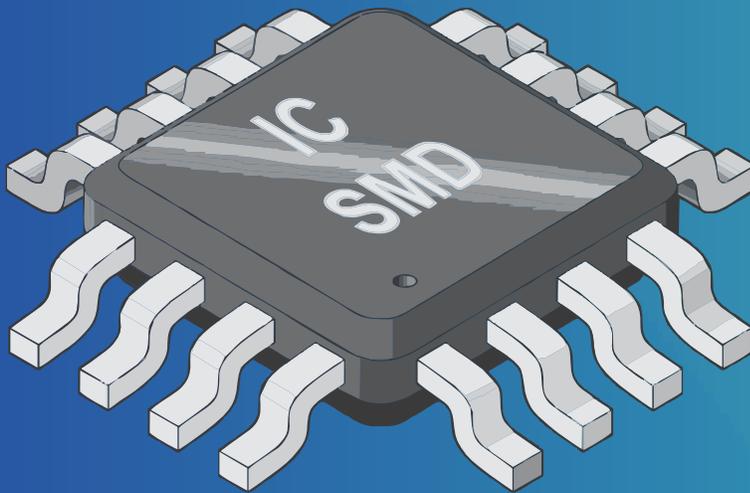




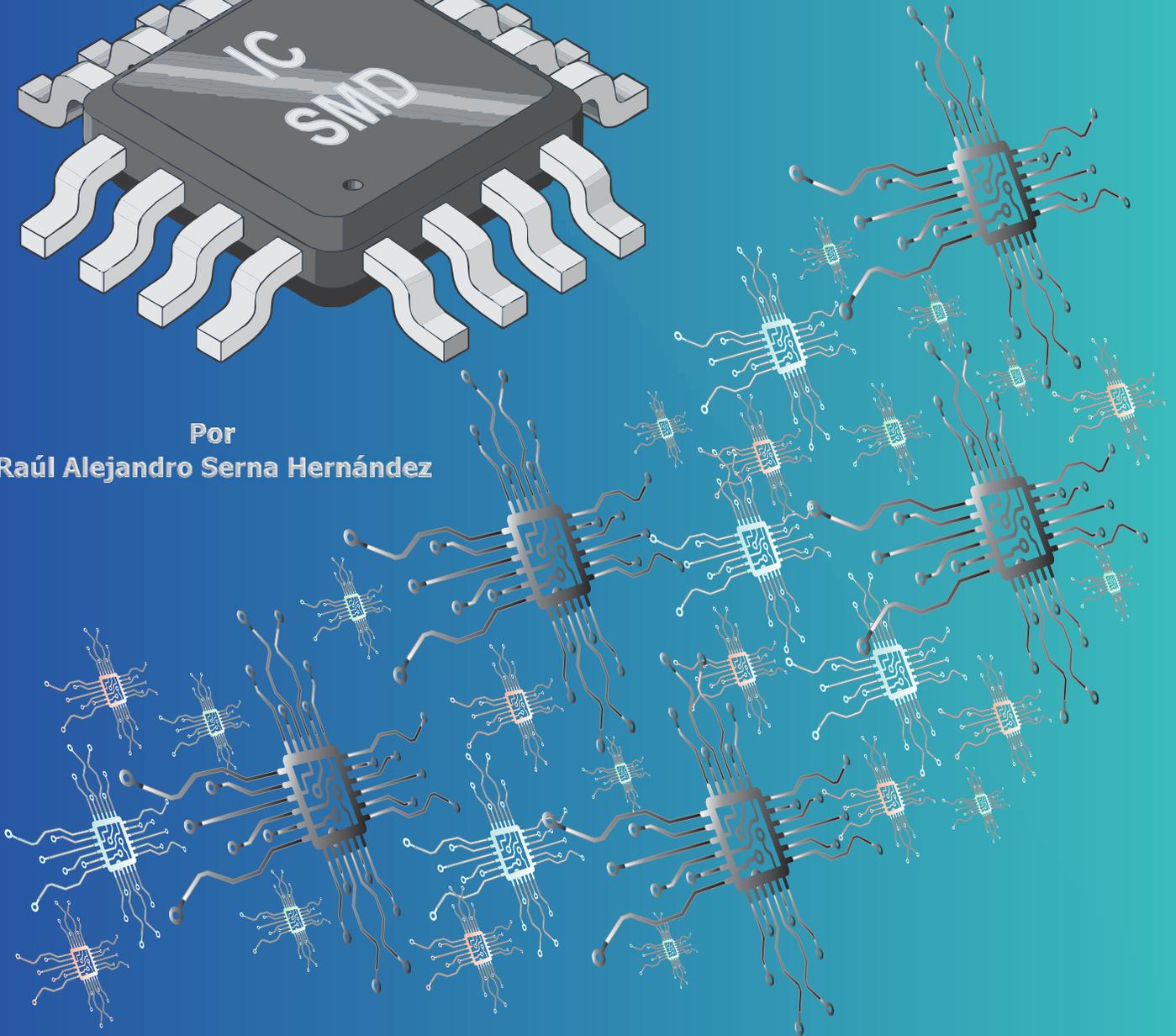
Universidad Autónoma del Estado de México
UAEM



Sistema de Control por procesamiento de imágenes de una máquina CNC para soldar circuitos electrónicos



Por
Raúl Alejandro Serna Hernández





UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
FACULTAD DE INGENIERÍA



**“Sistema de Control por procesamiento de imágenes de una máquina
CNC para soldar circuitos electrónicos”**

Tesis que para obtener el título de
Ingeniero en Electrónica

Presenta

Raúl Alejandro Serna Hernández

Asesor: **Dr. Javier Salas García**

Coasesor: **Dra. Mireya Salgado Gallegos**

RESUMEN

La presencia de circuitos electrónicos y la miniaturización de los mismos a través del uso de placas de circuito impreso PCB es actualmente parte integral de la actividad diaria en la sociedad. Por tanto, surge una necesidad para buscar un punto de equilibrio entre la calidad del producto final y las técnicas de inspección usadas durante el proceso de manufactura; es aquí donde los sistemas de inspección óptica automatizada AOI (Automated Optical Inspection) surgen como una alternativa viable para conseguir dichos estándares. Esta tesis propone que, mediante la generación de condiciones de iluminación idóneas a través de elementos de optoelectrónica y potencia junto con la aplicación de herramientas de procesamiento digital de imágenes que comprenden técnicas de detección de bordes, análisis de valores de intensidad y reconocimiento de patrones de reflexión es posible realizar un proceso de inspección de la calidad de uniones de soldadura mediante la interpretación del comportamiento gráfico de histogramas y el establecimiento de criterios de aceptabilidad para cada plano de color RGB, evaluados a través de una herramienta de programación como MATLAB; y de esta forma considerar el diseño y construcción del hardware y software desarrollado en el presente trabajo como un sistema de control óptico que de manera automática inspecciona y clasifica la calidad de soldadura de los dispositivos electrónicos con tecnología SMT presentes en una PCB.



"Desearía que no hubiera tenido que pasar en mi época", dijo Frodo.

"Yo también", dijo Gandalf,

"y todos los que viven para ver tiempos así. Pero no está en ellos decidir.

Todo lo que tenemos que decidir es qué hacer con el tiempo que se nos da"

J.R.R. Tolkien (1892-1973)

ÍNDICE

1	INTRODUCCIÓN	
1.1	Introducción a técnicas de inspección de uniones de soldadura en circuitos electrónicos	2
1.2	Panorama general de los sistemas de inspección óptica automatizada (AOI) ..	4
1.3	Objetivos del sistema de control	6
1.4	Hipótesis	7
1.5	Organización de la tesis	7
2	ANTECEDENTES	
2.1	Introducción	9
2.2	Estudio comparativo de sistemas AOI comerciales	10
2.3	Metodología del procesamiento de imágenes en aplicaciones CNC para soldar circuitos electrónicos	13
2.4	Requerimientos de un sistema de control por procesamiento de imágenes aplicado a procesos de soldadura e inspección de PCB's	15
3	METODOLOGÍA Y MATERIALES	
3.1	Descripción general del sistema	16
3.2	Sistema de detección y obtención de imágenes	19
3.2.1	Cámara	19
3.2.2	Arreglo circular de iluminación LED	21
3.2.3	Patrones de iluminación y reflexión	22
3.2.4	Especificaciones del sistema electrónico de alimentación	26
3.2.5	Algoritmos de funcionamiento	29
3.2.5.1	Programación PIC	31

3.2.5.2	Programación PC.....	33
3.2.6	Tarjeta de adquisición de datos	34
3.3	Interfaz con el usuario	35
3.3.1	Comunicación entre la PC y la tarjeta de adquisición de datos.....	37
4	RESULTADOS Y DISCUSIÓN	
4.1	Pruebas y observaciones del sistema de detección	38
4.2	Pruebas al sistema electrónico de alimentación	73
4.3	Desarrollo futuro	76
5	CONCLUSIONES	77
	BIBLIOGRAFÍA	81
	APÉNDICES	
A.	Programas informáticos	83
A.1	En el microcontrolador (PIC 18F4550)	83
A.2	En la PC	86

1. INTRODUCCIÓN

La industria de la fabricación de circuitos electrónicos ha experimentado un constante desarrollo durante las últimas décadas, así como una marcada evolución en las técnicas de manufactura e inspección relacionadas a los mismos. Derivado de este hecho, la utilización de circuitos electrónicos está presente en un sinnúmero de aplicaciones dentro de la actividad diaria del hombre, lo que ha provocado la demanda de diseños de alta complejidad, y por consecuencia con niveles de miniaturización y calidad elevados, lo cual se ha convertido en un requerimiento básico dentro de la industria en general.

La finalidad del presente capítulo se enfoca en ofrecer un panorama de las técnicas de inspección de uniones de soldadura en circuitos electrónicos disponibles actualmente, sus ventajas, desventajas y de la aplicación de los sistemas de inspección óptica automatizada *AOI (Automated Optical Inspection)* dentro de dichos procesos.

Con el planteamiento de los objetivos se mostrará cuán necesaria es la utilización de sistemas de inspección en procesos de manufactura, especialmente en la fabricación de placas de circuito impreso *PCB (Printed Circuit Board)*; y la motivación del proyecto de tesis para desarrollar un sistema automático, eficiente y de bajo costo, aplicado a un proceso de este tipo, en primera instancia para la detección de fallas y la inspección final del proceso de soldadura de los componentes de una PCB, así como de su posible implementación futura en el proceso de soldadura como un elemento de control automático.

1.1 INTRODUCCIÓN A TÉCNICAS DE INSPECCIÓN DE UNIONES DE SOLDADURA EN CIRCUITOS ELECTRÓNICOS

La inspección, entendida como el acto de observar algo cuidadosamente con el fin de obtener información, es un elemento clave en cualquier proceso de manufactura electrónica, incluyendo la fabricación de PCB's. En esta industria ha proliferado el uso de la tecnología SMT (*Surface Mount Technology*), que permite la disminución del tamaño de los componentes y el descenso de la densidad de empaquetado de las PCB's (Hao et al., 2013).

Un proceso de manufactura de PCB's puede consistir en cinco operaciones principales, las cuales son: 1) aplicación de soldadura, 2) colocación de componentes, 3) soldadura de uniones, 4) inspección de placa y 5) limpieza de placa; la Figura 1.1 muestra las etapas de dicho proceso (Ho and Ji, 2007).

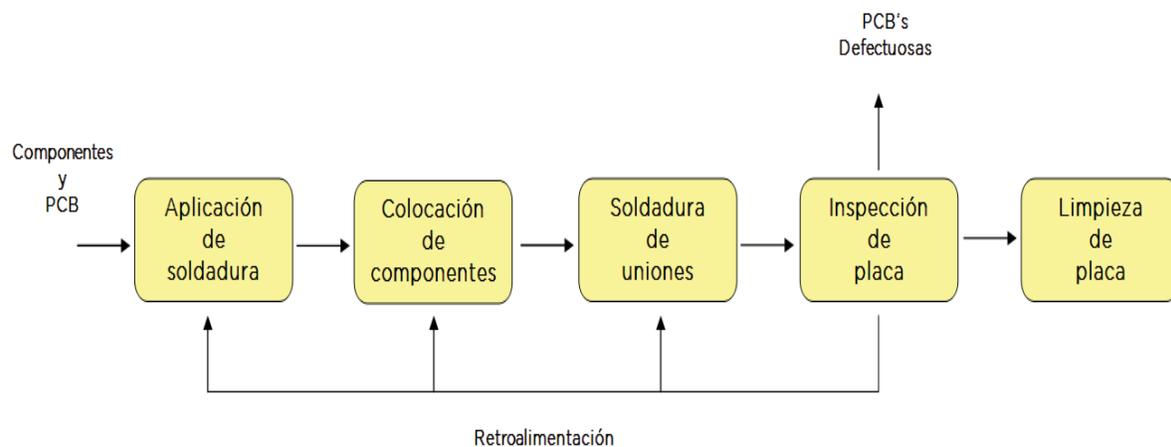


Figura 1.1 Diagrama a bloques de las etapas de un proceso de manufactura de PCB's.

La importancia de la inspección y la detección anticipada de errores fabricación antes de las fases posteriores de producción, cobra especial relevancia debido a que la detección y reparación de fallas aumenta de manera sustancial cada que el proceso supera una etapa determinada.

Los sistemas de inspección de circuitos electrónicos tienen como objetivo minimizar el número de fallas /defectos y la reducción de costos relacionados a la manufactura de sus productos. Dentro de los procesos comerciales de ensamblado de PCB's, la inspección de uniones de soldadura es una tarea esencial para cumplir con los objetivos mencionados, debido a que los componentes electrónicos dependen de las uniones de soldadura para proveer de conexión eléctrica hacia la PCB; por lo tanto, la calidad de la unión de soldadura es crítica para la calidad del producto final fabricado con dichos componentes electrónicos (Lee and Lo, 2002).

Debido a los requerimientos de la industria electrónica previamente mencionados, como lo son los diseños de alta complejidad y la miniaturización de circuitos, ha sido necesario modernizar también las técnicas aplicadas a los sistemas de inspección de uniones de soldadura. En primera instancia, el método de inspección manual directa ha quedado rebasado por las razones expuestas, siendo útil sólo en escenarios donde una falla general sea muy notoria. Algunos de los métodos utilizados para implementar sistemas de inspección en uniones de soldadura se pueden clasificar en: 1) pruebas eléctricas, 2) inspección de rayos X, 3) inspección térmica, y 4) inspección acústica (Yang, 2008, Yang and Ume, 2008); clasificación a la cual se añaden los sistemas de inspección óptica automatizada AOI (*Automated Optical Inspection*).

El proceso de inspección mediante pruebas eléctricas es útil para la detección modular de fallas mediante el uso de estímulos eléctricos controlados y la comparación de la respuesta con un rango permitido de resultados esperados (Tummala, 2000). Sin embargo, no es posible determinar de manera específica qué componente presenta deficiencias de soldadura, necesitando así de dos etapas, una de prueba y otra de localización concreta del defecto. La inspección por rayos X tiene su aplicación principalmente en los circuitos de múltiples capas y en arreglos de soldadura donde las uniones de soldadura no son visibles directamente, como ocurre en la fabricación de microprocesadores con arreglos *BGA (Ball Grid Array)*; presentan un costo elevado propio de un sistema tan específico, dificultad para la interpretación de las imágenes obtenidas (múltiples capas), baja velocidad de inspección y tiempos de adquisición de información relativamente largos. La inspección térmica se basa en la generación de imágenes a partir

de la radiación infrarroja emitida por un objeto y la temperatura asociada al mismo. La inspección acústica es un método no destructivo basado en emisiones y reflexiones sónicas usadas para detectar discontinuidades en uniones de soldadura. Ambas técnicas, en apariencia están limitadas a procesos de inspección, al ser complicada la interpretación de la información obtenida para su uso en un sistema de control del proceso de soldadura en tiempo real.

1.2 PANORAMA GENERAL DE LOS SISTEMA DE INSPECCIÓN ÓPTICA AUTOMÁTIZADA (AOI)

Los sistemas de inspección automática consisten en un método de análisis de la calidad de uniones de soldadura que se caracteriza por usar técnicas no destructivas y sin contacto. Este tipo de sistemas tienen como objetivo la detección y localización de cualquier defecto potencial en las uniones de soldadura que comprometan el funcionamiento de un circuito electrónico, como lo es una PCB. Para la implementación de los mismos, se requiere de la extracción de información proveniente de las uniones de soldadura en cuestión y de la superficie sobre la cual se encuentran las mismas (Mar et al., 2011).

Los estudios relacionados a la inspección de uniones de soldadura mediante métodos visuales automáticos inició durante los años 80's donde se reconoció a las uniones como el medio encargado de proveer conexión eléctrica entre dispositivos y circuitos electrónicos. Los dispositivos de tecnología SMT, hacen uso de las uniones de soldadura para conectar entre sí las patillas de los dispositivos y los pads de soldadura en una PCB; se entiende como pad, de su nombre en inglés "*land*", a una porción de un patrón conductivo usado generalmente para la conexión y/o unión de componentes (IPC, 2005). Por lo tanto, la calidad de las uniones de soldadura depende de la forma del flujo de soldadura alrededor del perímetro de la patilla del componente (Kim and Cho, 1995).

Como punto de partida para la aplicación de sistemas AOI, se recurre a clasificaciones que permiten determinar la calidad de las uniones de soldadura a inspeccionar, y los defectos asociados con éstas. Una clasificación aplicable a un sistema AOI para PCB's sería la siguiente: 1) soldadura aceptable/buena, la cual se puede subclasificar en: a) insuficiente pero aceptable, b) aceptable, c) excesiva pero aceptable; 2)

soldadura insuficiente, 3) soldadura en frío, 4) componente desplazado y 5) levantamiento tipo lápida (Hao et al., 2013, Kim and Cho, 1995).

Clasificaciones como la mencionada anteriormente, surgieron mediante técnicas de estimación usadas para medir la inclinación de la superficie que presentan las uniones de soldadura, usando arreglos de iluminación para obtener imágenes en escalas de grises, mediante las cuales fue posible medir los ángulos de inclinación de dichas uniones, en donde la calidad de la unión está determinada por dicho ángulo y por la forma de la misma; presentando una forma cóncava en las soldaduras aceptables, mientras que las inaceptables/malas presentaban un forma convexa (Loh and Lu, 1999).

La etapa siguiente de implementación de un sistema AOI se centra en la extracción de información proveniente de las uniones de soldadura y de los defectos potenciales que se pueden presentar en el proceso de manufactura. La operación automática de los sistemas AOI y el ser caracterizados como una técnica sin contacto, se deben al hecho de utilizar cámaras digitales, ya sean fotográficas o de video, como elemento central para la adquisición de imágenes provenientes de las uniones de soldadura. Este método basa su operación en el principio de la reflexión especular, que dentro del campo de la óptica se rige por las leyes de reflexión. En el caso específico de un circuito electrónico, las patillas de un componente SMT y las uniones de soldadura, pueden ser tratadas como superficies especulares, las cuales se comportan como un espejo plano que permite la generación de patrones de reflexión, lo cuales pueden ser detectados a través de un cámara (Hao et al., 2013). Los patrones de reflexión generados son acentuados mediante arreglos de iluminación adicionales; los cuales se forman usando fuentes de luz tanto monocromática como del tipo *RGB (Red Green Blue)*, las cuales según las leyes de reflexión, son reflejadas hacia la cámara en un ángulo específico determinado por la pendiente que presenta la superficie de la unión de soldadura a inspeccionar.

La finalidad de utilizar un sistema de iluminación dedicado es mantener niveles de iluminación suficientes que permitan revelar las regiones oscuras en la imagen obtenida de una unión de soldadura (Mar et al., 2011).

Un elemento relacionado con la reflexión de emisiones de luz sobre una PCB que facilita la adquisición y extracción de información de las imágenes generadas, es que tanto el empaquetado plástico de los chips SMT como la tablilla base de la PCB, difunden aleatoriamente los rayos de luz incidente en todas direcciones; por tanto, mezclan cada haz de color incidente, provocando que la cámara perciba el patrón de reflexión como si estuviera en su color natural en luz blanca (Kim and Cho, 1995).

Es en este punto cuando se observa cómo un sistema de inspección AOI aplicado a un proceso de manufactura de PCB's se puede implementar mediante el uso de un conjunto de elementos formado por el dispositivo de adquisición de imagen (cámara digital), el arreglo de iluminación y el dispositivo de procesamiento de los datos adquiridos por todo el sistema, como lo es una PC. Esto hace posible extraer información de las uniones de soldadura en 3D, mediante la cual es posible generar imágenes en 2D, tanto a color como en escala de grises (Hao et al., 2013), cuyo análisis posterior permitirá detectar los defectos de soldaduras establecidos en la clasificación de referencia.

La forma de las uniones de soldadura tiende a variar de manera importante dependiendo de las condiciones bajo las que estas se suelden, esto incluye la cantidad de soldadura y el nivel de calor aplicado durante el proceso; esto puede provocar gran variedad en las formas de las uniones, aspecto que puede complicar en análisis de las mismas. Por esta razón, la etapa final del sistema AOI consiste en procesar los perfiles en 2D y 3D generados por los patrones de iluminación, generalmente, mediante la comparación de la información obtenida contra parámetros de referencia predefinidos (clasificación de referencia). Donde finalmente, en función de estos criterios, es posible que un sistema determine de forma automática la calidad de las uniones inspeccionadas.

1.3 OBJETIVO DEL SISTEMA DE CONTROL

El objetivo principal de esta tesis, es implementar un sistema control óptico que mediante el procesamiento de imágenes y el análisis de patrones de iluminación, reflexión y detección de bordes, sea capaz tanto de determinar la calidad de la soldadura de PCB's terminadas, como de soldar las mismas en función de la forma y el tamaño de los elementos que la

componen, en escenarios donde no se cuente con los archivos de diseño o manufactura en formato *CAD/CAM (Computer Aided Design / Computer Aided Manufacturing)*.

Para la consecución del objetivo, se parte de un dispositivo robótico CNC para soldar circuitos electrónicos, previamente desarrollado; a cuya estructura se le añadirá un arreglo de iluminación LED en forma de anillo en conjunto con una cámara digital que se encargará de la generación de las condiciones de iluminación y la adquisición de imágenes resultantes de la PCB para su posterior procesamiento. El control del anillo y su etapa de alimentación se realizará mediante un microcontrolador PIC18F4550 (Microchip, 2009) encargado de la activación y regulación de la señal PWM de potencia que variará los niveles de iluminación generados por el anillo. El procesamiento de las imágenes obtenidas será realizado por la computadora y mediante la interfaz gráfica desarrollada se podrá monitorear el sistema en tiempo real y permitirá la interacción con el usuario para desplegar los resultados de la inspección y del proceso de soldadura, además de controlar el anillo de iluminación mediante una interfaz *USB (Universal Serial Bus)*.

1.4 HIPÓTESIS

Mediante la implementación de técnicas de análisis de patrones de iluminación y reflexión, así como de técnicas de detección de bordes se puede diseñar y construir un sistema de control automático para la calibración, ajuste e inspección de la calidad de soldadura de una máquina CNC para soldar circuitos integrados.

1.5 ORGANIZACIÓN DE LA TESIS

En esta sección se describen los capítulos que integran esta tesis, abordando la forma en que contribuyen al alcance de los objetivos planteados.

El capítulo 2 titulado “Antecedentes”, en función del panorama general de los sistemas AOI presentado en el capítulo 1, aborda las alternativas comerciales disponibles actualmente, analizando sus ventajas y desventajas. Además, se hace un análisis de los métodos y técnicas de procesamiento de imágenes cuya implementación es viable en un proceso de manufactura de circuitos electrónicos que utilice herramientas CNC.

La parte final de dicho capítulo detalla los requerimientos que debe tener el sistema de inspección propuesto para determinar la calidad de la soldadura de una PCB mediante el arreglo de iluminación y captura de imágenes.

En el capítulo 3, “Metodología y materiales”, se describen las características del sistema de detección y obtención de imágenes, detallando los elementos físicos del mismo, incluyendo el anillo de iluminación, la cámara, la etapa de alimentación y los patrones de reflexión mediante los cuales el sistema indicará la calidad de las uniones de soldadura. Además, se profundiza en las características de la interfaz gráfica de usuario y de la tarjeta de adquisición de datos, mediante las cuales se realizará el control de la operación del sistema de detección y obtención de imágenes, y el despliegue de resultados.

El capítulo 4, “Resultados y discusión”, tiene por objetivo describir las propiedades generales del sistema; y de cómo su funcionamiento junto con los datos obtenidos, pueden ser aplicados a un proceso de manufactura realizado por una máquina CNC.

Finalmente, el capítulo 5 tiene como objetivo la validación de las técnicas implementadas y del sistema propuesto, mediante la integración y comparación de los resultados obtenidos.

2. ANTECEDENTES

2.1 INTRODUCCIÓN

La automatización de procesos de inspección dentro de la industria electrónica busca asegurar la calidad con la que son fabricados los productos entregados al cliente final. En un proceso de manufactura de productos electrónicos aproximadamente el 30% de todas las tareas de producción están relacionadas con la inspección, de los cuales el 60% se realizan usando técnicas visuales. En el desglose de los defectos encontrados durante la inspección visual, un 30% corresponde a defectos de los componentes/piezas, un 50% a defectos de ensamblado y un 20% a errores de soldadura (Edinbarough et al., 2005).

A raíz de estas estadísticas, se han desarrollado diversas técnicas y procesos de inspección de circuitos electrónicos las cuales, como se vio anteriormente, se aplican a diferentes niveles de producción. La implementación de estos sistemas en procesos a baja escala está limitada generalmente a la inspección directa realizada por expertos en análisis de uniones de soldadura, debido a que implica un costo menor frente a los sistemas automáticos más complejos. Sin embargo, un estudio conducido por AT&T mostró una repetibilidad de decisión de sólo el 44% para el mismo experto al inspeccionar uniones de soldadura, y sólo un 6% de acuerdo entre el juicio de cuatro inspectores revisando la misma tarjeta; mostrando que las descripciones de variaciones permisibles son altamente cualitativas y sujetas a la interpretación de los expertos en inspección y conduce a incrementos en los costos de inspección manual que pueden ser hasta del 50% de los costos de ensamblado en algunos casos(Edinbarough et al., 2005).

Dentro del objeto de estudio de esta tesis, se busca la implementación de un sistema que combine las capacidades robóticas de una máquina CNC con un conjunto de técnicas de detección y procesamiento de imágenes, que permitan desarrollar un sistema de inspección óptica automática de bajo costo, aplicable a dichos procesos a baja escala.

En este capítulo se presenta una descripción de algunos sistemas comerciales de inspección óptica automática, junto con algunos dispositivos y técnicas descritos en la literatura científica. A continuación se describen conceptos relacionados a algunas técnicas de procesamiento de imágenes viables dentro de un proceso de esta naturaleza; y finalmente, se detallan los requerimientos necesarios para desarrollar el sistema de inspección que permite alcanzar los objetivos de esta tesis.

2.2 ESTUDIO COMPARATIVO DE SISTEMAS AOI COMERCIALES

Los experimentos, técnicas y dispositivos comerciales para inspecciones de uniones de soldadura desarrollados recientemente basan su funcionamiento en dos elementos principales: a) el elemento de adquisición de imágenes y b) el sistema de iluminación para realce de características; los cuales integran el soporte físico del sistema de inspección, que posteriormente se combina con la parte del procesamiento de la información obtenida por ambos elementos, siendo esta segunda parte la que en su mayoría determina y las características del proceso de soldadura que son viables para inspección.

Como parte de un análisis comparativo de sistemas de inspección óptica desarrollados recientemente, se presenta la Tabla 2.1, donde se muestran algunos ejemplos representativos de dichos sistemas de inspección AOI, tanto disponibles comercialmente como de algunos experimentos realizados a nivel académico; en dicha tabla se indican las características más relevantes de cada dispositivo incluyendo los elementos de adquisición de imágenes, sistemas de iluminación y las características de la PCB y del proceso de soldadura a inspeccionar.

Tabla 2.1 Comparativo de sistemas comerciales y experimentos de inspección óptica automática.

Fabricante / Autor	Sistema de iluminación	Elemento de adquisición de imágenes	Etapas de inspección viables	Dimensiones PCB	Características Sobresalientes
OMRON VT-RNSii	Anillo LED RGB con realce	Cámara de alta resolución de 10/15/20 μm	Soldadura por pasta Soldadura por reflujo Soldadura por ola Posicionamiento de componentes	330mm x 255mm y 510mm x 460mm	Inspección de producción a baja escala sobre línea. Procesamiento de imagen a color. Software de usuario (reparación/verificación). Reporte de componentes detectados como defectuosos.
OMRON VT-RNSii-ptH	Anillo LED RGB con realce	Cámara de alta resolución de 10/15/20 μm	Soldadura por pasta Soldadura por reflujo Soldadura por ola Posicionamiento de componentes	330mm x 255mm y 650mm x 550mm	Inspección de producción a baja escala fuera de línea. Procesamiento de imagen a color. Software de usuario (reparación/verificación). Reporte de componentes detectados como defectuosos.
OMRON VT-S500	Anillo LED RGB con realce 3D	Cámara de alta definición y alta velocidad y resolución de 10/15 μm	Posicionamiento de componentes Post-soldadura	510mm x 300 mm y 510mm x 610 mm	Procesamiento por modelado 3D. Monitoreo en tiempo real. Estación de reparación integrada. Software de usuario.
OMRON VT-S720	Anillo LED RGB con realce 3D	Cámaras multi-ángulo de alta definición y velocidad y resolución de 15/20 μm)	Soldadura por reflujo	460 mm x 510mm	Procesamiento por modelado 3D. Monitoreo en tiempo real. Estación de reparación integrada. Software de usuario.
Edinbarough et al. (2005)	N/A	Cámara estándar CCD (escala de grises).	Soldadura por pasta Soldadura por reflujo Soldadura por ola Posicionamiento de componentes	Modificable por programación	Control mediante PC y tarjeta de adquisición de datos. Brazo robótico con 6 grados de libertad. Software basado en redes neuronales. No detecta defectos de soldadura, sólo defectos por levantamiento.
Hao et al. (2013)	Arreglo Hemisférico RGB	Cámara CCD digital a color	Soldadura por pasta Soldadura por reflujo Soldadura por ola Soldadura por pasta Posicionamiento de componentes SMT	No especificado	Basado en el procesamiento de formas 3D representándolas mediante imágenes 2D. Software basado en redes neuronales. Amplia detección de defectos de soldadura y colocación.

Los cuatro primeros ejemplos mostrados en la tabla anterior, corresponden a maquinaria disponible comercialmente desarrollada por la empresa OMRON agrupada en dos pares. El primer par de modelos (RNSii /RNSii-ptH) pertenecen a una línea de inspección para manufactura a baja escala que basa su procesamiento en adquisición de imágenes en 2D, ambas cuentan con los mismos elementos y características de iluminación, adquisición y tratamiento de la información obtenida, diferenciándose en el tamaño admisible de las PCB a inspeccionar y en que el primero está diseñado para integrarse dentro de una línea de producción, mientras que el segundo aplica para procesos de inspección fuera de línea de menor escala. El siguiente par de modelos (VT-S500/ VT-S720) se basan en la tecnología de procesamiento 3D del fabricante en la cual realizan la inspección por cuantificación de la forma de la soldadura usando realce de color para realizar una reconstrucción 3D de las uniones para reducir los cálculos y procesamientos.

