



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

**“DISEÑO DE UN PROTOTIPO DE SISTEMA DE ALERTA**  
**TEMPRANA PARA INUNDACIÓN Y DESBORDAMIENTO,**  
**USANDO VISIÓN ARTIFICIAL Y REDES CELULARES PARA LA**  
**CIUDAD DE PUYO”**

**TRABAJO DE TITULACIÓN: DISPOSITIVO TECNOLÓGICO**

**Para optar al Grado Académico de:**

**INGENIERO EN ELECTRÓNICA, CONTROL Y REDES**  
**INDUSTRIALES**

**AUTOR: HENRY SANTIAGO SAA UREÑA.**

**TUTOR: ING. PABLO LOZADA.**

**Riobamba – Ecuador**  
**2018**

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES**  
**INDUSTRIALES**

El tribunal del trabajo de titulación certifica que: El trabajo de tipo de dispositivo tecnológico: “DISEÑO DE UN PROTOTIPO DE SISTEMA DE ALERTA TEMPRANA PARA INUNDACIÓN Y DESBORDAMIENTO, USANDO VISIÓN ARTIFICIAL Y REDES CELULARES PARA LA CIUDAD DE PUYO”, de responsabilidad del señor HENRY SANTIAGO SAA UREÑA, ha sido minuciosamente revisado por los miembros del tribunal del trabajo de titulación, quedando autorizada su presentación.

<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
Dr. Julio Santillan. C <b>VICEDECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA.</b>	.....	.....
Ing. Freddy Chávez. V <b>DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES.</b>	.....	.....
Ing. Pablo Lozada. Y <b>DIRECTOR DE TRABAJO DE TITULACION</b>	.....	.....
Dr. Jorge Luis Hernández. A <b>MIEMBRO DEL TRIBUNAL.</b>	.....	.....

Yo, Henry Santiago Saá Ureña, declaro que el presente Trabajo de Titulación es de mi autoría y que los resultados del mismo son auténticos y originales. Los textos constantes en el documento que provienen de otra fuente están debidamente citados y referenciados. Como autor, asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación.

Henry Santiago Saá Ureña

## **DEDICATORIA**

Agradezco eternamente a mi madre, por obsequiarme la vida, por su amor y esfuerzo constante, por su ejemplo de lucha y trabajo, por su apoyo incondicional en decisiones trascendentes en mi existencia de vida.

A mi hermana, Anita por su afecto, comprensión y motivación en momentos cruciales de mi vida, Carmen por su apoyo y comprensión.

A mis amistades y familiares, por los gratos recuerdos de alegrías, de angustias, que quedaran por siempre.

Henry Santiago Saá Ureña.

## **AGRADECIMIENTO**

Agradezco a la vida por cada momento de alegría, de paz, de euforia, por tantos recuerdos que vivirán por siempre en mi memoria.

A la Escuela Superior Politécnica de Chimborazo por la formación profesional que me ha brindado, atreves de la ciencia.

A las y los docentes, que me guiaron y permitieron caminar por esta senda, del conocimiento.

Henry Santiago

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	ix
INDICE DE FIGURAS.....	x
ÍNDICE DE ABREVIATURAS.....	xiii
RESUMEN.....	xv
ABSTRACT.....	xvi
INTRODUCCIÓN .....	17
<b>CAPITULO I</b>	
<b>1. Marco Teórico .....</b>	<b>23</b>
<b>1.1 Sistemas Embebidos (Sistemas Empotrados).....</b>	<b>23</b>
<i>1.1.1 Componentes de los Sistemas Embebidos .....</i>	<i>24</i>
<i>1.1.2 Sistemas Operativos Embebidos.....</i>	<i>25</i>
<b>1.2 Visión artificial .....</b>	<b>25</b>
<i>1.2.1 Arquitectura de sistemas de visión artificial .....</i>	<i>26</i>
<i>1.2.2 Dispositivo de captura. ....</i>	<i>27</i>
<i>1.2.3 Digitalización de capturas (señal analógica).....</i>	<i>27</i>
<i>1.2.4 Representación de una imagen.....</i>	<i>28</i>
<i>1.2.5 Componentes de las imágenes HSV (Hue/tonalidad, saturation/saturación, intensity/intensidad) a partir de una imagen RGB (Rojo, Verde, Azul).....</i>	<i>31</i>
<i>1.2.7 Rotación visión artificial.....</i>	<i>31</i>
<i>1.2.8 Extracción capa de color (Operador Bitwise_and ).....</i>	<i>32</i>
<i>1.2.9 Binarización.....</i>	<i>33</i>
<i>1.2.10 Detección de contornos .....</i>	<i>34</i>
<i>1.2.11 Operación de morfología .....</i>	<i>35</i>
<i>1.2.12 Operación erosión .....</i>	<i>35</i>
<i>1.2.13 Operación de dilatación .....</i>	<i>36</i>
<i>1.2.14 Apertura y cierre.....</i>	<i>36</i>
<b>1.3 OpenCV (Open Source Computer Vision) .....</b>	<b>37</b>
<i>1.3.1 OpenCV y Python.....</i>	<i>38</i>
<b>1.4 Medición de niveles de ríos .....</b>	<b>38</b>
<i>1.4.1 Lectura de Escala fija .....</i>	<i>39</i>
<b>1.5 Raspberry Pi3 .....</b>	<b>40</b>
<i>1.5.1 Hardware placa Raspberry RPi3 .....</i>	<i>41</i>
<i>1.5.2 Software, Sistemas Operativos de la Placa raspberry pi3.....</i>	<i>41</i>

1.5.3	<i>Raspbian Stretch v2_4_5</i> .....	42
1.5.4	<i>Módulo de cámara (Pi_camera)</i> .....	43
1.6	<b>Internet de las cosas (Internet Of Things)</b> .....	44
<b>CAPITULO II</b>		
2	<b>MARCO METODOLÓGICO</b> .....	45
2.1	<b>Secuencia de procesos que realiza el programa.</b> .....	45
2.2	<b>Adquisición de imágenes</b> .....	47
2.2.1	<i>Selección del dispositivo de captura (Cámara)</i> .....	47
2.2.2	<i>Captura de Escenario (Imagen)</i> .....	47
2.3	<b>Etapa para detección de placa de referencia</b> .....	49
2.3.1	<i>Pre-procesamiento de imagen (Escenario)</i> .....	50
2.3.2	<i>Lectura de imagen.</i> .....	50
2.3.3	<i>Giro de la imagen capturada (Escenario)</i> .....	50
2.3.4	<i>Filtro de Gauss (Filtro para eliminación de ruido)</i> .....	50
2.3.5	<i>Conversión de RGB a HSV</i> .....	51
2.3.6	<i>Filtro de color</i> .....	52
2.3.7	<i>Binarizado</i> .....	52
2.3.8	<i>Morfología</i> .....	53
2.3.9	<i>Dilatación de la morfología, closing(cerrada)</i> .....	53
2.3.10	<i>Detección de bordes y contornos Canny</i> .....	54
2.3.11	<i>Recortar imagen</i> .....	55
2.3.12	<i>Selección del contorno más grande</i> .....	55
2.3.13	<i>Análisis de etiqueta</i> .....	57
2.3.14	<i>Conteo de niveles</i> .....	58
2.3.15	<i>Respaldo de datos identificados</i> .....	59
2.3.16	<i>Análisis de datos</i> .....	59
2.3.17	<i>Altura de nivel de rio</i> .....	60
2.3.18	<i>Situación Actual de Rio</i> .....	61
2.3.19	<i>Taza de cambio de nivel de rio</i> .....	61
2.3.20	<i>Envío de información</i> .....	64
2.3.21	<i>Información vía Mail</i> .....	64
2.3.22	<i>Retransmisión de comunicados</i> .....	65
2.3.23	<i>Información a Servidor WEB</i> .....	67

