marinum* y los parásitos coccídeos como *Cryptosporidium* poseen una característica peculiar, las paredes celulares de estas bacterias contienen ácidos grasos, por ejemplo los ácidos micólicos, que les confieren la propiedad de ser ácido-alcohol resistente; es decir, una vez sido expuestos a la influencia de un colorante primario como la fucsina de Ziehl y a un calentamiento posterior para la fijación del color respectivo, sus paredes celulares adquieren la propiedad de resistir la decoloración con alcohol-ácido. Debido a ello, a este tipo de bacterias se les denomina **ácido-alcohol resistentes** (BAAR). Esta propiedad permite la fácil distinción de las bacterias ante un fondo que contraste con el color adquirido mediante la coloración como se muestra en la figura 2.2. De este procedimiento es el que se vale la baciloscopía para realizar el conteo respectivo de los bacilos y dar un diagnóstico inmediato y económico; inmediato debido a la sencillez del procedimiento y económico porque, con respecto a otras técnicas, esta implica un gasto mínimo en el uso de materiales.

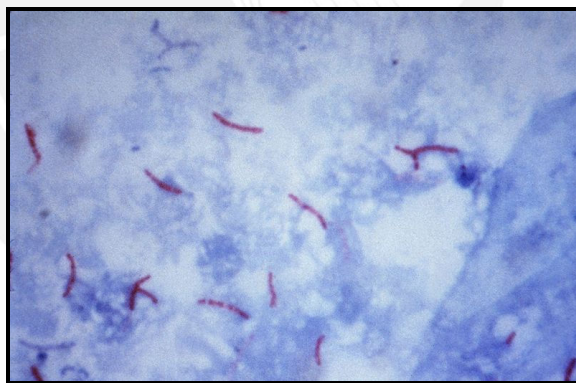


Fig.2.2 Bacilos de *M. Tuberculosis* después de aplicada la tinción de Ziehl-Neelsen

## 2.1.2 Procedimiento

La baciloscopía implica un procedimiento específico, pasos que se tienen que seguir para garantizar la seguridad del laboratorista y el correcto manejo de las herramientas para la generación de un diagnóstico fiable. Asimismo se toma en cuenta las condiciones ambientales que deben tener las habitaciones y las características técnicas de las herramientas usadas [3].

### 2.1.2.1 Ambiente y Material de trabajo de laboratorio

En este aspecto, se recomienda siempre tener un ambiente de tamaño considerable para el laboratorio, más aún si en el servicio de salud en donde se encuentra hay una gran afluencia de público. Asimismo, es necesario recordar que siempre se deben aplicar las normas de bioseguridad, para evitar riesgos al manipular potenciales muestras infectantes. Además, debe haber una adecuada distribución del laboratorio, en donde cada instrumento o herramienta debe ocupar un lugar determinado, de la misma forma, cada tipo de procedimiento debe ser hecho en un espacio idóneo que cumpla con las normas de bioseguridad. [3]

### 2.1.2.2 Preparación del Extendido

Antes de la realización de cualquier procedimiento que pueda comprometer la salud del laboratorista, es necesario, que este se lave las manos y que use un guardapolvo de protección, lo más largo posible. Luego, recordar que el procedimiento se debe realizar en un orden estricto, procesando series de cómo máximo 12 muestras [3].

El proceso de extendido se realiza sacando la muestra con un “bajalengua” y poniéndola suavemente sobre el portaobjetos, luego se hace un movimiento de vaivén a fin de extender la muestra, una vez extendida la muestra se procede a calentar los bordes del portaobjetos con un mechero, para luego, dejarlo secar a temperatura ambiente.

### 2.1.2.3 Coloración (Técnica de Ziehl-Neelsen)

La famosa técnica de coloración, es la más recomendada para el diagnóstico de enfermedades causadas por micobacterias. Su procedimiento es sencillo y rápido:

Una vez obtenidas las muestras secas se procede a cubrir la totalidad del portaobjetos con el colorante *fucsina básica fenicada filtrada* luego, se procede a calentar suavemente la muestra colorada, una vez que este seca, se le echa agua a baja presión para eliminar los restos de colorante [3].

Luego se procede a la decoloración, esta se realiza con ácido alcohol, con lo que se elimina el color fucsia de la muestra y solo los bacilos quedan pintados debido a la

propiedad que líneas arriba ya se explicó. Sin embargo, esto no es suficiente, pues se necesita un fondo que contraste con el color de los bacilos, para ello se usa azul metileno previamente filtrado que se distribuye por toda la muestra, y se deja secar a temperatura ambiente para que luego, la muestra final ya procesada totalmente sea analizada con el microscopio.

#### 2.1.2.4 Observación Microscópica de la muestra coloreada

La observación microscópica, tiene dos objetivos importantes:

- a. Determinar si en el extendido hay Bacilos Ácido Alcohol Resistentes (BAAR).
- b. Establecer su número aproximado.

#### 2.1.2.5 Método de Lectura

Los bacilos tienen forma de pequeños bastoncitos delgados, ligeramente curvos, todos de color rojo, algunas veces no están aislados, sino más bien forman grupos o cúmulos que dejan ver una mancha roja granulada.

El operario debe notar el número de bacilos en cada campo que analiza, para luego dar un promedio que conllevará a la generación de un diagnóstico.

#### 2.1.2.6 Informe de Resultados

Tabla 2.1 Informe de resultados bajo la terminología de cruces, según la cantidad de bacilos encontrados en la muestra médica (de TBC).

Resultado del examen microscópico	Informe
No se encuentran BAAR en los 100 campos observados	<b>No se observan bacilos ácido-alcohol resistentes</b>
Se observan de 1 a 9 BAAR en 100 campos observados	<b>Nº exacto de bacilos en 100 campos</b>
Se observa entre 10 y 99 BAAR en 100 campos observados	<b>Positivo (+)</b>
Se observan de 1 a 10 BAAR por campo en 50 campos observados	<b>Positivo (++)</b>
Se observan más de 10 BAAR por campo en 20 campos observados	<b>Positivo (+++)</b>



### 2.1.3 Aplicaciones:

Los procedimientos que implica la baciloscopía permite determinar enfermedades como: **lepra, tuberculosis, abscesos de piel, linfadenitis, osteomielitis y otras afecciones generalizadas**, pero con un alto porcentaje de afección pulmonar.

### 2.2 Microscopio Olympus CX31

El modelo Olympus CX31 es un microscopio óptico binocular de uso médico y educacional (Ver figuras 2.3 y 2.4). La importancia de este equipo es que en base a sus dimensiones y características se realiza el diseño del sistema automático.

El CX31 es un microscopio de campo claro completo que incluye un tubo de observación binocular con objetivos de 4X, 10X, 40X y uno opcional de 100X de inmersión en aceite, esta opción permite su uso en la baciloscopía.

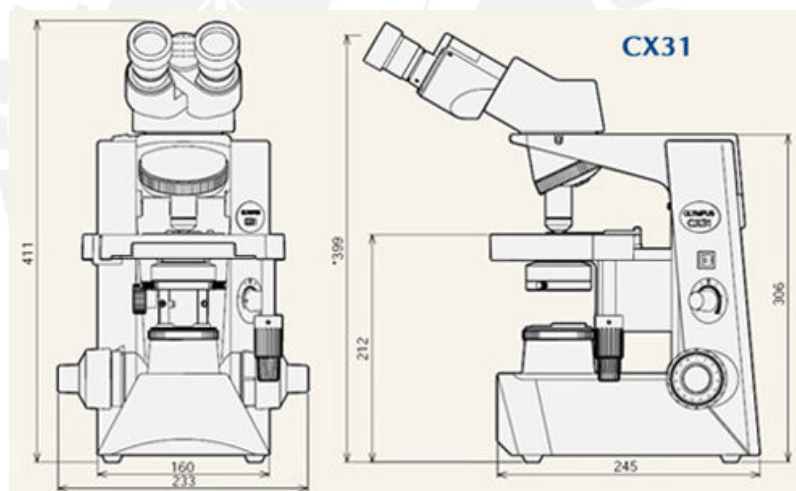


Fig. 2.3 Dimensiones del Microscopio Olympus CX31

Fuente:

[http://www.olympuslatinoamerica.com/spanish/seg/seg\\_product\\_detail\\_esp.asp?q=716&s=11&c=53&d=2](http://www.olympuslatinoamerica.com/spanish/seg/seg_product_detail_esp.asp?q=716&s=11&c=53&d=2)

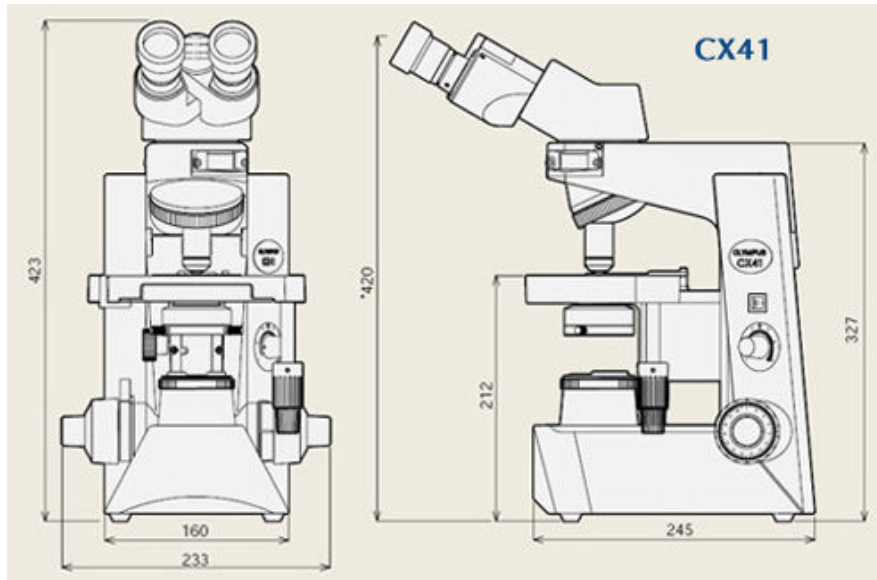


Fig. 2.4 Dimensiones del Microscopio Olympus CX31

Fuente:

[http://www.olympuslatinoamerica.com/spanish/seg/seg\\_product\\_detail\\_esp.asp?g=716&s=11&c=53&d=2](http://www.olympuslatinoamerica.com/spanish/seg/seg_product_detail_esp.asp?g=716&s=11&c=53&d=2)

### 2.2.1 Características y Beneficios

Estos microscopios son sólidos, fáciles de operar y confiables para el uso habitual en clínica y educación.

- Los objetivos Plan C Acromáticos corregidos al infinito y desarrollados recientemente ofrecen una uniformidad de visión excepcional produciendo imágenes claras y nítidas hasta el mismo borde del campo visual; (4x, 10x, 40x, 100x aceite)
- Oculares de F.N. 20 (el CX41 puede ser equipado con oculares de F.N. 20 ó 22)
- Tratamiento anti-hongos que ayuda a proteger las partes ópticas aún en regiones de alta humedad
- El revólver portaobjeto orientado hacia adentro cuádruple (CX31) mejora el acceso y facilita el cambio rápido del portaobjeto
- Condensador Abbe fijo de A.N. 1.25, centrable y enfocable para la iluminación Köhler

- Un bombillo halógeno de 6V/30W provee iluminación amplia para cualquier observación.
- El estativo diseñado ergonómicamente tiene los tornillos micro y macrométricos ubicados en una posición baja, una ranura para el analizador de luz polarizada lineal (sólo en el CX41) y mangos en el frente y la parte posterior para el transporte; tope de campo integrado
- Platina integrada con mandos a izquierda o derecha diseñado sin cremallera (CX31: solo derecha), controles X-Y posicionados bajos con empuñaduras ergonómicas para facilitar la operación.

### 2.3 Motor Paso a Paso

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos. En esta ocasión necesitamos mover pequeñas distancias entre campo y campo para adquirir las imágenes, así como para realizar el enfoque automático de la lente de inmersión, por ello se escogen este tipo de motores.

La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde  $90^\circ$  hasta pequeños movimientos de tan solo  $1.8^\circ$ , es decir, que se necesitarán 4 pasos en el primer caso ( $90^\circ$ ) y 200 para el segundo caso ( $1.8^\circ$ ), para completar un giro completo de  $360^\circ$ . En este caso el paso que tienen los motores a usar son de 7.2 grados permitiendo una resolución de 50 movimientos por giro completo ( $360^\circ$ ).

Estos motores poseen la característica de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizadas, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas. La propiedad de enclavamiento permite al sistema una mayor precisión pues al no haber movimiento alguno o instrucción enviada, este quedará en un lugar fijo y no habrá pequeños deslizamientos que generen variación con respecto a los puntos de referencia.

### 2.3.1 Principio de funcionamiento

#### 2.3.1.1 Motores de Pasos Bipolares

Existen diversos tipos de motores de pasos; sin embargo, en este caso, explicaremos el funcionamiento de los **motores de pasos bipolares**, por ser estos los más sencillos de controlar. Por tal razón se usan para el desarrollo de este trabajo.

Básicamente los motores de pasos Bipolares, están constituidos por un rotor que tiene adheridos imanes permanentes, y un estator conformado por cierto número de bobinas excitadoras (bobinados independientes).

Este tipo de motores normalmente posee cuatro salidas (Ver figura 2.8), por donde se ingresarán las señales que le permiten realizar el giro de un paso. Para ello se requiere de un puente H por cada bobina de motor, para realizar la inversión de sentido de corriente.

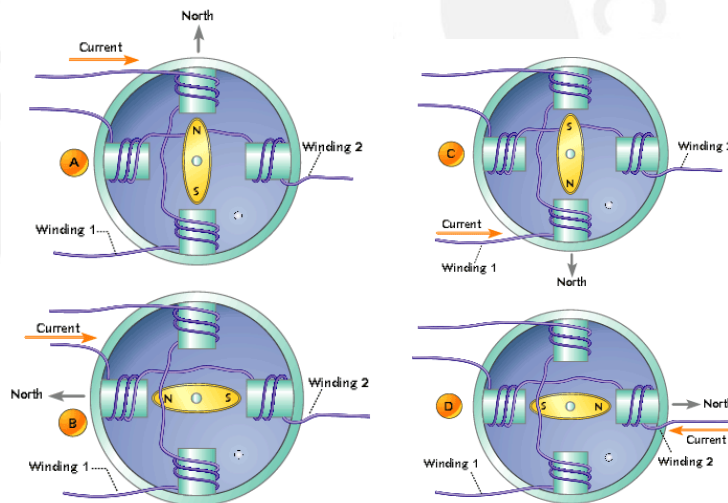


Fig. 2.8 Bobinado de un motor bipolar básico

Fuente: <http://www.forosdeelectronica.com/f19/introduccion-motores-paso-paso-289/>

Cada inversión del sentido de la corriente genera el giro de un paso en el motor. El sentido de giro depende del paso siguiente o del paso anterior. A continuación se muestra en la tabla la secuencia de pasos de un motor bipolar.

## 2.4 Comunicación Serial

La comunicación serial es un protocolo para comunicación entre dispositivos, muy difundido en los últimos años, debido a su facilidad de entendimiento y a las largas distancias que permite su implementación. El puerto serial RS-232 es el más común en las computadoras de escritorio, mediante este puerto se envían y reciben bytes (8 bits) de información un bit a la vez; si bien este tipo de comunicación es ligeramente más lento que una comunicación en paralelo, que permite una comunicación un byte a la vez, este tipo de comunicación es más sencillo y permite alcanzar mayores distancias de operación, por esas razones, principalmente por su sencillez, es que se elige este tipo de comunicación para la elaboración del proyecto.

### 2.4.1 RS-232

RS-232 (Estándar ANSI/EIA-232) es el tipo conector serial que se usan en las PCs IBM y compatibles (Ver figura 2.9). Este tipo de conector es utilizado en diversos dispositivos y aplicaciones, como conectar una impresora o un mouse, incluso en instrumentación industrial. Este tipo de conector ha sido desarrollado a lo largo de los años, permitiendo mejoras en su desempeño y velocidad de transmisión. RS-232 tiene una limitación, solo permite comunicaciones de punto a punto entre los dispositivos y el puerto serial de la computadora. El hardware de RS-232 se puede utilizar para comunicaciones seriales en distancias de hasta 15 metros.