Dentro de las investigaciones realizadas en el campo de la inspección óptica de circuitos electrónicos, el sistema propuesto por (Edinborough et al., 2005) consiste en un brazo robótico que se encarga de posicionar la cámara para adquisición sobre una región de interés entre la PCB y la patilla de cada componente, para su posterior análisis mediante software basado en redes neuronales; sin embargo, debido a la resolución de la cámara usada en esta investigación, el sistema sólo es capaz de detectar defectos por levantamiento de la patilla del componente y no los generados por deficiencias en la calidad de la soldadura. Por otro lado, los trabajos realizados por (Hao et al., 2013) combinan el uso del arreglo de iluminación LED de tres colores con la adquisición de imágenes mediante cámaras a color, además de realizar el procesamiento y clasificación de defectos mediante una red neuronal; este trabajo se centra en la inspección de componentes de montaje superficial y es aplicable para la inspección de uniones de varios procesos de soldadura (por ola, por reflujo, por pasta, etc.) donde gracias a las condiciones generadas por el sistema de iluminación, la información en 3D de las formas que tienen las uniones de soldadura pueden ser representadas mediante imágenes a color en 2D para su procesamiento.

2.3 METODOLOGÍA DEL PROCESAMIENTO DE IMÁGENES EN APLICACIONES CNC PARA SOLDAR CIRCUITOS ELECTRÓNICOS

Como se señaló en la sección 1.2, un sistema de inspección AOI para un proceso de manufactura de PCB's, es viable mediante el uso de un dispositivo de adquisición de imagen (cámara digital), el arreglo de iluminación y una PC para el procesamiento de datos. En función de estos elementos, la metodología a aplicar iniciará con el control de un anillo de iluminación. En dicha sección, se hizo mención de algunas clasificaciones usadas para determinar la calidad o los defectos presentes en las uniones de soldadura a inspeccionar. En la Figura 2.1 se muestran diferentes escenarios que se pueden presentar posterior a la etapa de soldadura de un componente SMT donde se indica el tipo de soldadura/defecto y las imágenes obtenidas de los mismos, junto con un modelo de los patrones de reflexión característicos de las uniones de soldadura y la vista lateral de los casos en cuestión.

Las uniones soldadas que se consideran como “aceptables”, presentan un patrón de reflexión secuencial RGB y la unión visualmente presenta una forma cóncava. Los componentes con soldadura insuficiente, presentan una región roja (R) muy marcada en la parte externa del pad, mientras que los casos de soldadura en frío presentan una región roja adicional cercana a la patilla del componente. Los defectos por desplazamiento y levantamiento presentan patrones similares, en donde el pad refleja los haces de luz de tal forma que presentan un modelo concéntrico con secuencia RBGR. Los modelos presentados, muestran cómo mediante una modificación externa de los niveles de iluminación usando anillo LED RGB permite capturar una imagen a color de los valores RGB reflectados por las uniones. La Figura 2.1 muestra la clasificación propuesta por (Hao et al., 2013), donde se observan imágenes típicas de uniones de soldadura asociadas con los defectos que éstas pueden presentar, junto con el modelo o patrón de reflexión que generan.

Las imágenes son funciones bidimensionales que cuantifican la intensidad de luz, siendo la matriz el modelo más común para la representación de las mismas; se eligió utilizar una herramienta como MATLAB para el desarrollo del sistema de visión propuesto, esto debido a que la unidad de procesamiento de dicho programa es la matriz, por lo que su utilización en el procesamiento de imágenes es una extensión natural de su lenguaje de programación (Cuevas et al., 2014).

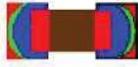
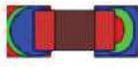
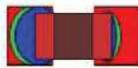
Tipo	Imagen	Modelo Soldadura	Vista Lateral
Soldadura Aceptable			
Soldadura en frío			
Soldadura Insuficiente			
Componente Desplazado			
Componente Erróneo			
Levantamiento tipo lápida			

Figura 2.1 *Imágenes típicas de muestra de uniones de soldadura, sus defectos asociados, el modelo de reflexión generado y las vistas laterales de éstos (modificado de (Hao et al., 2013)).*

La metodología a seguir para la implementación del sistema de control e inspección propuesto, se basará en el uso de las herramientas de procesamiento de imágenes de MATLAB. Específicamente, mediante la aplicación de operaciones de pixel, las cuales son realizadas sobre imágenes donde sólo se toma en cuenta el valor de un determinado pixel dentro de una imagen (Cuevas et al., 2014) y con base en las mismas, es posible aislar cada banda individual de color, comparar las matrices obtenidas contra un parámetro de referencia y determinar la calidad de la soldadura de cada componente, al realizar un barrido mediante un sistema robótico.

Las operaciones de pixel como la resta, son útiles para la segmentación y el mejoramiento de imágenes, así como para la detección de un cambio de posición en una imagen (Cuevas et al., 2010). Basándose en este hecho, se pueden analizar los valores de las matrices R, G y B del modelo de referencia y restar las imágenes obtenidas por el barrido del sistema robótico a la PCB; y de esta manera las diferencias obtenidas de los niveles de intensidad de cada banda color, determinarán si la calidad de la unión

inspeccionada es aceptable, al tener valores similares al modelo de referencia, o si esta es defectuosa, al presentar una patrón de reflexión diferente.

2.4 REQUERIMIENTOS DE UN SISTEMA DE CONTROL POR PROCESAMIENTO DE IMÁGENES APLICADO A PROCESOS DE SOLDADURA E INSPECCIÓN DE PCB'S

De acuerdo a la información expuesta en las secciones anteriores, en esta parte se enumeran los requerimientos para desarrollar un sistema capaz de inspeccionar las uniones de soldadura de los componentes electrónicos sobre una tarjeta de circuito impreso. De manera general, los requerimientos del sistema propuesto en la presente tesis son: a) que el sistema sea capaz de inspeccionar las uniones de soldadura de una PCB; b) que sea posible controlar las condiciones de iluminación de la placa para la correcta adquisición de imágenes; c) que el proceso pueda ser monitoreado mediante una interfaz de usuario y d) que la misma interfaz genere un reporte de los defectos detectados en función del procesamiento de las imágenes adquiridas.

El proyecto propuesto constituye el sistema de visión de un brazo robótico, el cual a su vez forma parte de una máquina para soldar circuitos integrados que utiliza gas hidroxilo como combustible generado a partir de la electrólisis del agua. Dicho sistema fue registrado ante el IMPI (*Instituto Mexicano de la Propiedad Industrial*) con solicitud número MX/a/2014/007507, el cual procedió al otorgamiento del grado de patente mediante el folio 49179 con fecha 4 de agosto de 2020.

El desarrollo del sistema mecánico para posicionar el elemento soldador no forma parte del alcance del presente trabajo; se busca que a partir de un sistema robótico previamente desarrollado sea posible integrar los elementos de iluminación y adquisición de imágenes con una pieza móvil que desplaza el elemento soldador, de tal manera que esta combinación sea posible realizar un barrido de la PCB terminada y capturar mediante imágenes a color los patrones de reflexión que permitan determinar la calidad de las uniones.

3. MATERIALES Y MÉTODOS

Las herramientas provistas por entornos software como MATLAB para el tratamiento y procesamiento de imágenes, posicionan a estas técnicas como una opción adecuada para la implementación de un sistema electrónico que incorpore elementos de visión máquina, que a su vez permita aplicarlo al proceso de manufactura realizado por una máquina CNC que suelda circuitos electrónicos. En este capítulo se sintetiza el conjunto de materiales involucrados en la realización del sistema propuesto y los métodos utilizados para llevar a cabo la inspección de PCB's.

3.1 DESCRIPCIÓN GENERAL DEL SISTEMA

Con el objetivo de explicar la construcción de manera general tanto del sistema como del proceso de inspección propuesto para determinar la calidad de la soldadura de una placa con circuitos electrónicos, la estructura del mismo se ha dividido en subsistemas. En la Figura 3.1 muestra el diagrama montaje de los distintos elementos del proyecto y su relación como sistema de visión para de la máquina CNC para soldar circuitos electrónicos; mientras que en la Figura 3.2 presenta un diagrama esquemático donde se indican todos los elementos que conforman el sistema de inspección óptica automática, dichos subsistemas se describen de manera breve a continuación:

Sistema de Iluminación: esta parte se encarga de modificar los niveles de iluminación de la PCB en general, este subsistema está compuesto por una serie de tres de anillos (Anillo R, Anillo G, Anillo B), donde a lo largo de su circunferencia están dispuestos un conjunto de LED's (*Light Emmiting Diode*) rojos, verdes y azules

respectivamente; el cual tiene el propósito de crear las condiciones propicias para la generación de los patrones de reflexión sobre las uniones de soldadura.

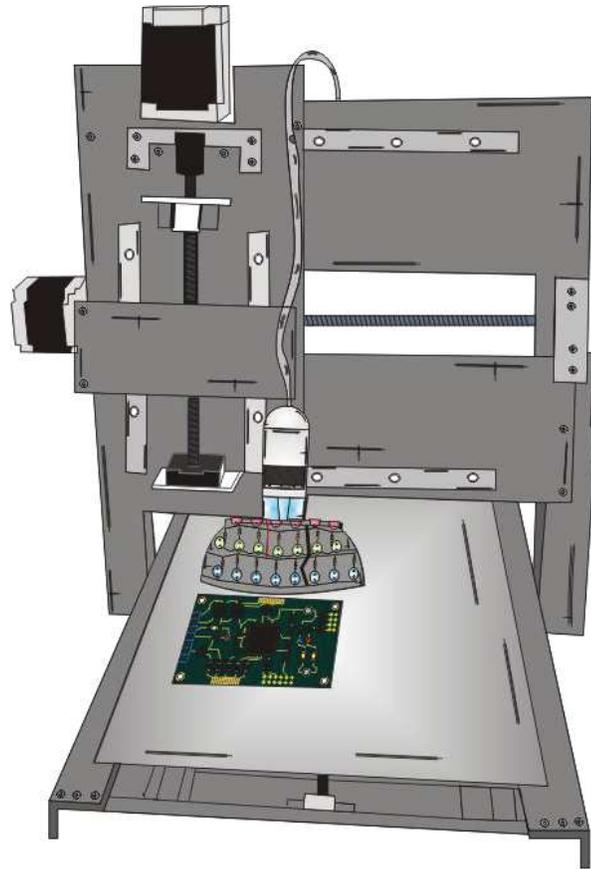


Figura 3.1 *Diagrama de montaje de los experimentos. Adaptación y montaje de los sistemas de iluminación y detección al sistema robótico del cual se compone la máquina CNC para soldar circuitos electrónicos.*

Circuito de Potencia: En esta parte está incluida la etapa de alimentación para el Anillo RGB, la cual es activada y controlada mediante la señal PWM generada desde el microcontrolador PIC (μ Controlador); dicha señal se encarga de activar un circuito manejador de compuerta (D1) para el control del disparo de un transistor (Q1) MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*). De esta forma, se genera una señal PWM de potencia mediante la modificación del ciclo de trabajo, y por lo tanto se controla la tensión que se le aplica a la carga; que en este caso será el anillo RGB, y por ende se verá como una variación en la intensidad del anillo LED RGB.

Sistema de detección: Esta parte es un complemento del subsistema de iluminación y está integrada por una cámara (Cámara/Microscopio) que actúa como el elemento de detección y captura de imágenes del sistema, cuyo montaje se realiza de manera concéntrica dentro del arreglo de anillos LED.

Interfaz: Es el elemento que realiza la conexión lógica y física entre el puerto *USB* de la computadora y el subsistema de iluminación. El sistema propuesto es programable, desde una computadora personal, y mediante una aplicación desarrollada en Visual Studio 2013, el usuario es capaz controlar el encendido del sistema de iluminación y la variación de los niveles de iluminación, además de poder monitorear el sistema en tiempo real y observar un reporte de las uniones de soldadura que el sistema detecte, así como de la comunicación con el sistema robótico preexistente (Tarjeta de Interfaz Sistema Robótico).

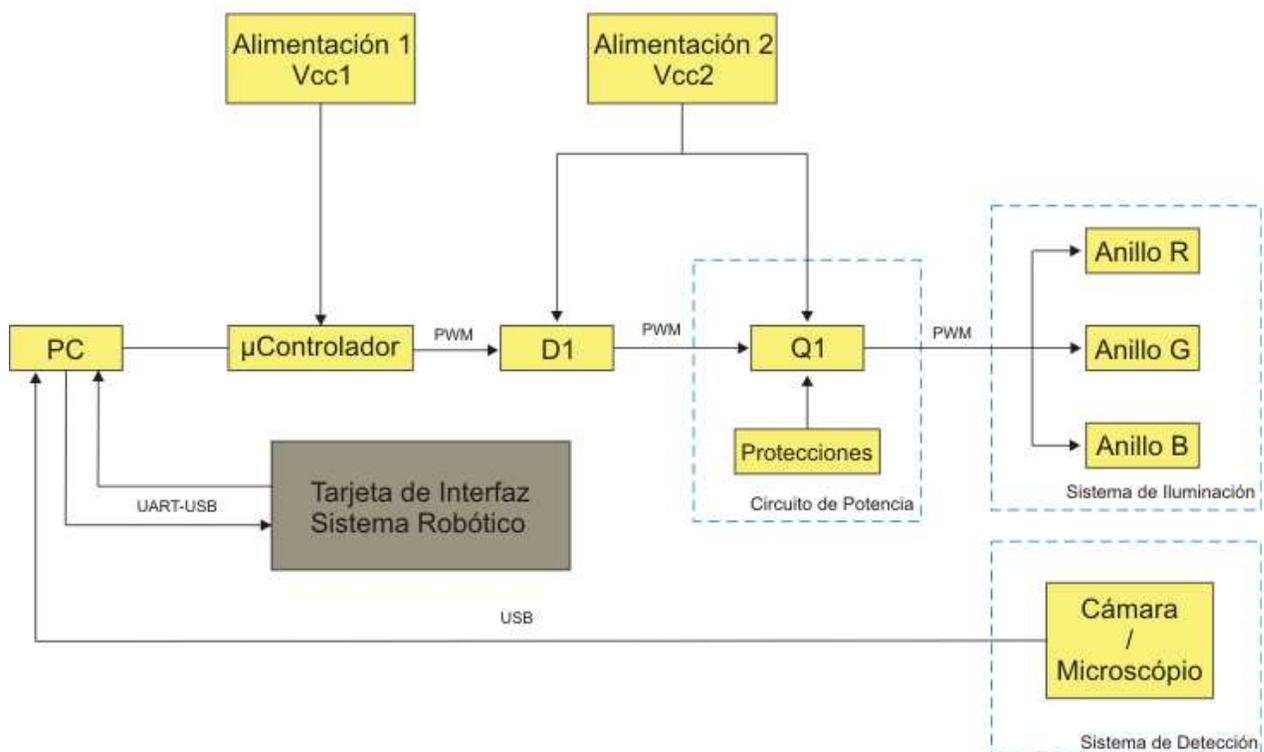


Figura 3.2 Diagrama esquemático del sistema de inspección óptica automática para la máquina CNC para soldar circuitos electrónicos.

3.2 SISTEMA DE DETECCIÓN Y ADQUISICIÓN DE IMÁGENES

3.2.1 Cámara

La detección y captura de imágenes de los componentes electrónicos, y en específico de las uniones de soldadura bajo inspección dentro de un sistema de inspección óptica automática se realiza a través de una cámara, cuyas características dependen de las técnicas de procesamiento de imágenes a utilizar en la etapa posterior. En la Figura 3.3 se muestra el elemento elegido para realizar la adquisición de imágenes del sistema; dado que las formas de los pads para componentes SMT están determinadas por las dimensiones del mismo y del proceso de soldadura utilizado (Heinrich et al., 1996), donde las dimensiones se encuentran en unidades de milímetros; se optó por utilizar un microscopio digital con amplificación variable modelo Dino-Lite AM2111 con resolución de 640x480 píxeles y con un sensor a color CMOS (*Complementary Metal Oxide Semiconductor*), cuyo bajo consumo de energía y su menor sensibilidad a la luz frente a otro tipo de tecnologías, lo vuelven un elemento viable para su aplicación dentro del sistema de inspección propuesto.



Figura 3.3 *Microscopio digital de amplificación variables para adquisición de imágenes.*

El rango de amplificación de dicho dispositivo, según especificaciones del fabricante es de 20x-200x, el microscopio se forma de una sola lente cuya amplificación se modifica de manera manual mediante un selector integrado que permite modificar el enfoque de la misma en relación a la distancia de trabajo.

Las pruebas de posicionamiento de la cámara se realizaron en función de los parámetros de operación proporcionados por el fabricante, mostrados en la Tabla 3.1, aspecto que se profundizará en la sección 4 “Resultados y Discusión”.

Tabla 3.1 *Parámetros de operación, amplificación, distancia de trabajo y campo de vista del microscopio AM211. Unidades en milímetros (Modificado de Dino-Lite 2017).*

Magnificación	Distancia de Trabajo	Campo de Vista (Eje X)	Campo de Vista (Eje Y)
20	51.7	21	15.7
30	24.1	14	10.5
40	11	10.5	7.9
50	3.8	8.4	6.3
60	-0.6	7	5.2
200	-0.4	2.1	1.6
210	0.6	2	1.5
220	1.7	1.9	1.4

El microscopio seleccionado cuenta con una interfaz USB 2.0 la cual se utilizará para realizar la conexión con la PC, y específicamente con MATLAB; valiéndose del paquete de herramientas *Image Acquisition Toolbox* se construirá un objeto de video (*videoinput*) de entrada, el cual representa la conexión entre MATLAB y el dispositivo de adquisición de imágenes conectado (Matworks, 2017), en este caso, el microscopio. Posteriormente, en cada secuencia de barrido se obtendrá una imagen o “*frame*” individual de cada sección de la PCB bajo inspección.

3.2.2 Arreglo circular de iluminación LED

Como se muestra en la Figura 3.4, el arreglo de iluminación está formado por una estructura de soporte tipo plato parabólico (1) fabricado de plástico, el cual permite la colocación de manera concéntrica de un conjunto de tres anillos: anillo de iluminación rojo (2), anillo de iluminación verde (3) y anillo de iluminación azul (4), cada uno de estos se compone de un total de 20 LED's de cada color; en la figura mencionada se observa la vista inferior del arreglo de iluminación, el diámetro (\emptyset) típico de cada LED es de 5 mm, colocando cada uno en la disposición circular mencionada con una separación aproximada de 18° de arco. Los aspectos relacionados al control de iluminación del mismo, así como el circuito de alimentación para su funcionamiento se explican a detalle en la sección 3.2.4 “Especificaciones del sistema electrónico de alimentación”.

En la parte superior de la estructura de soporte se realiza el montaje de uno de los extremos del microscopio (5), el cual se acopla a la parte central del plato donde se sitúa un orificio de 25 mm (\emptyset), espacio que permite a la lente del microscopio (6) realizar los ajustes pertinentes y tener un campo de vista aceptable para la inspección de las uniones y los componentes de la PCB.

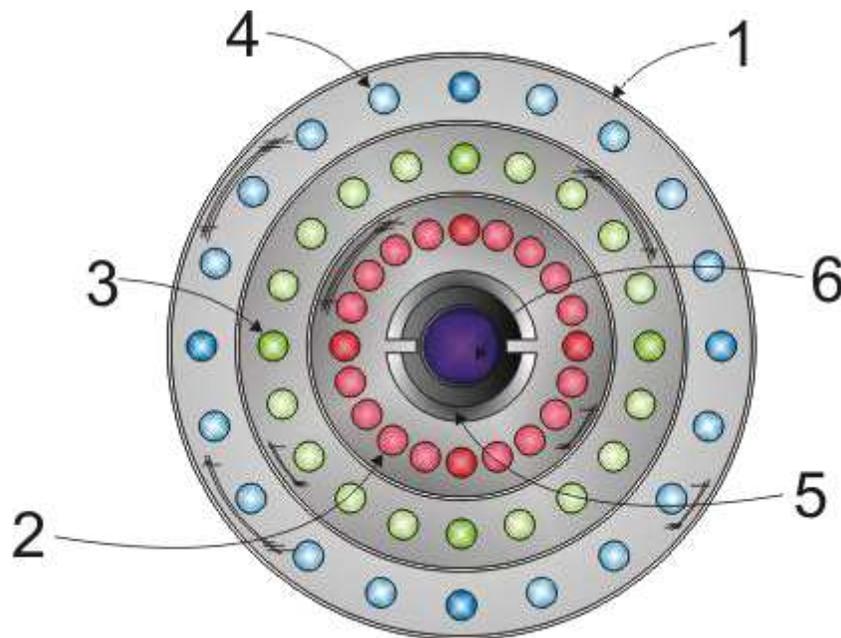


Figura 3.4 Vista inferior de la estructura que forma el arreglo circular de iluminación LED.

3.2.3 Patrones de iluminación y reflexión

Como parte de la metodología propuesta para el desarrollo del sistema inspección, ya se ha mencionado que se utilizarán como base las imágenes con un modelo de color RGB, el cual se basa en la combinación de los colores primarios rojo (R), verde (G) y azul (A), cuyo origen se remonta a la tecnología de la televisión y puede ser considerado como la representación fundamental del color en las computadoras, cámaras digitales y para el almacenamiento de imágenes. Además, la mayoría de los programas para el procesamiento de imágenes y de representación gráfica utilizan este modelo para la representación interna del color (Cuevas et al., 2014).

Los patrones de iluminación generados, y los patrones de reflexión de las uniones de soldadura a detectar e inspeccionar por el sistema, utilizan el modelo RGB debido a que se le considera como un formato de color aditivo, lo que significa que la combinación de los colores se basa en la adición de los componentes individuales considerando como base el negro. Este proceso puede ser visto como el traslape de 3 rayos de luz de colores rojo, verde y azul, los cuales son dirigidos hacia una superficie blanca, y cuya intensidad puede ser continuamente controlada; y donde la intensidad de los componentes de color determina tanto el tono como la iluminación de color resultante (Cuevas et al., 2014).

En la Figura 3.5 se muestra el diagrama de flujo con el cual se generó el programa para la determinación de la calidad de la soldadura de cada una de las imágenes obtenidas durante el barrido de la PCB ejecutado por el sistema robótico. De manera general se explica el funcionamiento de dicho programa, el cual está incluido en el Apéndice A.2; el proceso inicia con el barrido o movimiento de los sistemas de iluminación y detección con la ayuda del sistema robótico, en cada paso del mismo se crea un objeto de video para capturar un *frame* individual de las uniones de soldadura iluminadas por el Anillo LED RGB. La imagen en cuestión es guardada y se inicia el procesamiento de la misma, se realiza un aislamiento de cada banda individual de color junto con la utilización de la función *himst* del paquete de herramientas *Image Processing Toolbox* para calcular los histogramas de cada banda de color, dando como resultado un vector de dimensiones [256 x 1], los cuales son distribuciones que describen la frecuencia con la que se presentan los valores contenidos en cada imagen (Cuevas et al., 2014). Posteriormente, mediante la

función *findpeaks* del paquete *Signal Processing Toolbox* se detectan los máximos locales cada vector, junto con sus posiciones.

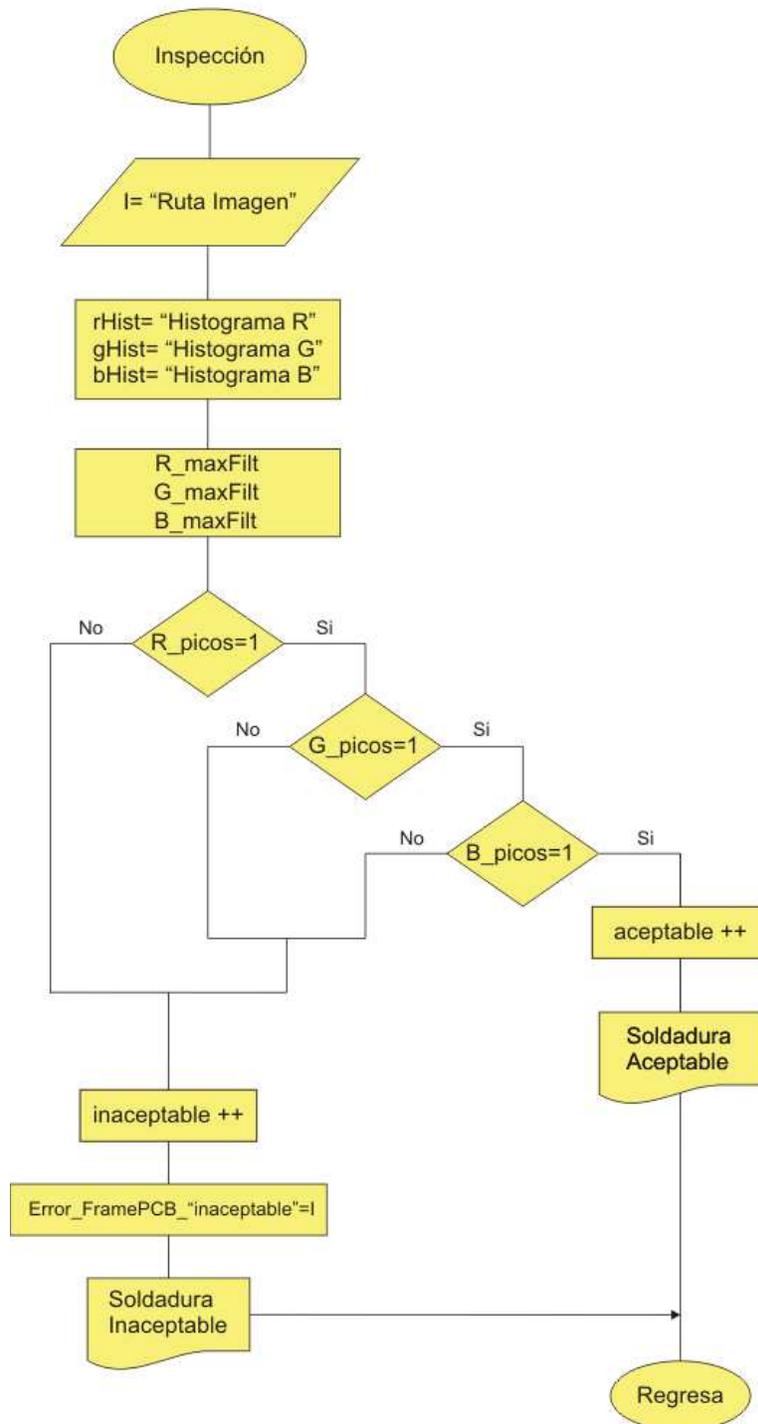


Figura 3.5 Diagrama de flujo del programa de tratamiento de imágenes obtenidas por el sistema de detección y determinación de la calidad de las uniones de soldadura.

La importancia de los histogramas recae en el hecho de que los patrones de reflexión pueden ser caracterizados mediante las gráficas que cada histograma individual genera. Como se mencionó en la sección 2.3, las uniones de soldadura aceptables, presentan un patrón de reflexión secuencial RGB, el cual se vería representado en los histogramas individuales con una gráfica con un valor máximo o pico pronunciado. De forma experimental, se determinó implementar una condición de filtro para suprimir los picos con valores menores al 80% del valor máximo detectado, estableciendo que, el histograma que presente sólo un pico máximo es un indicador de un patrón de reflexión aceptable para un determinado color; mientras que un histograma con más de un pico mayor a las condiciones de umbral establecidas, es indicador de un problema en la unión de soldadura detectada, la cual se clasifica como inaceptable; la Figura 3.6 muestra las gráficas de dos diferentes histogramas de uniones de soldadura caracterizadas como Aceptable e Inaceptable respectivamente, cuyo comportamiento es coherente con los criterios anteriormente mencionados.

En virtud de lo expuesto en el párrafo anterior, la segunda parte del diagrama presentado en la Figura 3.5, consiste en una secuencia de expresiones condicionales anidadas, que determinan el número de picos presentes en cada banda individual de color, en caso de cumplirse la condición para los tres colores, se determina que la unión es “Aceptable”; en caso contrario, si la condición no se cumple para determinado color el programa se marca la unión de soldadura como “Inaceptable”, guardando la información en una nueva imagen para su posterior despliegue en el reporte generado por la interfaz gráfica desarrollada.

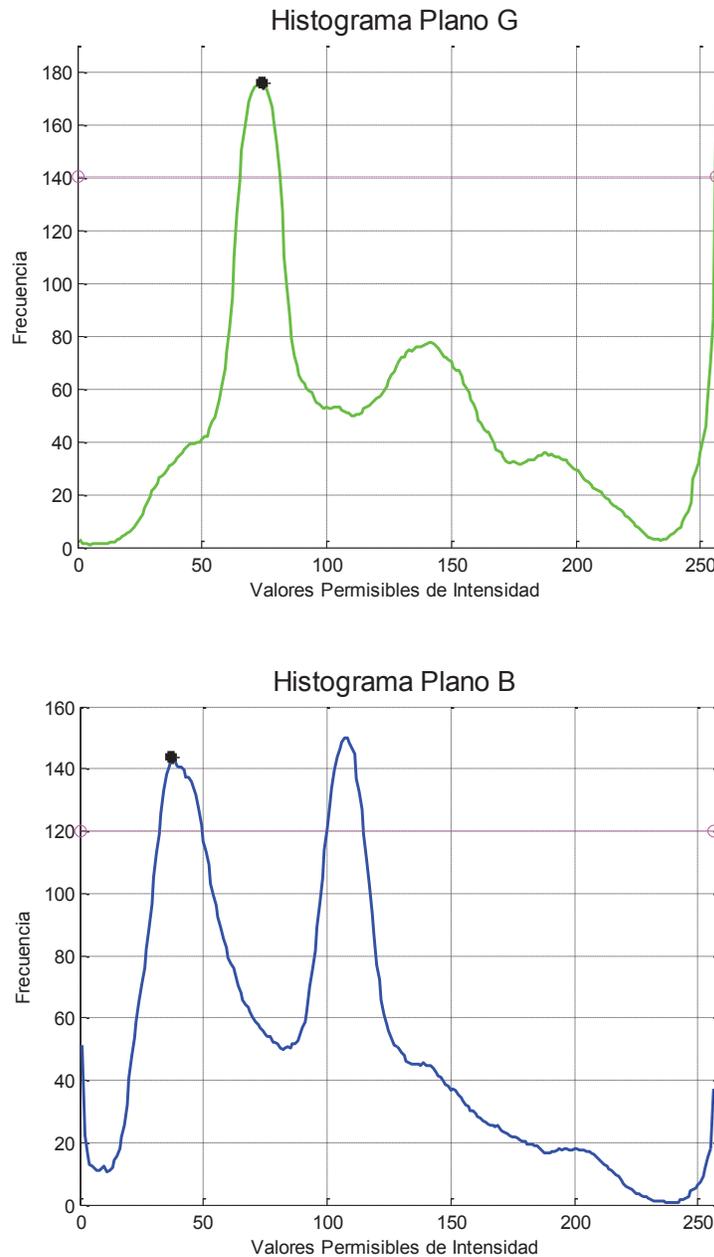


Figura 3.6 Gráficas que muestra el comportamiento de dos histogramas generados por el análisis de distintas uniones de soldadura, las cuales muestran la relación intensidad contra frecuencia de los pixeles con dichos valores de intensidad. En la gráfica de arriba se muestra el histograma del plano G de una unión considerada como aceptable, se puede observar que la forma de la misma sólo presenta un pico mayor al 80% del valor máximo de frecuencia obtenido (indicado con la línea color magenta). En la gráfica de abajo se muestra el histograma del plano B de una unión considerada como Inaceptable (Soldadura Insuficiente), se observa que se presentan diversos picos superiores al valor de umbral establecido.