### CAPITULO III

<b>3</b>	<b>DISCUSIÓN Y ANÁLISIS DE RESULTADOS.</b>	<b>68</b>
<b>3.1</b>	<b>Pruebas de funcionamiento del prototipo</b>	<b>69</b>
<b>3.1.1</b>	<i>Pruebas de detección de placa de referencia</i>	<b>69</b>
<b>3.1.2</b>	<i>Pruebas de conteo de niveles</i>	<b>77</b>
<b>3.1.3</b>	<i>Pruebas de envío de informes (Mails)</i>	<b>83</b>
<b>3.1.4</b>	<i>Pruebas de actualización de datos en la página web</i>	<b>86</b>
<b>3.2</b>	<b>Análisis de resultados de pruebas obtenidas</b>	<b>88</b>
<b>3.2.1</b>	<i>¿Puede usarse sistemas de visión artificial para satisfacer la necesidad de información sobre fenómenos naturales?</i>	<b>88</b>
<b>3.2.2</b>	<i>¿Cuanto influyen librerías y algoritmos, en conjunto con el desempeño de los procesos de visión artificial, en la funcionalidad de este sistema?</i>	<b>89</b>
<b>3.2.3</b>	<i>Beneficios en el uso de la red para información sobre situaciones de interés para evitar catástrofes</i>	<b>92</b>
	<b>CONCLUSIONES</b>	<b>96</b>
	<b>RECOMENDACIONES</b>	<b>97</b>
	<b>BIBLIOGRAFÍA</b>	<b>98</b>



## ÍNDICE DE TABLAS

<b>Tabla 1-1 Características Placa Raspberry Pi3 .....</b>	<b>41</b>
<b>Tabla 2-1 Características cámara RPi3 .....</b>	<b>43</b>
<b>Tabla 1-2 Referencias de alertas, rango de niveles. ....</b>	<b>59</b>
<b>Tabla 2-2 Datos de referencia para calculo .....</b>	<b>60</b>
<b>Tabla 3-2 Limites superiores de rangos de nivel .....</b>	<b>62</b>
<b>Tabla 4-2 Datos de refencias, para cálculo de información.....</b>	<b>62</b>
<b>Tabla 5-2 Datos de referencia de alertas.....</b>	<b>62</b>
<b>Tabla 6-2 Alertas y rangos .....</b>	<b>65</b>
<b>Tabla 1-3 Tiempos de detección placa .....</b>	<b>76</b>
<b>Tabla 2-3 Filtro Gaussiano .....</b>	<b>89</b>

## INDICE DE FIGURAS

Figura 1-1 Proceso de un sistema de vision artificial .....	26
Figura 2-1 Espectro Electromagnético.....	27
Figura 3-1 Sistema de coordenadas de pixeles de una imagen.....	28
Figura 4-1 Distribución de pixeles en vecindad a cuatro.....	28
Figura 5-1 Distribución de pixeles en vecindad a ocho.....	29
Figura 6-1 Representación de banda de Matiz vs RGB.....	30
Figura 7-1 Representación de cono de color HSV.....	30
Figura 8-1 Proceso de filtrado de imagen.....	32
Figura 9-1 Binarización .....	33
Figura 10-1 Detección de contornos .....	34
Figura 11-1 Erosión de una imagen binaria. ....	35
Figura 12-1 Dilatación de una imagen binaria .....	36
Figura 13-1 Proceso de cierre.....	37
Figura 14-1 Escalas para registro de niveles de río.....	39
Figura 15-1 Raspberry Pi3 .....	40
Figura 16-1 Raspbian Stretch v2_4_5.....	42
Figura 17-1 Cámara RPi3 8 Mp. ....	43
Figura 1-2 Diagrama de flujo que describe el sistema .....	45
Figura 2-2 Flujograma describe el algoritmo de visión implementado.....	46
Figura 3-2 Ventana de activación de la cámara .....	48
Figura 4-2 Ventana de activación de la cámara .....	48
Figura 5-2 Flujograma detección de placa de referencia .....	49
Figura 6-2 Imagen supresión de ruido, filtro de gauss. ....	51
Figura 7-2 Filtro de color rojo .....	52
Figura 8-2 Binarizacion.....	53
Figura 9-2 Dilatación morfológica CLOSING.....	54
Figura 10-2 Ventana de activación de la cámara .....	55
Figura 11-2 Detección del contorno más grande. ....	56
Figura 12-2 Detección del contorno más grande. ....	56
Figura 13-2 Análisis de etiqueta.....	57
Figura 14-2 Conteo de niveles a analizar .....	58
Figura 15-2 Bandeja de entrada mail, del dispositivo.....	66
Figura 16-2 Mail, bandeja de entrada (Usuarios) .....	66