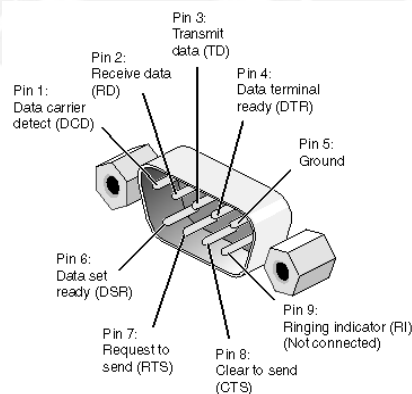


Fig. 2.9 Conector DB-9

Fuente: <http://iaci.foroactivo.com/microcontroladores-f9/problema-con-los-puertos-serie-para-hc11-t19.htm>

## 2.5 MATLAB

**MATLAB** (abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado (IDE) (Ver figura 2.10) con un lenguaje de programación propio (lenguaje M).

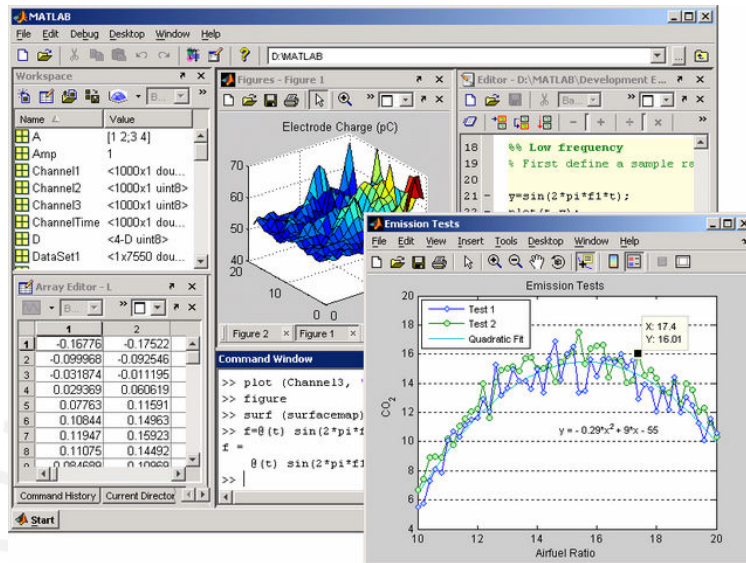


Fig. 2.10 Entorno de trabajo de MATLAB

Fuente: <http://www.mathworks.es/products/matlab/>

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

Las características resaltantes y de importancia para la elaboración del trabajo en cuestión son: la implementación de algoritmos que describirán la secuencia lógica de control de todo el sistema, comunicación con otros dispositivos hardware, lo cual permite el fácil el envío de instrucciones hacia los motores que se usarán a través de una comunicación vía puerto serial teniendo en cuenta el protocolo de comunicación respectivo que es el RS-232.



Por otro lado, este software posee los llamados “Toolbox” que son paquetes de instrucciones para fines específicos. Para el desarrollo del trabajo se usarán dos paquetes: Data Acquisition Toolbox, Image Acquisition Toolbox.

### 2.5.1 Data Acquisition Toolbox

El software de Adquisición de datos de Matlab provee al usuario de un completo set de herramientas para la entrada y salida tanto analógica como digital para una variedad de hardware de adquisición de Datos compatible con la PC [4].

Este set de herramientas permite configurar dispositivos externos y poder leer la información captada por ellos para un posterior procesamiento dentro del programa principal MATLAB® o a través de un ambiente gráfico llamado Simulink®.

Se puede visualizar la data adquirida, procesarla, realizar pruebas y ver los resultados bajo un entorno amigable y de fácil entendimiento, a través de comandos especializados para labores específicas. (Ver figura 2.11)

Esta herramienta nos ayudara, en el desarrollo del proyecto, ha enviar instrucciones determinadas para el manejo de motores que regirán los movimientos en tres dimensiones del sistema, tanto para el barrido de los campos dentro de la muestra como para el enfoque de la imagen.

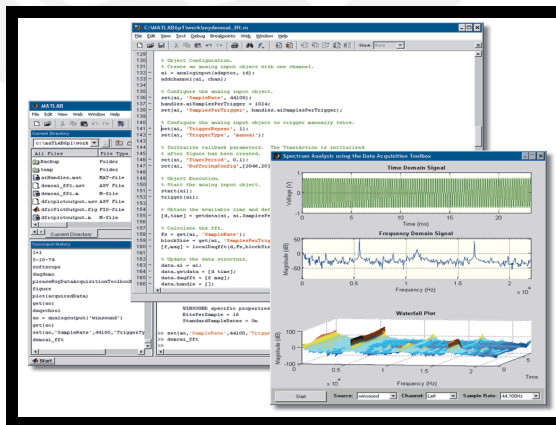


Fig. 2.11 Entorno de Trabajo mediante la caja de herramientas de Adquisición de Datos de MATLAB

Fuente: <http://www.mathworks.es/products/daq/>

### 2.5.2 Image Acquisition Toolbox

Este set de herramientas permite la captura de imágenes y video directamente al programa MATLAB® y al entorno gráfico Simulink® dependiendo de la compatibilidad del hardware compatible con PC usado [5].

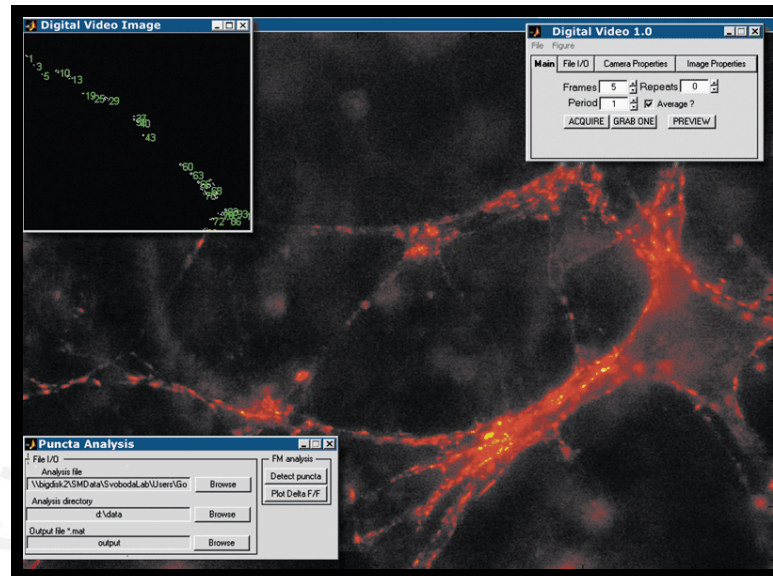


Fig. 2.12 Análisis de imagen mediante la caja de herramientas de Adquisición de imágenes de MATLAB

Fuente: <http://www.humusoft.cz/produkty/matlab/aknihovny/imaq/>

El reconocimiento es automático, y la configuración de las propiedades del hardware permite una mayor versatilidad para el trabajo que se quiera realizar. Asimismo, se puede tener una muestra previa de la imagen adquirida ya sea como imagen estática o a través de video, para un posterior procesamiento. (Ver figura 2.12)

En este caso en particular, se usa el toolbox de adquisición de imágenes para lograr el manejo de la cámara que captará las imágenes de los diversos campos de la muestra analizada, y las mostrará en tiempo real. Por otra parte, el procesamiento que se realiza a la imagen para cuantizar el grado de nitidez (enfoco) que se tenga, también se realiza gracias a este set de herramientas.

## 2.6 VMLab

VMLab es un software que permite la programación de microcontroladores en lenguaje ensamblador y la realización de simulaciones mediante su panel de control, osciloscopio virtual y Panel de comunicación serial. (Ver figura 2.13)

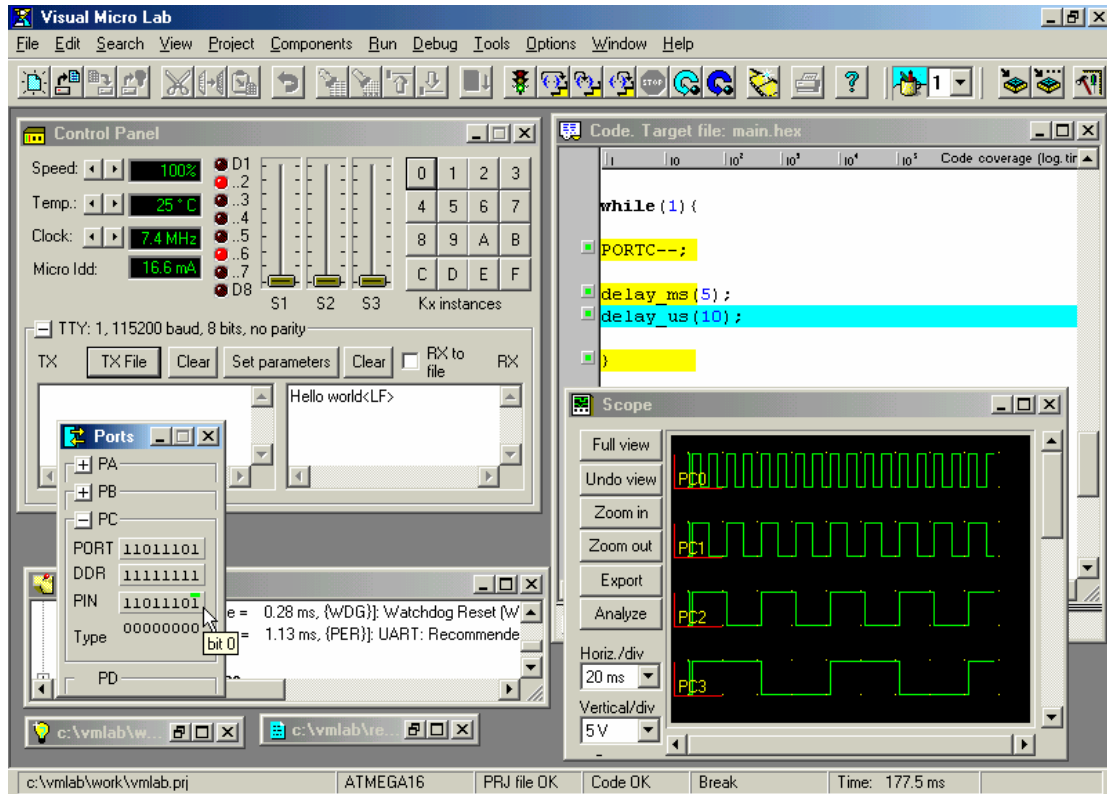


Fig. 2.13 Entorno de trabajo de VMLab

Fuente:

[https://writer.zoho.com/ImageDisplay.im?name=17231251163\\_70884100000002007\\_78860111.032.png&acclid=70884100000002007](https://writer.zoho.com/ImageDisplay.im?name=17231251163_70884100000002007_78860111.032.png&acclid=70884100000002007)

## CAPÍTULO 3

### DISEÑO DE LOS ALGORITMOS DEL SOFTWARE DE CONTROL PARA LA AUTOMATIZACIÓN DEL PROCESO DE ADQUISICIÓN DE IMÁGENES DE MUESTRAS MÉDICAS EN UN MICROSCOPIO ÓPTICO BINOCULAR

#### 3.1 Problemática

El uso de un microscopio óptico binocular para el análisis de muestras médicas requiere de cierto entrenamiento para tener la capacidad de poder manejarlo eficientemente. Sin embargo, ello depende de cuan meticuloso y preciso se quiera ser. Análisis de mayor precisión demandan un mayor tiempo de operación, y con ello un consumo de recursos que implican tiempo y dinero. Lo que se pretende con este trabajo es buscar la manera de hacer el proceso de análisis de las muestras de forma autónoma, tomando en cuenta los requerimientos mínimos que implica la operación convencional para la elaboración de un diagnóstico confiable.

Se identifican tres actividades principales para el proceso de análisis de imágenes en una muestra médica a través de la técnica de la baciloscopía:

- Constante seguimiento y visualización del objetivo a través de la lente del microscopio óptico binocular.

- Ajuste del acercamiento de la lente para el correcto enfoque de la imagen.
- Desplazamiento en el plano de la muestra, a través de puntos determinados y espaciados de manera uniforme para la visualización correcta de cada cuadro de imagen captada por la lente.

Estas tres actividades ordenadas y concatenadas convenientemente constituyen el proceso de conteo de bacilos por simple inspección a lo largo de toda la muestra llamado baciloscopía. Lo resaltante de todo este procedimiento es su repetitividad a través de diferentes muestras, ya que el proceso siempre es el mismo. Dicha lógica secuencial y repetitiva conlleva al planteamiento del diseño de un sistema que sea capaz de realizar la misma labor de adquisición de imágenes pero de manera automática, cumpliendo con los requerimientos del proceso manual para luego realizar el conteo respectivo de bacilos en la muestra total.

### **3.2 Objetivos**

#### **3.2.1 Objetivo General**

El objetivo principal de este trabajo es el diseño e implementación de los algoritmos del software de control de un sistema automático de adquisición de imágenes de una muestra médica a través de un microscopio óptico binocular.

Para cumplir con dicho objetivo principal, se plantean objetivos específicos que dividen el trabajo para una mejor distribución de la labor total.

#### **3.2.2 Objetivos Específicos:**

- Controlar una Cámara Web para la captura de imágenes, con una resolución de 640 x 480 píxeles para la obtención imágenes con un grado de nitidez aceptable por el laboratorista.
- Lograr un desplazamiento coordinado en 3 dimensiones:
  - El Plano XY: Lograr el desplazamiento automático a través de puntos equidistantes tanto en el plano X como en Y según el ancho y largo definido por el operario (X máximo: 30 campos, Y máximo: 10 campos)
  - El Eje Z: Lograr un movimiento inteligente según el aumento o disminución del enfoque de la imagen.

- Almacenar las imágenes capturadas bajo nombres concatenados, tomando en cuenta el orden de adquisición y su ubicación dentro de la muestra total.
- Crear un interfaz de usuario sencillo y de fácil entendimiento, con prevención de errores y mensajes que permitan el correcto uso del sistema.

### 3.3 Planteamiento de la Solución

Para el cumplimiento de los objetivos se plantea UN DIAGRAMA DE FLUJO GENERAL (ver figura 3.1). Este diagrama muestra de manera concisa los bloques principales de programación: Bloque de configuración de parámetros, Bloque de Barrido automático y Bloque de enfoque automático. El primer bloque inicializa y declara los parámetros a usar dentro de la lógica general del programa. El segundo permite desplazar automáticamente la muestra a través del plano XY para analizar campo a campo cada imagen. El tercer bloque realiza el enfoque automático de cada campo, teniendo en consideración el programa de análisis de foco realizado por otra persona (esto se detalla más adelante). Líneas abajo se pondrán en detalle cada parte del diagrama de flujo general; además el diagrama de flujo completo del trabajo se encuentra en los anexos, con ello se puede ver gráficamente cómo funciona la lógica de solución para cada bloque.

Cabe resaltar que estos bloques no son los únicos que intervienen dentro de la lógica general de la solución. Bloques como: captura de imagen, validación de limitador, análisis de enfoque, etc. Son desarrollados en el transcurso del capítulo como bloques auxiliares para el mejor control del sistema en general, previniendo errores y haciendo que la lógica secuencial de cada parte sea coherente y se lleguen a cumplir los objetivos planteados al principio de este capítulo.

El diagrama mostrado a continuación (Ver figura 3.1) es de ayuda para una comprensión a primera vista de la lógica del programa integral. Sin embargo esta gráfica no es suficiente pues es necesario detallar cada bloque que implica el código general.