3.2.4 Especificaciones del sistema electrónico de alimentación

En la Figura 3.2 presentada previamente, se indica que el sistema electrónico de alimentación se compone de 5 elementos principales, dado que la etapa de alimentación del sistema robótico no forma parte del alcance de la presente tesis. El propósito principal de la etapa de alimentación del sistema propuesto tiene la finalidad de energizar y controlar los elementos que interactúan con el sistema de iluminación formado por el Anillo LED RGB.

La primera parte la compone un microcontrolador PIC18F4550 el cual se alimenta con 5V suministrados desde una fuente de poder ATX. El PIC será utilizado, en lo que se refiere al sistema electrónico de alimentación como el dispositivo generador de la señal PWM de control, la cual se generará mediante el uso de uno de los módulos CCP (*Capture/Compare/PWM*) contenidos en el PIC. Siguiendo las instrucciones proporcionadas por el fabricante para la configuración de los registros asociados a dicho módulo en su modo PWM, primero se configura el registro PR2 (*Timer 2 Period Register*) para establecer el periodo de la señal PWM, el cual se fija en 1 ms, de la siguiente forma:

$$PR2 = \frac{T_{PWM}}{4T_{Osc} (\text{TMR2 Prescale Value})} - 1 \quad (3.1)$$

T_{PWM} : *Periodo de la señal PWM*

T_{Osc} : *Periodo de oscilación del microprocesador*

TMR2 Pr escale Value = Valor de pre-escalado del Timer 2

$$T_{PWM} = 1ms \quad (3.2)$$

$$f_{Osc} = 8MHz \quad (3.3)$$

$$T_{Osc} = \frac{1}{f_{Osc}} = \frac{1}{8MHz} = 0.125\mu s \quad (3.4)$$

$$TMR2 \text{ Pr escale Value} = 16 \quad (3.5)$$

$$PR2 = \frac{1ms}{4(0.125\mu s)(16)} - 1 = 125 - 1 = 124 \quad (3.6)$$

El valor obtenido de 124 para el registro PR2 indica el valor decimal máximo que el microcontrolador asociará con un ciclo de trabajo del 100%.

Además, como la operación del sistema requiere que la interfaz de usuario permita regular el ciclo de trabajo de la señal, y por ende del sistema de iluminación, se procede a calcular el valor de configuración del registro CCPR1L (*Capture/Compare/PWM Register 1 Low Byte*) y de la parte específica del registro CCP1CON<5:4>, de la siguiente manera:

$$CCPRxL : CCPxCON < 5 : 4 > = \frac{PWM_D}{T_{Osc} (TMR2 \text{ Pr escale Value})} \quad (3.7)$$

PWM_{DC} : *Ciclo de trabajo de la señal PWM en segundos*

Estableciendo un valor para el ciclo de trabajo arbitrario del 50% para realizar el procedimiento:

$$CCPR1L:CCP1CON<5:4> = \frac{1ms(0.5)}{0.125\mu s(16)} = 250 \quad (3.8)$$

En formato hexadecimal y binario:

$$CCPR1L : CCP1CON < 5 : 4 > = 0xFA = 0b1111010 \quad (3.9)$$

Acomodando el resultado obtenido en un formato de 10 bits, se utilizan los 8 bits más significativos para configurar el registro CCPR1L y los dos bits menos significativos para los bits 5 y 4 del registro CCP2CON respectivamente, quedando de la siguiente manera:

$$CCPR1L = 0b1111010 = 0x3E = 62 \quad (3.10)$$

$$CCP1CON = 0b00101100 = 0x2C = 44 \quad (3.11)$$

Se observa de la configuración del registro CCPR1L que el valor obtenido de 62 para un ciclo de trabajo del 50% es coherente con el valor máximo encontrado de 124 para el registro PR2; en la sección 3.2.5.2 “Programación PIC” se detallan los valores de escritura

al registro CCPR1L necesarios para lograr la modificación del ciclo de trabajo a través de la conexión USART que permite controlar la iluminación desde la interfaz de usuario en la PC.

La parte formal de la etapa de potencia inicia con el circuito manejador de compuerta con matrícula IR2110, que permite implementar un circuito de conmutación inferior para el control del disparo del MOSFET de potencia IRF640 que forma parte del circuito de potencia, cuya conmutación permite obtener a su salida la señal PWM de potencia necesaria para el control del anillo de iluminación LED RGB, la Figura 3.7 muestra los recursos electrónicos del sistema. El cálculo de las protecciones necesarias (*Red Snubber*) para la correcta operación del circuito de potencia se muestra a continuación:

$$I_o = \frac{P}{V} \quad (3.12)$$

$$I_T = I_o D \quad (3.13)$$

$$C_s = \frac{3I_T t_F}{V_{DS_{off}}} \quad (3.14)$$

$$R_s = \frac{1}{3C_s F_s} \quad (3.15)$$

I_o : Corriente que circula en la carga

P : Potencia de la carga

V : Voltaje de la fuente de alimentación V_{CC2}

I_T : Corriente del transistor

D : Ciclo de trabajo

C_s = Capacitor Red Snubber

R_s = Resistencia Red Snubber

$V_{DS,off}$: Voltaje Drain - Source en estado de apagado

f_s : Frecuencia de conmutación

t_F : Tiempo de bajada

Dado que los tres anillos de iluminación RGB están dispuestos en una conexión en paralelo con un total de 60 LED's, cada uno con un consumo de potencia típico de 1 W:

$$P = 1W(60) = 60W \quad (3.16)$$

$$I_o = \frac{60W}{12V} = 5A \quad (3.17)$$

$$I_T = 5A \times 1 = 5A \quad (3.18)$$

$$C_s = \frac{3(5A)(36ns)}{12V} = 45nF \quad (3.19)$$

$$R_s = \frac{1}{3(45nF)(1kHz)} = 7.41k\Omega \quad (3.20)$$

3.2.5 Algoritmos de funcionamiento

A manera de descripción del funcionamiento lógico del sistema de inspección, la programación del mismo se divide en dos partes principales: a) programación del microcontrolador y b) programación de la PC. La primera parte explica el envío y recepción de datos que realiza el PIC para recibir instrucciones por parte de la interfaz de usuario, así como de las señales de control para el sistema de iluminación; mientras que la segunda parte contempla el procedimiento de programación tanto de la interfaz gráfica de usuario desarrollada en Visual Studio 2013 y de los programas para tratamiento de imágenes desarrollados en MATLAB.

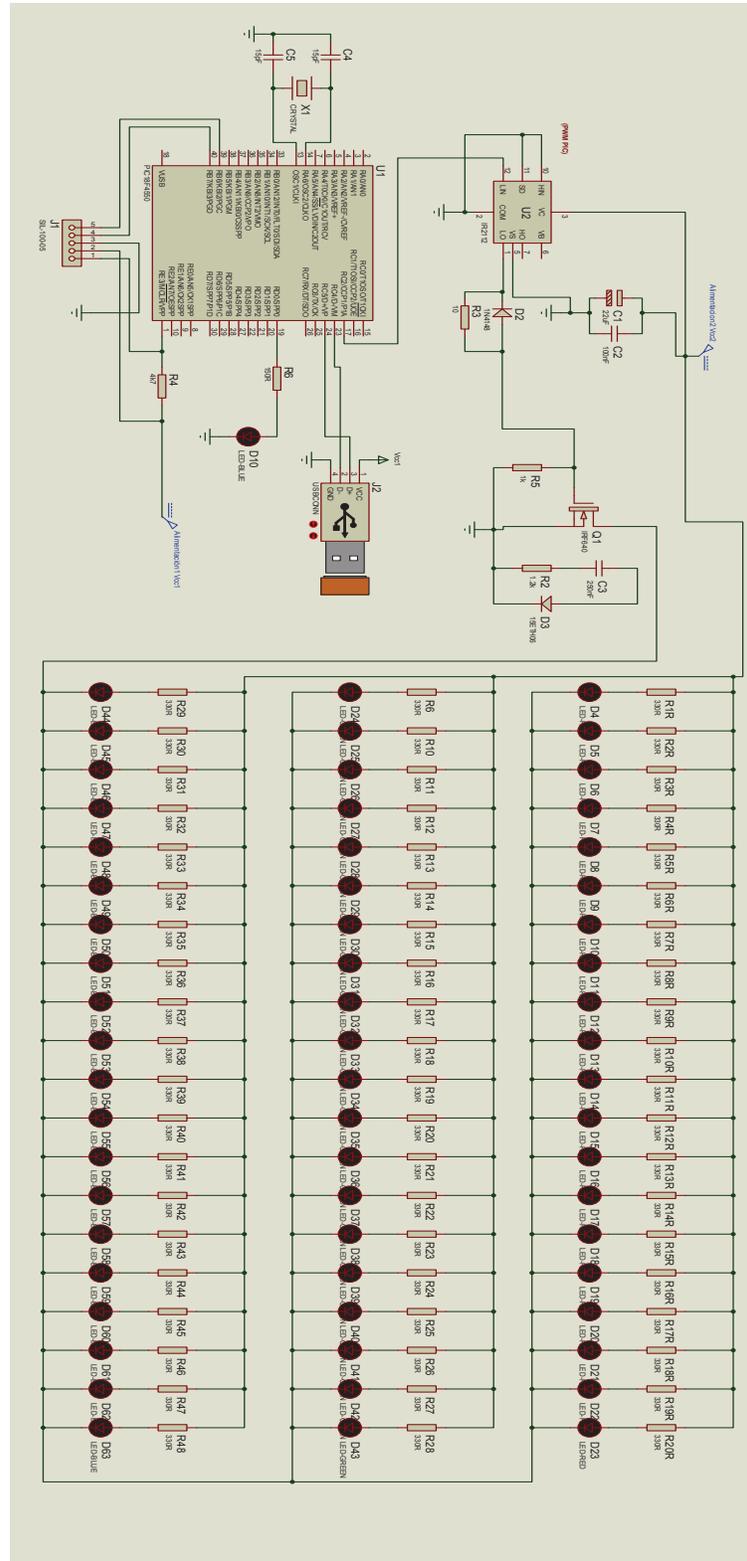


Figura 3.7 Diagrama del circuito eléctrico que muestra los recursos electrónicos del sistema.

3.2.5.1 Programación PIC

La función principal del microcontrolador es realizar la comunicación entre la tarjeta de interfaz con la PC, para de esta forma recibir las instrucciones enviadas desde la aplicación o interfaz de usuario desarrollada para el control del encendido y apagado del sistema de iluminación y del control de los niveles del mismo. La elección del PIC 18F4550 se debe a que la movilidad del brazo robótico del cual forma parte el sistema de control, al cual se hace referencia en la sección 2.4, basa su operación en el uso de dichos dispositivos, por lo que se pretende utilizar y optimizar los recursos que el sistema robótico provee.

La primera parte de la programación del microcontrolador recae en la configuración de los registros involucrados con el módulo EUSART (*Enhanced Universal Synchronous Asynchronous Receiver Transmitter*), el cual funciona como una SCI (*Serial Communication Interface*), que puede ser configurado de manera asíncrona y full-duplex, permitiendo la comunicación con dispositivos periféricos (*Microchip, 2009*). Dicho módulo se programa de forma asíncrona con un modo de recepción continuo, una velocidad de 9600 baudios, destacando la habilitación de las interrupciones de transmisión y recepción.

En la Figura 3.8 se puede observar el diagrama de flujo del programa principal del PIC, el cual permanece en un estado de espera mediante un bucle infinito, que está regido por las interrupciones globales y de periféricos del microcontrolador, la habilitación de la misma está dictada por el monitoreo constante de la bandera de interrupción “RX USART”.

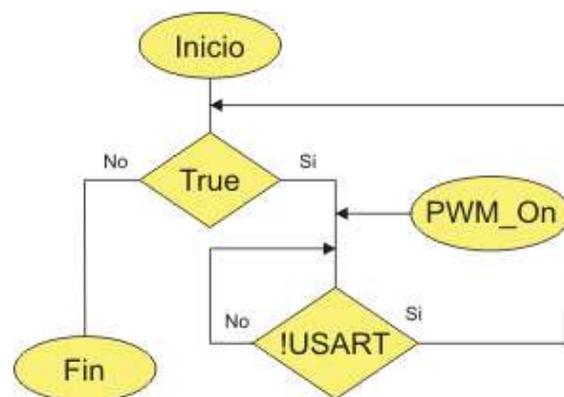


Figura 3.8 Diagrama de flujo del programa principal para la comunicación del microcontrolador y la PC.

La Figura 3.9 muestra el diagrama de flujo específico de la del código contenido dentro de la interrupción programada. Como se mencionó, al momento en que se recibe un dato a través del pin establecido en la configuración del módulo EUSART (RC7), la bandera de interrupción de recepción cambia su estado. A continuación se activa la salida PWM establecida y el caracter recibido se lee mediante los recursos que ofrece el módulo y se guarda en un variable utilizada para la operación de una estructura de control tipo switch.

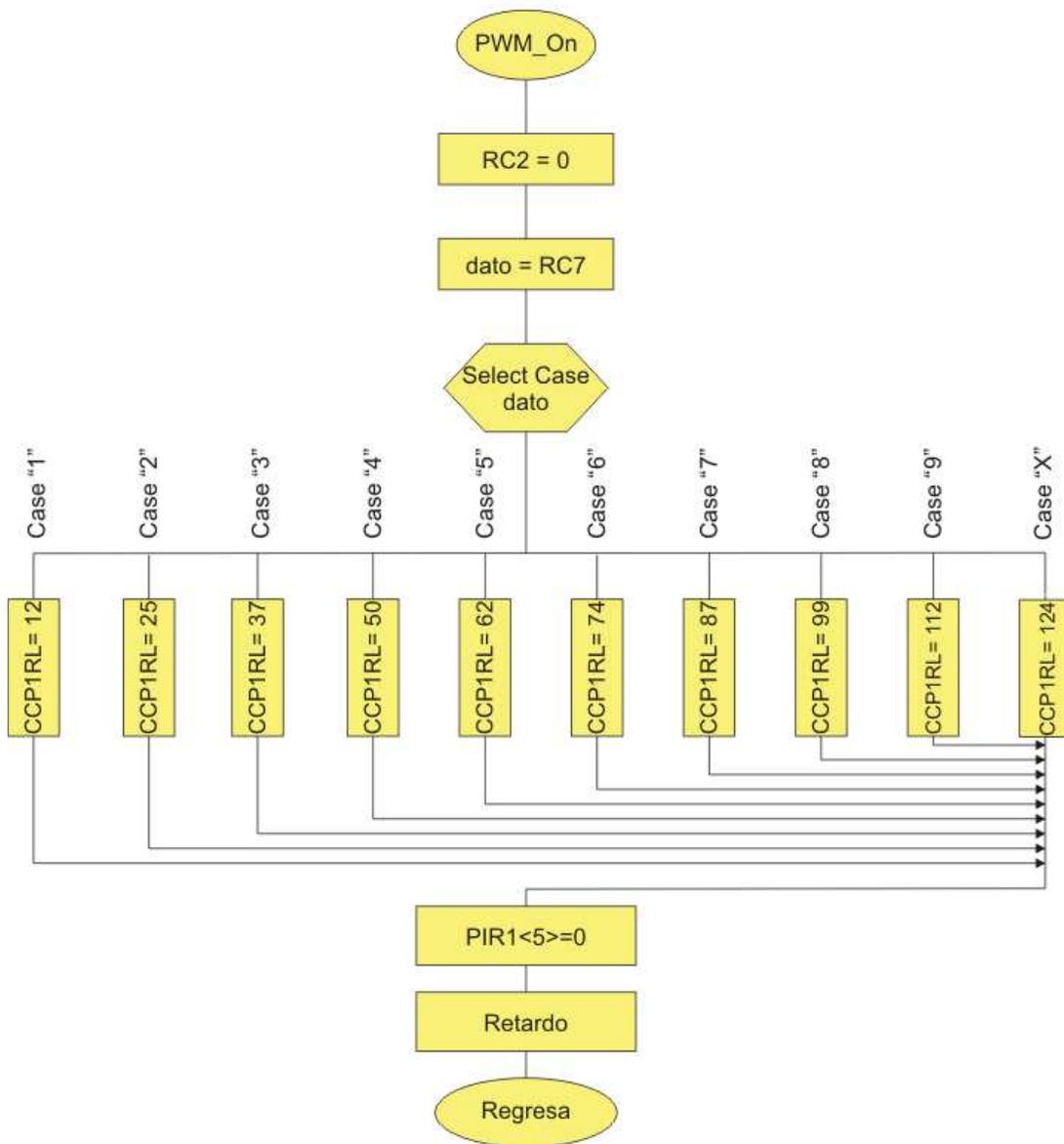


Figura 3.9 Diagrama de flujo de la interrupción periférica de recepción USART.

Como se mencionó en los cálculos realizados en la sección 3.2.4 “Especificaciones del sistema electrónico de alimentación”, se obtuvo un valor específico a escribir en el registro PR2 que corresponde a un ciclo de trabajo del 100%. En función de este valor y con el objetivo de que la interfaz de usuario permita modificar los niveles de iluminación del anillo LED, se eligió permitir al operador modificar el ciclo de trabajo en intervalos de 10% en un rango de 10 a 100%; en la Tabla 3.2 se resume la forma en que se asignaron caracteres a ser leídos por PIC, los cuales se envían desde la PC, que definen el valor escrito en el registro CCPR1L; y por ende, establecen el valor del ciclo de trabajo.

Tabla 3.1 Descripción de la asociación de caracteres leídos para la variable “dato” y la definición del ciclo de trabajo.

Carácter USART (dato)	Valor registro CCPR1L	Ciclo de Trabajo
1	12	10%
2	25	20%
3	37	30%
4	50	40%
5	62	50%
6	74	60%
7	87	70%
8	99	80%
9	112	90%
X	124	100%

Una vez que el módulo CCP establece la señal PWM en el pin de salida definido y con el valor de ciclo de trabajo indicado por el operador, la bandera de interrupción se limpia y el PIC regresa al programa principal, donde espera el envío de otro carácter a través de la comunicación serial. El código programado se muestra en el Apéndice A.1.

3.2.5.2 Programación PC

La programación de la PC busca establecer la comunicación entre ésta y la tarjeta de interfaz que controla el sistema de iluminación; así como de proveer también del medio o

puerto de conexión para el microscopio que integra el sistema de detección. En ambos casos, la comunicación se realiza de manera serial.

Para el primero se utiliza como interfaz un módulo UART que conecta la PC, utilizando un puerto COM virtual emulado mediante drivers, con el microcontrolador que rige el funcionamiento del anillo RGB. La conexión del microscopio se realiza a través de un cable estándar con terminación USB 2.0, el cual se conecta a cualquier puerto disponible de la PC de control; cabe mencionar que para el correcto reconocimiento del elemento de adquisición de imágenes en la PC y por consecuencia del *Image Acquisition Toolbox* de MATLAB, es indispensable la instalación de los controladores proporcionados por el fabricante del microscopio.

La interfaz desarrollada en Visual Studio 2013 sirve como herramienta de comunicación entre la computadora y el módulo UART. En el Apéndice A.2 se muestra el código del programa realizado en dicho paquete. La programación de la PC, específicamente el programa para el procesamiento de imágenes de las uniones de soldadura, fue explicado anteriormente, ya que forma parte de la sección 3.2.3 “Patrones de iluminación y reflexión”.

3.2.6 Tarjeta de adquisición de datos

La interfaz electrónica utilizada para el desarrollo de la tarjeta de adquisición de datos se basa en el uso de un microcontrolador PIC18F4550, dispositivo que se encarga de coordinar: (1) la comunicación establecida con la computadora para la transmisión de las instrucciones necesarias para el control del sistema de iluminación; (2) la comunicación y envío de instrucciones a la tarjeta del sistema robótico para el inicio y operación del sistema de inspección óptica automática. La configuración del sistema se lleva a cabo vía la PC usando la interfaz gráfica descrita en la sección 3.3.

La tarjeta de adquisición de datos contiene también los elementos que conforman el sistema electrónico de alimentación. Dicha etapa de alimentación se compone de un driver o manejador de compuerta con matrícula IR2110 (International Rectifier, 2005), el cual cuenta con una entrada lógica “HIN” que recibe la señal PWM generada desde el

microcontrolador; la configuración global del componente se realiza en función de las especificaciones del fabricante para la implementación de un circuito manejador de compuerta en conmutación inferior, en la cual se incluyen diodos de recuperación rápida, resistencias y capacitores.

El manejo de los niveles de corriente y voltaje para la operación del anillo se realiza a través del transistor MOSFET con matrícula IRF640 (International Rectifier, 2010), la señal generada por el driver es conectada a la compuerta del transistor en cuestión, el cual funciona como un circuito troceador reductor. En la tarjeta de adquisición de datos también están presentes los dispositivos electrónicos de protección que integran la red *Snubber* del transistor, elementos que se encargan de proteger el circuito contra picos de voltaje u oscilaciones de carácter inductivo; los cálculos de dichos elementos se establecieron en la sección 3.2.4.

La señal PWM de potencia generada a la salida de la tarjeta se conecta mediante un puente a la estructura de soporte del sistema de iluminación, sobre la cual están colocados cada uno de los LED's que conforman los tres anillos y las resistencias necesarias para la protección de los mismos.

3.3 INTERFAZ CON EL USUARIO

Dentro de los objetivos planteados previamente, se estableció que se desarrollaría una interfaz que permitiera al operador del sistema de inspección: 1) controlar el encendido y apagado del sistema de iluminación, además de permitir modificar los niveles de iluminación del anillo RGB, mediante el control del ciclo de trabajo de la señal PWM producida en conjunto por el PIC y el circuito electrónico de potencia y 2) monitorear el sistema en tiempo real, permitiendo al operador observar las uniones de soldadura clasificadas como defectuosas.

Las imágenes de la Figura 3.10 muestran la interfaz gráfica del sistema, en la cual el usuario puede en primera instancia realizar la conexión de la misma con el módulo UART de tarjeta de adquisición de datos a través de un puerto COM virtual como fue mencionado. Posteriormente, el usuario puede controlar la activación del sistema de iluminación, cuyo

funcionamiento se muestra mediante una indicación visual del encendido del anillo LED; esta acción es la encargada de enviar una señal vía UART para la activación del módulo PWM del PIC. Después del encendido, la interfaz permite la manipulación de una barra tipo “*scroll*” la cual controla el ciclo de trabajo de la señal PWM de potencia suministrada al anillo y varía en el rango de 10-100%, contando con dos indicaciones sensibles a dicha variación, las cuales se muestran en forma de una caja de texto y una barra de progreso.

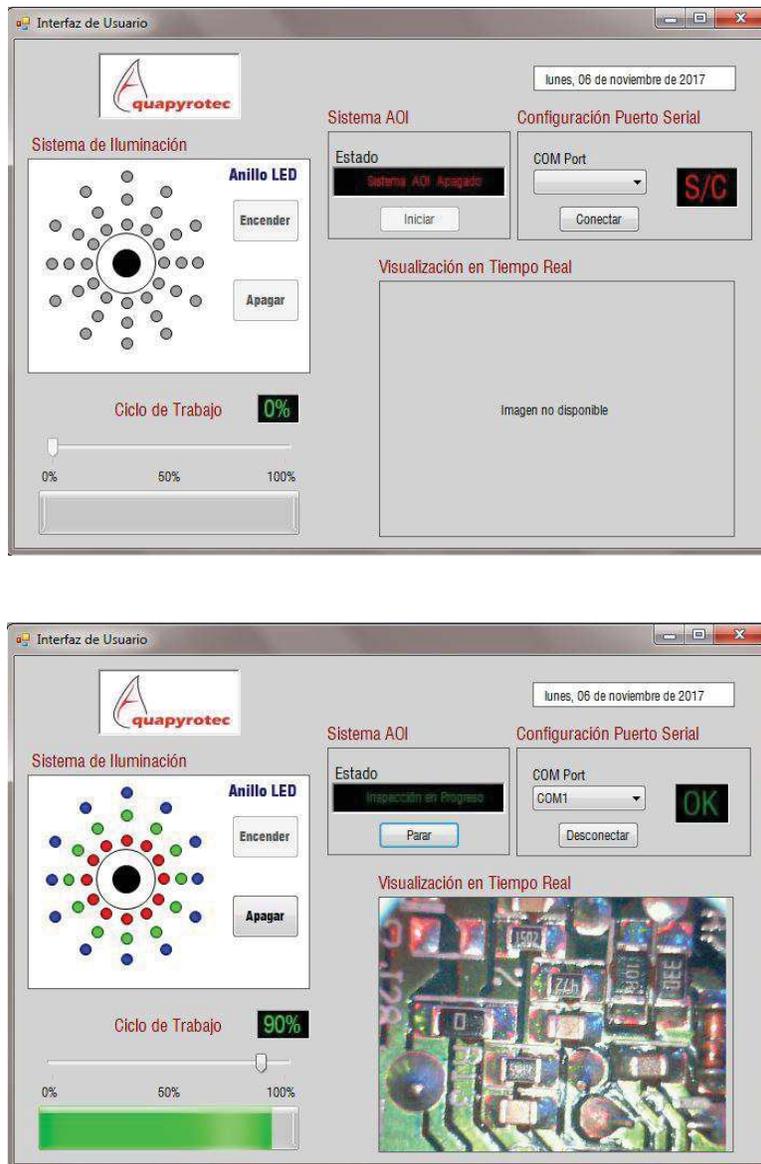


Figura 3.10 Aplicación de escritorio que es la interfaz de comunicación entre el usuario y los sistemas de iluminación y detección.

Una vez definidos los niveles de iluminación deseados, la operación del sistema de inspección (AOI) se inicializa mediante el botón “Iniciar”, el cual envía la orden a la tarjeta de interfaz del sistema robótico para que inicie el barrido de la PCB; en este punto la interfaz de usuario activa la ventana de Visualización en Tiempo Real, donde el operador puede observar el video capturado mediante el elemento de adquisición de imágenes, donde de manera paralela el sistema realiza el procesamiento de las imágenes capturadas.

3.3.1 Comunicación entre la PC y la tarjeta de adquisición de datos

En la Figura 3.11 se muestra un diagrama esquemático de los recursos del microprocesador utilizado en conjunto con los elementos que forman la tarjeta de adquisición de datos. El puerto C tiene doble funcionalidad, la primera es establecer los canales de comunicación del este dispositivo con la computadora, mientras que la segunda es proveer el pin de salida para la señal PWM de control necesaria para la operación al sistema de iluminación. El puerto RB0 provee una salida lógica que cumple las funciones de señalización visual, la cual indica la correcta operación de la tarjeta.

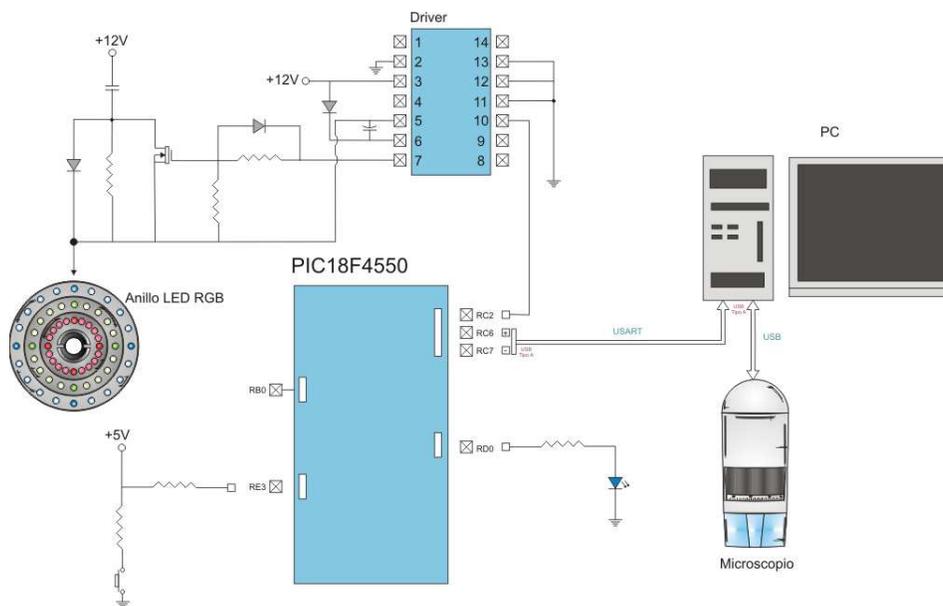


Figura 3.11 Recursos del microprocesador PIC18F4550 y su relación con los componentes del sistema electrónico de la tarjeta de adquisición de datos.

4. RESULTADOS Y DISCUSIÓN

4.1 PRUEBAS Y OBSERVACIONES DEL SISTEMA DE DETECCIÓN

En la Tabla 3.1, se presentaron los parámetros de operación respecto a la distancia de inspección que caracterizan el uso del microscopio seleccionado como elemento de adquisición de imágenes. La calibración del sistema de detección consistió inicialmente con el análisis de las dimensiones de los elementos involucrados en el proceso de detección, los cuales son 1) el tamaño promedio de las PCB's a inspeccionar cuyo rango oscilará entre: 50mm x 31mm -100mm x 80mm, y 2) el tamaño de los encapsulados con tecnología SMT usados en la placa. En la Tabla 4.1 se presentan las dimensiones típicas de distintos tipos chips SMT “planos”, generalmente utilizados para resistencias y capacitores cerámicos; además también se presentan algunos encapsulados *SOT (Small Outline Transistors)* usados para diodos/transistores/circuitos integrados, y encapsulados del tipo *QFP (Quad Flat Pack)* usados en microprocesadores (IPC, 2005).

La etapa de calibración del sistema de detección se llevó a cabo mediante un conjunto de pruebas cuya finalidad fue probar la respuesta del código desarrollado en MATLAB para la detección de los componentes SMT y que consiste en: 1) Detección del centroide de cada componente SMT y 2) Recorte del área de la imagen con cada componente.

Tabla 4.1 Dimensiones típicas de componentes con tecnologías SMT. Unidades en milímetros (IPC, 1999).