<b>Figura 17-2 Mensaje informativo de mail enviado por el dispositivo .....</b>	<b>67</b>
<b>Figura 18-2 Pagina WEB de monitoreo .....</b>	<b>67</b>
<b>Figura 1-3 Disposición de dispositivo .....</b>	<b>68</b>
<b>Figura 2-3 Algoritmo de detección de placa parte a .....</b>	<b>70</b>
<b>Figura 3-3 Algoritmo de detección de placa parte b .....</b>	<b>71</b>
<b>Figura 4-3 Ventana de monitoreo de procesos del programa.....</b>	<b>71</b>
<b>Figura 5-3 Placa de escenario en condiciones normales .....</b>	<b>72</b>
<b>Figura 6-3 Placa de escenario en condiciones normales .....</b>	<b>72</b>
<b>Figura 7-3 Placa de escenario en condiciones de alerta verde .....</b>	<b>73</b>
<b>Figura 8-3 Placa de escenario en condiciones de alerta amarilla .....</b>	<b>73</b>
<b>Figura 9-3 Placa de escenario en condiciones de alerta roja .....</b>	<b>74</b>
<b>Figura 10-3 Ventana procesos del programa, ausencia de placa .....</b>	<b>74</b>
<b>Figura 11-3 Ausencia de placa .....</b>	<b>75</b>
<b>Figura 12-3 Proceso de detección de niveles .....</b>	<b>78</b>
<b>Figura 13-3 Ventana de procesos de sistema .....</b>	<b>79</b>
<b>Figura 14-3 Resultado de procesos de niveles.....</b>	<b>79</b>
<b>Figura 15-3 Ventana de procesos de sistema .....</b>	<b>80</b>
<b>Figura 16-3 Resultado de procesos de niveles.....</b>	<b>80</b>
<b>Figura 17-3 Ventana de procesos de sistema .....</b>	<b>81</b>
<b>Figura 18-3 Resultado de procesos de niveles.....</b>	<b>81</b>
<b>Figura 19-3 Ventana de procesos de sistema .....</b>	<b>82</b>
<b>Figura 20-3 Resultado de procesos de niveles.....</b>	<b>82</b>
<b>Figura 21-3 Ventana de procesos de sistema .....</b>	<b>83</b>
<b>Figura 22-3 Resultado de procesos de niveles.....</b>	<b>83</b>
<b>Figura 23-3 Envío de informe de actividad “Normal”.....</b>	<b>84</b>
<b>Figura 24-3 Envío de informe de actividad “Verde”.....</b>	<b>84</b>
<b>Figura 25-3 Envío de informe de actividad “Amarilla”.....</b>	<b>85</b>
<b>Figura 26-3 Envío de informe de actividad “Roja”.....</b>	<b>85</b>
<b>Figura 27-3 Envío de informe de actividad “Error de placa”.....</b>	<b>85</b>
<b>Figura 28-3 Página de información.....</b>	<b>86</b>
<b>Figura 29-3 Imagen de comunicado, publicado en web.....</b>	<b>86</b>
<b>Figura 30-3 Imagen que muestra el sitio web.....</b>	<b>87</b>
<b>Figura 31-3 Imagen de datos publicados en el servidor web.....</b>	<b>87</b>
<b>Figura 32-3 Información adicional.....</b>	<b>88</b>
<b>Figura 33-3 Prototipo de SAT.....</b>	<b>89</b>

<b>Figura 34-3 Imagen en ausencia de filtro Gaussiano.....</b>	<b>90</b>
<b>Figura 35-3 Imagen en presencia de filtro Gaussiano.....</b>	<b>91</b>
<b>Figura 36-3 Zonas de interés de monitoreo.....</b>	<b>95</b>

## ÍNDICE DE ABREVIATURAS

<b>OpenCV:</b>	Acrónimo de las palabras Open Source Computer Vision
<b>PC:</b>	Computador Personal.
<b>MHz:</b>	Megahercios.
<b>CC:</b>	Voltaje de corriente Continua.
<b>AC:</b>	Voltaje de corriente alterna.
<b>CPU:</b>	Unidad central de proceso.
<b>CC:</b>	Corriente continúa.
<b>V:</b>	Voltaje.
<b>I/O:</b>	Entrada/Salida (Input/Output).
<b>RAM:</b>	Memoria de acceso aleatorio.
<b>ROM:</b>	Memoria de sólo lectura.
<b>A:</b>	Amperios.
<b>RTC:</b>	Reloj en tiempo real.
<b>USB:</b>	Bus Universal en Serie
<b>IP:</b>	Internet Protocolo
<b>SO:</b>	Sistema Operativo
<b>SATI:</b>	Sistema de alerta temprana de inundación
<b>Jpg:</b>	Joint Photographic Experts Group
<b>HSV:</b>	Hue/tonalidad, saturation/saturación, intensity/intensidad.

## ÍNDICE DE ANEXOS

**ANEXO A:** ECIPITACIÓN PROMEDIO ANUAL EN EL ECUADOR. FUENTE: MAE, 2013.

**ANEXO B:** DIAGRAMA DE FLUJO DE PROCEDIMIENTOS PARA GESTION DE RIESGOS

**ANEXO C:** DIAGRAMA DE FLUJO DE PROCEDIMIENTOS PARA GESTIÓN DE RIESGOS

**ANEXO D:** RADAR SISTEMA DE ALERTA TEMPRANA DE INUNDACIONES

## RESUMEN

Se diseñó un prototipo de sistema de alerta temprana para inundación y desbordamiento usando visión artificial y redes celulares para la ciudad de Puyo, para informar sobre actividad fluvial, simulando las situaciones del río Citayacu ubicado en el barrio Obrero en la ciudad de Puyo. El sistema basó su procesamiento en un computador de placa reducida, Raspberry Pi3B+, junto con un módulo PiCamera, los cuales se posicionaron de tal manera que se capturan imágenes del ambiente a ser analizado. Dichas imágenes entregan información de la altura en relación a los niveles que mantiene durante actividad fluvial normal. Las imágenes se procesan mediante algoritmos de segmentación codificados bajo lenguaje de programación Python en conjunto con librerías de visión artificial de OpenCV. Con el fin de usar técnicas de procesamiento para localizar en la captura, la placa de referencia, posteriormente los contornos correspondientes a las franjas relacionadas a niveles identificados, para informar sobre la situación que se generó. En base a estos datos, se emiten mails de información a grupos de socorro, a usuarios que se encuentran en zonas de riesgo de inundación, cercanas al lugar de monitoreo. Toda la información recolectada es enviada a un servidor web, permitiendo informar a quienes deseen conocer la situación que estuvo presentándose en relación a la actividad fluvial de la localización. Cuando los grupos de socorro deciden comunicar cierta información es enviada hacia el dispositivo y retransmitida a los diferentes usuarios, así como también publicado en el sitio web. El dispositivo presentó una alternativa de sistema de monitoreo ante procedimientos de monitoreo convencionales que realizan los grupos de riesgo en la localidad, además de información continua en la web. El prototipo se proyecta a evolucionar hacia nuevos sistemas que nos permitan adquirir datos sobre otras situaciones de interés para monitoreo, brindándonos mayor información ante posibles desastres naturales en cada localidad monitoreada.

**Palabras Claves:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INTELIGENCIA ARTIFICIAL>, <VISIÓN ARTIFICIAL>, <ACTIVIDAD FLUVIAL>, <INTERNET DE LAS COSAS (IOT)>, <OPENCV (SOFTWARE)>, <PLACA DE REFERENCIA>, <RECONOCIMIENTO DE CONTORNOS>, <SISTEMA EMBEBIDO>.

## ABSTRACT

A prototype of an early warning system for flood and overflow was designed using artificial vision and cellular networks for the Puyo city, to inform about fluvial activity, simulating the Citayacu River situations located in the Obrero neighborhood in the Puyo city. The system based its processing on a small board computer, Raspberry Pi3B +, together with a PiCamera module, which was positioned in such a way that images of the environment to be analyzed are captured. These images provide height information in relation to the levels maintained during normal fluvial activity. The images are processed by encoded segmentation algorithms under Python programming language in conjunction with OpenCV artificial vision libraries. In order to use processing techniques to locate in the capture, the reference plate, subsequently the contours corresponding to the strips related to identified levels, to report on the situation that was generated. Based on these data, information e-mails are sent to emergency groups, to users who are in flood risk zones, near the monitoring site. All the information collected is sent to a web server, allowing informing those who wish to know the situation that was presented in relation to the fluvial activity of the location. When the emergency groups decide to communicate certain information, it is sent to the device and retransmitted to different users, as well as published on the website. The device presented an alternative monitoring system for conventional monitoring procedures carried out by risk groups in the locality, in addition to continuous information on the web. The prototype is projected to evolve towards new systems that allow us to acquire data about other situations of interest for monitoring, giving us more information about possible natural disasters in each monitored location.