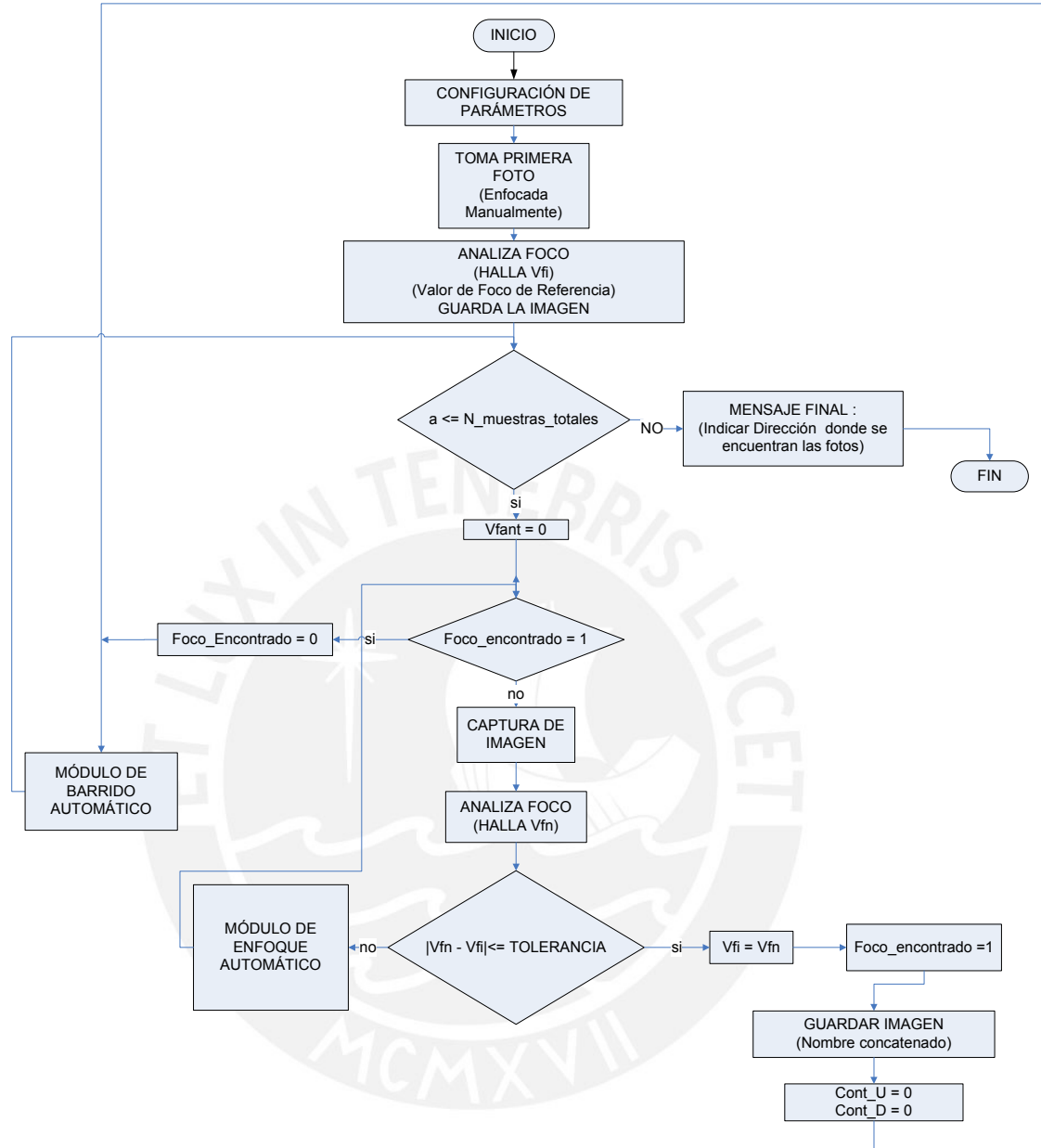


Fig. 3.1 DIAGRAMA DE FLUJO GENERAL

A continuación se describe cada bloque de trabajo, que en conjunto logran la elaboración de una lógica coherente del sistema, es decir, su automatización. Se detallan los recursos usados (hardware y software), el modo de solución (lógica usada), y desarrollo de cada bloque (código implementado en software de programación).

### 3.3.1 Captura de imagen:

Es necesario tomar en cuenta que el microscopio a utilizar es un microscopio óptico **binocular**, es decir, posee dos lentes de observación. Por tal razón, para la pre-visualización y captura de las imágenes en cada campo se usa una cámara web de forma tubular (Ver figura 3.2) que sea fácilmente adaptable a una lente del microscopio, en este caso se usa la lente derecha.

La cámara web a usar es una de forma tubular marca Halion. Sus características son:



Fig. 3.2 Cámara Web Halion de 5MP

- Interface: USB 2.0.
- 5.0 Mega Pixel.
- Plug and Play (no requiere de driver)
- Resolución de video: 640 x 480 / 800 x 600.
- Formato de vídeo: 24 BIT RGB.
- Rango de foco: 5 cm hasta infinito.
- Vídeo de Alta Calidad.
- Vídeo conferencias - Email (MSN / Yahoo).
- Compatible con Windows 95/98/Me/2000/XP/NT/VISTA - MAC.

Estas características nos permiten tener una imagen de nitidez aceptable y una pre-visualización en formato video que, conforme se realice el barrido de la muestra, nos dará una visión de la imagen que se está capturando y como se realiza el enfoque respectivo.

Para logra el control de la cámara web, la pre-visualización, la captura automática de la imagen, el análisis de enfoque y el almacenamiento con el nombre respectivo se usa el

software MATLAB®; específicamente *la Caja de Herramientas de Adquisición de Imágenes®* que este software posee. Es decir, bajo ciertos comandos usados correctamente se logrará el pleno control de la cámara web para el proceso automático integral que implica a todo el sistema.

### 3.3.2 Desplazamiento a través del Plano XY:

El Microscopio modelo CX31 de la compañía Olympus, como se dijo en el capítulo 2, posee perillas que se ubican en un mismo eje tubular (ver figura 3.3). Este arreglo dado permite adaptar 2 motores para cada eje respectivamente; de manera que mediante el uso de pequeñas y delgadas bandas dentadas se pueda transmitir el movimiento de motor a perilla.

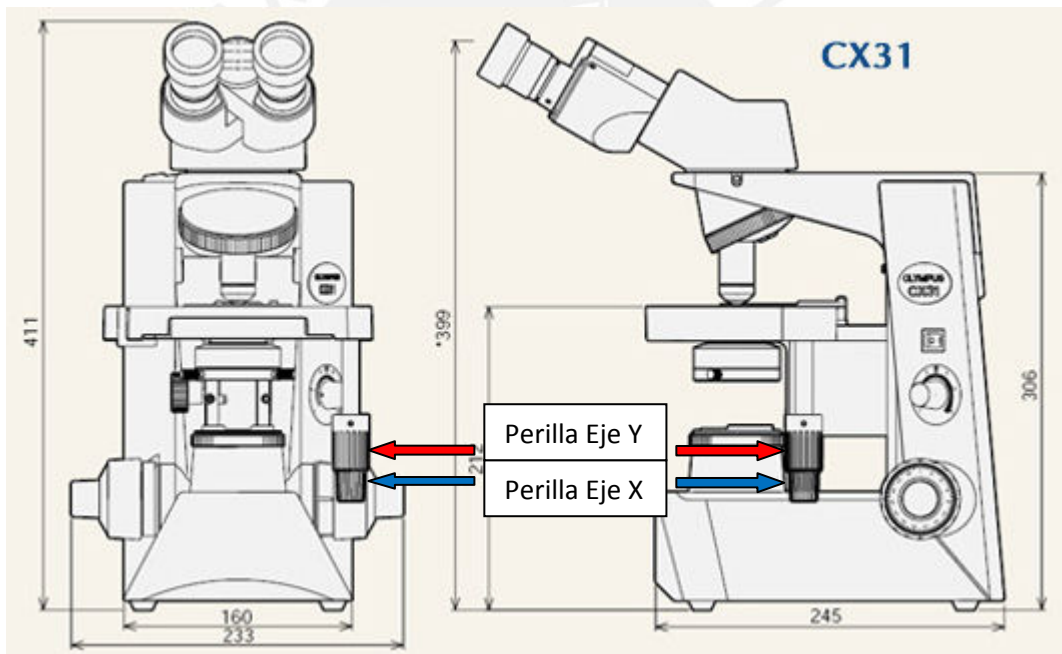


Fig. 3.3 Perillas de Ajuste en el plano XY del microscopio óptico binocular modelo CX31 de la compañía Olympus.

Fuente:

[http://www.olympuslatinoamerica.com/spanish/seg/seg\\_product\\_detail\\_esp.asp?g=716&s=11&c=53&d=2](http://www.olympuslatinoamerica.com/spanish/seg/seg_product_detail_esp.asp?g=716&s=11&c=53&d=2)

El planteamiento de la solución para este tipo de desplazamiento sugiere:

- El uso de motores de pasos de  $7.5^\circ$  por paso, debido a la exactitud que se requiere para manejar cada perilla y el enclavamiento característico de este tipo de motores, que permite que no haya movimientos sutiles que generen errores de ubicación cuando el sistema se mantenga estático.
- El uso de un microcontrolador para el envío de la secuencia de pasos (se vio en el capítulo 2) que permite el movimiento controlado de los motores. En este caso se usa un microcontrolador Atmega8 de la compañía ATMEL y el software de lenguaje ensamblador VMLab para la programación de cada motor.

### Desplazamiento a través del eje Z (Movimiento de la lente para el enfoque)

De la misma forma que para el eje X e Y, el microscopio posee 2 perillas especiales (ver figura 3.4), en este caso, para el ajuste de la lente de enfoque. Una perilla de ajuste largo, y una perilla de ajuste fino. En este caso solo se usa un motor para la perilla de ajuste fino, debido a que la perilla de ajuste largo puede ser manipulada por el operario sin mayor problema. La sensibilidad del movimiento y el desplazamiento por paso ( $7.5^\circ$ ) permite un correcta variación de enfoque conforme se realice el giro del motor.

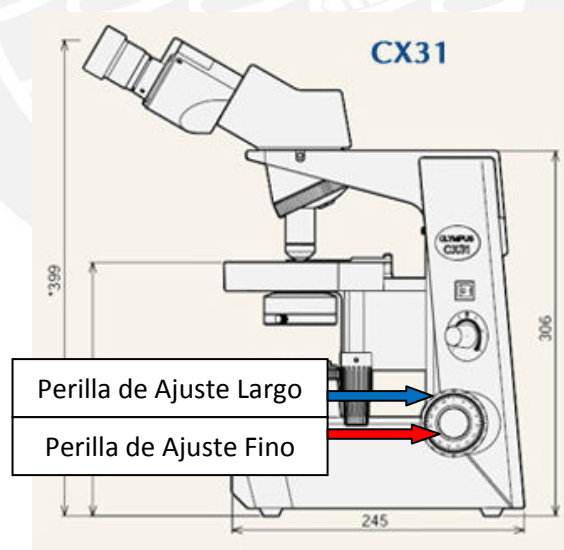


Fig. 3.4 Perillas de Enfoque del microscopio CX 31 de la compañía Olympus.

Fuente:

[http://www.olympuslatinoamerica.com/spanish/seg/seg\\_product\\_detail\\_esp.asp?q=716&s=11&c=53&d=2](http://www.olympuslatinoamerica.com/spanish/seg/seg_product_detail_esp.asp?q=716&s=11&c=53&d=2)

Al igual que los motores X e Y, se usa el microcontrolador Atmega8 de la compañía ATMEL para el envío de la secuencia de pasos y el software VMLab para la programación respectiva.

### Desplazamiento Coordinado en 3 dimensiones

Si bien líneas arriba se describe el modo de solución para lograr el control del movimiento de los motores, y el control de la cámara, es necesaria la creación de un programa principal que combine estas soluciones de la mejor forma a fin de que el sistema total tenga un comportamiento automático y se llegue a obtener la cantidad de imágenes por muestra analizada.

Para cumplir con todo ello se plantea una lógica de control total que será diseñada y programada en el software MATLAB® bajo el uso de lenguaje de programación M, propio del software a utilizar. Dicho programa permite el manejo de la cámara y la comunicación respectiva vía puerto serial con el microcontrolador para el correcto movimiento de los motores cuando sea necesario.

Un punto importante de esta solución es el código utilizado para la cuantización del enfoque de la imagen; para ello se toma parte del código implementado por el Alumno Joel Alanya Beltrán que forma parte del desarrollo del sistema en conjunto, así como Luis Ángel Ruiz Navarro, encargado del hardware del sistema.

Volviendo al punto del desarrollo de la lógica de control, es necesario explicar su planteamiento de la siguiente manera:

- **El software de análisis de enfoque de imagen**, implementado previamente, indica un valor numérico dependiendo de cuán clara esté la imagen. Esta propiedad permite analizar todos los campos y evaluar la nitidez de cada punto. Sin embargo, se requiere un punto de referencia, es decir, un valor inicial con el cual se puedan comparar los “valores de foco” siguientes y así poder mover la lente para un enfoque correcto. Por tal razón, un requisito indispensable del sistema es **la correcta posición del primer punto de evaluación** que debe ser hecha por el operario, asimismo, el laboratorista debe indicar cuántos puntos quiere analizar por eje, teniendo como opciones un máximo de 30 campos en el eje X y 10 campos en el eje Y.

- Luego de obtener **el primer valor de foco** el sistema empieza a trabajar automáticamente, es decir el desplazamiento a través del plano XY y el eje Z se concatenan para pasar de campo a campo e ir analizando la imagen y enfocando respectivamente.

A continuación se procede a explicar como será maneja cada movimiento:

### Movimiento en el plano XY (Barrido de la muestra campo a campo):

Para una mejor ubicación del movimiento de la muestra, se debe indicar que el plano de referencia es el cuarto cuadrante dentro de un plano cartesiano (Ver figura 3.5).

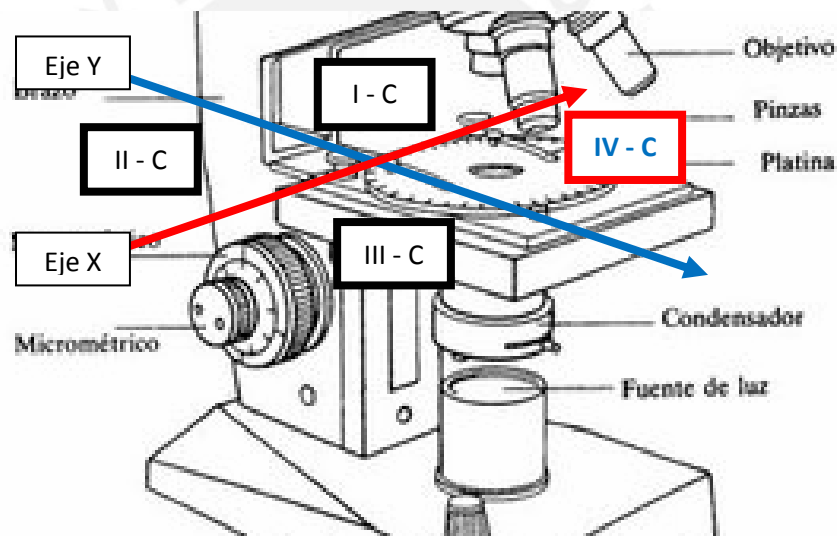


Fig. 3.5 Plano de acción del Barrido automático (IV cuadrante)

Fuente: [http://palangre009.blogspot.com/2009/09/quien-invento-el-microscopio\\_17.html](http://palangre009.blogspot.com/2009/09/quien-invento-el-microscopio_17.html)

Por otro lado, el microcontrolador atmega8 usado es programado de tal forma que reciba, vía comunicación serial (revisar capítulo 2) las letras A, D, W o S, (Ver figura 3.6) dichas letras indican un movimiento específico de los motores que será realizado por cada motor dependiendo de su ubicación:



A: Movimiento en el eje X Negativo.

D: Movimiento en el eje X Positivo.

W: Movimiento en el eje Y Positivo.

S: Movimiento en el eje Y Negativo.

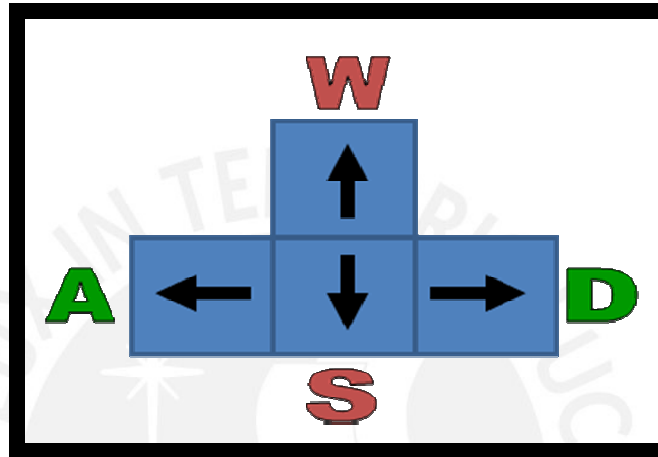


Fig. 3.6 Instrucciones para el movimiento de los motores en el plano XY.

### Movimiento en el eje Z (Enfoque automático):

Con la ayuda del **software de análisis de enfoque de imagen** y el **valor de foco del punto inicial** es que se realiza el desplazamiento vertical de la lente.