Código	Tamaño (LxW)
0201	0.6 x 0.3
0402	1.0 x 0.5
0603	1.6 x 0.8
0805	2.0 x 1.25
1206	3.2 x 1.6
1210	3.2 x 2.6
2020	5.08 x 5.08
2512	6.35 x 3.0
5664	5.6 x 6.4
SOT-23	3.0 x 1.4
SOIC-8	5.8 x 4.8
SOIC-14	8.8 x 3.8
SOT-89	4.5 x 2.5
SOD-123	3.6 x 1.5
QFP-44	10.0 x 10.0
QFP-64	12.0 x 12.0
QFP-100	23.2 x 17.8
DIL-32	20.4 x 14.4

Inicialmente se utilizó una imagen de muestra para calibración, la Figura 4.1 muestra la imagen de prueba (imagen a color en el espacio RGB) para la detección de diversos componentes SMT (QFP, SOIC, DIL). La imagen visualizada en MATLAB permite usar la función “*Data Cursor*”, la cual muestra la información de cada pixel que compone la imagen de la forma:

$$[X \ Y]=[96 \ 316] \quad (4.1)$$

$$[R \ G \ B]=[10 \ 21 \ 17] \quad (4.2)$$

Donde:

$[X \ Y]$: *Coordenadas de pixel*

$[R \ G \ B]$: *Valores de intensidad del pixel para cada plano de color*

Como se explicó en secciones anteriores, los píxeles de una imagen RGB se componen de la superposición de valores de intensidad en cada uno de los planos de color que lo integran. Dichos valores están dentro del rango $[0\ 255]$; donde mediante el análisis manual de la imagen bajo prueba se puede observar que los valores más bajos de dicho rango corresponden a las tonalidades más oscuras, mientras que los valores más altos corresponden a las tonalidades claras presentes en la PCB; por tanto, un pixel con valores de intensidad $[0\ 0\ 0]$ representará un pixel completamente negro, cuya detección es de interés para localizar el cuerpo de los componentes.

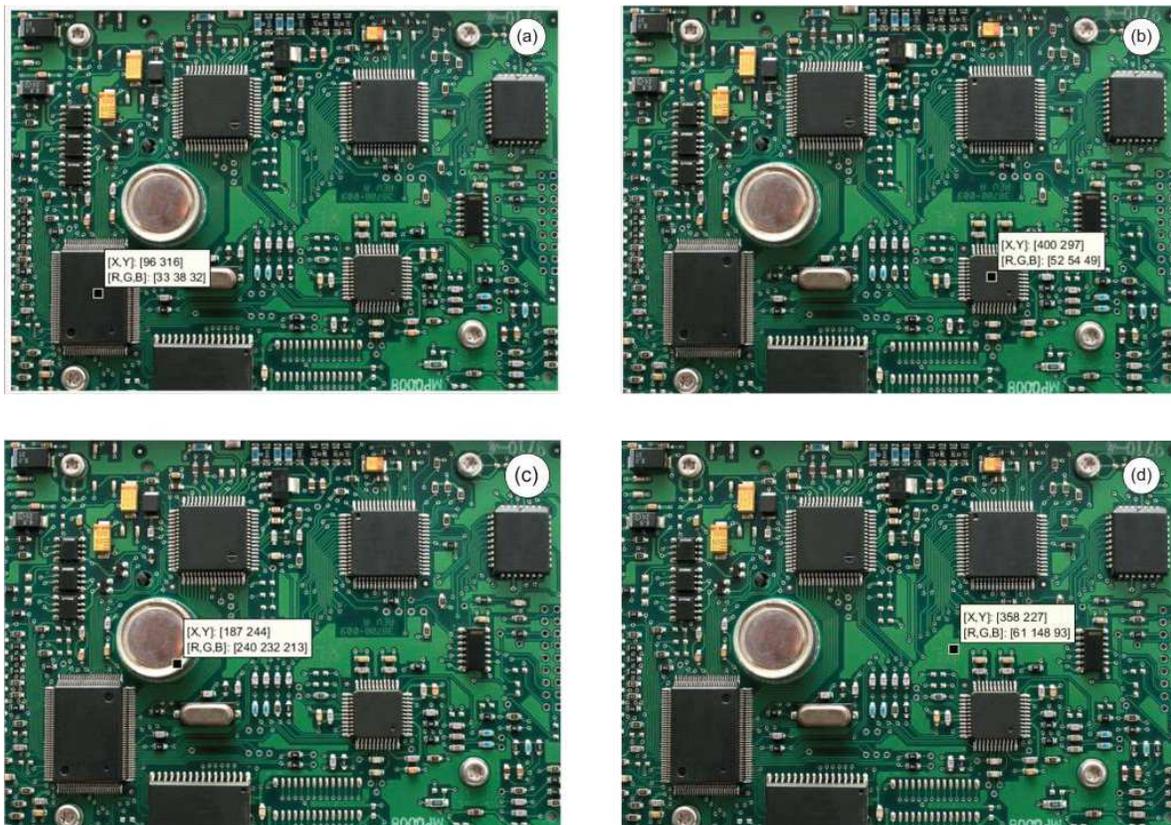


Figura 4.1 Imagen de muestra para calibración del sistema de detección. a) Valores de intensidad de componente SMT (1). b) Valores de intensidad de componente SMT (2). c) Valores de intensidad de capacitor. d) Valores de intensidad de placa.

Previo al procesamiento automático realizado por el software, fue necesario establecer ciertos valores de umbral en el código, lo cual se realizó de manera manual

siguiendo los criterios de detección de tonalidades mencionados. Nuevamente, mediante la función “*Data Cursor*” se analizaron los valores de intensidad de distintas regiones de la imagen original entre los que destacan: 1) Componentes SMT, 2) Componentes varios y 3) Pistas/placa. La Figura 4.1 muestra los valores de intensidad asociados a las regiones analizadas que fueron mencionadas. Las Figuras 4.1a y 4.1b muestran los valores de intensidad del cuerpo/encapsulado de los componentes SMT los cuales caen dentro del rango [0 55] para los tres planos de color; la Figura 4.1c muestra las intensidades detectadas en las regiones brillantes de un capacitor electrolítico de montaje “*through hole*”, donde la parte más clara del borde tiene valores superiores a 200; mientras la Figura 4.1d corresponde a los valores de intensidad de la placa de circuito impreso, los cuales presentan valores superiores a 100 en el plano de color asociado a la PCB, en este caso verde.

Los criterios previamente establecidos permiten al código modificar a la imagen original insertando valores binarios en cada uno de los tres planos de color, donde el valor “0” se asocia con la detección del color negro, y por ende del cuerpo del componente, mientras que el valor “255” se asocia a cualquier otro color; y generando una nueva imagen en blanco y negro dentro de los tres planos.

El procesamiento posterior analizó las regiones binarias para extraer los centroides de éstas, para la obtención de coordenadas (x, y) de dichos centros. En primera instancia, el programa convirtió la imagen modificada que es del tipo $M \times N \times 3$ de tipo *uint8* en una imagen binaria $M \times N$ de tipo *logical* que permite a la función “*bwareafilt*” agrupar un conjunto de píxeles según la relaciones de estos con sus vecinos, seleccionando sólo las regiones más grandes encontradas. Con este paso, la función “*regionprops*” para el análisis de las propiedades de las regiones encontradas, permiten crear una estructura “s”, donde uno de sus campos guarda las coordenadas de los centroides, que por tanto, son los centros del cuerpo de cada componente detectado, guardándolos en la matriz “*centroids*”.

```
centroids =
    67.4485    31.6162
    34.2485    20.1948
    91.2708   321.5289
    68.4795   148.3568
    69.0021   119.0934
    71.0244   182.3666
```

La Figura 4.2 muestra todas fases de procesamiento anteriormente descritas: 1) Discriminación de colores e intensidades (Figura 4.2a), 2) Binarización de la imagen modificada (Figura 4.2b), 3) Filtrado y detección de regiones binarias de interés (Figura 4.2c) y 4) Localización de los componentes y sus centros en la imagen original (Figura 4.2d). La matriz obtenida con los centros de cada componente, se utilizó para que el código escrito realice un recorte de la imagen original.

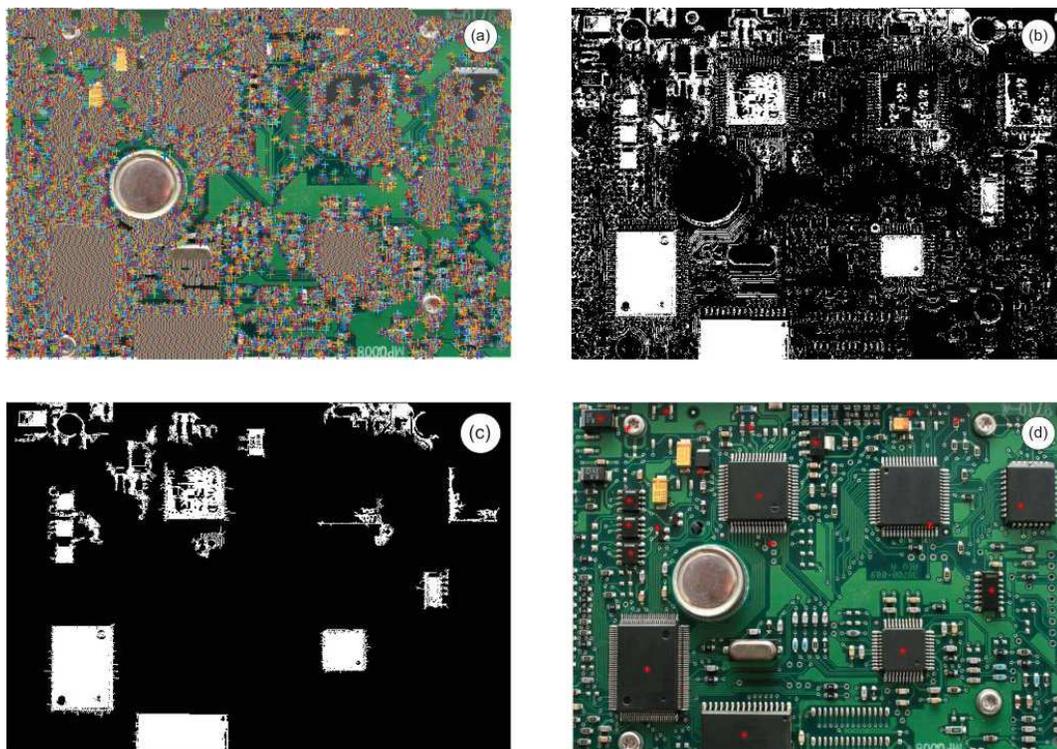


Figura 4.2 *Procesamiento de regiones binarias y obtención de centroides. a) Imagen original en el espacio RGB utilizada como referencia. b) Detección de componentes SMT mediante la selección de píxeles “negros”. c) Binarización de la imagen de referencia y extracción de las regiones detectadas. d) Visualización de la imagen original donde se marcan los centros de los componentes detectados.*

DeteccionNegros.m

```

%%Función de MATLAB para detectar los pixeles negros en una imagen RGB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Im = imread('C:\Users\Raúl\Documents\Tesis\Imágenes
Tesis\Resultados\rect_expzA2.jpg');
Im_mod = Im;
%%%Despliegue de imagen original%%%
figure; %Creación del objeto gráfico
subplot(2, 2, 1);
imshow(Im);
%% Se obtienen los valores de las dimensiones de la imagen
[m,n,bandas] = size(Im);
%%%Despliegue de imagen original con identificación de pixeles %negros
subplot(2, 2, 2);
imshow(Im)
hold on %Objeto gráfico fijo para efecto de comandos gráficos %tengan adicionales
%Extracción bandas individuales de color para el análisis del color %de cada pixel
en la imagen original bajo inspección
redChannel = Im(:, :, 1);
greenChannel = Im(:, :, 2);
blueChannel = Im(:, :, 3);
%Valores de umbral para cada banda/plano de color usados para la
%comparación y determinación de los pixeles negros detectados
umbral = 30;
umbral2 = 10;
umbral3 = 30;
%Ciclo de inspección para determinación de pixeles negros
for r=1:m
    for c=1:n
        if (redChannel(r, c) <= umbral) & (greenChannel(r, c) <= umbral2) &
(blueChannel(r, c) <= umbral3)
            sumaNegro = redChannel(r, c)+greenChannel(r, c)+ blueChannel(r, c);
            plot(c,r,'+')%Marca de indicación de pixel negro detectado sobre Im_mod
            Im_mod(r, c, 1) = 255;
            Im_mod(r, c, 2) = 255;
            Im_mod(r, c, 3) = 255;
        else %Valor cero para tonalidades de pixel distintas a negro
            Im_mod(r, c, 1) = 0;
            Im_mod(r, c, 2) = 0;
            Im_mod(r, c, 3) = 0;
        end
    end
end
hold off %Liberación de objeto gráfico
%%%Despliegue de imagen binarizada para análisis de componentes obtenidos
subplot(2, 2, 3);
imshow(Im_mod);
%%%Despliegue de imagen binarizada y proceso de filtración de áreas
%%%detectadas como componentes de la PCB
Im_bk = im2bw(Im_mod,0.5);
BW2 = bwareafilt(Im_bk,10);
subplot(2, 2, 4);
imshow(Im)
s = regionprops(BW2, 'centroid'); %Propiedades de regiones. Centros
centroids = cat(1, s.Centroid); %Concatenación de Arreglo
s1 = regionprops(BW2, 'area'); %Propiedades de regiones. Área
areas = cat(1, s1.Area); %Concatenación de Arreglo
hold on
plot(centroids(:,1),centroids(:,2), 'r*')
hold off

```

En esta sección se describe la forma en que el programa desarrollado obtiene el número de componentes detectados y los centros de éstos, de tal manera que posicione la herramienta de recorte de MATLAB “*imcrop*” mediante un ciclo de instrucción que se realiza tantas veces como centros de componentes sean detectados. La instrucción usada para este proceso se muestra a continuación:

```
iC= imcrop(Im, [centroids(i)-oX    centroids(i,2)-oY pX pY]);
```

Donde:

Im: Imagen original bajo análisis

centroids: Matriz de centroides de tamaño " $a \times 2$ "

i: Variable de conteo

a: Número de centros detectados

oX: Abscisa de origen del rectángulo de recorte

oY: Ordenada de origen del rectángulo de recorte

pX: Ancho en píxeles del rectángulo de recorte en el eje X

pY: Ancho en píxeles del rectángulo de recorte en el eje Y

Los valores *oX*, *oY*, *pX*, *pY* fueron obtenidos de manera empírica. La Tabla 4.1 muestra las dimensiones típicas de una serie de encapsulados con tecnología SMT, para la determinar dichos valores se realizó una caracterización de las dimensiones de los mismos, enfocándose en los encapsulados tipo SOT/SOD, QFP y los chips planos mencionados con anterioridad. El criterio de selección se realizó en función del área total en píxeles de las regiones binarias detectadas; por tanto, las dimensiones de rectángulo de recorte se establecen con base al tipo de componente detectado, guardando una nueva imagen sólo con el componente de interés, la cual es apta para ser procesada por el programa de tratamiento de imágenes para la determinación de la calidad de la soldadura descrito en el sección 3.2.3.

Los resultados obtenidos para la calibración de los códigos de detección se observan mediante el despliegue automático de las imágenes individuales de los componentes detectados. La Figura 4.3 muestra los recortes obtenidos de manera automática por el programa respecto a la imagen original de referencia y calibración.

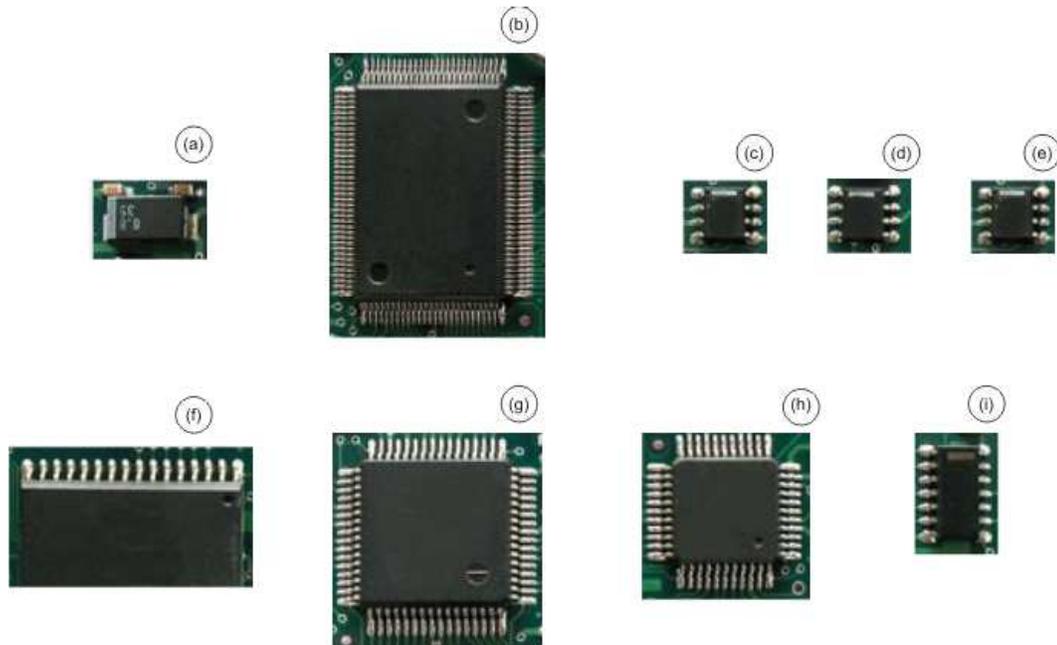


Figura 4.3 Recorte automático realizado por el sistema de detección. a) Componente SOD123 con dimensiones 3.6 x 1.5 mm (55 x 37 píxeles). b) Componente QFP-100 con dimensiones 23.2 x 17.2 mm (106 x 139 píxeles). c) Componente SOIC-8 con dimensiones 5.8 x 4.8 mm (40 x 35 píxeles). d) Componente SOIC-8 con dimensiones 5.8 x 4.8 mm (40 x 35 píxeles). e) Componente SOIC-8 con dimensiones 5.8 x 4.8 mm (40 x 35 píxeles). f) Componente SOIC-32 con dimensiones 20.4 x 14.4 mm (120 x 90 píxeles). g) Componente QFP-64 con dimensiones 12 x 12 mm (102 x 105 píxeles). h) Componente QFP-44 con dimensiones 10 x 10 mm (82 x 82 píxeles). i) Componente SOIC-14 con dimensiones 8.8 x 3.8 mm (40 x 60 píxeles).

Recortes.m

```

%%Función de MATLAB para realizar recortes de los componentes
[a b c] = size(centroids); %%Obtiene el número de centros detectados
folder = 'C:\Users\Raúl\Documents\Tesis\Imágenes Tesis\Recortes_RGB1\';

for i=1:a
    if (areas(i)>=7000 && areas(i)<=7500) %Componente Tipo QFP100
        pX = 106; %Píxeles ancho x
        pY = 139; %Píxeles largo y
        oX = 50; %Origen x
        oY = 70; %Origen y
        iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
        nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
        imwrite(iC,nameIm)
        figure
        imshow(iC)
    end
    if (areas(i)>=3200 && areas(i)<=3300)%Componente Tipo QFP-64
        pX = 102; %Píxeles ancho x
        pY = 105; %Píxeles largo y
        oX = 52; %Origen x
        oY = 55; %Origen y
        iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
        nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
        imwrite(iC,nameIm)
        figure
        imshow(iC)
    end
    if (areas(i)>=2500 && areas(i)<=2600)%Componente Tipo QFP-44
        pX = 82; %Píxeles ancho x
        pY = 82; %Píxeles largo y
        oX = 40; %Origen x
        oY = 40; %Origen y
        iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
        nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
        imwrite(iC,nameIm)
        figure
        imshow(iC)
    end
    if (areas(i)>=800 && areas(i)<=900) %Componente Tipo SOIC-14
        pX = 40; %Píxeles ancho x
        pY = 60; %Píxeles largo y
        oX = 23; %Origen x
        oY = 28; %Origen y
        iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
        nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
        imwrite(iC,nameIm)
        figure
        imshow(iC)
    end
    if (areas(i)>=495 && areas(i)<=515) %Componente Tipo SOD-123
        pX = 55; %Píxeles ancho x
        pY = 37; %Píxeles largo y
        oX = 30; %Origen x
        oY = 18; %Origen y
        iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
        nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
        imwrite(iC,nameIm)
        figure
        imshow(iC)
    end
    if (areas(i)>=470 && areas(i)<=494) %Componente Tipo SOIC-8
        pX = 40; %Píxeles ancho x
        pY = 35; %Píxeles largo y
        oX = 18; %Origen x
        oY = 16; %Origen y
        iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
        nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
        imwrite(iC,nameIm)
        figure
        imshow(iC)
    end
end

```

Recortes.m (Continuación)

```

if (areas(i)>=4000 && areas(i)<=4700) || (areas(i)>=3400 && areas(i)<=3500)
%Componente Tipo 0805 (Vertical)
    pX = 75; %Píxeles ancho x
    pY = 130; %Píxeles largo y
    oX = 39; %Origen x
    oY = 69; %Origen y
    iC= imcrop(Im,[centroids(i)-oX    centroids(i,2)-oY pX pY]);
    nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
    imwrite(iC,nameIm)
    figure
    imshow(iC)
end
if (areas(i)>=4900 && areas(i)<=5200) || (areas(i)>=3100 && areas(i)<=3200) ||
(areas(i)>=2200 && areas(i)<=2400)%Componente Tipo SOIC-32 y Tipo 0805 (Horizontal)
    pX = 123; %Píxeles ancho x
    pY = 61; %Píxeles largo y
    oX = 60; %Origen x
    oY = 29; %Origen y
    iC= imcrop(Im,[centroids(i)-oX    centroids(i,2)-oY pX pY]);
    nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
    imwrite(iC,nameIm)
    figure
    imshow(iC)
end
if (areas(i)>=7500 && areas(i)<=8000) || (areas(i)>=8100 && areas(i)<=8250)
%Componente Tipo 1210 (Horizontal)
    pX = 215; %Píxeles ancho x
    pY = 105; %Píxeles largo y
    oX = 119; %Origen x
    oY = 65; %Origen y
    iC= imcrop(Im,[centroids(i)-oX    centroids(i,2)-oY pX pY]);
    nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
    imwrite(iC,nameIm)
    figure
    imshow(iC)
end
if (areas(i)>=13000 && areas(i)<=14000) || (areas(i)>=7900 &&
areas(i)<=8100)%Componente Tipo SOT-23 Horizontal
    pX = 157; %Píxeles ancho x
    pY = 165; %Píxeles largo y
    oX = 70; %Origen x
    oY = 87; %Origen y
    iC= imcrop(Im,[centroids(i)-oX    centroids(i,2)-oY pX pY]);
    nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
    imwrite(iC,nameIm)
    figure
    imshow(iC)
end
if (areas(i)>=10000 && areas(i)<=11000) %Componente Tipo SOT-23 Vertical
(Mayo05)%%%%%%%%%%%%%%
    pX = 210; %Píxeles ancho x
    pY = 150; %Píxeles largo y
    oX = 117; %Origen x
    oY = 88; %Origen y
    iC= imcrop(Im,[centroids(i)-oX    centroids(i,2)-oY pX pY]);
    nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
    imwrite(iC,nameIm)
    figure
    imshow(iC)
elseif (areas(i)>=6400 && areas(i)<=6700)
    pX = 170; %Píxeles ancho x
    pY = 155; %Píxeles largo y
    oX = 84; %Origen x
    oY = 78; %Origen y
    iC= imcrop(Im,[centroids(i)-oX    centroids(i,2)-oY pX pY]);
    nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
    imwrite(iC,nameIm)
    figure
    imshow(iC)
end

```

La evaluación de los resultados obtenidos durante el proceso de inspección, se realizó de la siguiente forma: 1) Número de componentes presentes en la PCB, 2) Componentes detectados por el sistema, 3) Recortes Generados de cada componente individual detectado, 4) Análisis Correcto, que corresponde a los aciertos para la determinación de la calidad de la soldadura, 5) Efectividad Detección, que corresponde a la tasa de aciertos en la detección de componentes y 6) Efectividad Análisis, corresponde a la tasa de aciertos la calidad de la soldadura de los componentes detectados, como se muestra al final de esta sección en la Tabla 4.2.

En la sección 3.2.3 se expuso el método de análisis de la calidad de las uniones de soldadura en función del histograma producido por los valores de intensidad detectados por el programa de procesamiento de imágenes. Para la evaluación de los resultados descritos en el párrafo anterior, el sistema de detección analiza todos los valores producidos por el histograma.

Visualmente se observa que las gráficas que representan una unión de soldadura aceptable presentan sólo un valor pico de intensidad, el cual debe superar el valor de umbral establecido durante la etapa de calibración para cada color. La condición que debe cumplir cada recorte (componente individual) para considerarse dentro de la categoría “Detección Correcta”, es que el programa determine que el histograma es “Aceptable” en los tres espacios de color analizados.

En la Figuras 4.4, 4.5 y 4.6 se muestran los histogramas para cada uno de los espacios RGB de los componentes individuales de la imagen de prueba. En dichas imágenes se observan los valores de intensidad que el sistema de detección clasifica como valores pico y mediante los cuales determina la viabilidad de cada unión de soldadura, y es en función de estos datos arrojados que clasifica la unión de soldadura a partir del procesamiento de la imagen de la PCB.

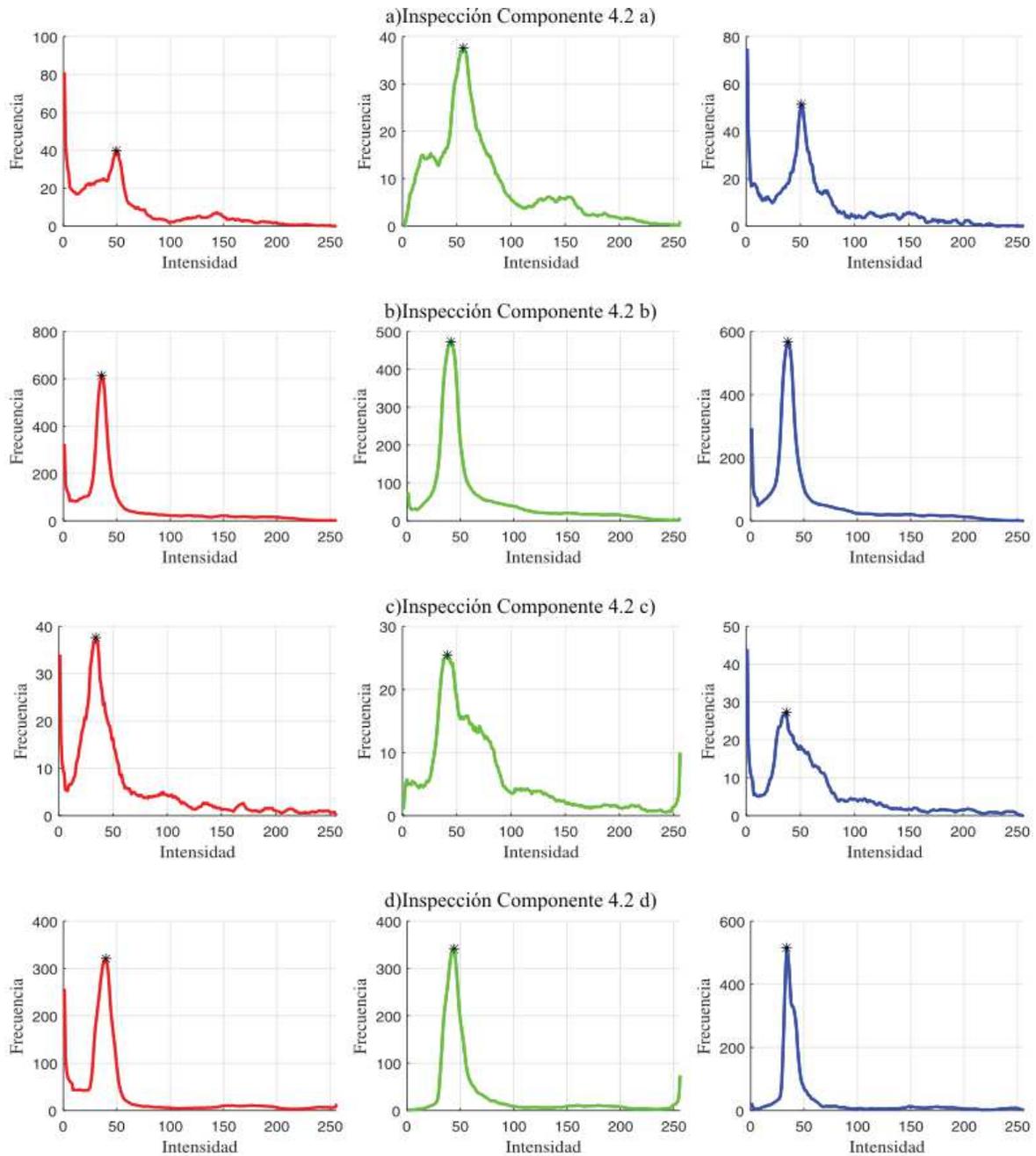


Figura 4.4 Histogramas de intensidad contra frecuencia en la imagen de prueba, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos a), b), c) y d) de la Figura 4.3 (Fuente: elaboración propia).