**KeyWords:** <TECHNOLOGY AND ENGINEERING SCIENCES>, <ARTIFICIAL INTELLIGENCE ^ <ARTIFICIAL VISION>, <FLUVIAL ACTIVITY>, <INTERNET OF THINGS (IOT)>, OPENCV (SOFTWARE)>, <REFERENCE PLATE <RECOGNITION OF CONTOURS >, <EMBEDED SYSTEM>



## INTRODUCCIÓN

La ciudad de Puyo al encontrarse ubicada en una zona geográfica, húmeda de alta precipitación de lluvias (véase anexo A), constantemente se encuentra situaciones de inundación, por descarga de aguas hacia los afluentes.

El presente proyecto de titulación se enfocó en diseñar una alternativa, ante la necesidad información, para conocer la actividad fluvial de un río, se tomó en cuenta para el desarrollo de este proyecto de tesis, eventualidades que se presentan ante lluvias prolongadas, simulando la situaciones que presenta el río Citayacu ubicado el en barrio Obrero en la ciudad de Puyo, que es afluente al río Puyo. Con este proyecto es una alternativa de SATI (sistema de alerta temprana de inundación) para mantener informada a la población y grupos de socorro, sobre la actividad fluvial que se presente.

La ciudad de Puyo se encuentra constantemente en presencia de lluvias, que se prolongan en ocasiones por más de doce horas, provocando que crezcan el cauce de los ríos y sus afluentes. Esta actividad fluvial genera problemas debido a que cuando se elevan los niveles de los ríos inunda zonas cercanas, afectando y poniendo en peligro a la población. Actualmente ante este tipo de eventualidades la población no cuenta con un medio que pueda informarle de la situación que esté generándose en su localidad.

Los grupos de socorro de la ciudad de Puyo, ante una situación de lluvia (evento adverso), se manejan con un procedimiento que se basa en dirigirse hacia los afluentes más cercanos y en base a simple inspección (observación), determinar la posible actividad que se podría dar o estar generándose ante la situación en ese momento del río, generar informes hacia otras dependencias (véase anexo B y anexo C) para después de este procedimiento informar a la población.

Los sistemas tradicionales que se han usado para vincularlos como sistemas de alerta temprana en la ciudad de Puyo, basan su funcionamiento en datos de la estacione meteorológica que aun esta en uso, pero ubicadas fuera del perímetro urbano de la ciudad por lo que no presenta fidelidad aquellos datos , en este procedimiento toman en cuenta la cantidad de gotas de lluvia por metro cuadrado y la distancia que describe el río desde cierto afluente hasta cierta zona, para hacer un análisis en un área de interés y así generar cierta información de la actividad del río.

## ANTECEDENTES

La ciudad de Puyo capital de la provincia de Pastaza, ubicada en la región amazónica mantiene durante todo el año un clima inestable. Al ser una zona húmeda y cercana a elevaciones parte de la cordillera de los andes, por la cual viajan vientos desde la sierra, que por consecuencia traen masas de nubosidades elevadas, hacia el oriente provocando lluvias prolongadas en ocasiones por más de doce horas.

La importancia y necesidad de un sistema de alerta temprana ante inundación y desbordamiento es de mantener informada a la población, en este caso a quienes se encuentran en zonas de mayor riesgo de inundación, así como también informar a las entidades correspondientes. Información que comunique datos sobre el estado de los ríos y permita considerar situaciones de emergencia, o procedimientos que deban ejecutarse ante tales eventualidades, siendo este el fin para el desarrollo de este proyecto.

La población de la ciudad de Puyo no cuenta con un sistema de información en tiempo real ante este tipo de eventos naturales. Mientras que la logística implementada por grupos de ayuda se basa en simple observación desde puentes ubicados sobre ríos en lugares cercanos.

En una situación crítica de lluvias, los grupos encargados de la seguridad ante riesgos, se movilizan con sus vehículos, observando el nivel de los ríos (simple observación). Dicha movilización es limitada a lugares cercanos y donde es posible ingresar con vehículos, recorriendo así los diferentes puentes ubicados sobre ríos dentro de la ciudad. Estas operaciones de inspección son realizadas cada cierto periodo de tiempo para emitir informes de las situaciones generadas (véase anexo B y anexo C) por el incremento en el caudal de los ríos lo que conlleva periodos de tiempo hasta poder emitir información sobre esta actividad fluvial. En ocasiones donde ya se ha producido una situación de emergencia, las patrullas de inspección comunican a la central de riesgo y envían grupos de ayuda.

Al encontrarse en una situación de inundación, continúan con las mismas rutinas de observación por varios lugares mientras no haya crecido demasiado el nivel de los ríos, de presentarse el caso, dejan de recorrer por el lugar que haya sido afectado (inundado).

Debido a que la ciudad de Puyo tiene terrenos con superficies bastante irregulares es común encontrar viviendas a nivel del río o cercanas a zonas de riesgo, que ante situaciones de lluvias prolongadas terminan siendo objeto de una situación de emergencia (vivienda inundada), y el hecho de que un vehículo transite por estas zonas, sean vehículos particulares o en estos casos vehículos de socorro, desplazan agua que sigue ingresando a las viviendas, afectando y provocando molestias y daños.

En otros casos, los grupos de ayuda se encuentran mayormente enfocados en atender, zonas que en relación a otras, fueron las primeras en pedir ayuda, o fueron las primeras que se encontraron en emergencia. Mientras sucede esto, en otros lugares de la ciudad se pueden presentar situaciones de mayor riesgo, por lo que otras entidades pudieran acudir en ayuda, por ejemplo, bomberos, policía, y demás. Sin embargo, la falta de información del estado de los caudales de los ríos a tiempo real provoca que estas instituciones no estén al tanto de información sobre el estado de otros barrios o ríos.

Sistemas parecidos están siendo usados en Catamarca, Argentina, estos sistemas son parte de un proyecto de la UNC (Universidad Nacional de Córdoba), que busca de informar a las entidades pertinentes sobre la profundidad de río (detección de nivel con sensores para detectar la altura del río) así como caudalímetros (véase anexo D), ubicando cada sistema en las márgenes del río, retransmitiendo esta información de forma inalámbrica hacia estaciones de monitoreo. (unciencia.unc.edu.ar, 2015)

## **FORMULACIÓN DEL PROBLEMA**

¿Es posible diseñar un sistema de alerta temprana para emergencia ante inundaciones, mediante un sistema monitoreo en tiempo real usando visión artificial?