Se toma como referencia el valor inicial y se va comparando con el valor que se tiene de la imagen del campo que se analiza. Sin embargo, el algoritmo de evaluación de enfoque nos da siempre un valor con 2 puntos decimales, es decir, se tiene una contabilización muy fina del valor de la nitidez, por lo que es difícil que las demás imágenes tengan el mismo valor de foco. Por tal razón, se crea una variable a la que llamamos **tolerancia** que nos permite tener un rango de proximidad hacia el valor de foco ideal (el de la primera imagen). Por ejemplo, si se tiene un valor de foco inicial de 1800, es factible tener un valor de tolerancia de 50. Ello significa que una imagen bien enfocada debe estar en el rango de 1750 y 1850, si se sobrepasa ese rango o el valor inicial de foco, se actualiza el valor de referencia a ese nuevo valor hallado; por el

contrario si la imagen hallada tiene un valor de foco menor que el mínimo valor del rango se procede a mover la lente para encontrar un mejor enfoque.

El movimiento de autoenfoco consta de las siguientes condiciones y pasos:

Si el valor de foco de la imagen es menor que el de la referencia y no se encuentra dentro del rango, se realiza un desplazamiento hacia arriba (movimiento inicial predefinido), es decir en dirección al eje Z positivo y se almacena ese valor de foco.

- i. Luego se analiza el siguiente valor de foco, si este es mayor que el anterior pero no se encuentra dentro del rango de enfoque entonces se procede a seguir en la dirección que se tenía (hacia arriba) y se guarda ese valor de foco.
- ii. Si luego se analiza la imagen y su valor de foco es mayor, se sigue en esa dirección (hacia arriba). Pero si el valor es menor, ello quiere decir que la nitidez empeora, entonces se procede a ir en la dirección contraria (hacia abajo).
- iii. Si luego el valor de foco empieza a mejorar se sigue en la misma dirección, según sea el caso, hacia arriba o hacia abajo. Es decir cuando el enfoque empieza a mejorar, se sigue en la misma dirección, pero si el enfoque empeora, se toma una dirección contraria.
- iv. Claro está que si el valor de foco esta dentro del rango de enfoque o es mayor que el valor de referencia, se procede a guardar la imagen capturada y a pasar al siguiente campo, con el valor de referencia actualizado o no según sea el caso.

Lógicamente, no se puede pasar al siguiente campo si no se ha obtenido una imagen bien enfocada. Sin embargo, se ha tomado en cuenta la posibilidad de que el sistema no logre encontrar una imagen bien enfocada, ello quiere decir que la lente estará oscilando infinitamente hasta encontrar una imagen perfecta, lo que implica pérdida de tiempo. Para evitar ese problema se crea una variable llamada **limitador** que me indica el máximo número de movimientos verticales que puede realizar la lente hasta encontrar una imagen clara, cuyo valor de foco se encuentre dentro del rango permitido.

La instrucción de movimiento hacia arriba o hacia abajo dependiendo de la situación, se hace, al igual que con los motores X e Y, a través de letras que son enviadas vía

puerto serial hacia el microcontrolador (Ver figura 3.7), que luego las interpreta y envía la trama correspondiente al motor para generar su giro.

U: Desplazamiento a través del eje Z Positivo

J: Desplazamiento a través del eje Z Negativo

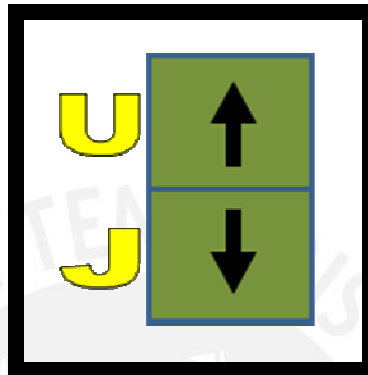


Fig. 3.7 Instrucciones para el movimiento del motor de control del eje vertical del microscopio

Si bien todo lo descrito anteriormente es para el mejor entendimiento del modo automático de operación del sistema, se decidió hacer también el modo manual del mismo que consiste básicamente en el manejo de los motores para el desplazamiento XY, enfoque de la lente y captura de imagen a través del uso del teclado; es decir, la letras que se enviaban vía puerto serial para el movimiento de cada motor, serán usadas para que con la tecla respectiva se pueda hacer lo mismo. De esta forma al apretar A, D, W, S, U, J se moverá cada motor según lo programado. Además se adiciona la tecla C para indicar que se quiere capturar la imagen que se está enfocando.

### 3.4 Desarrollo de la solución

A continuación, se procede a mostrar el desarrollo de la solución es decir, partes importantes del código diseñado tanto para el control de movimiento de los motores como para el control integral del sistema, es decir la lógica secuencial que gobierna a todo el sistema automático para la adquisición de las imágenes que conforman toda la muestra analizada.

**Algoritmo de Control de Movimiento de los motores X, Y, y Z (realizado con el software VMLab)**

Para controlar los motores de paso, es necesario conocer, en primer lugar, la secuencia de pasos que los gobiernan (ver tabla 3.1). En nuestro caso, los motores de pasos que se tiene son de 7.5° y su secuencia de pasos es la misma para todos:

Tabla 3.1 Secuencia de pasos para el motor de pasos bipolar usado.

	S1	S2	S3	S4
<b>Paso # 1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>Paso # 2</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>
<b>Paso # 3</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>
<b>Paso # 4</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>

Estos pasos serán almacenados en una tabla dentro del programa VMLab para luego, mediante punteros de memoria, acceder a ellos consecutivamente según las instrucciones recibidas. (Ver figura 3.8)

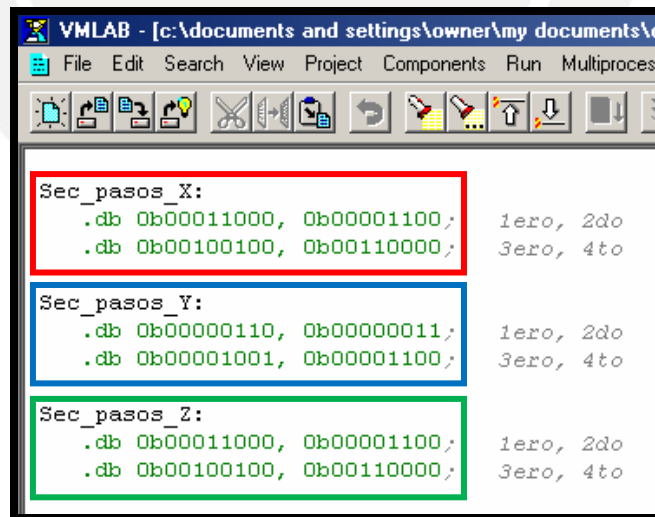


Fig. 3.8 Almacenamiento de la secuencia de pasos en tablas dentro de la memoria del atmega8.

El programa de control de los motores se inicia declarando la variable **datos** que guardará su valor en el registro 21, asimismo se configura el puntero de la pila para el

acceso a la tabla que contiene los pasos de cada motor. Además, se llaman a subrutinas que configuran **la comunicación serial y los puertos de entrada y salida** (Ver figura 3.9 y 3.10).

Las características de la comunicación serial son:

- i. 9600 baudios de Velocidad con una frecuencia de oscilación de 1MHz.
- ii. Comunicación asíncrona, sin paridad, un bit de parada y 8 bits de datos.
- iii. Se deshabilitan las interrupciones de comunicación.

```

////////////////////////////////////
////                               CONFIGURACION DEL SERIAL
////////////////////////////////////
CONFIGURA_SERIAL:

PUSH R17

;9600 baudios (fosc= 1MHZ)
ldi R16,high($0C)
out UBRRH,R16
ldi R16,low($0C)
out UBRRL,R16
;velocidad DOBLE, no multiprocesador
ldi R16,(0<<RXC | 0<<TXC | 1<<U2X | 0<<MPCM)
out UCSRA,R16
;Comunicación asíncrona, SIN PARIDAD, 1 bit de parada, 8 bits
ldi R16,(1<<URSEL | 0<<UMSEL | 0<<UPM1 | 0<<UPMO | 0<<USBS | 1<<UCS21 | 1<<UCS20)
out UCSRC,R16
;interrupciones de comunicación deshabilitadas, Tx y Rx habilitadas
ldi R16,(0<<RXCIE | 0<<TXCIE | 0<<UDRIE | 1<<RXEN | 1<<TXEN | 0<<UCS22 | 0<<TXB8)
out UCSRB,R16

POP R17

RET

```

Fig. 3.9 Configuración del puerto serial

La configuración de los puertos se realiza de la siguiente manera:

- i. Puerto B pines: 2-5 salidas para el motor X
- ii. Puerto C pines: 0-3 salidas para el motor Y
- iii. Puerto D pines: 2-5 salidas para el motor Z

```
////////////////////////////////////  
////                               CONFIGURACION DE PUERTOS  
////////////////////////////////////  
CONFIGURA_PUERTOS:  
  
push r16  
ldi r16, 0b00111100           ; PB2--5 salida para motor eje x  
out ddrb, r16  
ldi r16, 0b00001111         ; PC0--3 salida para motor eje y  
out ddrc, r16  
ldi r16, 0b00111100         ; PD2--5 salida para motor eje z  
out ddrd, r16  
pop r16  
ret
```

Fig. 3.10 Configuración de Puertos.

Luego de las configuraciones respectivas, se procede a esperar el dato que será enviado desde el programa principal para el movimiento de los motores, para ello se crea la subrutina llamada “**RECIBE\_DATO**” con ello, ni bien se recibe un dato, este se almacena en la variable **DATO**, con ello se realiza la comparación para saber qué movimiento realizar.

Se definen 6 movimientos para cada instrucción que se envía, en este caso son letras:

- A → Movimiento Negativo en el eje X
- D → Movimiento Positivo en el eje X
- S → Movimiento Negativo en el eje Y
- W → Movimiento Positivo en el eje Y
- J → Movimiento Negativo en el eje Z
- U → Movimiento Positivo en el eje Z

Entonces, estas letras son enviadas por el programa principal y el VMLab los recibe, los interpreta y envía la trama correspondiente a los motores de pasos:



```

////////////////////////////////////
:////                               MODO MANUAL
////////////////////////////////////

MANUAL:

RCALL RECIBE_DATO
CPI  DATO, 'D'      ; desplazarse a la derecha  (eje x)
BREQ X_DERECHA_MANUAL

CPI  DATO, 'A'      ; desplazarse a la izquierda (eje x)
BREQ X_IZQUIERDA_MANUAL

CPI  DATO, 'U'      ; desplazarse hacia arriba  (eje y)
BREQ Y_ARRIBA_MANUAL

CPI  DATO, 'S'      ; desplazarse hacia abajo  (eje y)
BREQ Y_ABAJO_MANUAL

CPI  DATO, 'U'      ; Subir                      (eje z)
BREQ Z_SUBE_MANUAL

CPI  DATO, 'J'      ; Bajar                      (eje z)
BREQ Z_BAJA_MANUAL

CPI  DATO, 'R'      ; retornar al menu principal
BREQ ARRANQUE
RJMP MANUAL

```

Fig. 3.11 Recepción y acción de las instrucciones para el movimiento de los motores.

Para el envío de la trama correspondiente, se usa un puntero de pila que me indica la última posición de memoria a la que se quedo apuntando, de manera que el siguiente paso es la siguiente dirección de memoria. Asimismo cabe resaltar que entre paso y paso que se dé en el motor, debe haber un retraso, para ello se crea una subrutina llamada **RETRASO** (figura 3.12), con la cual se genera un retraso de 100 milisegundos entre paso y paso.

```

//////////////////////////////////////////////////////////////////
//////////          RETARDO DE 100ms          //
//////////////////////////////////////////////////////////////////
RETARDO:
PUSH R16
PUSH R17
PUSH R18

CLR R16
CLR R17
CLR R18

RUTINA:
INC R16
CPI R16, 80
BREQ LEVEL_1
RJMP RUTINA

LEVEL_1:
INC R17
CLR R16
CPI R17, 250
BREQ LEVEL_2
RJMP RUTINA

LEVEL_2:
CLR R16
CLR R17
INC R18
CPI R18, 2
BRNE RUTINA

POP R18
POP R17
POP R16
RET

X_DERECHA_MANUAL:
CLR R17
LOOPX:
INC R18
CPI R18, 4
BREQ RSTX
RSTRX:
LDI R31,HIGH(Sec_pasos_X*2)
LDI R30,LOW(Sec_pasos_X*2)
LDI R16, 0
ADD R30, R18
ADC R31, R16
LPM R16, Z
OUT PORTB, R16
RCALL RETARDO
INC R17
CPI R17, $01
BRNE LOOPX
RJMP OPERACION

RSTX:
LDI R18, 0
RJMP RSTRX

```

Fig. 3.12 Retardo entre paso y paso y ejemplo de movimiento eje X

### Algoritmo de Control para el Modo Automático y Manual del Sistema (realizado con el software MATLAB)

Antes de ver el algoritmo de control total del sistema es necesario explicar el desarrollo de una función creada para la comunicación serial entre el programa principal y el microcontrolador que enviará la trama correspondiente para el movimiento de los motores.

#### Función “senddata” para el envío de instrucciones:

Este algoritmo implementado en el software MATLAB permite enviar una instrucción almacenada en la variable **datos** a través del puerto serial “s” (este puerto se configura al inicio del programa principal a través del código: **s = serial ('com4')** por ejemplo). Para ello, convierte el valor de la variable **datos** en una cadena de caracteres a través del comando “**num2str**”, luego se transmite esa cadena codificada a la variable **trama**. Después de ello, se transfiere las propiedades de la variable **trama** a la variable **t** y a través del comando **whos**, de esta forma se configura **el tamaño de salida del buffer** de acuerdo al tamaño de la trama (número de bytes), y **la velocidad de transmisión a 9600 baudios** de acuerdo a la velocidad de recepción del

microcontrolador (ver figura 3.13). Luego, se abre el puerto s, se envía la trama y se procede a cerrar el puerto.

```
function senddata(datos,s)
    datos = num2str(datos);
    trama=[datos];
    t=whos('trama');
    set(s,'outputBufferSize',t.bytes);
    set(s,'BaudRate',9600);
    fopen(s);
    fprintf(s,'%s',trama)
    fclose(s);
```

Fig. 3.13 Código de subprograma de comunicación serial

Una vez explicada la función creada para la comunicación serial, se procede a explicar el funcionamiento y la lógica del algoritmo de control integral del sistema.

Se inicializa el programa borrando todas las variables existentes y cerrando todas las ventanas auxiliares e inicializando los contadores de desplazamiento: i (eje X), j (eje Y), k (eje Z), a (contador de número de campos través de la muestra).

Asimismo, se declaran al inicio las variables:

#### Variable de selección de operación:

**selector:** Indica tres opciones: T: modo automático, M: modo manual, F: Finalizar la operación. Se inicializa con un valor 0.

#### Variables para el plano XY:

**muestras\_i:** Cantidad de campos a analizar en el eje X.

**muestras\_j:** Cantidad de campos a analizar en el eje Y.

**sentido\_i:** Bandera que me indica el sentido del movimiento a través del eje x en el plano: Hacia la derecha  $\rightarrow(0)$ , hacia la izquierda  $\leftarrow(1)$ .

### **Variables para el Autoenfoque:**

**cont\_U:** Cantidad de movimientos hacia arriba (Up) de la lente.

**cont\_D:** Cantidad de movimientos hacia abajo (Down) de la lente.

**Vf\_ant:** Variable que almacena el valor de foco anterior al actual de un campo.

**Tolerancia:** valor que me indica la fluctuación positiva y negativa con respecto al valor de foco de referencia.

**limitador:** Valor que me indica el máximo número de movimientos permitidos en el eje Z para lograr enfocar una imagen.

**Lim:** Bandera que me indica si se llegó (1) o no (0) al máximo número de movimientos verticales permitidos.

**Cont\_fallas:** Variable que almacena la cantidad de campos en donde no se logró enfocar correctamente la imagen.

**foco\_encontrado:** Bandera que indica si se ha encontrado (1) o no (0) una imagen correctamente enfocada.

Una vez declaradas las variables que serán de apoyo al sistema global, se procede a preguntar al usuario si se desea iniciar el modo automático o el modo manual del sistema para ello se identifica al modo automático con una letra "T" y al modo manual con una letra "M". Hay una opción adicional "F" que indica el final de la operación del sistema (ver figura 3.14 y 3.15).