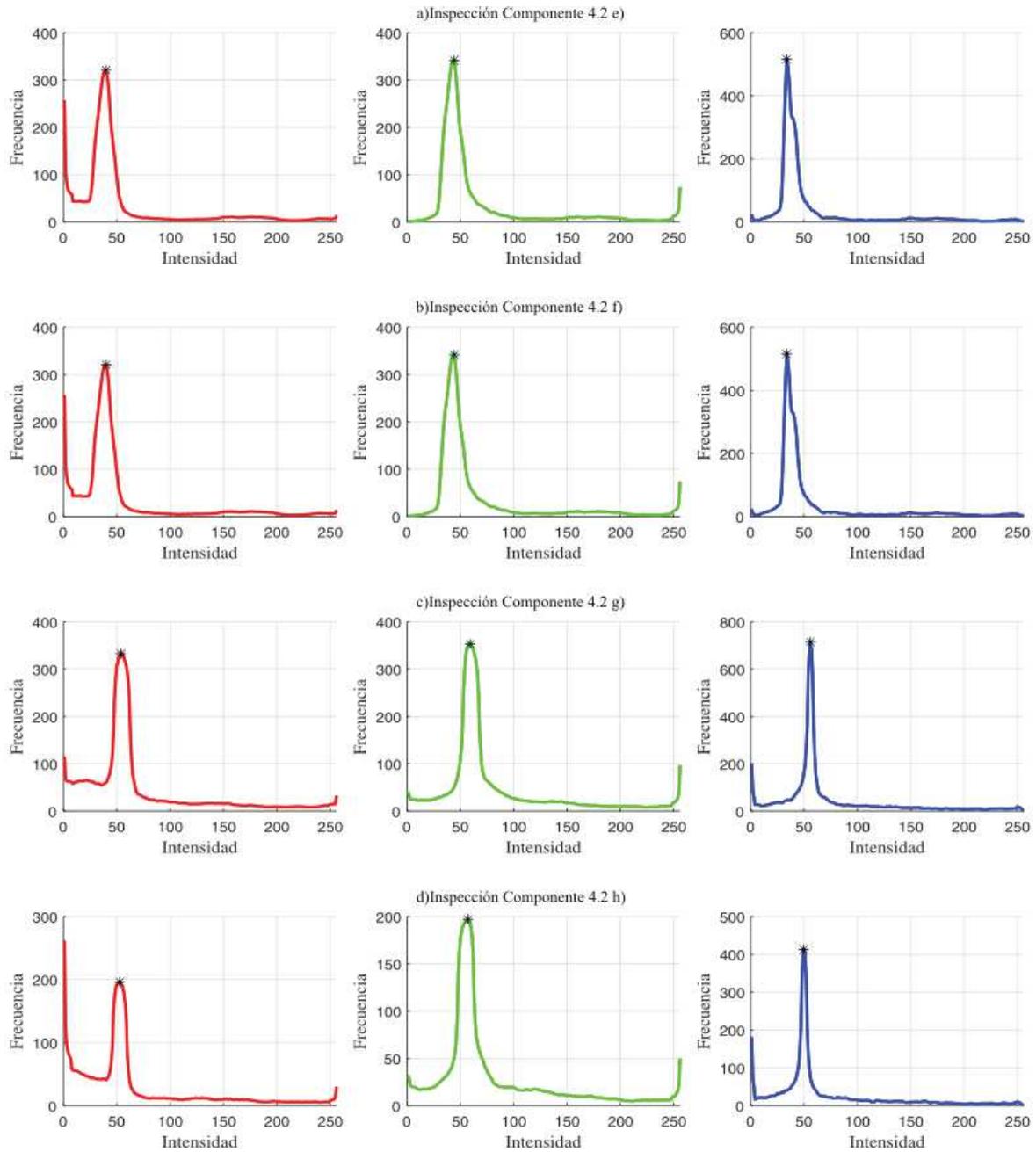


Figura 4.5 Histogramas de intensidad contra frecuencia en la imagen de prueba, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos e), f), g) y h) de la Figura 4.3 (Fuente: elaboración propia).

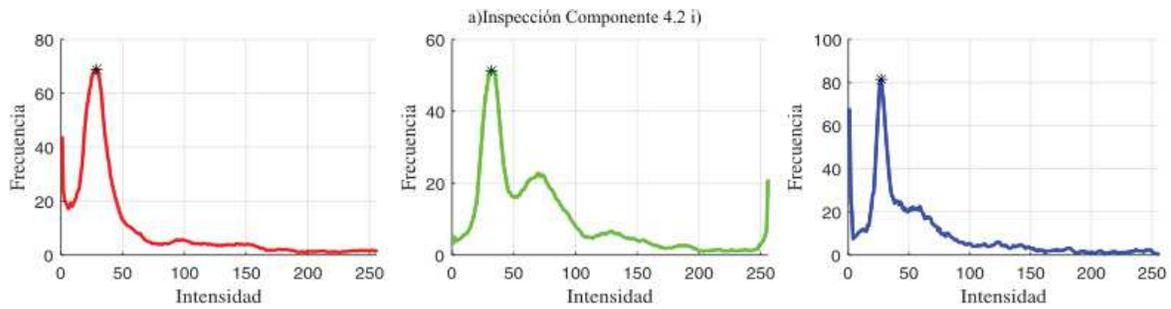


Figura 4.6 Histogramas de intensidad contra frecuencia en la imagen de prueba, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en el inciso i) de la Figura 4.3 (Fuente: elaboración propia).

La generación de los histogramas de las Figuras 4.4, 4.5 y 4.6 marcan la conclusión de la etapa de calibración, llevada a cabo mediante el análisis de una imagen de prueba seleccionada deliberadamente debido a que las condiciones de la misma se consideraban idóneas para realizar los primeros experimentos. Los resultados obtenidos presentados a continuación son producto de la aplicación del mismo procedimiento realizado en la fase de calibración sobre imágenes reales obtenidas mediante el elemento de adquisición de imágenes. Dichos resultados se muestran en dos partes, la primera abarca el análisis de la imagen obtenida de la PCB bajo inspección sin la intervención del sistema de iluminación; mientras que en la segunda, contiene el análisis de la misma PCB, esta vez con el anillo de iluminación RGB encendido para fines de comparación del desempeño del sistema completo bajo ambas condiciones de iluminación.

HistogramasRGB.m

```

%%Función para graficación de histogramas de los planos de RGB(concatenada)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I=imread('C:\Users\Raúl\Documents\Tesis\Imágenes
Tesis\Recortes_RGB1\RecortePCB_9.jpg');
%Se verifica que la imagen I sea RGB (que contenga los tres planos)
if(size(I, 3) ~=3)
    error('La imagen debe ser RGB')
end
close all
%Se establecen 256 valores que representan la profundidad del tipo de dato
%permisible
nBins = 256;
%Se encuentran los histogramas para cada plano RGB
[rHist,x] = imhist(I(:,:,1),nBins);
[gHist,y] = imhist(I(:,:,2),nBins);
[bHist,z] = imhist(I(:,:,3),nBins);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PLANO ROJO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure%Creación del objeto gráfico
xlabel('Intensidad','FontSize', 12, 'FontName', 'times') % Etiqueta
ylabel('Frecuencia','FontSize', 12, 'FontName', 'times') % Etiqueta
xlim([0 256]) %Límites eje X
fig = figure(1);
fig.Units = 'centimeters';
fig.Position(3) = 8;
fig.Position(4) = 6;
set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
set(fig,'defaultAxesColorOrder',[0 0 0]);
fig.PaperPositionMode = 'auto';
grid
rHist = smooth(rHist,6);
hold on;
plot(1:256, rHist, '-r','LineWidth',2);%Graficación del Histograma
%valores máximos del vector del histograma y sus localizaciones
Rpks = findpeaks(rHist);
Rumbral = 0.8*(max(Rpks));%Condición para filtrar de picos menores a 80%
[Rpks,Rlocs] = findpeaks(rHist,'MINPEAKHEIGHT',Rumbral,'Threshold',.1);
%Condición para suprimir los picos cuando la cercanía es menor a 5 unidades
%de intensidad en caso de que el programa detecte picos muy próximos
if length(Rpks)>= 2
    [r,s] = max(Rlocs);%Banderas para graficación correcta del máximo útil
    Rxumb = r-min(Rlocs);
    if Rxumb <=6
        Rpks = max(Rpks);
        Rlocs = Rlocs(s);
    end
end
Rpicos = Rpks(Rpks>=Rumbral);%Determinación del número de picos encontrados
plot(Rlocs, Rpks,'*k', 'LineWidth',1)
hold off
print(fig,'C:\Users\Raúl\Documents\Tesis\Imágenes
Tesis\Histogramas\Histograma_R','-depsc') %Exportación a formato .eps (CorelDraw)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PLANO VERDE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure%Creación del objeto gráfico
xlabel('Intensidad','FontSize', 12, 'FontName', 'times') % Etiqueta
ylabel('Frecuencia','FontSize', 12, 'FontName', 'times') % Etiqueta
xlim([0 256]) %Límites eje X
fig = figure(2);
fig.Units = 'centimeters';
fig.Position(3) = 8;
fig.Position(4) = 6;
set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
fig.PaperPositionMode = 'auto';
grid
gHist = smooth(gHist,15);
hold on;
plot(1:256, gHist, '-g','LineWidth',2);%Graficación del Histograma

```

HistogramasRGB.m (Continuación)

```

%Valores máximos del vector del histograma y sus localizaciones
Gpks = findpeaks(gHist);
Gumbral = 0.8*(max(Gpks));%Condición para filtrar de picos menores a 80%
[Gpks,Glocs] = findpeaks(gHist,'MINPEAKHEIGHT',Gumbral,'Threshold',0.2);
%Condición para suprimir los picos cuando la cercanía es menor a 5 unidades
%de intensidad en caso de que el programa detecte picos muy próximos
if length(Gpks)>= 2
    [r,s] = max(Glocs);%Banderas para graficación correcta del máximo útil
    Gxumb = r-min(Glocs);
    if Gxumb <=6
        Gpks = max(Gpks);
        Glocs = Glocs(s);
    end
end
Gpicos = Gpks(Gpks>=Gumbral);%Determinación del número de picos encontrados
plot(Glocs, Gpks,'*k', 'LineWidth',1)
hold off
print(fig,'C:\Users\Raúl\Documents\Tesis\Imágenes
Tesis\Histogramas\Histograma_G','-depssc') %Exportación a formato .eps (CorelDraw)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PLANO AZUL%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure%Creación del objeto gráfico
xlabel('Intensidad','FontSize', 12, 'FontName' , 'times') % Etiqueta
ylabel('Frecuencia','FontSize', 12, 'FontName' , 'times') % Etiqueta
xlim([0 256]) %Límites eje X
fig = figure(3);
fig.Units = 'centimeters';
fig.Position(3) = 8;
fig.Position(4) = 6;
set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
fig.PaperPositionMode = 'auto';
grid
bHist = smooth(bHist,6);
hold on;
plot(1:256, bHist, '-b','LineWidth',2);%Graficación del Histograma
%valores máximos del vector del histograma y sus localizaciones
Bpks = findpeaks(bHist);
Bumbral = 0.8*(max(Bpks));%Condición para filtrar de picos menores a 80%
[Bpks,Blocs] = findpeaks(bHist,'MINPEAKHEIGHT',Bumbral,'Threshold',0.1);
%Condición para suprimir los picos cuando la cercanía es menor a 5 unidades
%de intensidad en caso de que el programa detecte picos muy próximos
if length(Bpks)>= 2
    [r,s] = max(Blocs);%Banderas para graficación correcta del máximo útil
    Bxumb = r-min(Blocs);
    if Bxumb <=6
        Bpks = max(Bpks);
        Blocs = Blocs(s);
    end
end
Bpicos = Bpks(Bpks>=Bumbral);%Determinación del número de picos encontrados
plot( Blocs, Bpks,'*k', 'LineWidth',1)
hold off
print(fig,'C:\Users\Raúl\Documents\Tesis\Imágenes
Tesis\Histogramas\Histograma_B','-depssc') %Exportación a formato .eps (CorelDraw)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Ciclo para determinación de la calidad de la unión de soldadura
if length(Rpicos)== 1
    disp ('Rojo Aceptable')
    if length(Gpicos)== 1
        disp ('Verde Aceptable')
        if length(Bpicos)== 1
            disp ('Azul Aceptable')
            disp ('Unión de Soldadura Aceptable') %Se cumple la condición cada color
        else
            disp ('Unión de Soldadura Inaceptable')
        end
    else
        disp ('Unión de Soldadura Inaceptable')
    end
else
    disp ('Unión de Soldadura Inaceptable')
end
end

```

La inspección de la nueva imagen consistió de igual forma en el análisis de los píxeles que conforman la imagen RGB base y valores están dentro del rango [0 255] para que el código escrito continúe detectando las tonalidades oscuras y claras presentes en la PCB; y por tanto, como se mencionó, sea posible detectar cuerpo de los componentes. La Figura 4.7 muestra los valores de intensidad asociados a las regiones analizadas como fue mencionado previamente en esta sección.

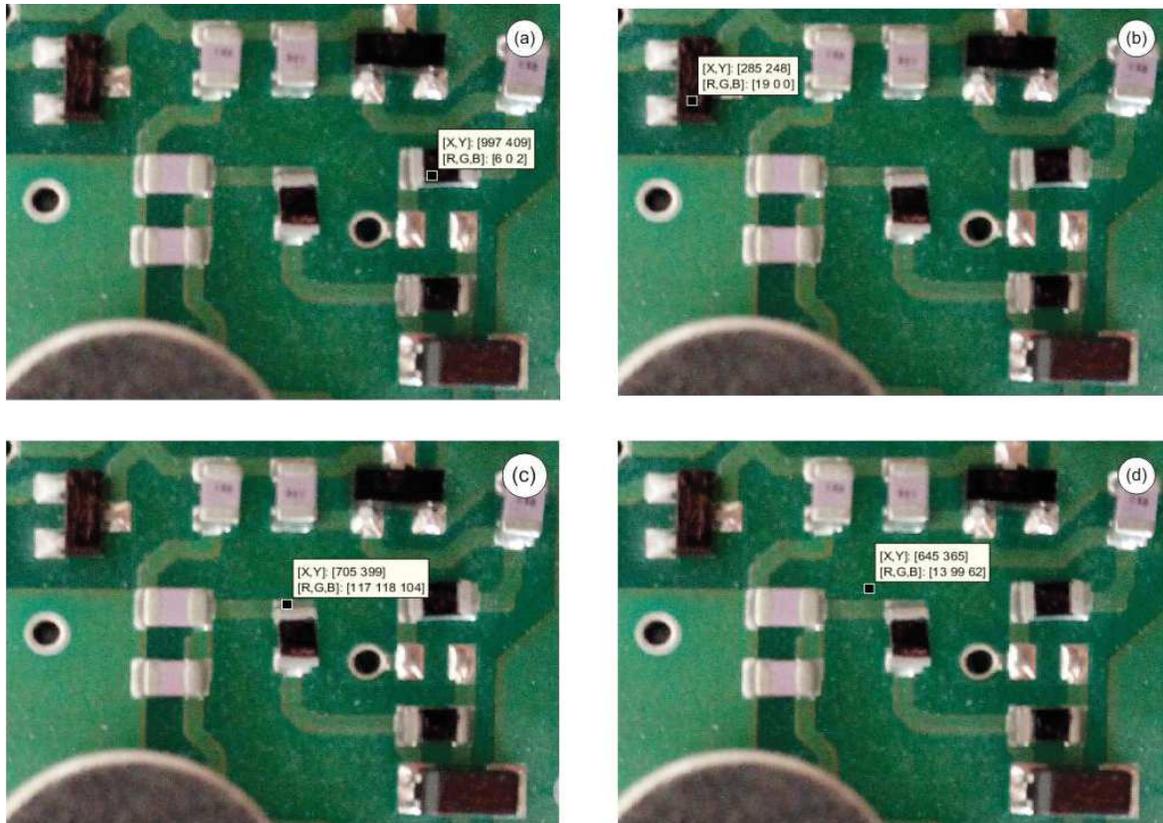


Figura 4.7 Imagen real obtenida por el sistema de adquisición de imágenes. a) Valores de intensidad de componente SMT (1). b) Valores de intensidad de componente SMT (2). c) Valores de intensidad de unión de soldadura. e) Valores de intensidad de placa.

La Figura 4.8 muestra todas fases de procesamiento anteriormente descritas: 1) Discriminación de colores e intensidades (Figura 4.8a), 2) Binarización de la imagen modificada (Figura 4.8b), 3) Filtrado y detección de regiones binarias de interés (Figura

4.8c) y 4) Localización de los componentes y sus centros en la imagen original (Figura 4.8d). Para este caso a diferencia de la Figura 4.2, se utiliza una imagen de una PCB obtenida directamente por el elemento de adquisición de imágenes.

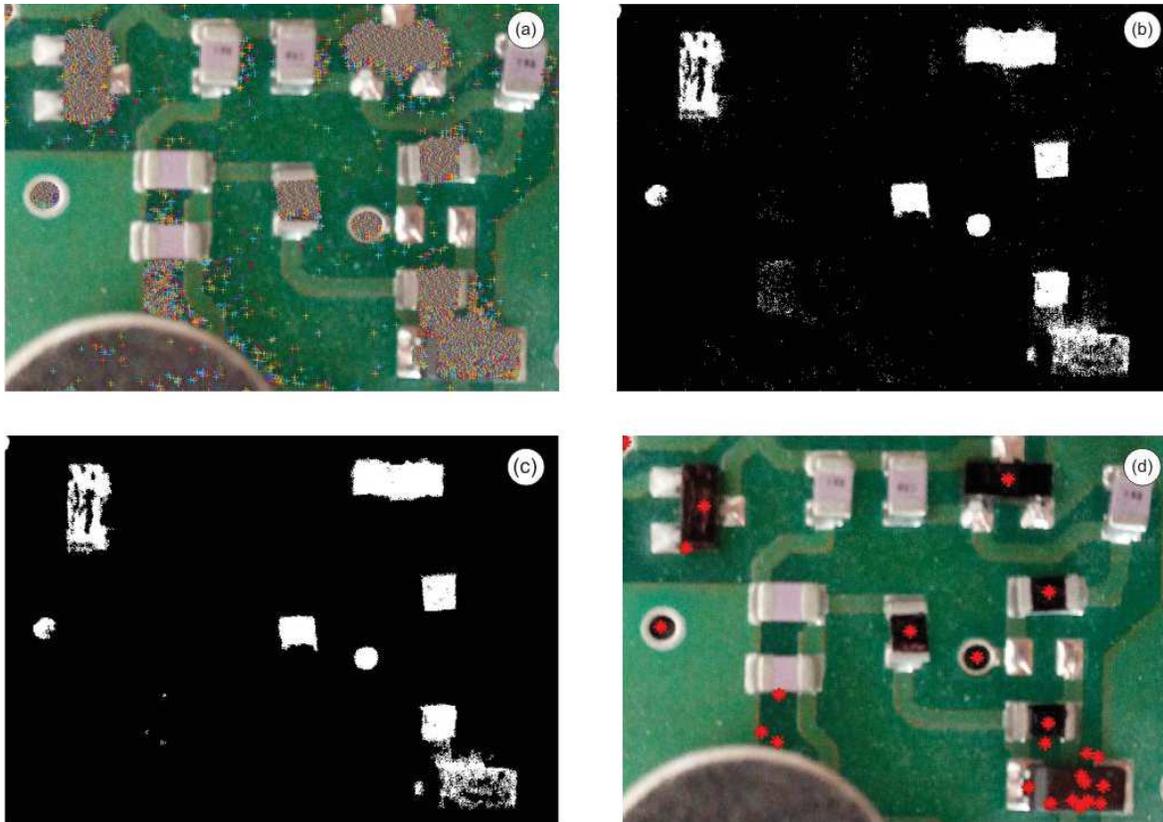


Figura 4.8 *Procesamiento de regiones binarias y obtención de centroides de la imagen real obtenida por el sistema de adquisición de imágenes. a) Imagen original en el espacio RGB utilizada como referencia. b) Detección de de componentes SMT mediante la selección de pixeles “negros”. c) Binarización de la primera imagen prueba obtenida por el sistema y extracción de las regiones detectadas. d) Visualización de la imagen original donde se marcan los centros de los componentes detectados.*

De igual manera que en la etapa de calibración, los resultados obtenidos se observan mediante el despliegue automático de las imágenes individuales de cada componentes detectados. La Figura 4.9 muestra los recortes generados por el programa respecto a la imagen original obtenida por la cámara.

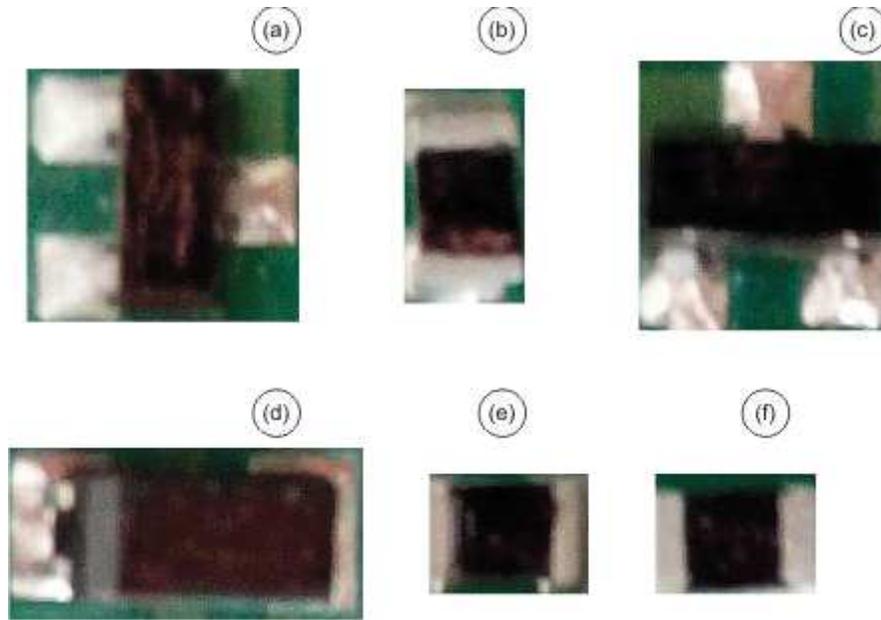


Figura 4.9 Recorte automático realizado por el sistema de detección sobre la imagen real obtenida por el sistema de adquisición de imágenes. a) Componente SOT-23(vertical) con dimensiones 3.0 x 1.4mm (205 x 190 pixeles). b) Componente 0805(vertical) con dimensiones 2.0 x 1.25mm (90 x 162 pixeles). c) Componente SOT-23(horizontal) con dimensiones 3.0 x 1.4mm (190 x 203 pixeles). d) Componente 1210 con dimensiones 3.2 x 2.6 mm (266 x 130 pixeles). e) Componente 0805(horizontal) con dimensiones 2.0 x 1.25mm (120 x 90 pixeles). f) Componente 0805(horizontal) con dimensiones 2.0 x 1.25mm (120 x 90 pixeles).

Siguiendo el mismo procedimiento establecido en la fase de calibración, en la Figuras 4.10 y 4.11 se muestran los histogramas generados para cada uno de los espacios RGB de los componentes individuales detectados por el programa, que fueron obtenidos de la imagen real provista por el sistema de adquisición de imágenes.

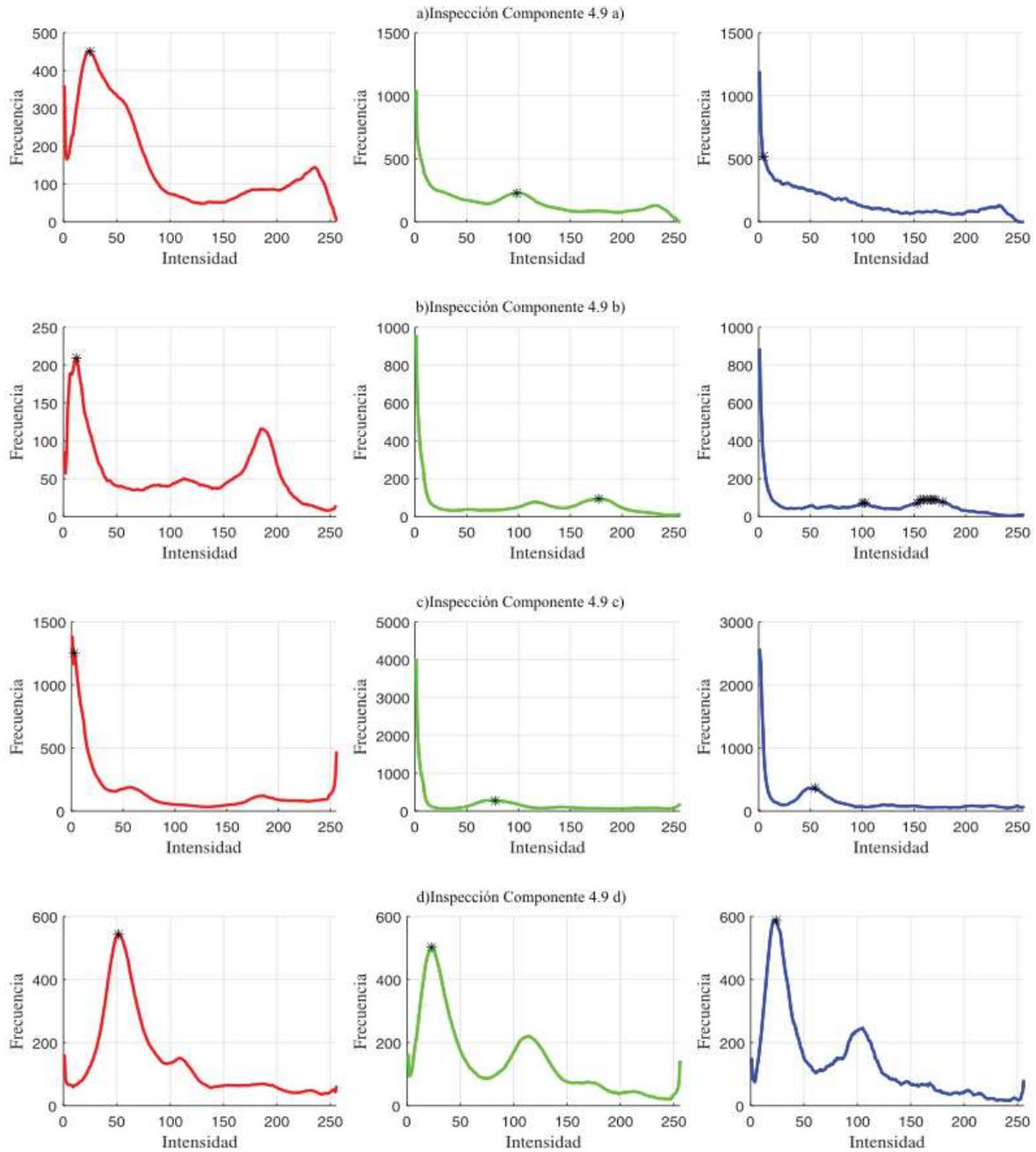


Figura 4.10 Histogramas de intensidad contra frecuencia de píxeles en la imagen real obtenida por el sistema de adquisición de imágenes, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos a), b), c) y d) de la Figura 4.9 (Fuente: elaboración propia).

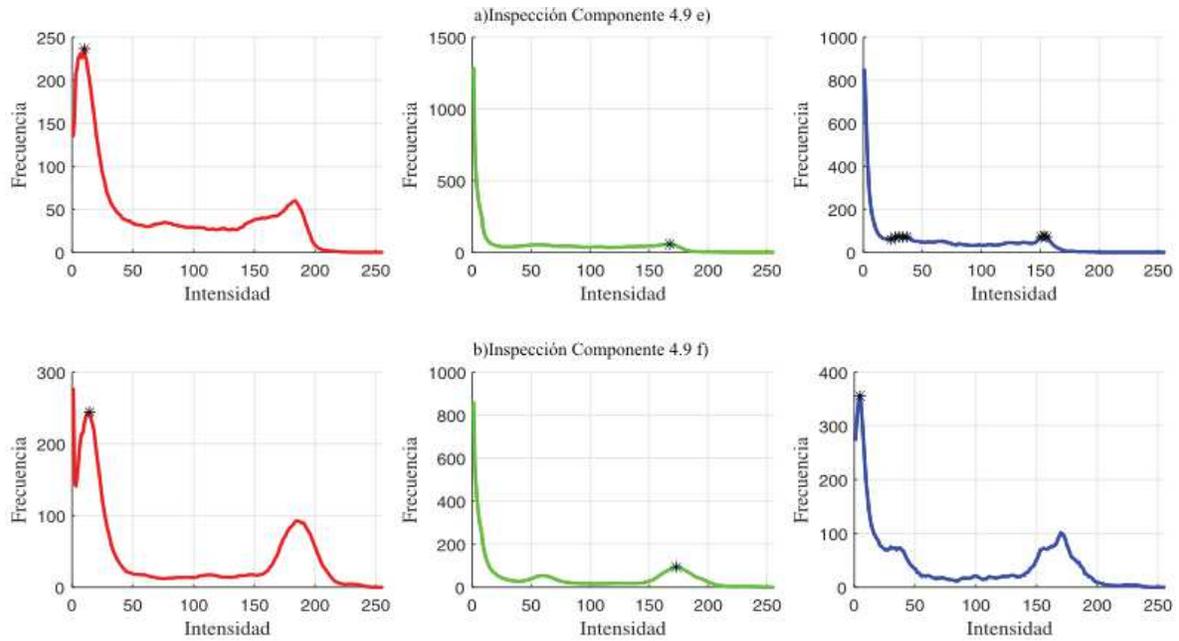


Figura 4.11 Histogramas de intensidad contra frecuencia de píxeles en la imagen de real obtenida por el sistema de adquisición de imágenes, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos e) y f) de la Figura 4.9 (Fuente: elaboración propia).

A continuación se muestra el desempeño del sistema de detección con las condiciones de iluminación modificadas por el anillo RGB, para la observación de los efectos de éste sobre la PCB bajo prueba y de la respuesta del código generado. La Figura 4.12 muestra los valores de intensidad asociados a las regiones a analizar.

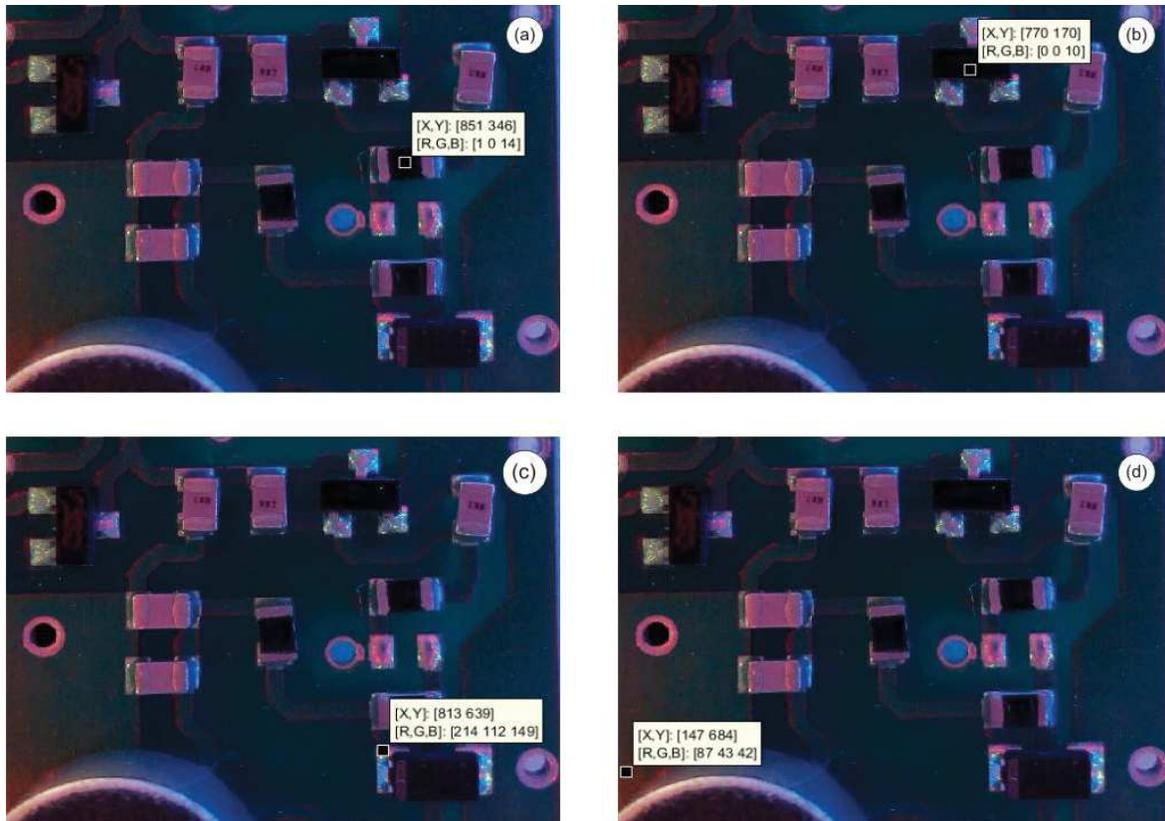


Figura 4.12 Imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido. a) Valores de intensidad de componente SMT 0805 (1). b) Valores de intensidad de componente SMT SOT-23 (2). c) Valores de intensidad de unión de soldadura. d) Valores de intensidad de placa.