## **JUSTIFICACION TEORICA**

El proyecto propone, diseñar un prototipo que monitoree el nivel de la superficie del agua, tomando como referencia la situación en un afluente al río Puyo en un tramo ubicado en la zona urbana de la ciudad de Puyo, y dependiendo del tipo de situación, alertar a los moradores del sector donde está monitoreándose, enviar estos datos a los usuarios de la zona de riesgo así como también mantener informados a grupos de socorro para que tomen decisiones.

El monitoreo se basa en principio, en obtener imágenes que me permitan definir el nivel del río en relación a un nivel de referencia señalado (Placa de referencia), tomando esta fase como muestreo cada cierto tiempo y emitiendo datos sobre el estado del río vía mail, así como a un servidor web, usando como medio de transmisión de información, redes celulares.

Basado en la información extraída, identificar una tasa de cambio en relación al nivel del río, determinando niveles críticos y alertas, para mantener informada y prevenir a la población cercana, o que se encuentre dentro de la zona de riesgo, ante una posible inundación y un posible desbordamiento por medio también de paquetes de información (e-mail) por redes celulares.

## **JUSTIFICACIÓN APLICATIVA**

Los métodos usados por los organismos de riesgo en la ciudad de Puyo son poco eficientes, debido a la extensión de la ciudad y cantidad de ríos que cruzan por la misma, ante la presencia de lluvias que se mantienen por prolongados periodos de tiempo.

Los organismos como la Cruz roja realizan recorridos, por los diferentes puentes observando el nivel de agua de los diferentes ríos, conllevando invertir mucho tiempo hasta recorrer afluentes y ríos principales que cruzan la ciudad para tener un panorama referente de la situación, ante posibles inundaciones, teniendo que repetir todo este proceso cada cierto tiempo, para tener un cierto estado de la situación hídrica, ante eventos prolongados de lluvia. En el caso de presentarse situaciones de inundación en cierta zona, los grupos de socorro acuden como ayuda, mientras otro grupo se dedica a seguir observando el nivel de los ríos por cada sector posible, si en algún sector el agua ha sobrepasado el nivel de la calzada (calle) los vehículos dejan de pasar de volver a pasar por el lugar, por la posible molestia y daños que puede generar el desplazamiento de agua consecuencia del transitar de vehículos.

En este caso la ayuda de otros organismos de socorro es tardía, por la ausencia de información sobre la situación y sobre los sectores afectados, así como también de nuevos posibles sectores en emergencia, o posibles desbordamientos en ciertas zonas, a su vez no tienen ninguna información real de cómo está cambiando el nivel del río, como un referente y prever posibles sectores afectados o posibles planes de contingencia.

Con este proyecto pretendo cubrir la problemática, de ausencia de información (datos) que nos permita conocer el estado de los ríos, tasa de incremento, para distribuir eficientemente ayuda y personal de socorro, así como identificar zonas críticas y predecir posibles zonas en emergencia, a su vez manteniendo informada a la población y demás organismos de rescate.

## **OBJETIVOS**

### **OBJETIVOS GENERALES:**

Diseñar un prototipo de sistema de alerta temprana para inundación y desbordamiento usando visión artificial y redes celulares para la ciudad de Puyo.

### **OBJETIVOS ESPECÍFICOS:**

- Determinar el mejor hardware y software que se usará en el prototipo.
- Diseñar un prototipo de monitoreo mediante visión artificial, como un sistema de alerta temprana.
- Obtener datos de la tasa de incremento de nivel de la superficie del agua simulando el afluente Citayacu del río Puyo.
- Generar comunicación de la estación de monitoreo con la estación base mediante redes celulares.
- Realizar pruebas de funcionamiento del sistema de monitoreo.
- Analizar y concluir los resultados obtenidos.

## CAPITULO I

### 1. MARCO TEÓRICO

#### 1.1 Sistemas embebidos (sistemas empotrados)

Un sistema empotrado es un sistema, donde su función principal no es computacional, pero es controlado por un equipo integrado que lo computa. Este equipo o dispositivo, puede estar compuesto por micros controladores o microprocesadores. El termino empotrado implica que se encuentra alojado dentro del sistema del equipo, no se lo puede mantener a simple vista, y forma parte de un conjunto más grande (de un todo). (Pérez, 2009, p.04)

Los sistemas computacionales se componen de hardware y software embebido, como elementos importantes. El sistema es independiente o parte de uno más grande, puesto que su software está embebido suele alojarse en memoria estática (ROM) por lo que es indiferente de la memoria dinámica (RAM). (Pérez, 2009, p.04)

Un sistema incrustado (embebido), distinguen tres características que lo definen y lo distinguen de otros sistemas computacionales.

- El sistema ejecuta tareas específicas de forma repetitiva, a diferencia de sistemas computacionales ordinarios que ejecuta cierta cantidad de programas.
- Presenta ciertas limitaciones de implementación como pueden ser de diseño (Problemas métricos), de tamaño, de costo desempeño, de consumo energético, etc. Un sistema embebido por lo general debe ser poco costoso, mantener un tamaño reducido, mantener buen desempeño de procesamiento datos a tiempo real. (Pérez, 2009, p.05)

- Capacidad de funcionamiento ante cambios ambientales, sin interrumpir su operación ni afectar funciones otorgadas.

### ***1.1.1 Componentes de los Sistemas Embebidos***

Se contemplan tres elementos principales, en sistemas empotrados:

- **Hardware (Equipo)**

Referente a los componentes físicos (Hardware), el equipo que usado en sistemas embebidos es parte importante para el desempeño y procesos que requiera el software, para poder ejecutar con fidelidad los procesos para los que fue diseñado el sistema. Muchos sistemas empotrados han mostrado una evolución en relación a sistemas comunes de cómputo como el tamaño, el consumo energético, su funcionalidad, eficiencia en capacidad de procesamiento, etc.

- **Software (Sistema)**

Software es el sistema que se ejecutara en segundo plano, el usuario no lo puede ver, realizando procesos repetitivos para los que fue programado.

El software que corre en sistemas empotrados es programado bajo restricciones de interés para su diseño:

- Que use en lo posible un mínimo espacio de memoria.
- Tamaño limitado de paquetes para procesar, debido a la velocidad que prestan los procesadores actuales en el mercado para este tipo de sistemas.