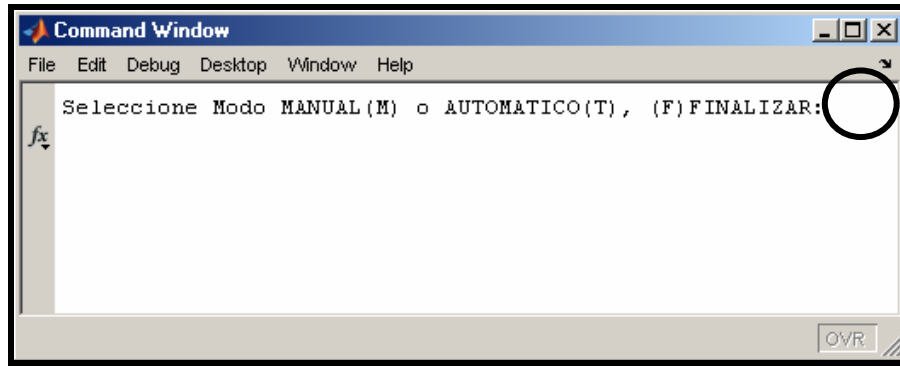


Fig. 3.14 Menu Inicial del programa

```

selector = 0;
while selector ~= 'F'
    selector = input('Seleccione Modo MANUAL (M) o AUTOMATICO (T), (F) FINALIZAR: ');
    if selector == 'M'
        uiwait(msgbox('BIENVENIDO AL MODO MANUAL DE OPERACION DEL MICROSCOPIO', 'MENSAJE DE BIENVENIDA', 'HELP', 'modal'));
        %% CONFIGURACIÓN DE CÁMARA Y MUESTRA PREVIA DE IMAGEN VIA VIDEO:
        imaghwin('winvideo', selector_camara);
        vid = videoinput('winvideo', selector_camara, 'YUY2_640x480');
        preview(vid);
        datos = 'M';
        senddata(datos, s);
        while selector == 'M'
            closepreview;
        elseif selector == 'T'
            datos = 'M'; % se cambia para limitar al maximo el codigo del atmega8.
            senddata(datos, s);
            while selector == 'T'
                closepreview;
            elseif selector == 'F'
                uiwait(msgbox('OPERACION FINALIZADA', 'MENSAJE DE FINALIZACIÓN', 'HELP', 'modal'));
            else
                uiwait(msgbox('ERROR DE SELECCION DE MODO (LETRAS MAYUSCULAS POR FAVOR)', 'ERROR DE SELECCION', 'ERROR', 'modal'));
            end
        end
    end
end
    
```

Annotations in the code block:

- A red circle highlights the condition `selector ~= 'F'` in the `while` loop, with a red arrow pointing to a box labeled "SELECTOR DE FINAL DE OPERACIÓN".
- A blue circle highlights the condition `selector == 'M'` in the inner `while` loop, with a blue arrow pointing to a box labeled "SELECTOR DE MODO MANUAL".
- A green circle highlights the condition `selector == 'T'` in the inner `while` loop, with a green arrow pointing to a box labeled "SELECTOR DE MODO AUTOMÁTICO".

Fig. 3.15 Código de Validación de opciones de operación

### 3.4.1 Modo Automático:

El modo automático inicializa con un mensaje de bienvenida y luego pidiendo al usuario el número de campos a analizar, tanto en el eje X como en el eje Y (ver figura 3.16 y 3.17). Estos datos son multiplicados para la obtención de la variable `N_muestras_totales` que me indica el número total de muestras que se va a analizar.

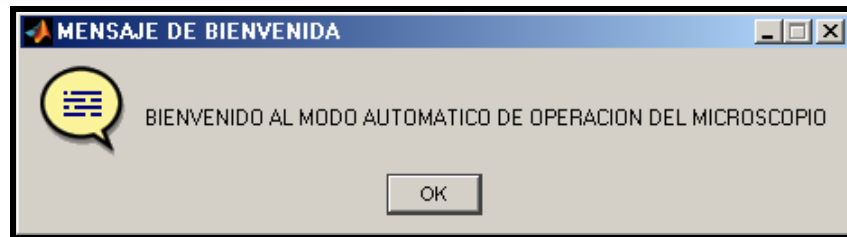


Fig. 3.16 Mensaje de Bienvenida al modo automático de Operación

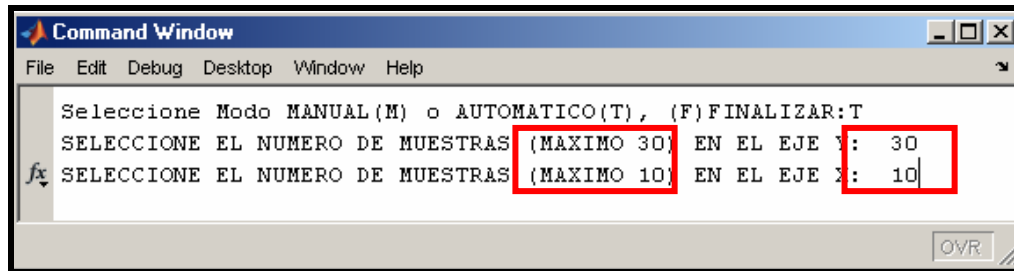


Fig. 3.17 Menú inicial para fijar los valores de los campos a analizar

Al guardar los datos de campos a analizar por eje y la variable `N_muestras_totales`, se procede a la captura y almacenamiento de la imagen del primer punto, colocado por el operario, de esta manera se obtiene el valor de foco de referencia; luego se procede a mandar la instrucción para el desplazamiento al siguiente campo. Para ello, primero se configura la cámara.

#### Configuración de la cámara:

```
imaqhwinfo('winvideo',selector_camara);
vid=videoinput('winvideo',selector_camara,'YUY2_640x480');
preview(vid);
```

La primera línea de código permite reconocer el dispositivo de captura de imagen conectado a la computadora, la segunda línea configura la cámara para que capture una imagen de 640 x 480 píxeles (la más grande que puede capturar), y la tercera línea permite una pre-visualización en video de la imagen que se está enfocando.

Una vez configurada la cámara se procede a la captura, análisis y almacenamiento de la imagen.



### Captura de la imagen:

```
I=getsnapshot(vid);
img = ybcr2rgb(I);
```

El primer comando permite capturar la imagen y almacenarla en una variable "I"; sin embargo al capturar la imagen, esta se ve afectada por la luz que incide en el ambiente, para quitar esa luz se utiliza el segundo comando, de manera que al mostrar la imagen esta salga tal y como se pre-visualizó.

El análisis de la imagen se realiza a través del código desarrollado por otro alumno, de manera que no esta dentro de nuestro alcance explicarlo (Ver figura 3.18), sino simplemente utilizarlo.

```

%% PROGRAMA DE ANALISIS DE FOCO:
I2 = rgb2gray(I);
%-----
%la gradiente de la imagen
%obtienes el contorno de la imagen
%usando el metodo de deteccion SOBEL
%-----
[BW,thresh,gv,gh] = edge(I2,'sobel');
%TENIENDO LOS VALORES DE LAS GRADIENTES gv y gh
%PODEMOS HALLAR LA MAGNITUD FOCAL
        %gv es el valor de la mascara vertical x imagen I
gv1=abs(gv); %gv1 es el valor absoluto
        %valor de la mascara horizontal x imagen I
gv2=abs(gh); %valor absoluto
Mfoco=plus(gv1,gv2); %suma de absolutas
Vfi=sum(Mfoco(:)); % suma de la matriz
%% guardar primera imagen captada
a = a+1;
  
```

VALOR DE FOCO DE REFERENCIA

Fig. 3.18 Código de Análisis de foco de cada imagen

### Almacenamiento de la imagen:

```
file=strcat('image', num2str(a));
file=strcat(file, '.jpg');
imwrite(img, file, 'quality', 100);
```

Para almacenar la imagen con un nombre concatenado, es decir, con un nombre que indique el número de campo a que pertenece, se hace un pequeño artificio: el comando “**num2str**” me permite cambiar el valor de la variable a de número a una cadena de caracteres, el comando “**strcat**” me permite concatenar cadena de caracteres, y el comando “**imwrite**” me permite guardar una imagen poniendo primero la variable en donde esta almacenada la imagen (img), luego el nombre (que hemos formado con el comando strcat) y la variable que se quiere modificar, en este caso la calidad (quality), y el valor de esa variable , que para este caso es 100 (la más alta calidad). Entonces si se esta en el campo número 30 el nombre para la imagen guardada será: “imagen30.jpg” (.jpg es el formato de la imagen).

Luego de la captura, análisis y almacenamiento de la primera imagen empieza el trabajo del algoritmo de automatización. Dentro de este código se encuentran dos bloques que trabajan conjuntamente: Bloque de barrido automático de la muestra a través del plano XY, y Bloque de ajuste automático del enfoque de la imagen.

#### **Bloque de Barrido Automático:**

Para ello es necesario tomar en cuenta que el barrido se realiza en forma de zigzag; es decir, se barre campo a campo en una dirección a través del eje X hasta llegar al ultimo punto, luego se baja un campo en el eje Y, y se procede a regresar campo a campo hasta el primer punto del eje X, luego otra vez se procede a bajar en el eje Y, y a desplazarse campo a campo en dirección al ultimo punto del eje X. Así sucesivamente hasta llegar al último campo. (Ver figura 3.19)

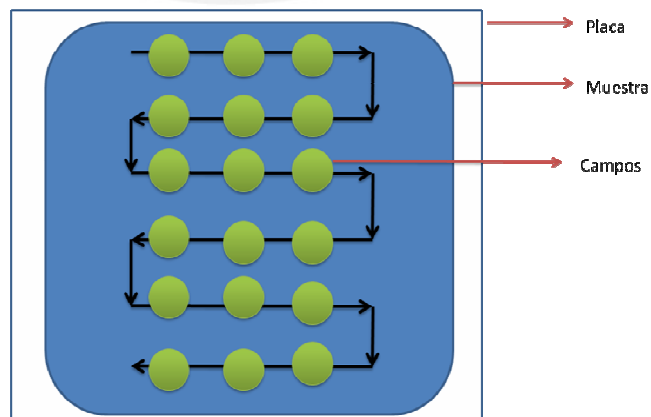


Fig. 3.19 Esquema de barrido automático a través del plano XY

El método que se usa para realizar el barrido automático está basado en la dirección de desplazamiento a través del eje X. De acuerdo a ello se decidirá que dirección seguir, ya sea en el eje X o en el eje Y.

Se usa una bandera llamada “**sentido\_i**” que me indica el sentido de desplazamiento a través del eje X: sentido positivo (0) y sentido negativo (1). Como se ve en la figura el primer sentido que toma el barrido es el sentido 0 de manera que al principio de programa se indica ese valor. Luego conforme se avance a través de la muestra se sensa el punto en el que se encuentra el campo analizado (“i” me indica el punto en el eje X y “j” me indica el punto en el eje Y) y se compara con los valores de las variables **muestras\_i** y **muestras\_j**, que indican el máximo número de campos en X e Y respectivamente. Entonces:

- Si se tiene  **$i = 1$**  y  **$\text{sentido}_i = 0$**  ( $\rightarrow$ ) se envía a través del puerto serial una “**D**” que me indica dirección **X positiva**, y se **aumenta** en 1 el valor de i.
- Si se tiene  **$i = \text{muestras}_i$**  y  **$\text{sentido}_i = 1$**  ( $\leftarrow$ ) se envía a través del puerto serial una “**A**” que me indica dirección **X negativa**, y se **disminuye** en 1 el valor de i.
- Si se tiene  **$1 < i < \text{muestras}_i$**  y  **$\text{sentido}_i = 0$**  ( $\rightarrow$ ) se envía una “**D**” para indicar dirección X positiva, y **aumenta** el valor de i en 1.
- Si se tiene  **$1 < i < \text{muestras}_i$**  y  **$\text{sentido}_i = 1$**  ( $\leftarrow$ ) se envía una “**A**” para indicar dirección X negativa, y **disminuye** el valor de i en 1.
- Si se tiene  **$i = \text{muestras}_i$**  y  **$\text{sentido}_i = 0$**  ( $\rightarrow$ ) eso quiere decir que se llegó al **último punto en el eje X**, por lo tanto se verifica primero si  **$1 < j < \text{muestras}_j$** ; si es así, se envía una “**S**” indicando movimiento negativo en el eje Y, y **se cambia el valor de sentido\_i a 1** indicando movimiento en dirección **negativa** en el eje X. En cambio, **si se tiene un valor de  $j = \text{muestras}_j$  se procede a finalizar el barrido.**
- Si se tiene  **$i = 1$**  y  **$\text{sentido}_i = 1$**  ( $\leftarrow$ ) eso quiere decir que se llegó al **primer punto en el eje X**, por lo tanto se verifica primero si  **$1 < j < \text{muestras}_j$** ; si es así, se envía una “**S**” indicando movimiento negativo en el eje Y, y **se cambia el valor de sentido\_i a 0** indicando movimiento en dirección **positiva** en el eje X. En cambio, **si se tiene un valor de  $j = \text{muestras}_j$  se procede a finalizar el barrido.**

De esta forma se realiza el barrido automático a través de toda la muestra analizando, claro está, el valor de la nitidez de la imagen en cada campo (Ver figura 3.20).

```

%% BARRIDO AUTOMATICO :
if i == muestras_i;
    if j == muestras_j;
        j = 1;
        i = 1;
    else
        j = j + 1;
        datos = 'S';
        senddata(datos,s);
        if sentido_i == 0;
            sentido_i = 1;
        else
            i = i - 1;
            datos = 'A';
            senddata(datos,s);
        end
    end
else
    if sentido_i == 0;
        i = i + 1;
        datos = 'D';
        senddata(datos,s);
    else
        if sentido_i == 1;
            if i == 1;
                if j == muestras_j;
                    j = 1;
                    i = 1;
                else
                    j = j + 1;
                    datos = 'S';
                    senddata(datos, s);
                    if sentido_i == 1;
                        sentido_i = 0;
                    end
                end
            end
        end
    end
end
end
end
end
    
```

Uso de la variable "sentido\_i" para el correcto envío de "S", "A" o "D" vía puerto serial según sea el caso

Validación de los valores topes de los contadores "i" y "j"

Envío de las instrucciones "S", "A", "D" vía puerto serial según corresponda

Fig. 3.20 Código detallado de barrido automático a través del plano XY. Los cuadros indican la acción determinada de cada código.

### Bloque de ajuste automático del enfoque de la imagen:

El bloque de ajuste automático de enfoque, está basado principalmente en el desplazamiento a través del eje Z regido por el valor de foco de la imagen, comparado con el valor de referencia y el valor de foco de un estado anterior en el mismo campo.

Es decir, se tienen 3 variables que se comparan:  $V_{fn}$  (valor de foco de la imagen actual enfocada),  $V_{fant}$  (valor de foco de un punto anterior en el mismo campo), y  $V_{fi}$  valor de referencia (tomado del punto inicial colocado por el operario).

Asimismo, se hace uso de las variables ya descritas anteriormente: **cont\_U**, **cont\_D**, **Tolerancia**, **limitador**, **Lim**, **Cont\_fallas**, **foco\_encontrado**.

Se valida primeramente el valor de la variable **a**, para ver si se llega a analizar el número de campos totales, si el valor de **a** es mayor al valor de  $N_{muestras\_totales}$ , entonces se finaliza la operación; pero si el valor de **a** es menor o igual al valor de  $N_{muestras\_totales}$  entonces se pasa a evaluar el valor de la bandera **foco\_encontrado**, si este es **1** se procede a pasar al siguiente campo, ese desplazamiento depende del bloque de barrido automático; por el contrario, si el valor es **0**, se procede a hacer la búsqueda de una imagen correctamente enfocada.

El procedimiento para la búsqueda de un enfoque correcto es el siguiente:

- Se parte desde la validación del valor de la variable **Lim**, **si es 1**, ello quiere decir que se tuvo el máximo número de movimientos verticales permitidos, entonces se procede a enviar la instrucción para el pase a un siguiente campo (BLOQUE DE BARRIDO AUTOMÁTICO), se inicializa (**pone a 0**) los valores de **cont\_U** y **cont\_D** para que en el siguiente campo se vuelvan a contar los movimientos verticales de la lente y, por último, se procede a **augmentar en 1** el contador de fallas: **cont\_fallas**. Por el contrario, si el valor de **Lim** es **0**, entonces se procede a capturar la imagen del campo analizado (Ver figura 3.21).