La Figura 4.13 muestra todas fases de procesamiento establecidas: 1) Discriminación de colores e intensidades (Figura 4.13 a), 2) Binarización de la imagen modificada (Figura 4.13 b), 3) Filtrado y detección de regiones binarias de interés (Figura 4.13 c) y 4) Localización de los componentes y sus centros en la imagen original (Figura 4.13 d). Para este caso, a diferencia de la Figura 4.7, se utiliza la imagen de la misma PCB obtenida directamente por el elemento de adquisición de imágenes y el sistema de iluminación encendido.

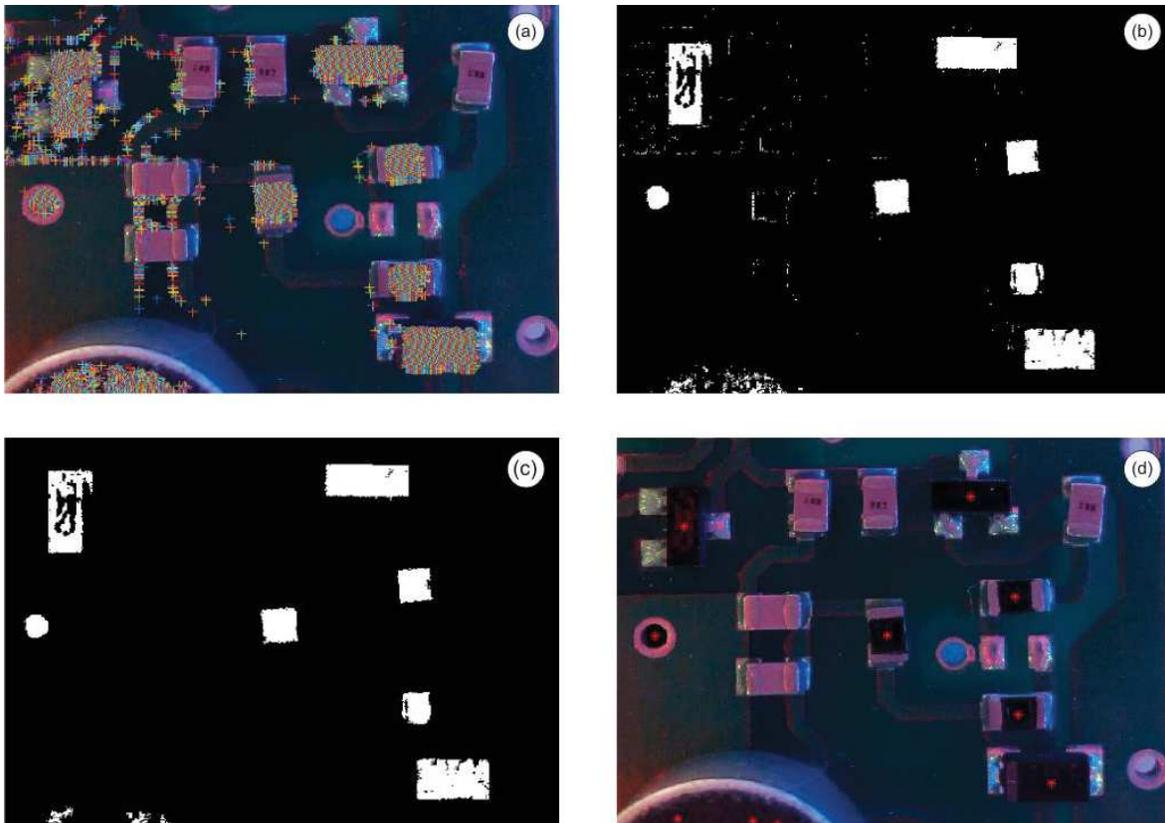


Figura 4.13 *Procesamiento de regiones binarias y obtención de centroides de la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido. a) Imagen original en el espacio RGB utilizada como referencia. b) Detección de componentes SMT mediante la selección de píxeles “negros”. c) Binarización de la primera imagen prueba obtenida por el sistema y extracción de las regiones detectadas. d) Visualización de la imagen original donde se marcan los centros de los componentes detectados.*

Los efectos producidos por el sistema de iluminación se observan tanto en los valores de intensidad en la imagen original al “resaltar” las secciones negras de los componentes SMT, que se traduce en valores más cercanos a $[0\ 0\ 0]$ para cada plano respecto a las dos imágenes analizadas previamente; lo cual tuvo como consecuencia que el programa detectara y generara regiones binarias más uniformes y por tanto, mayor precisión en la localización del centro de cada componente. La Figura 4.14 muestra los recortes generados por el programa respecto a la imagen original obtenida por la cámara.

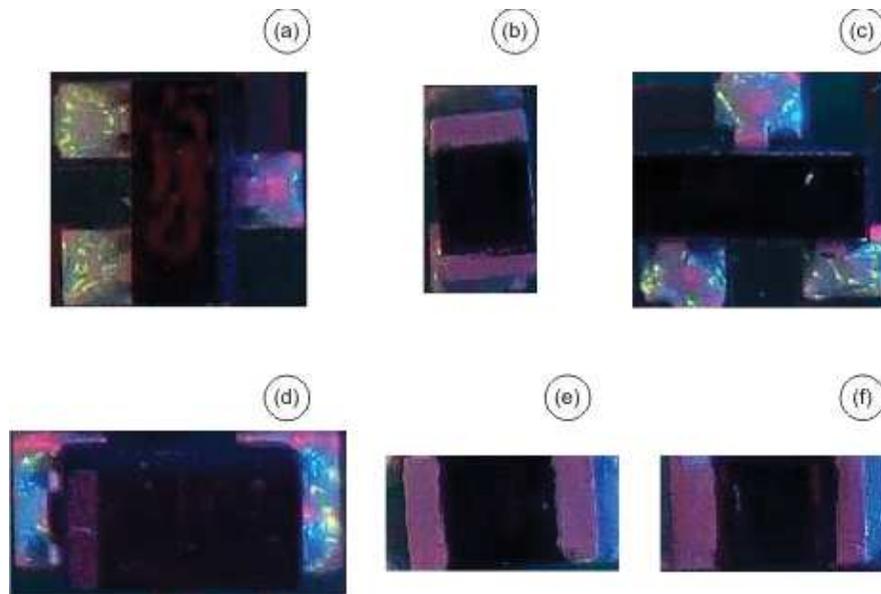


Figura 4.14 Recorte automático realizado por el sistema de detección sobre la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido. a) Componente SOT-23(vertical) con dimensiones 3.0 x 1.4mm (205 x 190 pixeles). b) Componente 0805(vertical) con dimensiones 2.0 x 1.25mm (90 x 162 pixeles). c) Componente SOT-23(horizontal) con dimensiones 3.0 x 1.4mm (190 x 203 pixeles). d) Componente 1210 con dimensiones 3.2 x 2.6 mm (266 x 130 pixeles). e) Componente 0805(horizontal) con dimensiones 2.0 x 1.25mm (120 x 90 pixeles). f) Componente 0805(horizontal) con dimensiones 2.0 x 1.25mm (120 x 90 pixeles).

De nueva cuenta, la última fase dentro de la inspección de la PCB se muestra en las Figuras 4.15 y 4.16 que contienen los histogramas generados para cada uno de los espacios RGB de cada componentes individual detectado por el programa, obtenidos de la imagen real provista por el sistema de adquisición de imágenes y con el sistema de iluminación encendido.

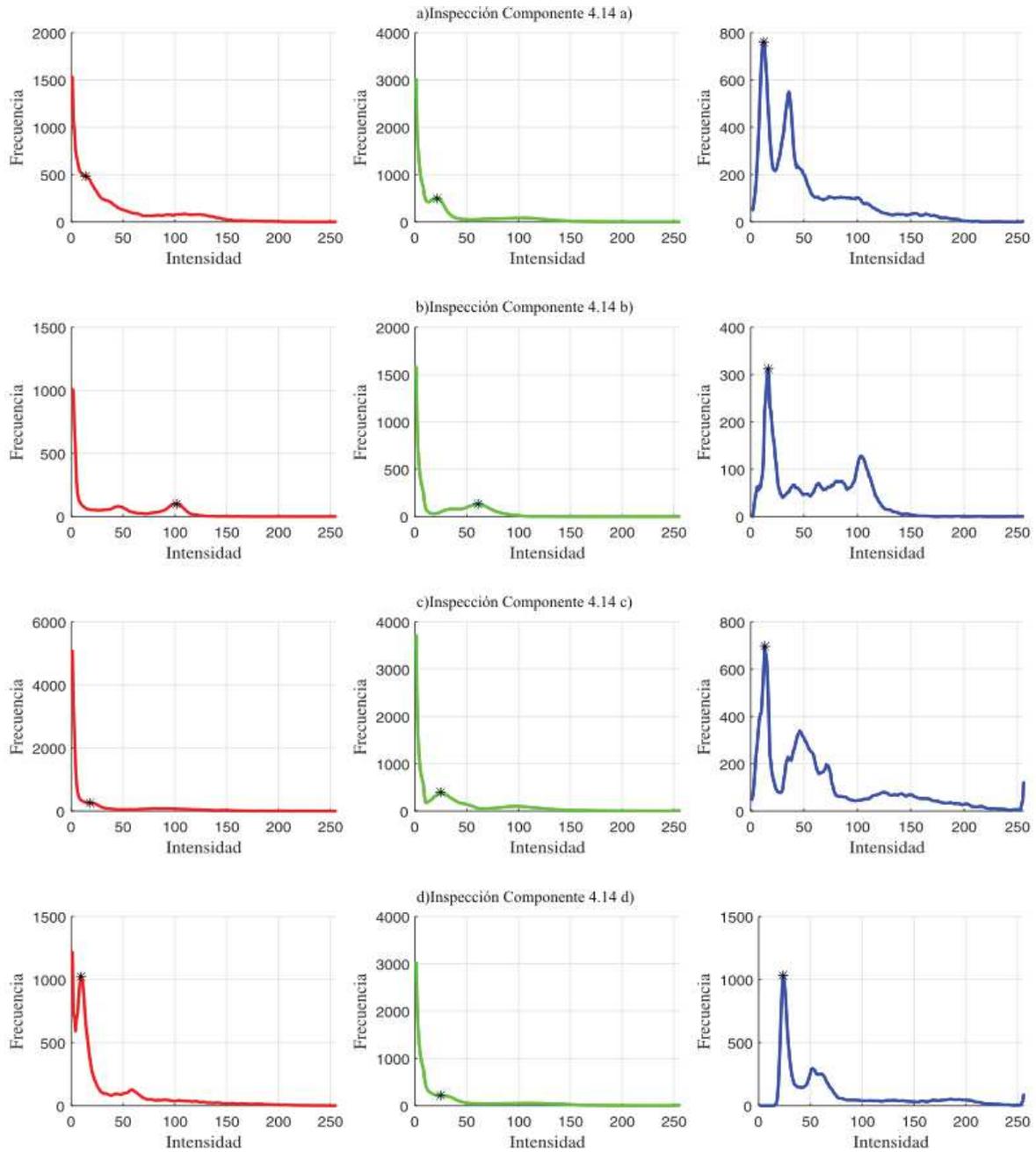


Figura 4.15 Histogramas de intensidad contra frecuencia de píxeles en la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos a), b), c) y d) de la Figura 4.14 (Fuente: elaboración propia).

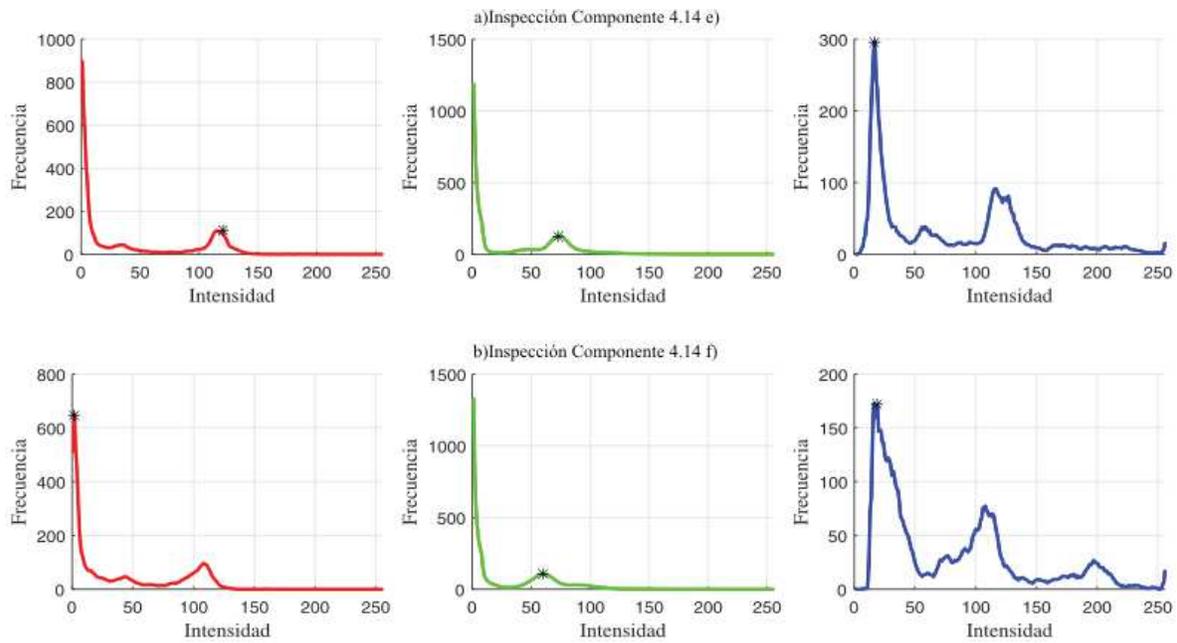


Figura 4.16 Histogramas de intensidad contra frecuencia de pixeles en la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos e) y f) de la Figura 4.9 (Fuente: elaboración propia).

A continuación se muestra una última prueba al desempeño del sistema de detección con las condiciones de iluminación modificadas por el anillo RGB, para la observación de los efectos de éste sobre la PCB bajo prueba; en este experimento, ésta contiene uniones soldadas de manera deficiente para evaluar la respuesta del código generado para el análisis de uniones de soldadura inaceptables. La Figura 4.17 muestra los valores de intensidad asociados a las regiones a analizar.

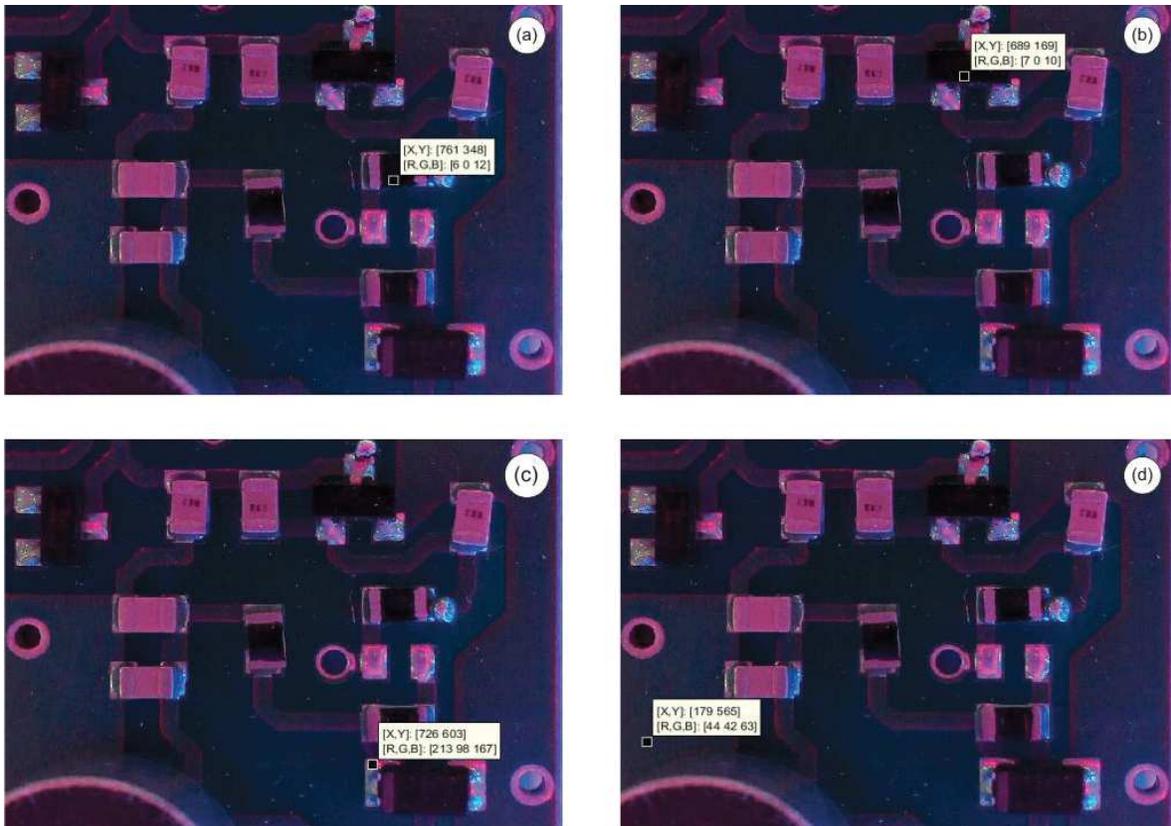


Figura 4.17 Imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido para uniones de soldadura inaceptables. a) Valores de intensidad de componente SMT 0805 (1). b) Valores de intensidad de componente SMT SOT-23 (2). c) Valores de intensidad de unión de soldadura. d) Valores de intensidad de placa.

La Figura 4.18 muestra todas fases de procesamiento establecidas: 1) Discriminación de colores e intensidades (Figura 4.18 a), 2) Binarización de la imagen modificada (Figura 4.18 b), 3) Filtrado y detección de regiones binarias de interés (Figura 4.18 c) y 4) Localización de los componentes y sus centros en la imagen original (Figura 4.18 d). Para este caso, la PCB bajo análisis contiene uniones de soldadura deficientes y por tanto, preclasificadas como inaceptables.

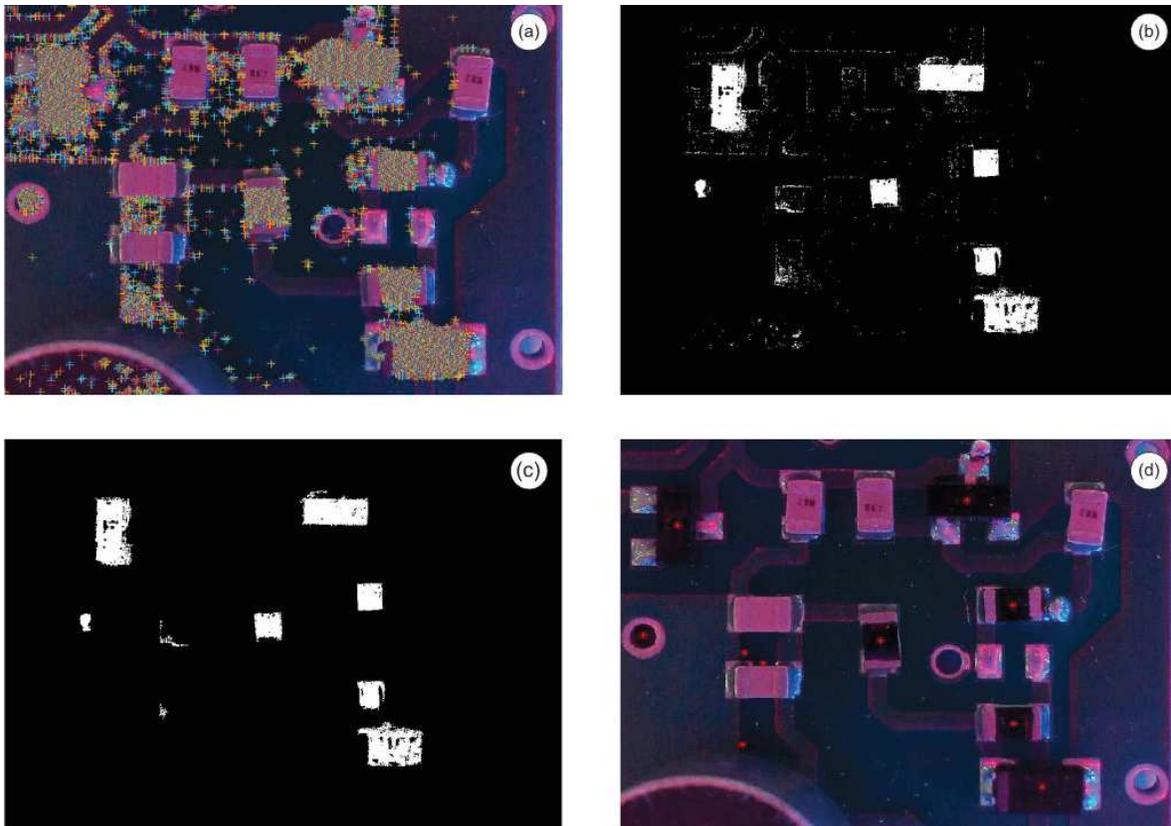


Figura 4.18 *Procesamiento de regiones binarias y obtención de centroides de la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido para uniones de soldadura inaceptables. a) Imagen original en el espacio RGB utilizada como referencia. b) Detección de componentes SMT mediante la selección de píxeles “negros”. c) Binarización de la primera imagen prueba obtenida por el sistema y extracción de las regiones detectadas. d) Visualización de la imagen original donde se marcan los centros de los componentes detectados.*

La Figura 4.19 muestra los recortes generados por el programa respecto a la imagen original obtenida por la cámara. Dicha imagen contiene dos uniones soldadas de manera deficiente, presentando exceso de soldadura como defecto; y por tanto, están preclasificadas como uniones de soldadura inaceptables, las cuales son mostradas en las Figuras 4.19 c) y 4.19 e).

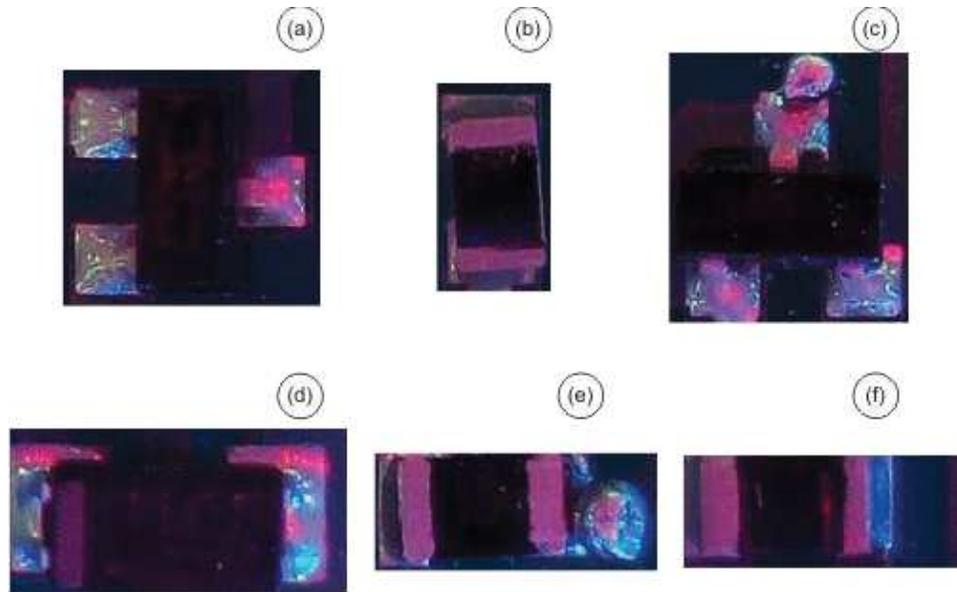


Figura 4.19 Recorte automático realizado por el sistema de detección sobre la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido. a) Componente SOT-23(vertical) con dimensiones 3.0 x 1.4mm (205 x 190 pixeles). b) Componente 0805(vertical) con dimensiones 2.0 x 1.25mm (90 x 162 pixeles). c) Componente SOT-23 con exceso de soldadura (horizontal) con dimensiones 3.0 x 1.4mm (190 x 203 pixeles). d) Componente 1210 con dimensiones 3.2 x 2.6 mm (266 x 130 pixeles). e) Componente 0805con exceso de soldadura (horizontal) con dimensiones 2.0 x 1.25mm (120 x 90 pixeles). f) Componente 0805(horizontal) con dimensiones 2.0 x 1.25mm (120 x 90 pixeles).

La última fase de inspección de la PCB se muestra en las Figuras 4.20 y 4.21 que contienen los histogramas generados para cada uno de los espacios RGB de cada componentes individual detectado por el programa, obtenidos de la imagen real provista por el sistema de adquisición de imágenes y con el sistema de iluminación encendido.

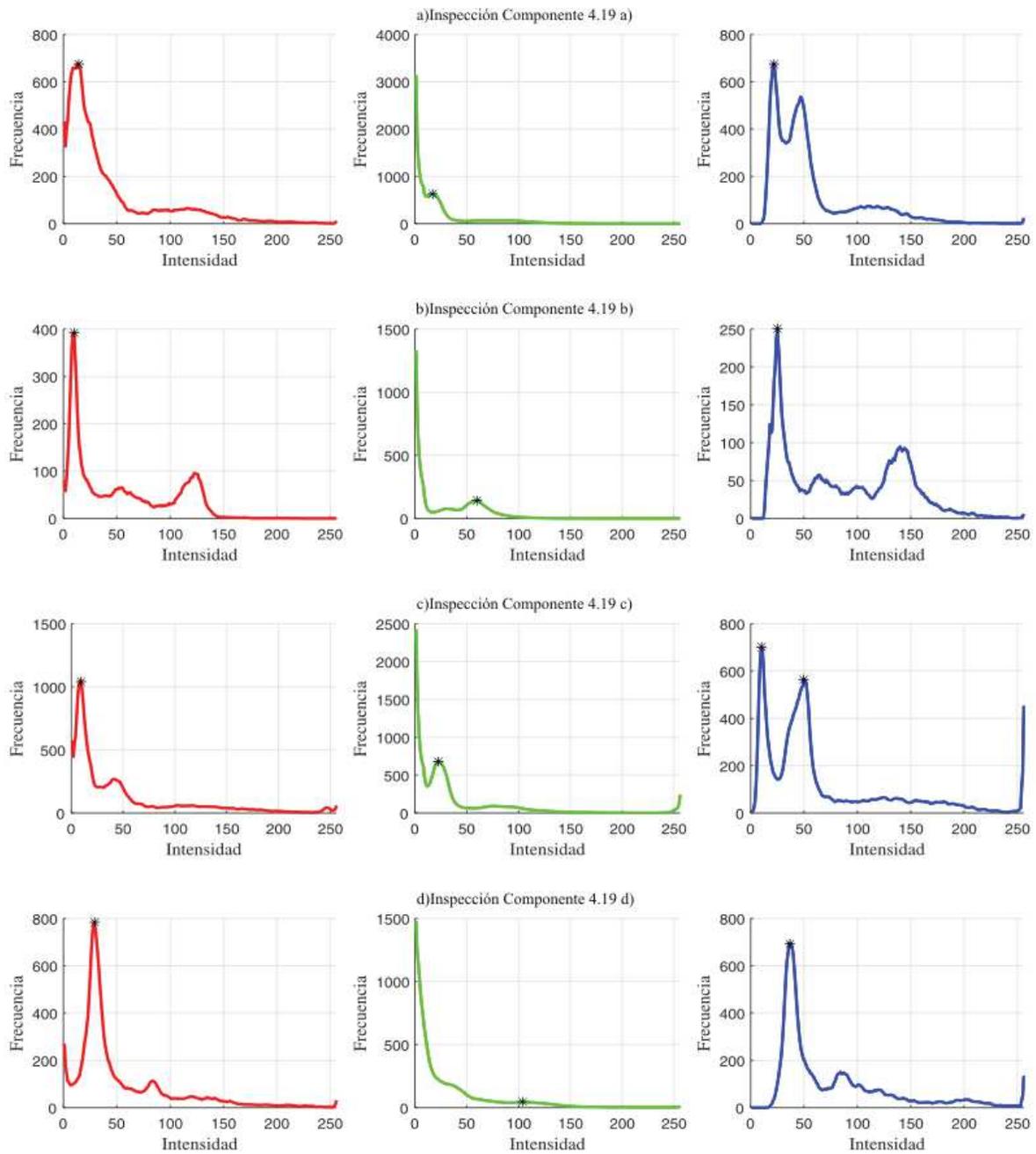


Figura 4.20 Histogramas de intensidad contra frecuencia de píxeles en la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos a), b), c) y d) de la Figura 4.14 (Fuente: elaboración propia).

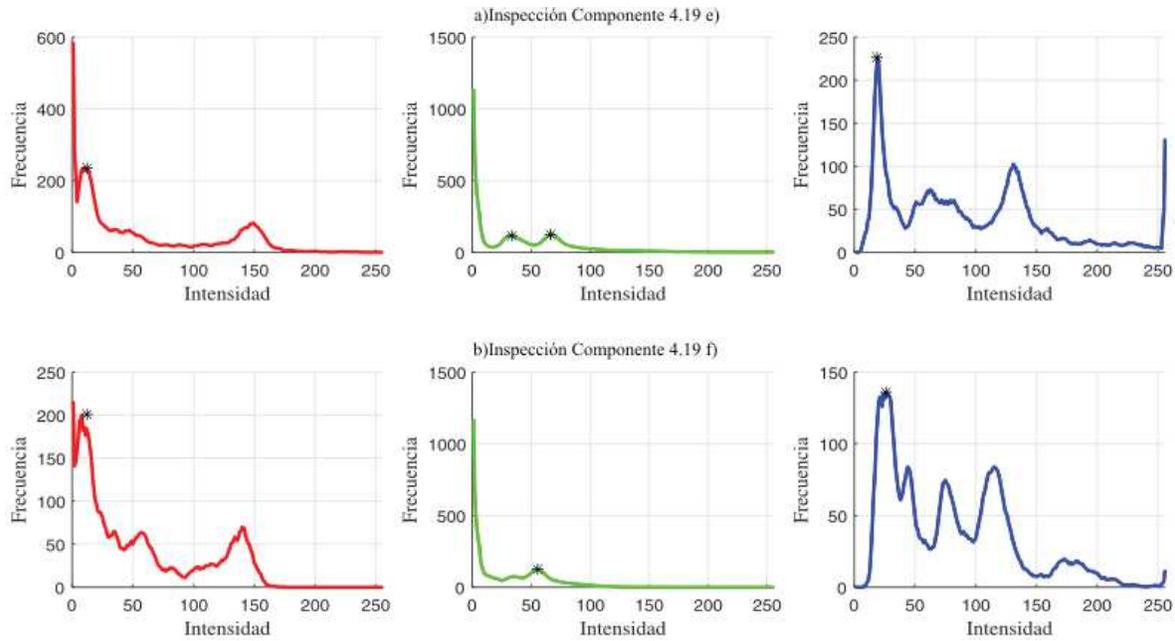


Figura 4.21 Histogramas de intensidad contra frecuencia de píxeles en la imagen real obtenida por el sistema de adquisición de imágenes con el anillo RGB encendido, para la determinación de la calidad de la soldadura de los componentes detectados en la PCB bajo inspección. Los resultados se muestran en grupos de tres gráficas para cada uno de los componentes indicados en los incisos e) y f) de la Figura 4.9 (Fuente: elaboración propia).