- Buscar optimizar el consumo energético en etapas de ejecución o espera de procesos, a más de buscar evitar el deterioro de los componentes. (Pérez, 2009, p.04)

### ***1.1.2 Sistemas Operativos Embebidos***

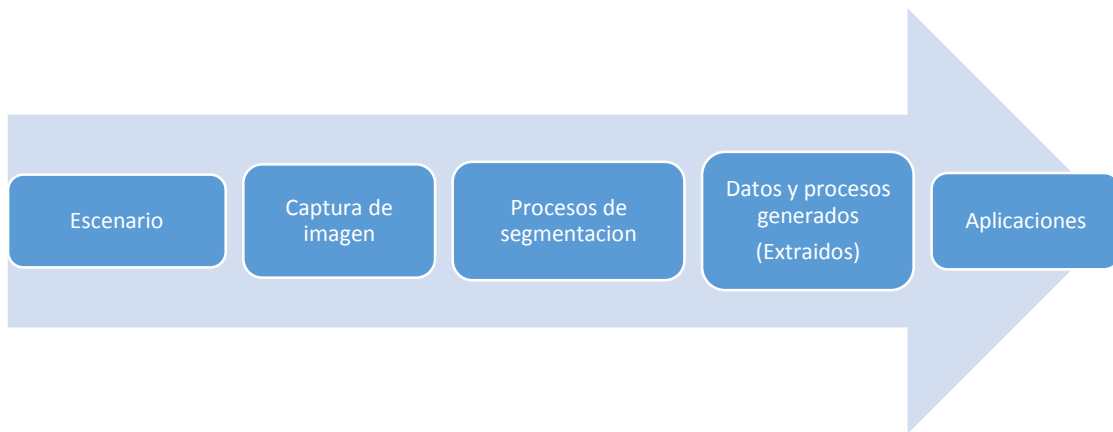
Los sistemas operativos son el medio por el cual interactúa el usuario con el sistema, pudiendo por medio de la interfaz administrar recursos. Los sistemas empotrados han mostrado características de trabajar a tiempo real.

Pérez lo define sistemas embebidos: Software reducido, diseñado para ser usado con un algún sistema operativo particular, pudiendo ser una versión simplificada de un SO (sistema operativo) que podría estar presente o usarse en cualquier computador. (Pérez, 2009, p.11)

## **1.2 Visión artificial**

La visión como un sentido esencial en el ser humano, no exclusivo, ya que su posible ausencia no imposibilita su capacidad intelectual. El significado del término acuñado como visión artificial, radica en brindar a los sistemas programados o autónomos una prestación para procesar datos de un ambiente para buscar interpretarlos en cierto grado, así como para el ser humano identificarlos proporciona información de su ambiente. (Pajares y De la Cruz, 2007, p. 01)

Puede entenderse gráficamente en la (Figura 1-1).



**Figura 1-1 Procesos de sistema de vision artificial .**

Realizado por: Saá H. 2018

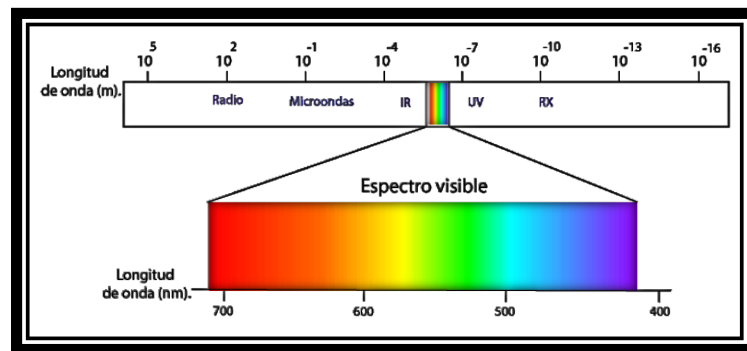
### ***1.2.1 Arquitectura de sistemas de visión artificial***

Se han identificado cinco componentes principales en este tipo de sistemas:

- Dispositivos destinados a la captura: Dispositivo óptico que recepta señales analógicas irradiadas desde el medio a ser capturado en este caso del escenario objeto de análisis.
- Conversión de la señal Análoga a Digital: Esta señal análoga obtenida de la captura, pasa por un proceso de digitalización, donde la información es destinada a diferentes formatos.
- Memoria de almacenamiento: Espacio de memoria donde se archivarán datos, información, archivos.
- Dispositivo de procesamiento: Elemento destinado a procesar paquetes de información para extraer de ellos, datos de interés, en este caso trabajan a la par con la memoria del dispositivo.
- Monitor: Permite la visualizar los procesos que se estén ejecutando.

### 1.2.2 Dispositivo de captura.

Es un elemento destinado a captar la radiación de luz reflejada desde los elementos parte del escenario que se desea captura como imagen, permitiendo convertir una señal analógica en un archivo digital que proporciona una imagen que representa el espectro electromagnético (Figura 1-2) que fue reflejado por el escenario capturado. (Sucar y Gómez, 2003, p. 12)



**Figura 2-1 Espectro de luz Visible**

Fuente: <http://www.aulaclie.es/fotografia-photoshop/graficos/espectro1.gif>

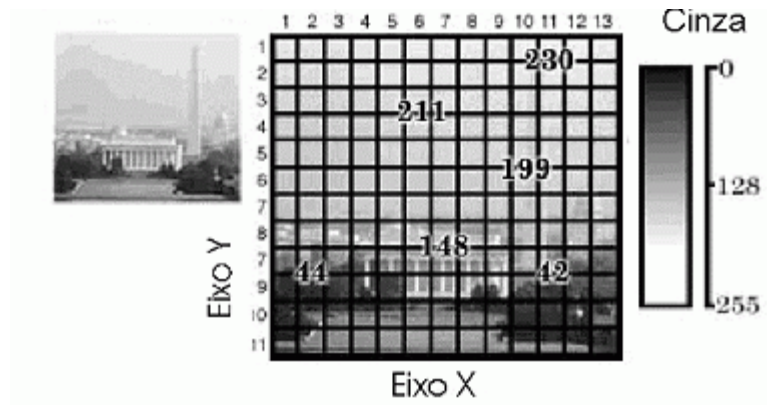
Como dispositivos destinados a capturas se encuentran: cámaras fotográficas, sistemas de escaneo, sensores de distancia, sensores ultrasónicos, sistemas de rayos X, sistemas de tomografías y resonancias, etc.

### 1.2.3 Digitalización de capturas (señal analógica).

Digitalización, proceso destinado a convertir una señal analógica en una archivo digital interpretable, registrando esta exposición a radiación, de los sensores ópticos, en un archivo digital interpretable, archivado en una matriz de señal discreta de M x N píxeles.

### 1.2.4 Representación de una imagen

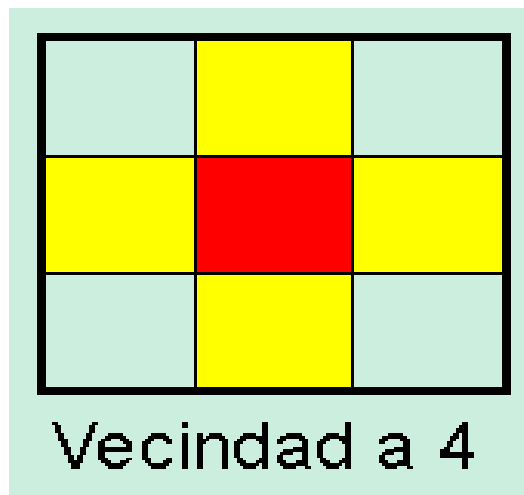
Los píxeles de un archivo de imagen, guardan información que registró la captura, organizándola de tal manera que busque reinterpretar el escenario al que fue expuesto (Figura 1-3).