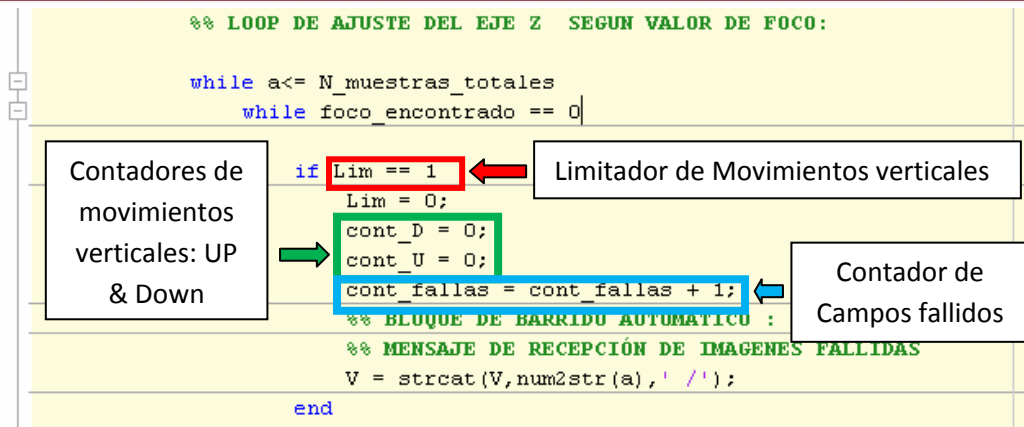


Fig. 3.21 Validación de límite de movimientos verticales según la bandera Lim

- Luego de capturar la imagen, esta es procesada por el algoritmo de análisis de foco, dando como resultado un Valor de foco actual: Vfn. Este valor tiene 2 comparaciones:
  - Se compara con el Valor de foco de referencia (**Vfi**) si este valor está **dentro del rango permitido** dado por la variable **tolerancia**, entonces el enfoque es válido y se procede a **guardar la imagen** y a cambiar el valor de la bandera **foco\_encontrado** a **1** para el paso a otro campo en la muestra; también, se vuelven a **inicializar (poner a 0)** los contadores de movimientos verticales: **cont\_U** y **cont\_D**. Además, si el valor de foco encontrado es **mayor que el valor de referencia (Vfi)**, este valor de referencia se actualiza y toma el mayor valor encontrado, teniéndose así un **valor de Vfi nuevo** (Ver figura 3.22).



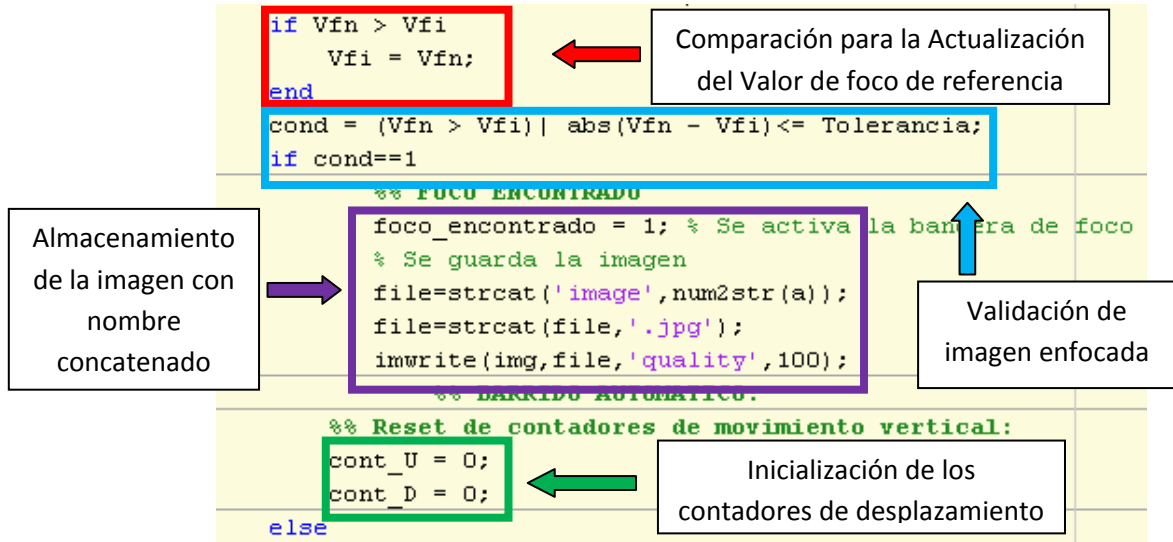


Fig. 3.22 Validación de foco encontrado y almacenamiento de imagen con nombre concatenado.

- De la misma forma, se compara el valor actual de foco  $V_{fn}$ , con un valor anterior ( $V_{fant}$ ), que puede ser, el valor de foco del campo anterior o el valor de foco de un punto vertical anterior del mismo campo analizado.

Esta comparación realizada, conlleva a un desplazamiento coherente a través del eje z en busca del foco correcto; para ello se usa, al igual que en el bloque de barrido automático, una bandera de desplazamiento llamada **BD**, cuyo valor “0” indica un desplazamiento de dirección ascendente, y el valor “1” indica un desplazamiento de dirección descendente.

De esa forma es que se procede a evaluar el movimiento vertical según el enfoque:

- Si el valor de  **$V_{fn}$**  es mayor a  **$V_{fant}$**  y no se ha realizado movimiento vertical alguno en ese campo ( **$cont_D = cont_U = 0$** ), entonces se procede a realizar un movimiento hacia arriba (**se envía “U” vía puerto serial**) y se pone la bandera **BD** a **0**, este es un movimiento de inicio simplemente. Se **actualiza el valor de  $V_{fant}$** , tomando este el valor de  $V_{fn}$  (Ver figura 3.23).

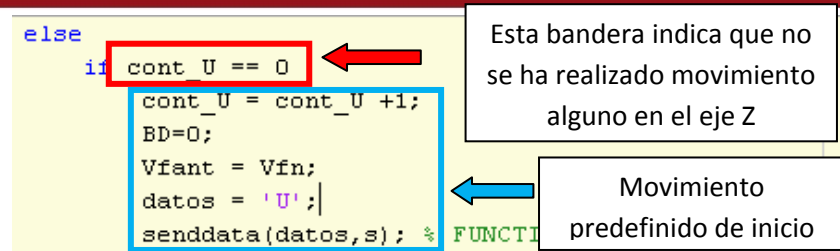


Fig. 3.23 Movimiento de inicio de enfoque automático.

- Luego se vuelve a obtener un valor **Vfn**, **si este es mayor que Vfant**, ello quiere decir que el enfoque mejora, por lo tanto el desplazamiento es el correcto, así que se mantiene la dirección ascendente, **se envía “U”** vía puerto serial y se **aumenta el contador cont\_U en 1**. Pero **si el valor de Vfn es menor**, se cambia de dirección ya que el enfoque comienza a empeorar, de manera que se **envía una “D”** vía puerto serial, **se aumenta en 1 el contador cont\_D** y se cambia el valor de **la bandera BD a 1** indicado **dirección Z negativa**. En ambos casos, luego del envío de la instrucción vía puerto serial, se suman los desplazamientos verticales (**cont\_D + cont\_U**), si estos son mayores al valor del **limitador** se procede a activar las banderas **Lim y foco\_encontrado (1)**. La primera bandera indica que se alcanzó el número de desplazamientos permitidos en el eje Z y el segundo simplemente simulará que se encontró una imagen enfocada (pero no es así) simplemente para realizar el paso a un siguiente campo.
- Luego, conforme se vaya avanzando, se mantendrá el sentido de desplazamiento si el enfoque mejora, o se cambiará de sentido si este empeora, siempre tomando en cuenta el **limitador**.

```

else
    cont_U = cont_U + 1;
    Vfant = Vfn;
    %% Transmitemos Dato 'D' indicando que se
    %% mueva un paso hacia abajo
    datos = 'U';
    senddata(datos,s); %% FUNCION #####
    %% Limitador de tomas:
    L = cont_D + cont_U;
    if L > limitador
        Lim = 1;
        foco_encontrado = 1;
    end
end
end
else
    if BD==1
        BD = 0;
        cont_U = cont_U + 1;
        Vfant = Vfn;
        %% Transmitemos Dato 'U' indicando que se
        %% mueva un paso hacia arriba
        datos = 'U';
        senddata(datos,s); %% FUNCION #####
        %% Limitador de tomas:
        L = cont_D + cont_U;
        if L > limitador
            Lim = 1;
            foco_encontrado = 1;
        end
    end
    else
        BD=1;
        cont_D = cont_D + 1;
        Vfant = Vfn;
        %% Transmitemos Dato 'D' indicando que se
        %% mueva un paso hacia abajo
        datos = 'D';
        senddata(datos,s); %% FUNCION #####
        %% Limitador de tomas:
        L = cont_D + cont_U;
        if L > limitador
            Lim = 1;
            foco_encontrado = 1;
        end
    end
end
end
end
end
end
end
end
end

```

Actualización del valor de foco anterior Vfant

Evaluación de movimientos verticales según el valor del limitador

Envío de instrucciones para el ajuste de enfoque de la lente

Evaluación y actualización del sentido de desplazamiento a través del eje Z

Fig. 3.24 Algoritmo codificado de Enfoque automático

### 3.4.2 Modo Manual:

Una vez planteada la lógica de movimientos de cada motor a través de tramas enviadas vía puerto serial, tramas que son códigos ASCII de letras específicas; se puede tomar estas instrucciones para que, a través de teclas correspondientes a las letras usadas, se pueda manejar el sistema con el fin de permitir una exploración, por parte del usuario, de la muestra que se va a analizar, sin tener la necesidad de tener contacto con el microscopio.

Partiendo de ello, se diseña el sistema manual de operación.

Se parte desde que en el Menú inicial se escoge la letra “M” de modo manual (Ver figura 3.25). En seguida aparece un mensaje de bienvenida, y luego configura la cámara web y se muestra una pre-visualización de la imagen enfocada en el microscopio; todo ello con la misma codificación usada par el modo automático.



Fig. 3.25 Mensaje de bienvenida Modo Manual

Luego, se muestra un menú en donde el sistema se mantiene a la espera de la instrucción que se quiera dar. Las teclas a usar son: **A, D, W, S, U, J y C**. Estas instrucciones de almacenan en la variable modo\_manual.

Si se presiona la tecla A (Movimiento eje X negativo) y luego “enter”:

- Se evalúa si el contador de movimientos en el eje X:  $i$ , está dentro del rango de movimiento permitido, es decir menor o igual que el valor de la variable **muestras<sub>i</sub>**, y mayor que el **mínimo valor de X: 1**. Si esto se cumple, se envía a través del puerto serial la letra “A” y se **disminuye en 1 el valor del contador  $i$** , sino se muestra un mensaje de error (Ver figura 3.26).

```

elseif modo_manual == 'A'
    %% MOVIMIENTO EJE X NEGATIVO :
    if (i >= 2) && (i <= muestras_i)
        i = i - 1;
        datos = 'A';
        senddata(datos,s); % FUNCTION #####
    else
        msgbox('MOVIMIENTO NO PERMITIDO : FUERA DE RANGO', 'ERROR EN EJE X', 'ERROR')
    end
end
    
```

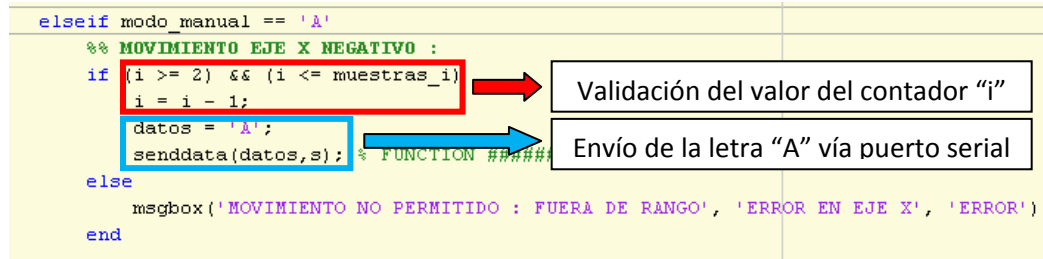


Fig. 3.26 Movimiento en eje X negativo

Si se presiona la tecla D (Movimiento eje X positivo) y luego “enter”:

- Se evalúa si el contador de movimientos en el eje X: i, está dentro del rango de movimiento permitido, es decir menor que el valor de la variable **muestras\_i**, y mayor que cero. Si esto se cumple, se envía a través del puerto serial la letra **“D”** y se **aumenta en 1 el valor del contador i**, sino se muestra un mensaje de error (Ver figura 3.27).

```

%% MOVIMIENTO EJE X POSITIVO :
if (i <= (muestras_i -1)) && (i > 0)
    i = i + 1;
    datos = 'D';
    senddata(datos,s); % FUNCTION #####
else
    msgbox('MOVIMIENTO NO PERMITIDO : FUERA DE RANGO', 'ERROR EN EJE X', 'ERROR')
end
    
```

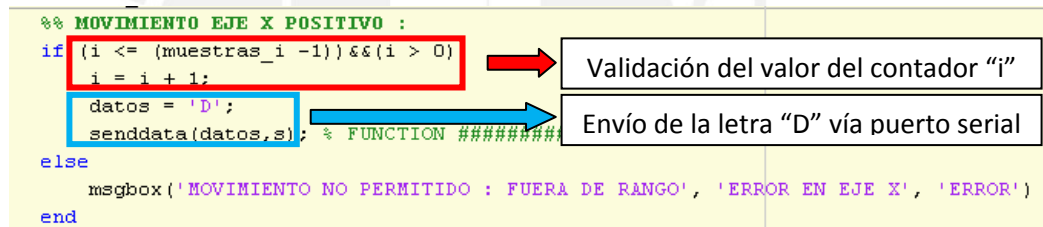


Fig. 3.27 Movimiento en eje X positivo

Si se presiona la tecla W (Movimiento eje Y positivo) y luego “enter”:

- Se evalúa si el contador de movimientos en el eje Y: j, está dentro del rango de movimiento permitido, es decir menor o igual que el valor de la variable **muestras\_j**, y mayor o igual que 2. Si esto se cumple, se envía a través del puerto serial la letra **“W”** y se **disminuye en 1 el valor del contador j**, sino se muestra un mensaje de error (Ver figura 3.28).

```

%% MOVIMIENTO EJE Y POSITIVO :
if (j >= 2) && (j <= muestras_j)
    i = i - 1;
    datos = 'W';
    senddata(datos,s); % FUNCTION #####
else
    msgbox('MOVIMIENTO NO PERMITIDO : FUERA DE RANGO', 'ERROR EN EJE Y', 'ERROR')
end
    
```

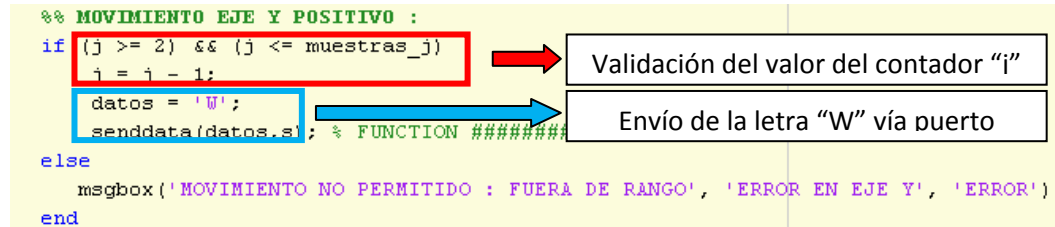


Fig. 3.28 Movimiento en eje Y positivo

Si se presiona la tecla S (Movimiento eje Y negativo) y luego “enter”:

- Se evalúa si el contador de movimientos en el eje Y: j, está dentro del rango de movimiento permitido, es decir menor o igual que el valor de la variable **muestras\_j menos 1**, y mayor o igual que 0. Si esto se cumple, se envía a través del puerto serial la letra “S” y se **incrementa en 1 el valor del contador j**, sino se muestra un mensaje de error (Ver figura 3.29).

```

%% MOVIMIENTO EJE Y NEGATIVO :
if (j <= (muestras_j -1) && (j > 0))
    j = j + 1;
    datos = 'S';
    senddata(datos,s); % FUNCTION #####
else
    msgbox('MOVIMIENTO NO PERMITIDO : FUERA DE RANGO', 'ERROR EN EJE Y', 'ERROR')
end
    
```

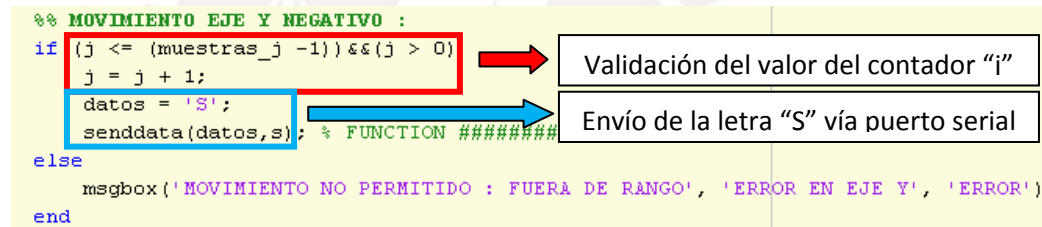


Fig. 3.29 Movimiento en eje Y negativo

Si se presiona la tecla U (Movimiento eje Z positivo) y luego “enter”:

- Simplemente se envía la letra “U” a través del puerto serial y se **incrementa el contador k en 1** (Ver figura 3.30).