A lo largo de la sección 4.1 se presentaron 4 pruebas o experimentos para evaluar el desempeño del sistema de detección para propósitos de calibración y ajustes del código desarrollado en MATLAB y para la evaluación de la influencia de las condiciones de iluminación generales así como de la inclusión del sistema de iluminación RGB construido. Para cada una de las pruebas se presentaron las imágenes originales en el espacio RGB a inspeccionar con muestras de los valores presentes de los píxeles de interés en un formato $[X, Y], [R, G, B]$; siendo las coordenadas del píxel en cuestión y sus valores de intensidad para cada plano de color en cada una de las Figuras (4.1, 4.7 y 4.12) respectivamente. En las Figuras 4.2, 4.8 y 4.13 se presentan las etapas de procesamiento realizadas por el código para la detección de componentes y los resultados arrojados, para después desplegar cada uno de los componentes detectados por el sistema en forma de recortes de la imagen

original inspeccionada, mostrados en las Figuras 4.3, 4.9 y 4.14; para finalmente presentar grupos de 9, 6 y 6 gráficas para cada experimento, las cuales muestran los histogramas de intensidad de cada plano de color para cada una de los componentes SMT detectados en las imágenes.

La primera prueba se realizó para fines de calibración del sistema, mediante ajustes varios al código escrito originalmente. Se seleccionó deliberadamente la imagen de prueba de la Figura 4.1 con una PCB con dimensiones 850mm x 850mm. La imagen se considera adecuada para una primera instancia de prueba dado que no presenta variaciones en cuanto a las condiciones de iluminación y debido a la variedad de componentes presentes en la placa permitió un punto de partida adecuado para observar los alcances del sistema de detección respecto a la diferencia de dimensiones entre los componentes presentes en la placa.

Las Figuras 4.1 y 4.2 muestran la PCB de calibración que contiene 157 componentes con tecnología SMT, de las cuales el 100% de las mismas se encuentran clasificadas como “Soldaduras Aceptables”. El programa es capaz de detectar 51 componentes, de los cuales, dadas las dimensiones de la PCB, se pueden obtener 9 recortes de manera automática para la determinación de la calidad de los mismos, únicamente para empaquetados tipo QFP44 y QFP-100 y SOIC-14 con dimensiones superiores a 8 x 8 mm. De los recortes individuales obtenidos que se muestran en la Figura 14.3, el programa logra una detección del 32.48 % de los componentes; para el rubro del análisis de calidad y clasificación de uniones de soldadura lo que se observa en las gráficas de las Figuras 4.5, 4.6 y 4.7, es que los histogramas presentan un solo pico para cada plano de color que cumple con las condiciones propuestas inicialmente para clasificar las uniones de dichos componentes como aceptables en todos de los casos, y por tanto, logrando un 100% de efectividad en el análisis e inspección

La evaluación real del desempeño del sistema completo continuó con el análisis de una imagen obtenida *in situ* por el elemento de adquisición. La PCB para este segundo experimento contaba con dimensiones 48 x 37 mm; la cual cuenta con componentes con empaquetado 0805, 1210, SOT-23, etc.; cuyas dimensiones son inferiores a 4 mm, por lo cual se eligió un área efectiva para inspección de 14 x 21 mm, como se muestra en las

Figuras 4.7 y 4.8, donde de nueva cuenta todas las uniones de soldadura se encuentran preclasificadas como “Soldaduras Aceptables”. La sección de la PCB inspeccionada contiene 11 componentes de tecnología SMT, de los cuales el sistema es capaz de detectar automáticamente 6 de ellos, mostrados en la Figura 4.9. El rendimiento del sistema tiene un porcentaje de 54.54 % de efectividad en cuanto a la detección de componentes sobre la placa; sin embargo, cabe resaltar que los 5 componentes no detectados presentan un color blanco en el cuerpo del empaquetado, características que no se contemplan los objetivos y códigos propuestos inicialmente.

En las gráficas mostradas en las Figuras 4.10 y 4.11 se presentan los histogramas para el segundo experimento, se puede observar que estos presentan variaciones y curvas más irregulares, sobre todo en el plano B y en menor medida en el plano G; lo cual tiene como resultado que el sistema realice clasificaciones de “Soldadura Inaceptable” en uniones consideradas como aceptables como es el caso mostrado en las Figuras 4.10 b) y 4.10 e). El segundo experimento por tanto, logra un 66.66% de efectividad en el análisis e inspección.

El tercer experimento se realizó sobre la misma imagen y con la misma colocación y la misma sección de interés de la PCB que se indicaron para la segunda prueba, con la diferencia de que para este caso se activó el anillo RGB del sistema de iluminación, como se muestra en la Figura 4.12. Los efectos observados debido a la modificación de las condiciones de iluminación fueron el “realce” de tonalidades oscuras (negros), las cuales presentaron valores más cercanos a 0 en cada uno de los planos, lo que es coherente con el valor teórico esperado, y la presencia de valores más bajos para la tonalidad que prevalece en la PCB, en este caso asociada al plano G; se observó que este efecto depende de la elevación del anillo sobre la PCB, pero debido que este cambio de iluminación acentuó los tonos negros, el desempeño del código de detección de componentes no se ve afectado, sino que presenta un comportamiento favorable como se observa en la Figura 4.13 al presentar regiones binarias más uniformes y definidas a comparación de las observadas en el segundo experimento.

La inspección realizada sobre los 11 componentes de tecnología SMT en la sección de la PCB, nuevamente fue capaz de detectar automáticamente 6 de ellos, mostrados en la Figura 4.14, con un rendimiento del sistema del 54.54 % de efectividad nuevamente y con

la misma consideración de los componentes con empaquetado con tonalidades distintas a negro. En las gráficas mostradas en las Figuras 4.15 y 4.16 se presentan los histogramas del tercer experimento, se puede observar mejores condiciones de iluminación del plano B que permite su correcto análisis. El cambio de iluminación producido por el anillo afecta en cierta medida las curvas de los histogramas del plano R; sin embargo, el sistema es capaz de detectar los picos establecidos en la condiciones de clasificación para considerar a la unión como aceptable. El impacto en el plano G es mínimo en relación al experimento anterior.

Como se observa en las gráficas presentadas en las Figuras 4.15 y 4.16, todos los componentes detectados se clasifican “Soldadura Aceptable” según los parámetros establecidos, logrando un 100% de efectividad en el análisis e inspección de las uniones de soldadura, lo cual es coherente con la hipótesis establecida en cuanto a la utilidad del anillo RGB dentro del proceso de inspección óptica.

Finalmente, el cuarto experimento realizado analiza la misma sección de la PCB de los dos experimentos anteriores, con la diferencia que para este caso presenta dos componentes con defectos en las uniones de soldadura (exceso de soldadura) lo cual se considera como “Soldadura Inaceptable”. El objetivo de esta última prueba fue el probar la eficiencia del sistema y la respuesta del código ante la presencia de uniones de soldadura que contengan algún defecto.

La Figura 4.17 y 4.18 muestra los mismos efectos observados debido a la modificación de las condiciones de iluminación, como lo son el “realce” de tonalidades oscuras y el comportamiento sigue siendo favorable en cuanto a la determinación de regiones binarias para la detección de componentes.

La inspección, de nueva cuenta, realizada sobre los 11 componentes de tecnología SMT, nuevamente fue capaz de detectar automáticamente 6 de ellos, mostrados en la Figura 4.19, en la cual se observan dos componentes con uniones de soldadura consideradas inaceptables. La prueba arroja un rendimiento del sistema del 54.54 % de efectividad nuevamente y con la misma consideración de los componentes con empaquetado con tonalidades distintas a negro.

Las gráficas presentadas en las Figuras 4.20 c) y 4.21e) muestran los histogramas de los componentes individuales obtenidos en las Figuras 4.19 c) y 4.19 e), los cuales corresponden a las dos uniones de soldadura defectuosas presentes en la PCB. Como se observa, el comportamiento gráfico presenta más de un pico para los planos de color G y B respectivamente, y por tanto el sistema las clasifica como “Soldadura Inaceptable” según los parámetros establecidos; siendo coherente con la preclasificación; y nuevamente logrando un 100% de efectividad en el análisis e inspección de las uniones de soldadura variadas, tanto aceptables como inaceptables.

En la Tabla 4.2 se muestran los resultados obtenidos en los cuatro experimentos realizados. La información mostrada en la segunda columna contiene el número total de componentes presentes en la PCB bajo inspección, la tercera columna contiene el número de componentes detectados por el sistema, mientras que la cuarta indica el número de componentes individuales que el programa puede aislar forma de recortes individuales. La columna “Análisis Correcto” corresponde al número de uniones de soldadura analizadas y clasificadas correctamente del total de componentes aislados según los parámetros establecidos inicialmente. Finalmente, las dos últimas columnas muestran de manera porcentual la efectividad del código propuesto separada en dos rubros, efectividad detección de componentes SMT sobre la imagen, y efectividad de inspección y análisis de la unión de soldadura de cada recorte individual.

Tabla 4.2 Resultados, desempeño y tasas de efectividad de los tres experimentos para prueba del desempeño del sistema de inspección óptica automática.

Tipo	Número de Componentes	Componentes Detectados	Recortes Generados	Análisis Correcto	Efectividad Detección	Efectividad Análisis
Experimento 1 Calibración	157	51	9	9	32.48%	100%
Experimento 2 Sin Anillo RGB	11	6	6	4	54.54%	66.66%
Experimento 3 Con Anillo RGB	11	6	6	6	54.54%	100%
Experimento 4 Con Anillo RGB	11	6	6	6	54.54%	100%

4.2 PRUEBAS AL SISTEMA ELECTRÓNICO DE ALIMENTACIÓN

Dentro de la etapa de pruebas al sistema electrónico de alimentación, los procedimientos se enfocan en determinar el funcionamiento de los elementos que componen el sistema y de las señales de control y potencia que intervienen dentro de éste.

Según los cálculos realizados en la sección 3.2.4, se estableció y programó que la señal PWM producida por el PIC cuenta con una amplitud de 5V suministrados desde la fuente de poder ATX “Alimentación Vcc1”, y con un periodo de 1 ms, por tanto teniendo una frecuencia de 1 kHz; siendo esta la señal de activación del circuito de conmutación inferior del IR2110. Este driver produce a su salida una señal de salida con niveles de voltaje de 12V, los cuales son suministrados también por la fuente de poder ATX “Alimentación Vcc2”; en la Figura 4.22 a) se muestra la medición del periodo y de la amplitud pico a pico de la señal generada por el driver.

Según especificaciones del fabricante, el MOSFET IRF640, es capaz de manejar una corriente máxima $I_D = 18A @ V_{GS} = 10V$. En la Figura 4.22 se observa que el periodo de la señal coincide con los parámetros programados para la señal de control; se observa también que las señales mostradas en las Figuras 4.17 a) y 4.17 b), presentan formas de onda inversas una de otra, esto debido al comportamiento de los estados de corte y saturación presentes en el pin “*Drain*” con relación al pin de control “*Gate*” del MOSFET (Boylestad et al., 2003). De esta manera se realiza la comprobación del correcto funcionamiento del circuito de potencia.

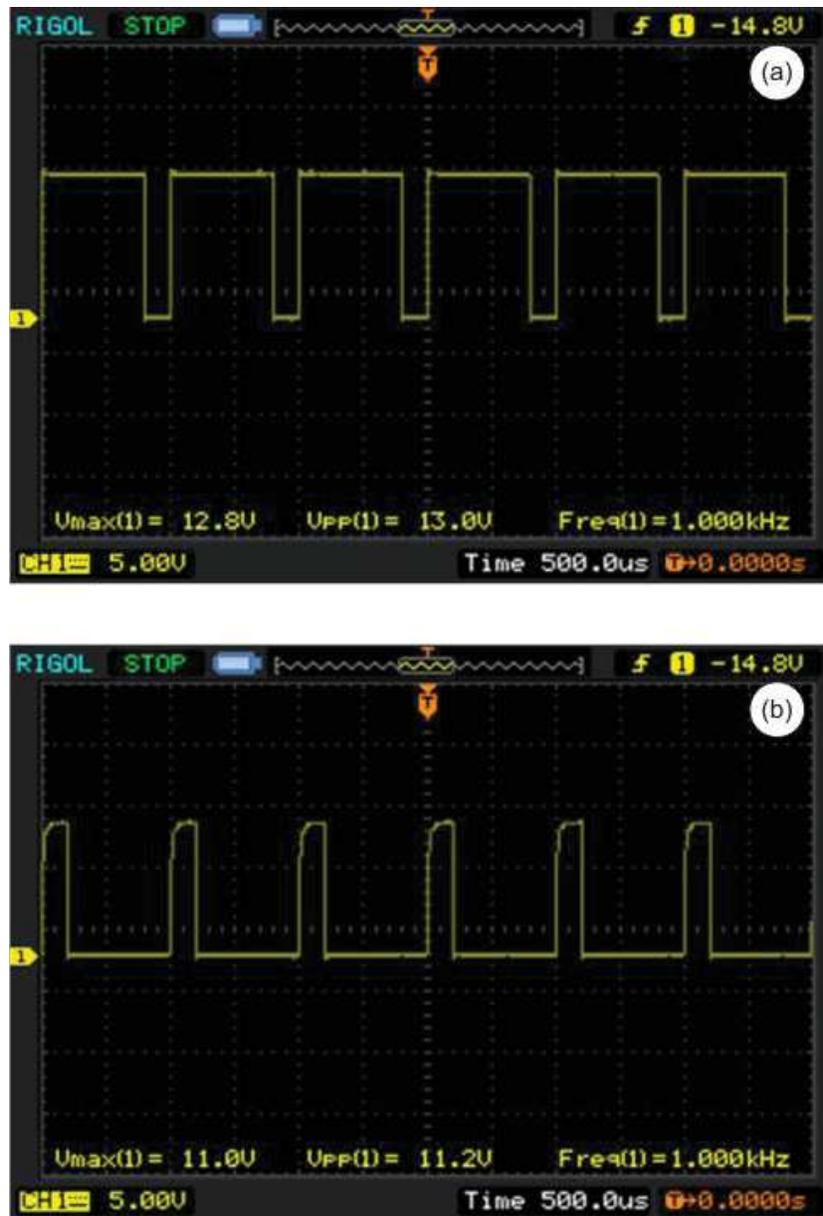


Figura 4.22 Señales de funcionamiento del sistema electrónico de potencia con ciclo de trabajo $D=80\%$. a) Señal de control PWM, forma de onda de la tensión eléctrica medida en la salida de conmutación inferior del driver IR2110. b) Señal de potencia PWM, forma de onda de la tensión eléctrica medida de Drain a Source en el MOSFET IRF640.

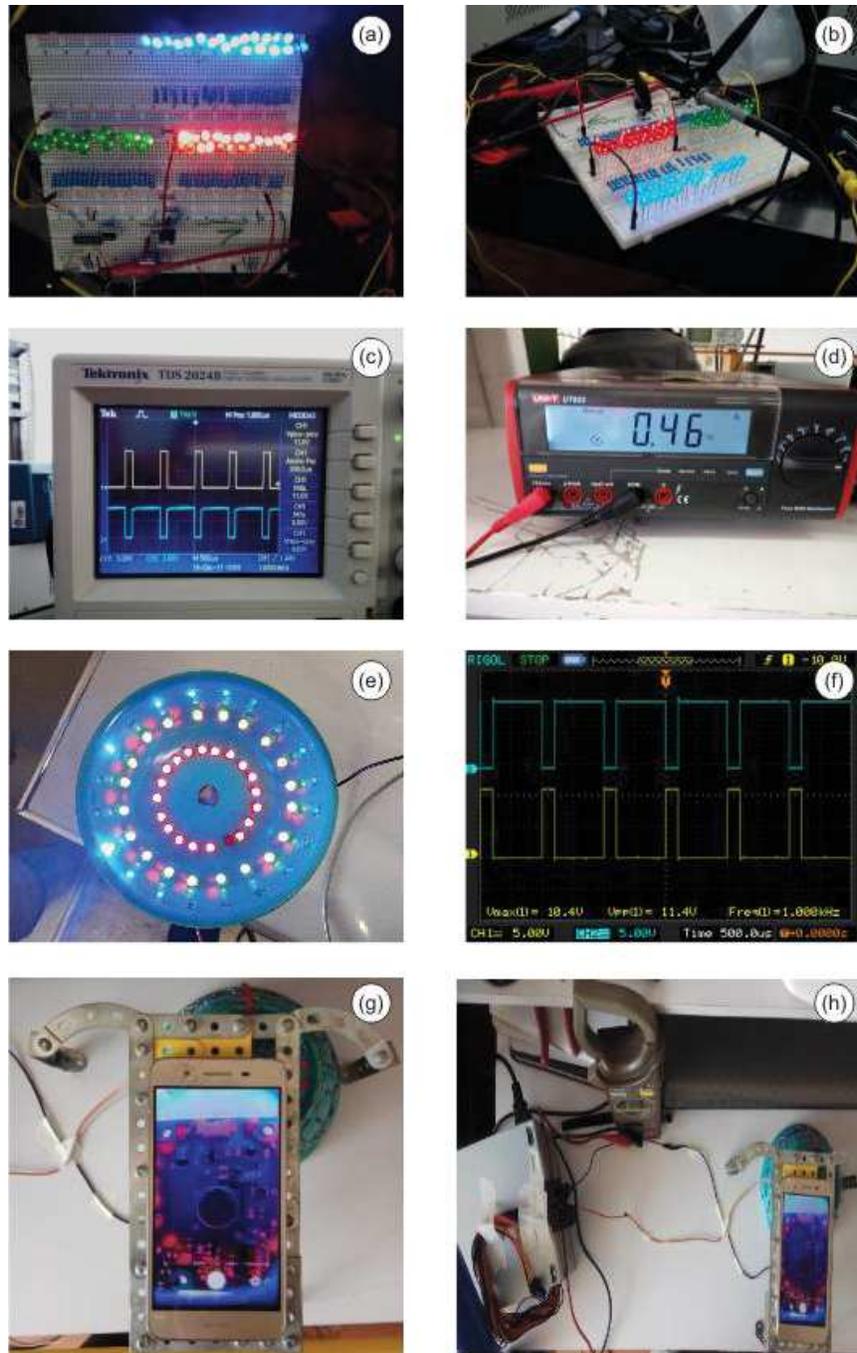


Figura 4.23 Pruebas prototipo al sistema electrónico de alimentación. a) Armado de etapa de potencia en protoboard. b) Observaciones al sistema de iluminación mediante variación del ciclo de trabajo de la señal de potencia. c) Señales observadas en osciloscopio. d) Mediciones de consumo de corriente del anillo de iluminación. e) Montaje final de LED's sobre estructura de soporte. f) Formas de onda obtenidas. g) y h) Montaje de prueba para observación del funcionamiento de anillo RGB y obtención de imágenes de prueba.

En la Figura 4.23 se muestra el proceso de construcción y prueba del sistema electrónico de alimentación. Las Figuras 4.23 a) y 4.23 b) muestran el funcionamiento del circuito de conmutación inferior que alimenta y regula el anillo RGB, el cual es activado por la señal de control PWM producida por el microcontrolador PIC18F4550. La Figura 4.23 e) muestra el montaje de los LED's sobre la estructura de soporte seleccionada para fabricar el anillo RGB propuesto. Las Figuras 4.23 g) y 4.23 h) muestran las primeras pruebas realizadas para comprobar la correcta construcción y conexión del anillo y las condiciones de iluminación que produce el mismo para la generación de imágenes de prueba.

4.3 DESARROLLO FUTURO

El sistema de inspección óptica automática pretende ser integrado a un sistema robótico para soldar circuitos integrados, por lo que como alternativa de desarrollo posterior, sería la modificación de las dimensiones de dichos materiales para integrarlos de manera conjunta en el portaherramientas antes mencionado, permitiendo total autonomía al sistema robótico, permitiendo que éste pueda soldar, inspeccionar y corregir los defectos de soldadura en una sola etapa.

De igual manera, como desarrollo futuro se contempla la caracterización de los valores de pixel e intensidades asociados a componentes con color de empaquetado distinto a negro, que permita modificar e incluir instrucciones en el código para su detección, y de esta forma incrementar la robustez del sistema de detección y viabilidad para su implementación a nivel comercial.

5. CONCLUSIONES

Una vez finalizados los cuatro experimentos y presentados los resultados en el capítulo anterior, se concluye que el sistema control óptico construido es capaz de producir patrones de iluminación y reflexión viables, los cuales le permiten aplicar los códigos propuestos para realizar el procesamiento de imágenes y la detección de bordes de los elementos que componen una PCB y, con base a los datos finales obtenidos el sistema es capaz realizar de forma automática un proceso de inspección de la calidad de soldadura de dispositivos electrónicos y por tanto puede ser considerado como una trayectoria de lazo cerrado dentro de un sistema de control de una máquina CNC para soldar circuitos integrados; marcando así la consecución del objetivo general establecido inicialmente y con base en los resultados obtenidos, comprobando la validez de la hipótesis planteada.

Los sistemas de inspección óptica automática (AOI) se presentan como un área de oportunidad para realizar la evaluación de la calidad de las uniones de soldadura en un circuito electrónico. La flexibilidad en cuanto a las opciones para seleccionar el elemento para adquirir las imágenes, hace del sistema propuesto una opción viable para realizar la tarea de inspección. El presente trabajo de tesis utilizó como elementos de adquisición de imágenes dos tipos distintos de cámara, la primera de un teléfono celular y la segunda fue una cámara tipo microscopio. La primera se usó por motivos de prueba y como muestra de que el código ofrece la facilidad de adquirir imágenes desde varios dispositivos, que representa una ventaja dado que puede ser viable el uso de los algoritmos de inspección propuestos, aún sin usar la totalidad de los elementos de la máquina completa; y la segunda, debido a la ergonomía de la misma respecto a las características de la etapa de iluminación.

El desarrollo, diseño y construcción del presente trabajo se basó en la comprensión y aplicación de fenómenos ópticos sobre superficies especulares, como lo es una unión de soldadura, así como el entendimiento de cómo, mediante la implementación de elementos electrónicos de iluminación y potencia, es posible generar y adquirir imágenes adecuadas para ser procesadas mediante herramientas de software para determinar la calidad de una unión de soldadura y por extensión de una placa de circuito impreso PCB. El sistema de inspección propuesto es capaz de determinar la calidad de una unión de soldadura basada en el procesamiento mediante software de los patrones de reflexión producido en la misma mediante una fuente de iluminación RGB. El análisis de los datos obtenidos respecto a las uniones de soldadura inspeccionadas indican que el sistema puede considerarse como una opción viable de bajo costo y fácil implementación en un proceso de manufactura de PCB; por lo tanto, las ideas establecidas en la hipótesis se consideran válidas.

Los datos obtenidos sugieren la viabilidad del sistema para ser utilizado a nivel comercial debido a que presenta la versatilidad de poder emplearse tanto en inspecciones en línea y fuera de línea durante un proceso de manufactura. Para el primer caso, como ya fue mencionado, el proyecto forma parte de la patente MX/a/2014/007507; por lo que implementado dentro de dicha máquina se pretende realizar de manera completa el proceso de manufactura para soldar e inspeccionar PCB's y dado el valor agregado que ofrece el sistema completo, la tecnología propuesta tendría oportunidad de comercialización como parte integral del sistema robótico. Para la inspección fuera de línea, dado que el código puede analizar imágenes provenientes de distintos elementos de adquisición o analizar un conjunto de imágenes/archivos obtenidos previamente, es viable emplearlo en un proceso comercial de soldadura donde sea necesario verificar la calidad de placas previamente manufacturadas, como etapa previa a la salida de las placas para venta o uso específico.

La etapa de instrumentación electrónica usada para el cumplimiento de los objetivos particulares establecidos se llevó a cabo mediante un microcontrolador. El PIC18F4550 se encargó en primera instancia de comunicar el sistema electrónico de alimentación con la computadora a través de una interfaz USB, lo que permitió transmitir las señales tanto de encendido como apagado del anillo de iluminación RGB, así como de las señales de control

y la modulación por pulsos para manipular el funcionamiento de la circuito de potencia y por tanto, de la intensidad bajo la cual operaba el anillo de iluminación.

La calibración del sistema, en lo que se refiera al programa diseñado para el análisis de las uniones de soldadura y para probar el funcionamiento de la etapa de adquisición de imágenes, se realizó mediante el estudio de imágenes obtenidas de dos elementos de adquisición diferentes como se indica en el primer párrafo del presente capítulo. Esto se hizo, en primera instancia para realizar los ajustes en el código generado para la inspección; y en segunda instancia para comprobar que el funcionamiento del elemento de adquisición propuesto, era adecuado y viable para ser integrado al sistema final. Para determinar la calidad de las uniones de soldadura, se generaron gráficas bidimensionales mediante MATLAB, las cuales que son histogramas de intensidad contra frecuencia de la imagen de cada componente inspeccionado en cada una de las porciones del espectro visible rojo, verde y azul (RGB) con el cual se iluminaba la PCB bajo análisis; donde el comportamiento de cada histograma determinó, de cumplirse para los tres casos, que la soldadura era aceptable.

El sistema de inspección óptica automática propuesto y desarrollado prueba su utilidad mediante la interpretación de la información proporcionada por las gráficas de intensidad que el sistema es capaz de generar. Los resultados obtenidos en función del comportamiento de cada histograma, permitieron clasificar los inspeccionados como una soldadura aceptable o no aceptable. Los experimentos o las pruebas usadas para el desarrollo del programa de inspección se realizó a través de uniones de soldadura preclasificadas como aceptable y no aceptables, que permitieron crear un histograma de control para cada caso y mediante el comportamiento de los niveles de intensidad RGB en cada uno de ellos, para después realizar la calibración de los distintos valores de umbral que permiten al programa llevar a cabo la inspección.

El procedimiento de inspección implementado muestra que puede ser utilizado como etapa de control de calidad al integrarse en una máquina o proceso de fabricación de PCB. Como limitación, el sistema no puede determinar en primera instancia los componentes cuyo tamaño en la imagen bajo inspección es menor a 35x35 píxeles; para realizar su clasificación es necesaria una segunda toma, realizando un acercamiento y obteniendo otra

imagen mediante el microscopio o el elemento de adquisición seleccionado, y de esta forma es posible analizar los componentes restantes. Los datos obtenidos muestran que el sistema es una alternativa viable que puede sustituir la demandante y desgastante inspección visual, que representa el método más común para manufactura de PCB a pequeña escala; así como de métodos por pruebas eléctricas, térmicas, acústicas o mediante rayos X usadas en otras de escalas y presupuestos de producción.

BIBLIOGRAFÍA

- BOYLESTAD, R., NASHELSKY, L. & BARRAZA, C. 2003. *Electrónica: teoría de circuitos y dispositivos electrónicos*, Prentice-Hall Hispanoamericana.
- CANNY, F. J. 1986. *A computational approach to edge detection*, IEEE Trans. Pattern Annuall Machine Intell, 8(6):679-698.
- CHAPMAN, S. J. 2007. *MATLAB Programming for Engineers*, Thomson.
- CUEVAS, E., ZALDÍVAR, D. & PÉREZ, M. 2014. *Procesamiento digital de imágenes usando MatLAB y Simulink*, Mexico, Alfaomega.
- EDINBAROUGH, I., BALDERAS, R. & BOSE, S. 2005. A vision and robot based on-line inspection monitoring sytem for electronic manufacturing. *Computers in Industry*, 56.
- HAO, W., XIANMIN, Z., YONGCONG, K., GAOFEI, O. & HONGWEI, X. 2013. Solder joint inspection based on neural network combined with genetic algorithm. *Optik - International Journal for Light and Electron Optics*, 124, 4110-4116.
- HEINRICH, S., SCHAEFER, M. & SCHROEDER, S. 1996. Prediction of Solder Joint Geometries in Array-Type Interconnects. *Journal of Electronic Packaging*, 118.
- HO, W. & JI, P. 2007. *Optimal production planning for PCB assembly*, Springer.
- IPC 2005. Generic requirements for surface mount design and land pattern standard IPC-7351. 92.

- KIM, J. & CHO, H. 1995. Neural network-based inspection of solder joints using circular illumination. *Image and Vision Computing*, 13.
- LEE, Z.-S. & LO, R.-C. 2002. Application of vision image cooperated with multi-light sources to recognition of solder joints of PCB. *IEEE Transactions on InTAAI artificial intelligence and applications*, 425-30.
- LOH, H. & LU, M. 1999. Printed circuit board inspection using image analysis. *IEEE Transactions on Industry Applications*, 35.
- MAR, N., YARLAGADDA, P. & FOOKES, C. 2011. Design and development of automatic visual inspection system for PCB manufacturing. *Journal of Robotic and Computer-Integrated Manufacturing*, 27, 962.
- MICROCHIP 2009. PIC18F2455/2550/4455/4550. 28/40/44-Pin, High-Performance, Enhanced Flash,
USB Microcontrollers with nanoWatt Technology. Microchip Technology Inc.
- SCHINCARIOL, R. A. & SCHWARTZ, F. W. 1990. An experimental investigation of variable density flow and mixing in homogeneous and heterogeneous media. *Water Resour. Res.*, 26, 2317–2329.
- TUMMALA, R. 2000. *Fundamental of Microsystems Packaging*, McGraw Hill Professional.
- YANG, J. 2008. Quality Inspection and Reliability Study of Solder Bumps in Packaged Electronic Devices: Using Laser Ultrasound and Finite Element Methods. *Electronic Components and Technology Conference Georgia Institute of Technology*, 74.
- YANG, J. & UME, I. 2008. Thermomechanical reliability study of flip-chip solder bumps: Using laser ultrasound technique and finite element method. *Electronic Components and Technology Conference*, 622.