Si se presiona la tecla J (Movimiento eje Z positivo) y luego “enter”:

- Simplemente se envía la letra “J” a través del puerto serial y se **disminuye el contador k en 1** (Ver figura 3.30).



```

elseif modo_manual == 'U'
    %% MOVIMIENTO EJE Z POSITIVO
    datos = 'U';
    senddata(datos,s);
    k = k + 1;
elseif modo_manual == 'J'
    %% MOVIMIENTO EJE Z NEGATIVO
    datos = 'J';
    senddata(datos,s);
    k = k - 1;
    
```

Envío de "U" y aumento del contador

Envío de "J" y decremento del contador

Fig. 3.30 Movimiento en eje Z negativo y Positivo

Si se presiona la tecla C (captura de imagen) y luego "enter":

- o Como la cámara ya está configurada se procede a la captura de la imagen, esta se procesa para quitar el efecto de la luz ambiental y se guarda con un nombre concatenado solo para las imágenes del modo manual, para ello previamente se ha **aumentado en 1** el valor de la variable **im** que me indica cuantas imágenes en este modo se han tomado. Luego se analiza la imagen con el algoritmo de evaluación de foco y como mensaje se envía el valor de foco que se obtiene y una visualización de la imagen capturada (Ver figura 3.31).

```

elseif modo_manual == 'C'
    %% CAPTURA DE IMAGEN
    % pause(3);
    I=getsnapshot(vid);
    img = ycbcr2rgb(I);
    im = im + 1;
    %% guardar imagen
    file=strcat('image_modo_manual',num2str(im));
    file=strcat(file, '.jpg');
    imwrite(img,file,'quality',100);
    imshow(img);
    %% ANALISIS DE FOCO
    I2= rgb2gray(I);
    [BW,thresh,gv,gh] = edge(I2,'sobel');
    gv1=abs(gv);
    gv2=abs(gh);
    Mfoco=plus(gv1,gv2); %suma de las dos matrices
    Vfoco=sum(Mfoco(:)); % suma de todos los valores de la matriz
    msje_foco = strcat('EL VALOR DE FOCO DE LA IMAGEN ES : ',num2str(Vfoco));
    uiwait(msgbox(msje_foco,'ANALISIS DE FOCO','HELP'));
    
```

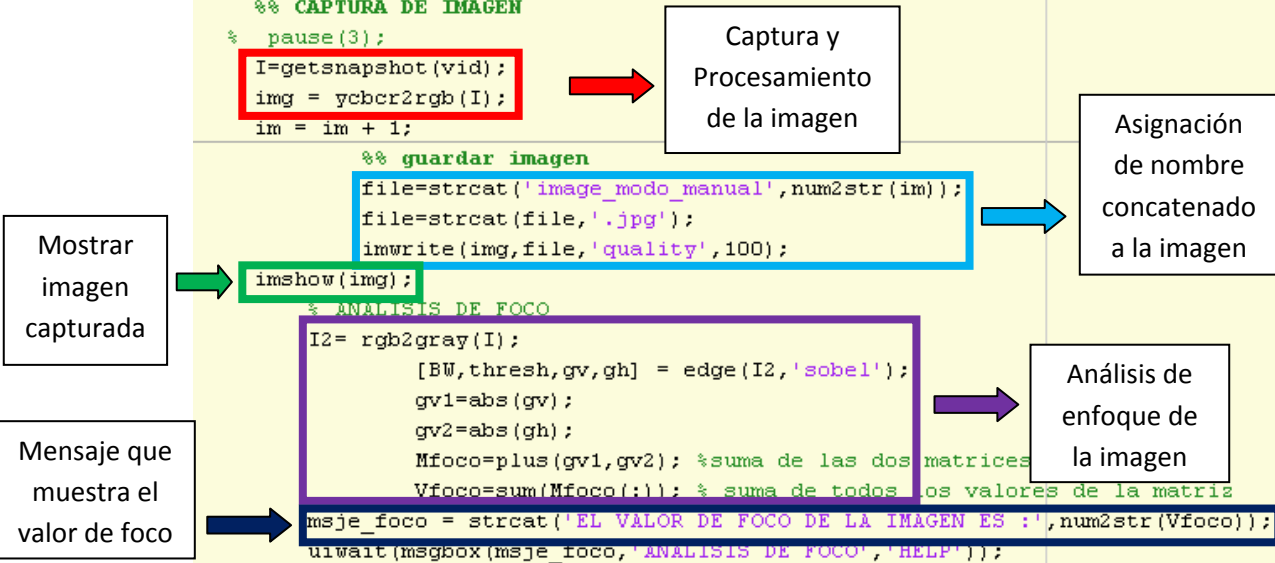


Fig. 3.31 Captura de imagen por instrucción: "C"

## CAPÍTULO 4

### EVALUACIONES Y RESULTADOS

En el presente capítulo se muestran las pruebas hechas al algoritmo realizado y como responde a las diversas pruebas a las que se le ha sometido. Básicamente lo que se busca analizar es el tiempo de respuesta del software y como este maneja el desplazamiento correcto a través del plano XY y el autoenfoco que realiza para la búsqueda de imágenes claras. Asimismo, se evalúa la capacidad de almacenamiento que tiene y el orden para manejar los subsistemas (cámara, motores y software de análisis de foco) en el modo automático.

#### **4.1 Tiempo de Operación del sistema y Cantidad e imágenes fallidas según los parámetros: Tolerancia y Limitador.**

En el capítulo 3 se explicó claramente que parámetros regían el movimiento a través del eje Z para la búsqueda de una imagen clara. Estos parámetros eran: **tolerancia y limitador**. El primero indicaba el rango en el que se consideraba, a partir del valor de foco de referencia, si una imagen era válida o no, es decir, si el valor de foco era lo suficientemente alto para tener una imagen nítida. El segundo parámetro limitaba la

cantidad máxima de movimientos en el eje Z para la búsqueda de una imagen correctamente enfocada.

En base a ello se realizan las siguientes pruebas, pues se varían la tolerancia y el limitador par ver el tiempo de operación del sistema.

Para cada caso, se analizan 40 campos en la muestra, tomando en cuenta un valor de referencia de foco igual a 1650 (sin unidades, según tamaño de imagen, en este contexto: 640x480 píxeles, un valor de foco de 1650 indica una imagen clara ), en base a ello se varían la tolerancia y el limitador de movimientos verticales: el tiempo se toma partir de la adquisición de la primera imagen hasta el momento en que aparece el mensaje de “operación finalizada”. (Ver Tabla. 4.1)

Tabla 4.1 Tiempo de análisis (en minutos y segundos) de 40 muestras con valores variables de Tolerancia y Limitador.

		TOLERANCIA				
		50	75	100	125	150
LIMITADOR	10	02:01	02:04	01:58	01:54	01:51
	15	01:58	02:00	01:55	01:50	01:49
	20	01:58	01:59	01:46	01:46	01:47
	25	01:57	01:49	01:48	01:47	01:48
	30	01:54	01:48	01:45	01:47	01:46

Como se ve en la Tabla 4.1, cuando la tolerancia es pequeña el tiempo se incrementa pues, es difícil encontrar una imagen correctamente enfocada, eso se puede dar porque el sistema realiza todos los movimientos verticales permitidos y no encuentra una imagen clara, allí es donde se pierde tiempo. Eso ocurre cuando la tolerancia es 50 y 75, sin embargo cuando la tolerancia es 100 o mayor, el tiempo de análisis se reduce haciendo el proceso más eficiente. El cuadro rojo dentro de la tabla indica los mejores tiempos que se pueden lograr; esto se da cuando el limitador tiene un valor entre 20 y 30 y la tolerancia está entre 100 y 150. Es necesario resaltar que es casi imperceptible para el operario la variación de un valor de foco de 50 por lo que si se tiene una imagen de valor de foco 1650 y otro de 1600, la variación de claridad no será notada; lo mismo sucede con los valores de tolerancia de 75 y 100, pero si se tiene una

variación de valor de foco de mayor a 100, la claridad de la imagen si se ve ligeramente menguada. (Ver Figuras 4.1, 4.2 y 4.3)

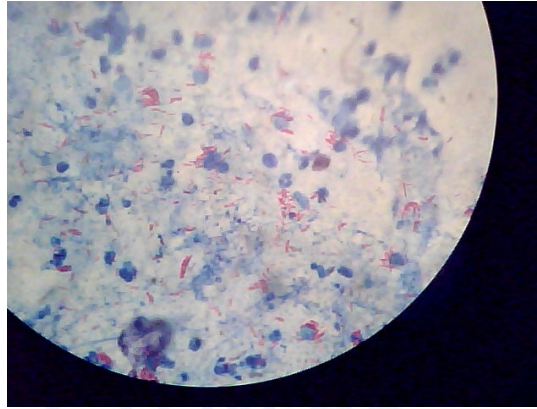


Figura 4.1 Imagen con valor de foco: 1680.23

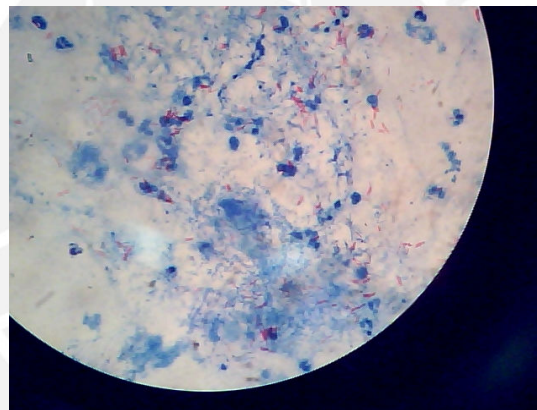


Figura 4.2 Imagen con valor de foco: 1795.26

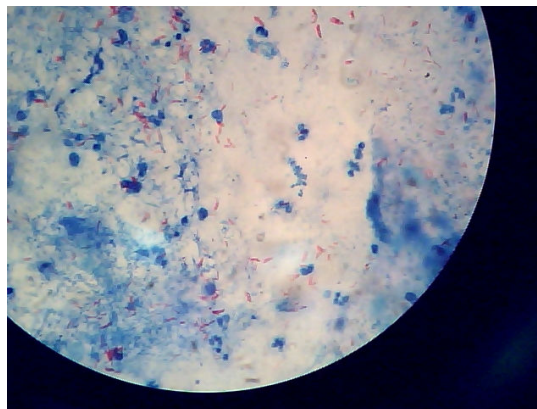


Fig. 4.3 Imagen con valor de foco: 1852.37

Asimismo se contabilizan la cantidad de imágenes fallidas en cada 40 muestras, ya que puede darse el caso de que debido a que se requiere un alto valor de precisión de valor de foco (valor pequeño de tolerancia), el limitador no sea lo suficientemente alto para dejar realizar la cantidad necesaria de movimientos para un enfoque correcto. (Ver Tabla 4.2)

Tabla 4.2 Número de imágenes fallidas en 40 muestras con valores variables de Tolerancia y Limitador

		TOLERANCIA				
		50	75	100	125	150
LIMITADOR	10	11	8	4	1	1
	15	9	6	3	1	0
	20	5	2	1	0	0
	25	3	1	0	0	0
	30	2	0	0	0	0

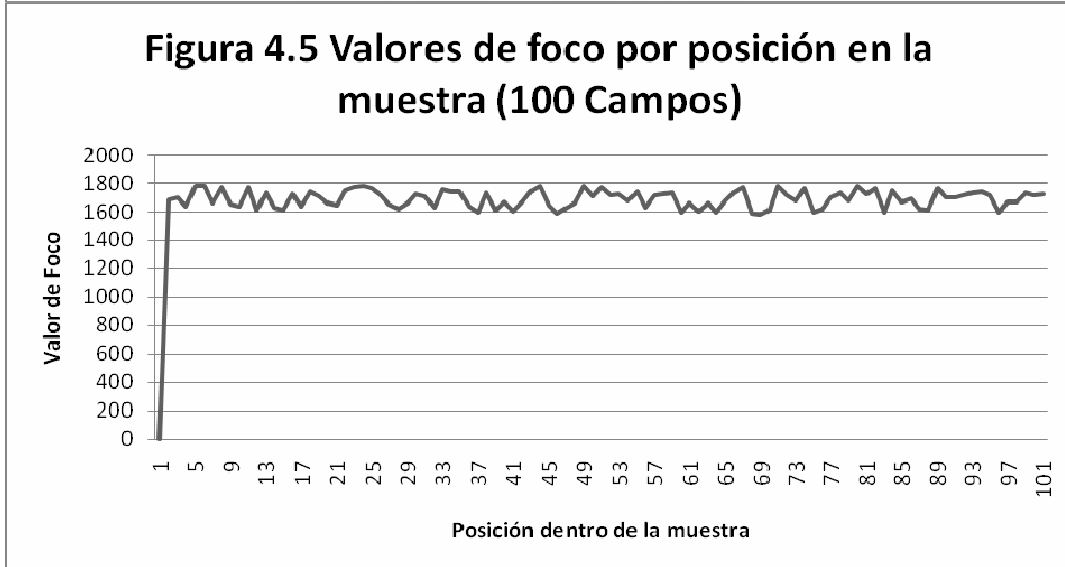
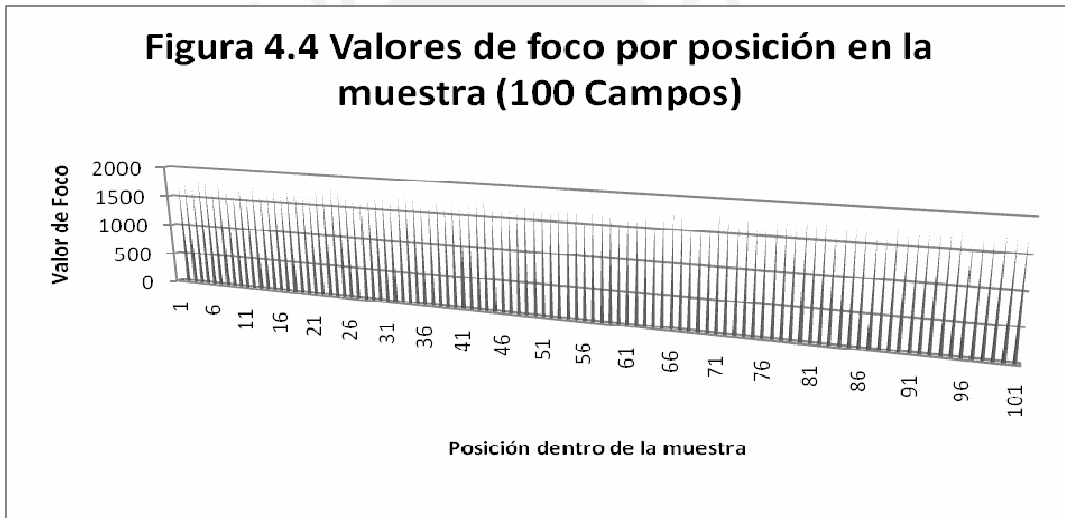
Como se observa en la tabla 4.2, cuando se tienen valores de tolerancia y de limitador pequeños, es más probable obtener imágenes fallidas ya que el sistema buscará una imagen muy cercana a la de referencia pero con un número muy limitado de movimientos. Por otra parte, si se tienen valores de tolerancia y limitador altos, el sistema no encuentra errores de imagen. Sin embargo es necesario tener en cuenta que a mayor tolerancia, se permite la captura de imágenes no tan claras, entonces, es necesario encontrar un punto de equilibrio que permita tener imágenes suficientemente claras bajo un tiempo de análisis aceptable. Según las tablas 4.1 y 4.2 el valor de tolerancia 100 y el valor de limitador 25 permiten tener imágenes claras, sin riesgo a imágenes fallidas, bajo un tiempo de 01:46 minutos para 40 muestras analizadas, es decir, si se realiza una prueba para lograr el análisis de 300 muestras (número máximo de muestras que puede analizar el sistema) esto tomará un tiempo promedio de 13:15 minutos.