APÉNDICES

A. PROGRAMAS INFORMÁTICOS

A.1. EN EL MICROCONTROLADOR (PIC 18F4550)

```
1. Principal_PruebaUSART.c
2. # include <p18f4550.h>
3. # include <usart.h>
4. # include <sw_uart.h>
5. # include <delays.h>
6. # include <string.h>
7. #include <delays.h>
8. #include <pwm.h>
9.
10. #pragma romdata CONFIG
11.     rom const unsigned char CONFIG1L = 0xFF;
12.     rom const unsigned char CONFIG1H = 0xF2;
13.     rom const unsigned char CONFIG2L = 0xFC;
14.     rom const unsigned char CONFIG2H = 0x18;
15.     rom const unsigned char CONFIG3L = 0xFF;
16.     rom const unsigned char CONFIG3H = 0xFF;
17.     rom const unsigned char CONFIG4L = 0x0B;
18.     rom const unsigned char CONFIG4H = 0xFF;
19.     rom const unsigned char CONFIG5L = 0xFF;
20.     rom const unsigned char CONFIG5H = 0xFF;
21.     rom const unsigned char CONFIG6L = 0xFF;
22.     rom const unsigned char CONFIG6H = 0xFF;
23.     rom const unsigned char CONFIG7L = 0xFF;
24.     rom const unsigned char CONFIG7H = 0xFF;
25. #pragma romdata
26.
27.
28. char ReadUSART( void );
29. void PWMOn(void);
30. void VectorInterrupcion(void);
31. char dato;
32.
33. #define      PWM          LATCbits.LATC2
34. #define      TRIS_PWM    TRISCbits.TRISC2
35.
36. //Función de Interrupción para activar PWM mediante la RX USART
37. #pragma interrupt PWM_On
38.     void PWM_On(void)
39.     {
40.         if(PIR1bits.RCIF==1)//Revisa la bandera de interrupción de RX USART
41.         {
42.
```

```
43.         TRIS_PWM = 0;           //Salida RC2 Señal PWM
44.
45.         dato = ReadUSART();
46.         while(BusyUSART()); //Espera hasta que el módulo se desocupe
47.
48.         switch(dato)
49.         {
50.             case '1':
51.                 putsUSART( "\n\r Ciclo de Trabajo: 10% " );
52.                 CCPR1L = 12;
53.                 break;
54.
55.             case '2':
56.                 putsUSART( "\n\r Ciclo de Trabajo: 20% " );
57.                 CCPR1L = 25;
58.                 break;
59.
60.             case '3':
61.                 putsUSART( "\n\r Ciclo de Trabajo: 30% " );
62.                 CCPR1L = 37;
63.                 break;
64.
65.             case '4':
66.                 putsUSART( "\n\r Ciclo de Trabajo: 40% " );
67.                 CCPR1L = 50;
68.                 break;
69.
70.             case '5':
71.                 putsUSART( "\n\r Ciclo de Trabajo: 50% " );
72.                 CCPR1L = 62;
73.                 break;
74.
75.             case '6':
76.                 putsUSART( "\n\r Ciclo de Trabajo: 60% " );
77.                 CCPR1L = 74;
78.                 break;
79.
80.             case '7':
81.                 putsUSART( "\n\r Ciclo de Trabajo: 70% " );
82.                 CCPR1L = 87;
83.                 break;
84.
85.             case '8':
86.                 putsUSART( "\n\r Ciclo de Trabajo: 80% " );
87.                 CCPR1L = 99;
88.                 break;
89.
90.             case '9':
91.                 putsUSART( "\n\r Ciclo de Trabajo: 90% " );
92.                 CCPR1L = 112;
93.                 break;
94.
95.             case 'X':
96.                 putsUSART( "\n\r Ciclo de Trabajo: 100% " );
97.                 CCPR1L = 124;
98.                 break;
99.         }
100.
101.         PIR1bits.RCIF=0;       //Limpia la bandera de interrupción
102.         Delay10KTCYx(25);
103.     }
104.
105. }
```

```

106.
107. //Vector de interrupción de alta prioridad
108. #pragma code Interrupcion=0x0008
109. void VectorInterrupcion (void)
110. {
111.     _asm GOTO PWM_On _endasm
112. }
113.
114. #pragma code
115.
116.
117. void main(void)
118. {
119.     INTCONbits.GIE=0;
120.     OSCCON =0x7A; //Configuración Oscilador 8 MHz
121.     TRISCbits.TRISC6 = 0; // set TX (RC6) as output pin 25
122.     TRISCbits.TRISC7 = 1; // and RX (RC7) as input pin 26
123.     INTCON=0xD0; //Configuración Registro INTCON=0b11010000
124.     T2CON = 0b00000111 ;//Postscale 1:1; Timer2 ON; Prescaler=16
125.     TMR2 = 0; //Inicializa en 0 TMR2
126.     CCP1CON= 0b00101100; //Modo PWM Salida Simple
127.     PR2 = 124; //Ciclo de Trabajo 100%
128.     RCONbits.IPEN=0; // Deshabilita Prioridades.-
129.     INTCONbits.PEIE=1; // Habilita Interrupción de periféricos
130.     INTCONbits.GIE=1; // Habilita Interrupción Global.
131.
132.
133.     OpenUSART(USART_TX_INT_OFF & USART_RX_INT_ON & USART_ASYNC_MODE &
134. USART_EIGHT_BIT & USART_CONT_RX & USART_BRGH_HIGH, 51);
135. //Interrupciones TX/RX(habilitada)
136. //Modo Asíncrono, Longitud 8 bits, Modo de recepción continuo.
137. //Baud rate 9600->SPBRG=51 @ Fosc= 8MHz
138.
139. //putsUSART( "\n\r Prueba Control Anillo RGB > " );
140.
141. while (1)
142. {
143.     while (!DataRdyUSART());
144.     putsUSART( "\n\r Ingrese Ciclo de trabajo: " );
145. }
146. }

```

A.2. EN LA PC

```

1. Código_Final.m
2. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3. %%Función de MATLAB para detectar los pixeles negros en una imagen RGB
4. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5.
6. Im = imread('C:\Users\Raúl\Documents\Tesis\Imágenes
   Tesis\Resultados\rect_expzA2.jpg');
7. Im_mod = Im;
8. %%%Despliegue de imagen original%%%
9. figure; %Creación del objeto gráfico
10. subplot(2, 2, 1);
11. imshow(Im);
12. %% Se obtienen los valores de las dimensiones de la imagen
13. [m,n,bandas] = size(Im);
14.
15. %%%Despliegue de imagen original con identificación de pixeles negros%%
16. subplot(2, 2, 2);
17. imshow(Im)
18. hold on %Objeto gráfico fijo para efecto de comandos gráficos tengan
   adicionales
19.
20. %Extracción bandas individuales de color para el análisis del color de cada
   pixel
21. %en la imagen original bajo inspección
22. redChannel = Im(:, :, 1);
23. greenChannel = Im(:, :, 2);
24. blueChannel = Im(:, :, 3);
25. %Valores de umbral para cada banda/plano de color usados para la
26. %comparación y determinación de los pixeles negros detectados
27. umbral = 30;
28. umbral2 = 10;
29. umbral3 = 30;
30.
31. %Ciclo de inspección para determinación de pixeles negros
32. for r=1:m
33.     for c=1:n
34.         if (redChannel(r, c) <= umbral) & (greenChannel(r, c) <=
           umbral2) & (blueChannel(r, c) <= umbral3)
35.             sumaNegro = redChannel(r, c)+greenChannel(r, c)+ blueChannel(r,
           c);
36.             plot(c,r,'+')%Marca de indicación de pixel negro detectado
           sobre Im_mod
37.             Im_mod(r, c, 1) = 255;
38.             Im_mod(r, c, 2) = 255;
39.             Im_mod(r, c, 3) = 255;
40.         else %Valor cero para tonalidades de pixel distintas a negro
41.             Im_mod(r, c, 1) = 0;
42.             Im_mod(r, c, 2) = 0;
43.             Im_mod(r, c, 3) = 0;
44.         end
45.     end
46. end
47. hold off %Liberación de objeto gráfico
48.
49. %%%Despliegue de imagen binarizada para análisis de componentes obtenidos
50. subplot(2, 2, 3);
51. imshow(Im_mod);
52.
53. %%%Despliegue de imagen binarizada y proceso de filtración de áreas
54. %%%detectadas detectadas como componentes de la PCB

```

```

55. Im_bk = im2bw(Im_mod,0.5);
56. BW2 = bwareafilt(Im_bk,10);
57. subplot(2, 2, 4);
58. imshow(Im)
59. s = regionprops(BW2, 'centroid'); %Propiedades de regiones. Centros
60. centroids = cat(1, s.Centroid); %Concatenación de Arreglo
61. s1 = regionprops(BW2, 'area'); %Propiedades de regiones. Área
62. areas = cat(1, s1.Area); %Concatenación de Arreglo
63. hold on
64. plot(centroids(:,1),centroids(:,2), 'r*')
65. hold off
66.
67. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68. %%Función de MATLAB para realizar recortes de los componentes
69. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70.
71. [a b c] = size(centroids); %%Obtiene el número de centros detectados
72. folder = 'C:\Users\Raúl\Documents\Tesis\Imágenes Tesis\Recortes_RGB1\';
73.
74. for i=1:a
75.     if (areas(i)>=7000 && areas(i)<=7500) %Componente Tipo QFP100
76.         pX = 106; %Píxeles ancho x
77.         pY = 139; %Píxeles largo y
78.         oX = 50; %Origen x
79.         oY = 70; %Origen y
80.         iC= imcrop(Im,[centroids(i)-oX centroids(i,2)-oY pX pY]);
81.         nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
82.         imwrite(iC,nameIm)
83.         figure
84.         imshow(iC)
85.     end
86.     if (areas(i)>=3200 && areas(i)<=3300) %Componente Tipo QFP-64
87.         pX = 102; %Píxeles ancho x
88.         pY = 105; %Píxeles largo y
89.         oX = 52; %Origen x
90.         oY = 55; %Origen y
91.         iC= imcrop(Im,[centroids(i)-oX centroids(i,2)-oY pX pY]);
92.         nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
93.         imwrite(iC,nameIm)
94.         figure
95.         imshow(iC)
96.     end
97.     if (areas(i)>=2500 && areas(i)<=2600) %Componente Tipo QFP-44
98.         pX = 82; %Píxeles ancho x
99.         pY = 82; %Píxeles largo y
100.         oX = 40; %Origen x
101.         oY = 40; %Origen y
102.         iC= imcrop(Im,[centroids(i)-oX centroids(i,2)-oY pX pY]);
103.         nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
104.         imwrite(iC,nameIm)
105.         figure
106.         imshow(iC)
107.     end
108.     if (areas(i)>=800 && areas(i)<=900) %Componente Tipo SOIC-14
109.         pX = 40; %Píxeles ancho x
110.         pY = 60; %Píxeles largo y
111.         oX = 23; %Origen x
112.         oY = 28; %Origen y
113.         iC= imcrop(Im,[centroids(i)-oX centroids(i,2)-oY pX pY]);
114.         nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
115.         imwrite(iC,nameIm)
116.         figure
117.         imshow(iC)

```

```

118.         end
119.         if (areas(i))>=495 && areas(i)<=515) %Componente Tipo SOD-123
120.             pX = 55; %Pixeles ancho x
121.             pY = 37; %Pixeles largo y
122.             oX = 30; %Origen x
123.             oY = 18; %Origen y
124.             iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
125.             nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
126.             imwrite(iC,nameIm)
127.             figure
128.             imshow(iC)
129.         end
130.         if (areas(i))>=470 && areas(i)<=494) %Componente Tipo SOIC-8
131.             pX = 40; %Pixeles ancho x
132.             pY = 35; %Pixeles largo y
133.             oX = 18; %Origen x
134.             oY = 16; %Origen y
135.             iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
136.             nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
137.             imwrite(iC,nameIm)
138.             figure
139.             imshow(iC)
140.         end
141.         if (areas(i))>=4000 && areas(i)<=4700) || (areas(i))>=3400 &&
areas(i)<=3500) %Componente Tipo 0805 (Vertical)
142.             pX = 75; %Pixeles ancho x
143.             pY = 130; %Pixeles largo y
144.             oX = 39; %Origen x
145.             oY = 69; %Origen y
146.             iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
147.             nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
148.             imwrite(iC,nameIm)
149.             figure
150.             imshow(iC)
151.         end
152.         if (areas(i))>=4900 && areas(i)<=5200) || (areas(i))>=3100 &&
areas(i)<=3200) || (areas(i))>=2200 && areas(i)<=2400) %Componente Tipo
SOIC-32 y Tipo 0805 (Horizontal)
153.             pX = 123; %Pixeles ancho x
154.             pY = 61; %Pixeles largo y
155.             oX = 60; %Origen x
156.             oY = 29; %Origen y
157.             iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
158.             nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
159.             imwrite(iC,nameIm)
160.             figure
161.             imshow(iC)
162.         end
163.         if (areas(i))>=7500 && areas(i)<=8000) || (areas(i))>=8100 &&
areas(i)<=8250) %Componente Tipo 1210 (Horizontal)
164.             pX = 215; %Pixeles ancho x
165.             pY = 105; %Pixeles largo y
166.             oX = 119; %Origen x
167.             oY = 65; %Origen y
168.             iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
169.             nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
170.             imwrite(iC,nameIm)
171.             figure
172.             imshow(iC)
173.         end
174.         if (areas(i))>=13000 && areas(i)<=14000) || (areas(i))>=7900 &&
areas(i)<=8100) %Componente Tipo SOT-23 Horizontal (Mayo05) %%%%%%%%%%%
175.             pX = 157; %Pixeles ancho x

```

```

176.         pY = 165; %Píxeles largo y
177.         oX = 70; %Origen x
178.         oY = 87; %Origen y
179.         iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
180.         nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
181.         imwrite(iC,nameIm)
182.         figure
183.         imshow(iC)
184.     end
185.     if (areas(i)>=10000 && areas(i)<=11000) %Componente Tipo SOT-23
        Vertical (Mayo05) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
186.         pX = 210; %Píxeles ancho x
187.         pY = 150; %Píxeles largo y
188.         oX = 117; %Origen x
189.         oY = 88; %Origen y
190.         iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
191.         nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
192.         imwrite(iC,nameIm)
193.         figure
194.         imshow(iC)
195.     elseif (areas(i)>=6400 && areas(i)<=6700)
196.         pX = 170; %Píxeles ancho x
197.         pY = 155; %Píxeles largo y
198.         oX = 84; %Origen x
199.         oY = 78; %Origen y
200.         iC= imcrop(Im,[centroids(i)-oX   centroids(i,2)-oY pX pY]);
201.         nameIm = [folder 'RecortePCB_' num2str(i) '.jpg'];
202.         imwrite(iC,nameIm)
203.         figure
204.         imshow(iC)
205.     end
206. end
207.
208.     %%Función para graficación de histogramas de los planos de
    RGB(concatenada)
209.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
210.     I=imread('C:\Users\Raúl\Documents\Tesis\Imágenes
    Tesis\Recortes_RGB1\RecortePCB_9.jpg');
211.     %Se verifica que la imagen I sea RGB (que contenga los tres planos)
212.     if(size(I, 3) ~=3)
213.         error('La imagen debe ser RGB')
214.     end
215.     close all
216.     %Se establecen 256 valores que representan la profundidad del tipo
    de dato
217.     %permisible
218.     nBins = 256;
219.     %Se encuentran los histogramas para cada plano RGB
220.     [rHist,x] = imhist(I(:,:,1),nBins);
221.     [gHist,y] = imhist(I(:,:,2),nBins);
222.     [bHist,z] = imhist(I(:,:,3),nBins);
223.
224.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PLANO
    ROJO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
225.     figure%Creación del objeto gráfico
226.     xlabel('Intensidad','FontSize', 12, 'FontName' , 'times') % Etiqueta
227.     ylabel('Frecuencia','FontSize', 12, 'FontName' , 'times') % Etiqueta
228.     xlim([0 256]) %Límites eje X
229.     fig = figure(1);
230.     fig.Units = 'centimeters';
231.     fig.Position(3) = 8;
232.     fig.Position(4) = 6;

```

```

233.     set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
234.     set(fig,'defaultAxesColorOrder',[0 0 0]);
235.     fig.PaperPositionMode = 'auto';
236.     grid
237.     rHist = smooth(rHist,6);
238.     hold on;
239.     plot(1:256, rHist, '-r','LineWidth',2);%Graficación del Histograma
240.     %valores máximos del vector del histograma y sus localizaciones
241.     Rpks = findpeaks(rHist);
242.     Rumbal = 0.8*(max(Rpks));%Condición para filtrar de picos menores a
80%
243.     [Rpks,Rlocs] =
findpeaks(rHist,'MINPEAKHEIGHT',Rumbal,'Threshold',.1);
244.     %Condición para suprimir los picos cuando la cercanía es menor a 5
unidades
245.     %de intensidad en caso de que el programa detecte picos muy próximos
246.     if length(Rpks)>= 2
247.         [r,s] = max(Rlocs);%Banderas para graficación correcta del
máximo útil
248.         Rxumb = r-min(Rlocs);
249.         if Rxumb <=6
250.             Rpks = max(Rpks);
251.             Rlocs = Rlocs(s);
252.         end
253.     end
254.     %%%%%%
255.     Rpicos = Rpks(Rpks>=Rumbal);%Determinación del número de picos
encontrados
256.
257.     plot( Rlocs, Rpks,'*k', 'LineWidth',1)
258.     hold off
259.
260.     %Exportación a formato .eps para CorelDraw
261.     print(fig,'C:\Users\Raúl\Documents\Tesis\Imágenes
Tesis\Histogramas\Histograma_R','-depsc')
262.
263.     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PLANO
VERDE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
264.     figure%Creación del objeto gráfico
265.     xlabel('Intensidad','FontSize', 12, 'FontName' ,'times') % Etiqueta
266.     ylabel('Frecuencia','FontSize', 12, 'FontName' ,'times') % Etiqueta
267.     xlim([0 256]) %Límites eje X
268.     fig = figure(2);
269.     fig.Units = 'centimeters';
270.     fig.Position(3) = 8;
271.     fig.Position(4) = 6;
272.     set(gca,'LooseInset', max(get(gca,'TightInset'), 0.02))
273.     fig.PaperPositionMode = 'auto';
274.     grid
275.     gHist = smooth(gHist,15);
276.     hold on;
277.     plot(1:256, gHist, '-g','LineWidth',2);%Graficación del Histograma
278.     %Valores máximos del vector del histograma y sus localizaciones
279.     Gpks = findpeaks(gHist);
280.     Gumbral = 0.8*(max(Gpks));%Condición para filtrar de picos menores a
80%
281.     [Gpks,Glocs] =
findpeaks(gHist,'MINPEAKHEIGHT',Gumbral,'Threshold',0.2);
282.     %Condición para suprimir los picos cuando la cercanía es menor a 5
unidades
283.     %de intensidad en caso de que el programa detecte picos muy próximos
284.     if length(Gpks)>= 2

```

```

285.         [r,s] = max(Glocs);%Banderas para graficación correcta del
           máximo útil
286.         Gxumb = r-min(Glocs);
287.         if Gxumb <=6
288.             Gpks = max(Gpks);
289.             Glocs = Glocs(s);
290.         end
291.     end
292.     %%%%%%
293.     Gpicos = Gpks(Gpks>=Gumbral);%Determinación del número de picos
           encontrados
294.     plot( Glocs, Gpks, '*k', 'LineWidth',1)
295.     hold off
296.     %Exportación a formato .eps para CorelDraw
297.     print(fig,'C:\Users\Raúl\Documents\Tesis\Imágenes
           Tesis\Histogramas\Histograma_G','-depsc')
298.
299.     %%%%%%%%%%PLANO
           AZUL%%%%%%%%%
300.     figure%Creación del objeto gráfico
301.     xlabel('Intensidad','FontSize', 12, 'FontName' , 'times') % Etiqueta
302.     ylabel('Frecuencia','FontSize', 12, 'FontName' , 'times') % Etiqueta
303.     xlim([0 256]) %Límites eje X
304.     fig = figure(3);
305.     fig.Units = 'centimeters';
306.     fig.Position(3) = 8;
307.     fig.Position(4) = 6;
308.     set(gca, 'LooseInset', max(get(gca, 'TightInset'), 0.02))
309.     fig.PaperPositionMode = 'auto';
310.     grid
311.     bHist = smooth(bHist,6);
312.     hold on;
313.     plot(1:256, bHist, '-b', 'LineWidth',2);%Graficación del Histograma
314.     %valores máximos del vector del histograma y sus localizaciones
315.     Bpks = findpeaks(bHist);
316.     Bumbral = 0.8*(max(Bpks));%Condición para filtrar de picos menores a
           80%
317.     [Bpks,Blocs] =
           findpeaks(bHist, 'MINPEAKHEIGHT', Bumbral, 'Threshold', 0.1);
318.     %Condición para suprimir los picos cuando la cercanía es menor a 5
           unidades
319.     %de intensidad en caso de que el programa detecte picos muy próximos
320.     if length(Bpks)>= 2
321.         [r,s] = max(Blocs);%Banderas para graficación correcta del
           máximo útil
322.         Bxumb = r-min(Blocs);
323.         if Bxumb <=6
324.             Bpks = max(Bpks);
325.             Blocs = Blocs(s);
326.         end
327.     end
328.     Bpicos = Bpks(Bpks>=Bumbral);%Determinación del número de picos
           encontrados
329.     plot( Blocs, Bpks, '*k', 'LineWidth',1)
330.     hold off
331.     %Exportación a formato .eps para CorelDraw
332.     print(fig,'C:\Users\Raúl\Documents\Tesis\Imágenes
           Tesis\Histogramas\Histograma_B','-depsc')
333.
334.     %%%%%%%%%Ciclo para determinación de la calidad de la unión de
           soldadura
335.     if length(Rpicos)== 1
336.         disp ('Rojo Aceptable')

```

```
337.         if length(Gpicos)== 1
338.             disp ('Verde Aceptable')
339.             if length(Bpicos)== 1
340.                 disp ('Azul Aceptable')
341.                 %Se cumple la condición de tener un sólo pico para cada
color
342.                 disp ('Unión de Soldadura Aceptable')
343.             else
344.                 disp ('Unión de Soldadura Inaceptable')
345.             end
346.         else
347.             disp ('Unión de Soldadura Inaceptable')
348.         end
349.     else
350.         disp ('Unión de Soldadura Inaceptable')
351.     end
```

```
1. Module1.vb
2. Module Module1
3.     Public time01 As Integer
4.     Public pwm As Integer
5.     Public duty As Integer
6.     Public lectura As String
7. End Module
8. Form1.vb
9. Imports System
10. Imports System.IO.Ports
11. Imports Microsoft.VisualBasic.PowerPacks
12. Public Class InterfazAnillo
13.
14.     Dim myPort As Array
15.     Private Sub Encender_Click(sender As Object, e As EventArgs) Handles
Encender.Click
16.         If SerialPort1.IsOpen Then
17.
18.             Encender.Enabled = False
19.             Apagar.Enabled = True
20.             PictureBox2.Visible = True
21.             Label11.Visible = False
22.             PictureBox2.Image =
Image.FromFile("C:\Users\Raúl\Documents\MATLAB\Prueba_Anillo_001.png")
23.             TrackBar1.Enabled = True
24.             ProgressBar1.Value = 0
25.             TrackBar1.Value = 0
26.             time01 = 0
27.             SerialPort1.Write(0)
28.
29.             OvalShape3.BackColor = Color.Red
30.             OvalShape4.BackColor = Color.Red
31.             OvalShape5.BackColor = Color.Red
32.             OvalShape6.BackColor = Color.Red
33.             OvalShape7.BackColor = Color.Red
34.             OvalShape8.BackColor = Color.Red
35.             OvalShape9.BackColor = Color.Red
36.             OvalShape10.BackColor = Color.Red
37.             OvalShape11.BackColor = Color.Red
38.             OvalShape12.BackColor = Color.Red
39.             OvalShape13.BackColor = Color.Red
40.             OvalShape14.BackColor = Color.Red
41.
42.             OvalShape15.BackColor = Color.Lime
43.             OvalShape16.BackColor = Color.Lime
44.             OvalShape17.BackColor = Color.Lime
45.             OvalShape18.BackColor = Color.Lime
46.             OvalShape19.BackColor = Color.Lime
47.             OvalShape20.BackColor = Color.Lime
48.             OvalShape21.BackColor = Color.Lime
49.             OvalShape22.BackColor = Color.Lime
50.             OvalShape23.BackColor = Color.Lime
51.             OvalShape24.BackColor = Color.Blue
52.             OvalShape25.BackColor = Color.Lime
53.             OvalShape26.BackColor = Color.Lime
54.
55.             OvalShape27.BackColor = Color.Blue
56.             OvalShape28.BackColor = Color.Blue
57.             OvalShape29.BackColor = Color.Blue
58.             OvalShape30.BackColor = Color.Lime
59.             OvalShape31.BackColor = Color.Blue
60.             OvalShape32.BackColor = Color.Blue
61.             OvalShape33.BackColor = Color.Blue
```

```
62.         OvalShape34.BackColor = Color.Blue
63.         OvalShape35.BackColor = Color.Blue
64.         OvalShape36.BackColor = Color.Blue
65.         OvalShape37.BackColor = Color.Blue
66.         OvalShape38.BackColor = Color.Blue
67.     Else
68.         Label14.Text = "Error"
69.         Label14.ForeColor = Color.Red
70.     End If
71.
72.
73.
74.
75. End Sub
76.
77. Private Sub TrackBar1_ValueChanged(sender As Object, e As EventArgs)
    Handles TrackBar1.ValueChanged
78.
79.     Label7.Text = Format(TrackBar1.Value * 10, "#") & "%"
80.     ProgressBar1.Value = TrackBar1.Value * 10
81.     SerialPort1.Write(TrackBar1.Value)
82.
83. End Sub
84.
85.
86. Private Sub SerialPort1_DataReceived(sender As Object,
    e As IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
87.     If (Encender.Enabled = True) Then
88.         lectura = SerialPort1.ReadLine()
89.         duty = Convert.ToInt32(lectura)
90.
91.     End If
92. End Sub
93.
94. Private Sub Apagar_Click(sender As Object, e As EventArgs) Handles
    Apagar.Click
95.     Apagar.Enabled = False
96.     PictureBox2.Visible = False
97.     Label11.Visible = True
98.     Label11.Text = "Imagen no disponible"
99.     PictureBox2.BorderStyle = BorderStyle.FixedSingle
100.     SerialPort1.Write(0)
101.     ProgressBar1.Value = 0
102.     TrackBar1.Value = 0
103.     time01 = 0
104.     Encender.Enabled = True
105.     TrackBar1.Enabled = False
106.     ProgressBar1.Enabled = False
107.
108.     OvalShape3.BackColor = Color.Gray
109.     OvalShape4.BackColor = Color.Gray
110.     OvalShape5.BackColor = Color.Gray
111.     OvalShape6.BackColor = Color.Gray
112.     OvalShape7.BackColor = Color.Gray
113.     OvalShape8.BackColor = Color.Gray
114.     OvalShape9.BackColor = Color.Gray
115.     OvalShape10.BackColor = Color.Gray
116.     OvalShape11.BackColor = Color.Gray
117.     OvalShape12.BackColor = Color.Gray
118.     OvalShape13.BackColor = Color.Gray
119.     OvalShape14.BackColor = Color.Gray
120.
121.     OvalShape15.BackColor = Color.Gray
```

```
122.         OvalShape16.BackColor = Color.Gray
123.         OvalShape17.BackColor = Color.Gray
124.         OvalShape18.BackColor = Color.Gray
125.         OvalShape19.BackColor = Color.Gray
126.         OvalShape20.BackColor = Color.Gray
127.         OvalShape21.BackColor = Color.Gray
128.         OvalShape22.BackColor = Color.Gray
129.         OvalShape23.BackColor = Color.Gray
130.         OvalShape24.BackColor = Color.Gray
131.         OvalShape25.BackColor = Color.Gray
132.         OvalShape26.BackColor = Color.Gray
133.
134.         OvalShape27.BackColor = Color.Gray
135.         OvalShape28.BackColor = Color.Gray
136.         OvalShape29.BackColor = Color.Gray
137.         OvalShape30.BackColor = Color.Gray
138.         OvalShape31.BackColor = Color.Gray
139.         OvalShape32.BackColor = Color.Gray
140.         OvalShape33.BackColor = Color.Gray
141.         OvalShape34.BackColor = Color.Gray
142.         OvalShape35.BackColor = Color.Gray
143.         OvalShape36.BackColor = Color.Gray
144.         OvalShape37.BackColor = Color.Gray
145.         OvalShape38.BackColor = Color.Gray
146.     End Sub
147.
148.     Private Sub InterfazAnillo_Load(sender As Object,
e As EventArgs) Handles MyBase.Load
149.         DateTimePicker1.Visible = False
150.         TrackBar1.Enabled = False
151.         Encender.Enabled = False
152.         Apagar.Enabled = False
153.         iniciarB.Enabled = False
154.         myPort = IO.Ports.SerialPort.GetPortNames()
155.         ComboBox1.Items.AddRange(myPort)
156.         Label11.Text = "Imagen no disponible"
157.
158.     End Sub
159.
160.     Private Sub Timer1_Tick(sender As Object, e As EventArgs)
Handles Timer1.Tick
161.         Dim fecha_pick As String
162.         fecha_pick = DateTimePicker1.Text
163.         Fecha.Text = fecha_pick
164.     End Sub
165.
166.     Private Sub iniciarB_Click(sender As Object, e As EventArgs)
Handles iniciarB.Click
167.         If Apagar.Enabled = True Then
168.             If iniciarB.Text = "Iniciar" Then
169.                 iniciarB.Text = "Parar"
170.                 Label13.Text = "Inspección en Progreso"
171.                 Label13.ForeColor = Color.Green
172.             ElseIf iniciarB.Text = "Parar" Then
173.                 iniciarB.Text = "Iniciar"
174.                 Label13.Text = "Sistema AOI Apagado"
175.                 Label13.ForeColor = Color.Red
176.             End If
177.         Else
178.             Label13.Text = "ERROR: Anillo Apagado!"
179.             Label13.ForeColor = Color.Red
180.         End If
181.     End Sub
```

```
182.
183.     Private Sub conectarB_Click(sender As Object, e As EventArgs)
        Handles conectarB.Click
184.         If conectarB.Text = "Conectar" Then
185.             conectarB.Text = "Desconectar"
186.             Label14.Text = "OK"
187.             Label14.ForeColor = Color.Green
188.             SerialPort1.PortName = ComboBox1.SelectedItem
189.             SerialPort1.Open()
190.             Encender.Enabled = True
191.             iniciarB.Enabled = True
192.         ElseIf conectarB.Text = "Desconectar" Then
193.             conectarB.Text = "Conectar"
194.             Label14.Text = "S/C"
195.             Label14.ForeColor = Color.Red
196.             SerialPort1.Close()
197.         End If
198.     End Sub
199. End Class
```