En conclusión, según las pruebas realizadas, los valores óptimos de tolerancia y limitador son 100 y 25 respectivamente. Estos valores permiten un correcto enfoque en un tiempo tolerante, como se explicará mas adelante.

### 4.2 Valores de Foco promedio por Cantidad de Muestras Totales

Se procede a analizar procesos integrales (Tolerancia: 100, limitador: 25), es decir procesos que implican 100, 200 y 300 campos, según ello se ven los valores de foco promedios por campo. Esto permite ver cuán eficiente es el sistema para encontrar imágenes correctamente enfocadas y cuanto tiempo toma un proceso completo, según la cantidad de campos que se quiera analizar en ambos ejes: X e Y, cabe recordar que estos valores son ingresados por el usuario con una máximo de 10 campos en eje Y, y 30 en el eje X.

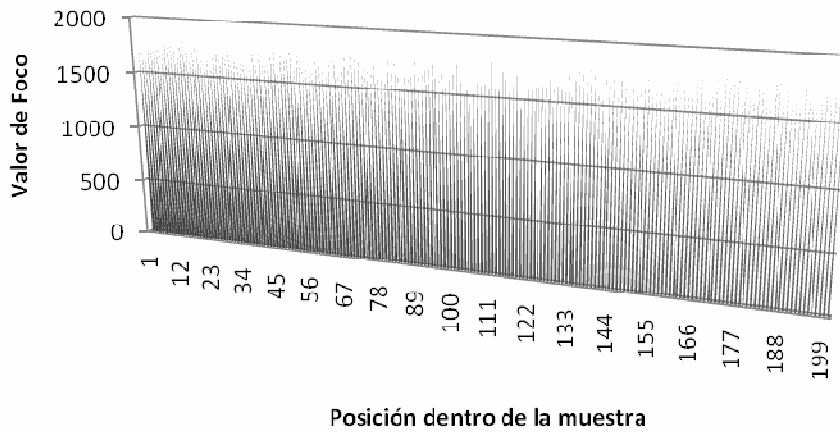
**Análisis de 100 Campos (10 en eje X y 10 en eje Y, Tolerancia = 100 , limitador = 25)**



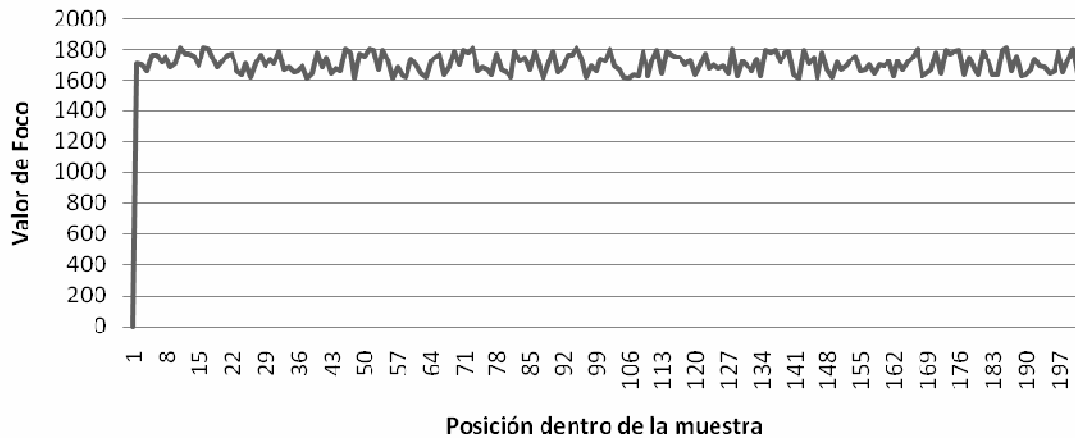


Análisis de 200 Campos (20 en eje X y 10 en eje Y, Tolerancia = 100 , limitador = 25)

**Figura 4.6 Valores de foco por posición en la muestra (200 Campos)**

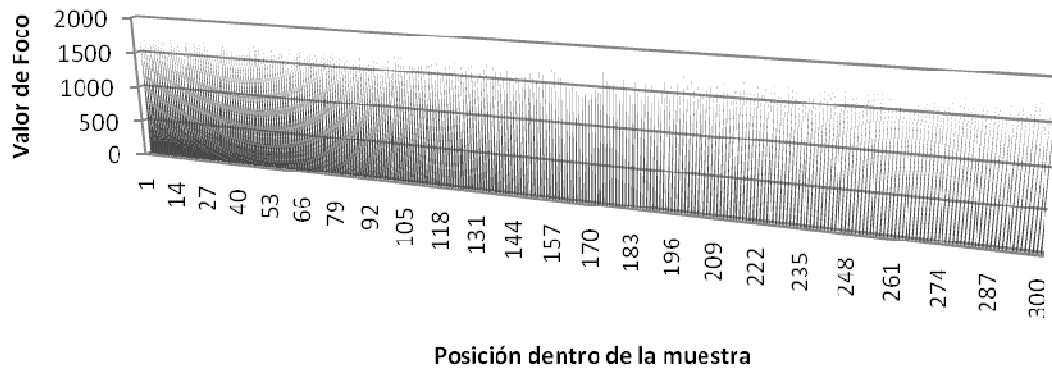


**Figura 4.7 Valores de foco por posición en la muestra (200 Campos)**

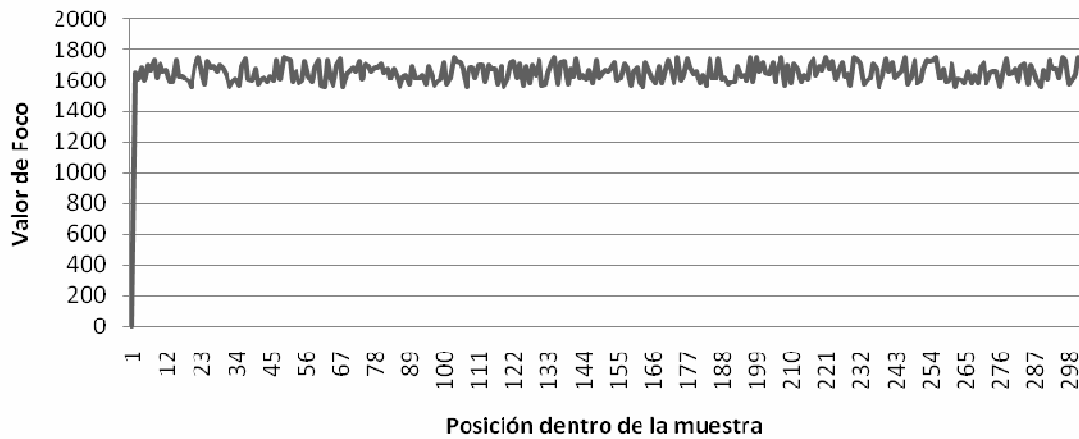


Análisis de 300 Campos (30 en eje X y 10 en eje Y, Tolerancia = 100 , limitador = 25)

**Figura 4.8 Valores de foco por posición en la muestra (300 Campos)**



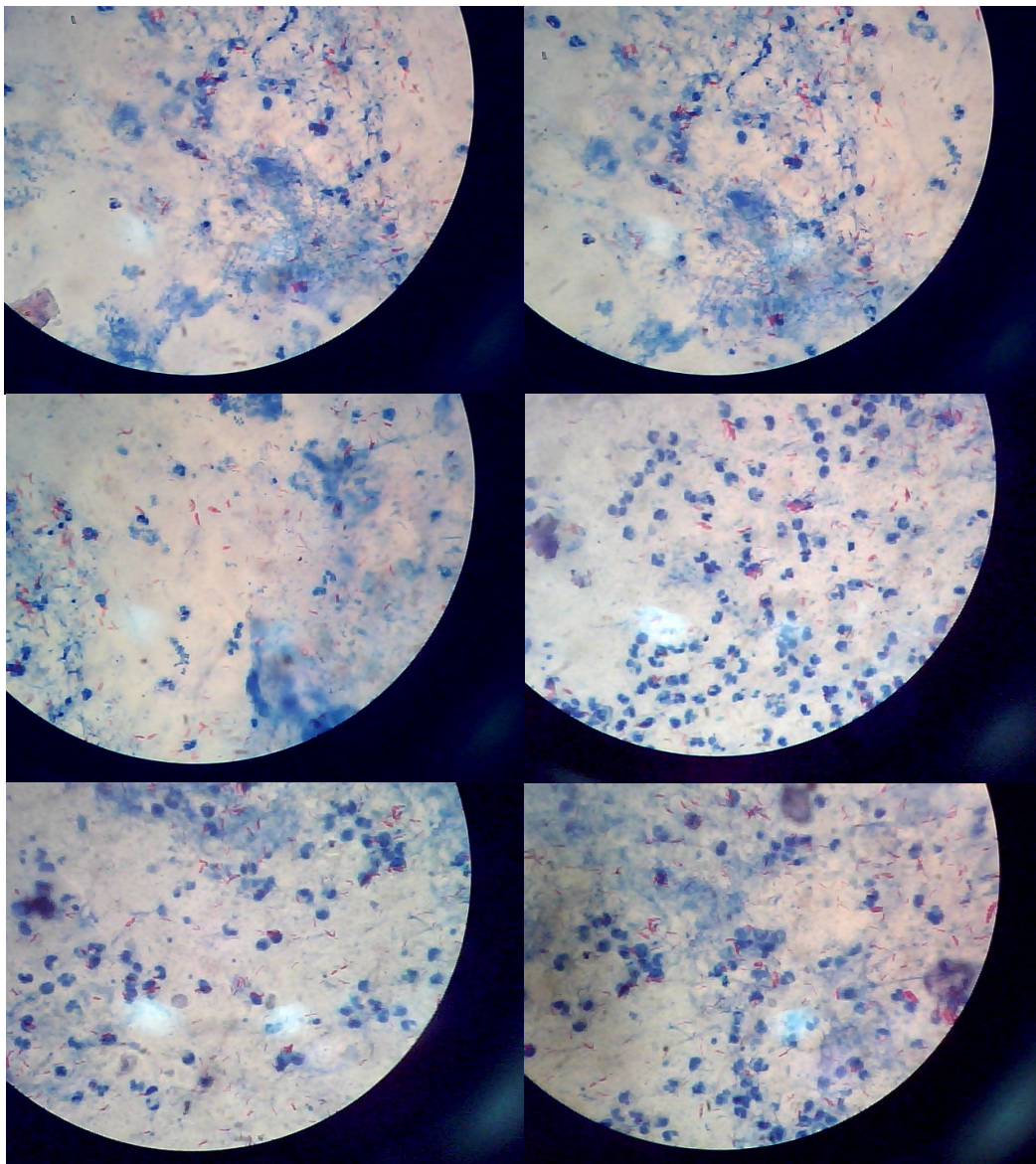
**Figura 4.9 Valores de foco por posición en la muestra (300 Campos)**



Entonces, en todas las gráficas se verifica el correcto enfoque y la generación de imágenes claras, ello quiere decir que el programa realiza la operación de enfoque de manera correcta cumpliendo con los parámetros de Tolerancia y Limitador.

En conclusión, se puede decir que el resultado de las evaluaciones verifican el correcto funcionamiento del algoritmo implementado, cumpliendo con los márgenes de claridad deseadas. Los valores de foco indican la claridad de la imagen, bajo esa premisa se verifica que en todas las pruebas realizadas se obtuvieron valores de foco que oscilaban entre 1550.12 a 1790.23 según un punto de referencia de de 1650, lo que implica una nitidez alta de las imágenes en toda la muestra.

A continuación, se muestran parte de las imágenes obtenidas de una muestra analizada:



**CONCLUSIONES:**

- Un Valor de foco implica un número adimensional, que describe la claridad de la imagen, según la sección 4.1 un valor de foco aceptable para una imagen de 640 x480 píxeles varía entre 1580 y 1790. Bajo esos conceptos, se puede decir que se logró el enfoque del 95% de las imágenes tomando un margen de error de enfoque (Tolerancia) de 100. Lo que implica un grado aceptable de claridad de la imagen teniendo en cuenta que el valor de foco promedio es de 1680.
- El movimiento a través del plano XY se realiza correctamente, (concatenando movimientos en X y en Y), en el caso más crítico, con 30 campos en el eje X y 10 en el eje Y, lo que implica 300 campos analizados.
- El tiempo de análisis promedio es de 40 imágenes en 1 minuto y 46 segundos, lo que hace un promedio general de 13 minutos y 15 segundos, por cada 300 campos. Un tiempo aceptable con respecto al usado por un operador en el área de baciloscopia de un hospital convencional (20 minutos para 300 campos analizados).

A continuación, se muestran los posibles alcances que tendría la inserción del sistema en un área de trabajo convencional:

- El sistema permitirá una reducción en el tiempo que toma el barrido de las muestras mediante un proceso convencional realizado por el operador (técnico) del microscopio. Esto permite que el operador aproveche dicho tiempo para realizar tareas de mayor complejidad.
- La reducción de tiempo implica un rápido diagnóstico, lo que conllevaría a una eficiente atención del paciente y a la pronta toma de medidas respecto a su enfermedad.

- La implementación del modo manual del sistema permite tener una posible aplicación en telemedicina debido a que el usuario puede controlar el microscopio mediante el interfaz dentro de la computadora.



### RECOMENDACIONES:

- El sistema tendría una mayor eficiencia si se trabajara con un microscopio trinocular, ya que la adaptación de una cámara sería más fácil, permitiendo una mayor precisión en la adaptación mecánica, una mejor calidad de imagen y una menor exigencia en el proceso de enfoque.
- El sistema puede ser implementado en otro software, mediante otro sistema operativo que permita el uso de programas de licencia libre como LINUX®, debido a que el uso del software Matlab® y los Toolbox correspondientes implica un gasto extra.





## BIBLIOGRAFIA

- [1] David Pertuz Arroyo, Héctor R. Ibañez Grandas. Sistema de adquisición automática de imágenes para microscopio óptico. *Revista Ingeniería y Desarrollo*, 2007.  
[http://www.scielo.org.co/scielo.php?pid=S0122-34612007000200003&script=sci\\_arttext&tlng=en](http://www.scielo.org.co/scielo.php?pid=S0122-34612007000200003&script=sci_arttext&tlng=en)
- [2] R. Nava, G. Cristóbal, E. Gil, B. Escalante-Ramírez, M. Desco. Sistema automático para la adquisición y análisis de imágenes. XXVI Congreso Anual de la Sociedad Española de Ingeniería Biomédica. Valladolid.2008.  
[http://hggm.es/image/pdfs/CASEIB%202008\\_pp%20493-496.pdf](http://hggm.es/image/pdfs/CASEIB%202008_pp%20493-496.pdf)
- [3] Asencios L, Quispe N, Sanabria H, Yi Chu A. Manual de Normas y Procedimientos en Bacteriología de Tuberculosis. Serie de Normas Técnicas N° 10 Ministerio de Salud, Instituto Nacional de Salud. Red Nacional de Laboratorios de Salud.1995. 17-30.
- [4] Organización Panamericana de la salud. Manual para el diagnóstico bacteriológico de la tuberculosis.2008 <http://www.paho.org/Spanish/AD/DPC/CD/tb-labs-bacilosopia.pdf>
- [4] MathWorks, Inc. Data Acquisition Toolbox 2.16. 2010.  
<http://www.mathworks.es/products/daq/>
- [5] MathWorks, Inc. Image Acquisition Toolbox 3.5. 2010.  
<http://www.mathworks.es/products/imaq/>
- [6] A.Gilat. Matlab: una introducción con ejemplos prácticos. Barcelona, 2006.
- [7] J. Romero, M. Dolores, R. Zalaya (2002). Fundamentos Matemáticos par la Ingeniería con Matlab. Valencia.
- [8] S. Attaway (2009). MATLAB a practical approach. Canada
- [9] B. Hahn, D. T. Valentine (2007). ESSENTIAL MATLAB for engineers and scientists, third ed. Italy.
- [10] B.R. Hunt, R.L. Lipsman, J.M. Rosemberg (2003). A guide to Matlab for beginners and experienced users, first ed. United States of America.