**Figura 3-1 Sistema de coordenadas de píxeles de una imagen**

Fuente: <http://www.dpi.inpe.br/~carlos/Academicos/Cursos/Pdi/digitalizacao.gif>

Cada píxel se describe tras el valor de su coordenada (X, Y), relacionado con sus píxeles vecinos, horizontales como verticales, este arreglo se lo conoce como vecindad a cuatro. (Figura 1-4).



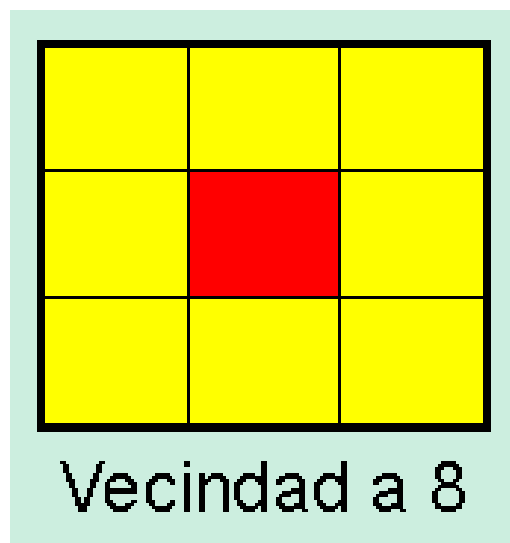
**Figura 4-1 Distribución de píxeles en vecindad a cuatro**

Fuente: <http://www.monografias.com/trabajos107/proceso-formacion-imagenes/28.png>

Los píxeles presentan una distancia unitaria desde su posición original, las coordenadas se asignan por la Ecuación 1-1.

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \quad \text{Ecuación 1-1}$$

En los píxeles distribuidos a los diagonales, se los conoce como vecindad a ocho (Figura 1-5), el cálculo de coordenadas obedece a la Ecuación 1-2.



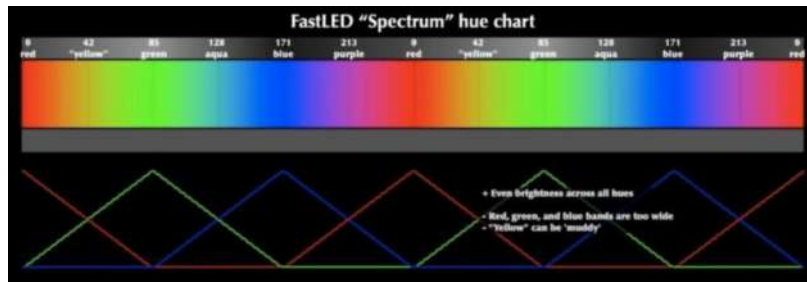
**Figura 5-1 Distribución de píxeles en vecindad a ocho**

Fuente: <http://www.monografias.com/trabajos107/proceso-formacion-imagenes/28.png>

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1) \quad \text{Ecuación 1-2}$$

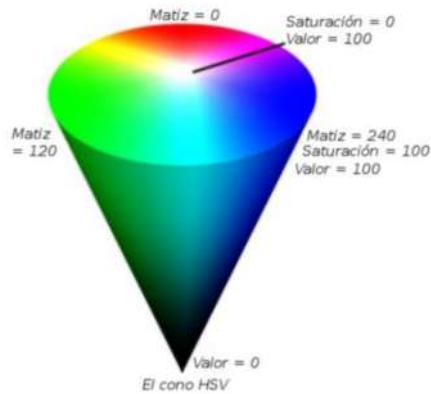
### ***1.2.5 Componentes de las imágenes HSV (Hue/tonalidad, saturation/saturación, intensity/intensidad) a partir de una imagen RGB (Rojo, Verde, Azul).***

El interés en el modelo HSV se basa en que la información de intensidad y color pueden separarse, y el matiz y la saturación (Figura1-6), en particular están muy relacionados con la percepción de color del ojo humano (Figura1-7).



**Figura 6-1** Representación de banda de Matiz vs RGB

Fuente: <https://github.com/FastLED/FastLED/wiki/FastLED-HSV-Colors>



**Figura 7-1** Representación de cono de color HSV

Fuente: <https://migsar.wordpress.com/tag/color-cromatico/>

Cada color de la imágenes RGB pueden representarse con 24 bits (8 bits por componente), variando de 0 a 255 con diferencias de 1, o lo que es lo mismo con diferencias de  $1/255$ . El tratamiento con imágenes HSV es ligeramente diferente, la intensidad de la imagen es semejante a como lo es RGB de 0 a 255 o de 0 a 1. Cada pixel de la imagen en relación al matiz tiene sus valores expresados en grados. Expresando con 8 bits 256 valores en un rango de 0 a 360 grados, con incrementos de  $360/255$ , del mismo modo, los pixeles de saturación pueden verse con valores de 0 a 1 en incrementos de  $1/255$ . (Pajares y De la Cruz, 2007, p.129)

### **1.2.6 Segmentación**

Las imágenes obtenidas de un dispositivo óptico a simple captura, no presenta las mejores prestaciones como para basarnos en la información presente en esas imágenes, y con el fin de mejorar su definición y demás caracteres se necesita usar técnicas que optimicen dichos caracteres.

Segmentación se presenta como un proceso, técnicas mediante las cuales podemos modificar las imágenes en busca de extraer información relevante para poder ser usada en distintos procesos. Este proceso es una pauta aplicable para otras técnicas como detección de contornos, bordes, regiones, continuidad o discontinuidad. (Pajares y De la Cruz, 2007, p.08)

### **1.2.7 Rotación visión artificial**

La aplicación práctica de este algoritmo es el proceso de rotación, usado cuando se requiere posicionar cierta información detectada en un punto determinado para realizar posteriores procesos, como por ejemplo el uso de funciones lógicas and, or, xor, etc; con máscaras para obtener o eliminar ciertas partes de este, procesos de unión o fusión con otras imágenes, etc.

Si introducimos coordenadas polares tendremos:

$$x = r \cos \theta \quad y = r \sin \theta \quad u = w \cos \phi \quad v = w \sin \phi \quad \text{Ecuación 1-3}$$

Donde  $f(x, y)$  y  $F(u, v)$  son ahora  $f(r, \theta)$  y  $F(w, \phi)$  respectivamente, la sustitución directa bien en el caso continuo o discreto produce:

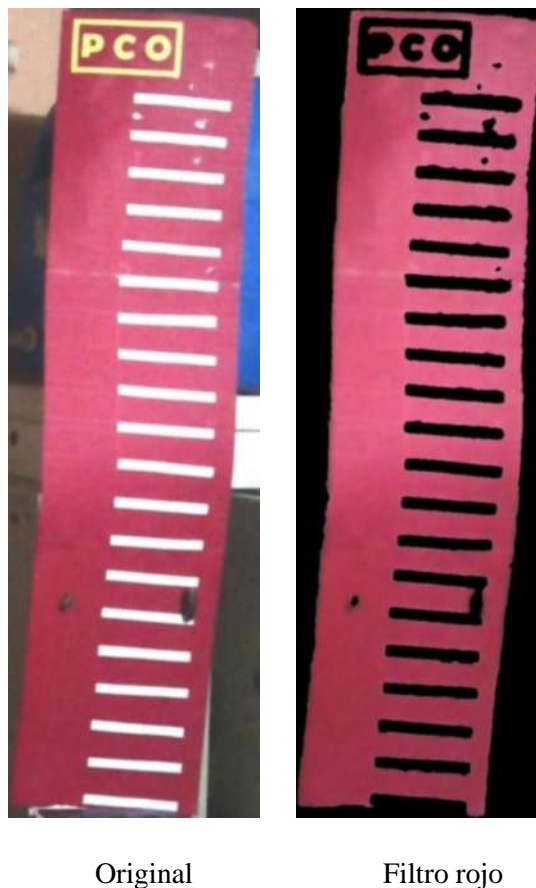
$$f(r, \theta + \theta_0) \Leftrightarrow F(w, \phi + \theta_0) \quad \text{Ecuación 1-4}$$

Rotando  $f(x, y)$  por un ángulo  $\theta_0$  se rota  $F(u, v)$  por el mismo ángulo, como describe la Ecuación 1-4. (Pajares y De la Cruz, 2007, p.27)

### 1.2.8 Extracción capa de color (Operador Bitwise\_and )

Los operadores comparten algo en común: son operadores de "bit a bit". Es decir, operan con números (normalmente), pero en lugar de tratar ese número como si fuera un solo valor, lo tratan como si fuera una cadena de bits, escrita en binario de dos complementos. Un binario de complemento de dos es igual que la representación binaria clásica para enteros positivos, como puede evidenciarse en la Figura 1-8.

El operador "bitwise and" opera con cada bit de la salida es 1 si el bit correspondiente de x AND de y es 1; de lo contrario, es 0. (Python.org, 2011)



**Figura 8-1** Proceso de filtrado de imagen

Realizado por: Saá H. 2018



### 1.2.9 Binarización

Binarizar conlleva un proceso en el que se cuantifica el nivel de gris presente en la imagen a tratar, interactuando con un valor de umbral THRESH\_BINARY. Se compara si el valor de nivel de gris es menor al umbral el pixel toma un valor de cero (negro) (Ecuación 1-3), caso contrario un valor de uno (blanco) (Ecuación 1-4). (Ezqueda, 2002, p.24)

$$S[x, y] = 0(0), E[x, y] \leq T \quad \text{Ecuación 1-3}$$

$$S[x, y] = 1(255), E[x, y] > T \quad \text{Ecuación 1-4}$$

La dificultad de usar binarización radica en buscar un el valor de umbral que mejor se ajuste a las imágenes para asignar valores de entre cero y uno, debido a que al manipular el valor de umbral de forma aleatoria buscando un valor que mejor bina rice la imagen, en este punto se puede optar por buscar el mejor valor que se adapte a la situación. En la Figura 1-9:



(a) Imagen original.



(b) Escala de grises.



(c) Binarización umbral.

**Figura 9-1** Binarización

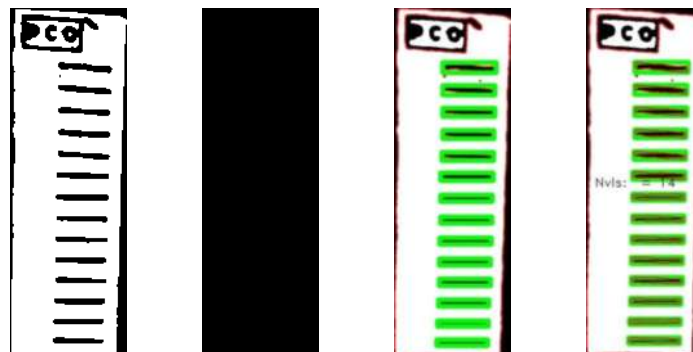
Realizado por: Saá H. 2018

### 1.2.10 Detección de contornos

Detección de bordes o contornos, técnica bastante usada para en el procesamiento de imágenes para extraer regiones, límites de una forma u objeto presentes en la imagen, usados en varios procesos para determinar una región de análisis (contornos). Su función es variar Hue (intensidad) en un pixel, si de una condición de blanco (255) a negro (0) o al contrario, generar una región (contorno).

Se entiende contorno de un objeto, en una imagen, como la región correspondiente al mínimo conjunto de píxeles, diferenciando el contorno de lo que se definiría como fondo. Por lo general estos contornos corresponden a un conjunto de puntos que presentan discontinuidad en relación a los píxeles vecinos que podría ser cambios de textura, matiz o brillo. (Vélez et al., 2003, p. 128)

Al aplicar este algoritmo para identificar contornos, la función muestra los límites del objeto, donde su uso es de gran utilidad para variedad aplicaciones y procesos como en este caso fue implementado para conteo como puede apreciarse en la Figura 1-10.



**Figura 10-1** Detección de contornos

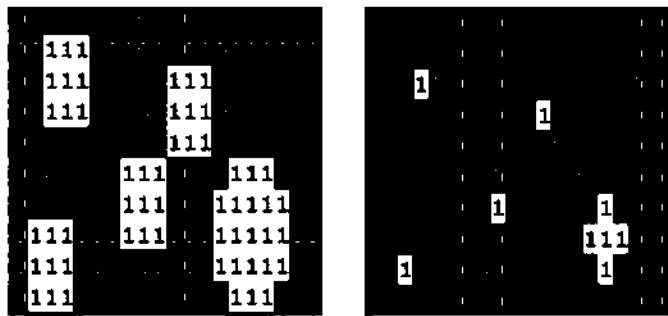
Realizado por: Saá H. 2018

### 1.2.11 Operación de morfología

Los procesos de operación morfológicos permiten darle un realce geométrico a los conjuntos de píxeles que denotan una región de interés.

### 1.2.12 Operación erosión

El proceso morfológico de erosión, modifica la imagen original, eliminando píxeles vecinos, tomando un valor de uno en la imagen original, como se aprecia en la Figura 1-11.



**Figura 11-2** Operación de erosión para imagen binaria.

Fuente: <http://eia.udg.es/~rafa/docencia/morfologia.pdf>

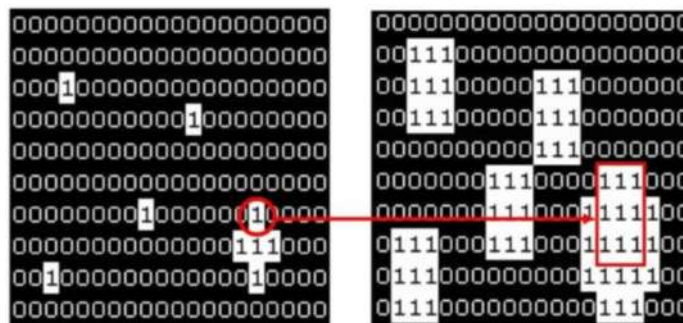
Operación matemática que define la operación de erosión:

$$A \ominus B = \{x / B_x \subseteq A\} \quad \text{Ecuación 1-5}$$

### 1.2.13 Operación de dilatación

Dilatación es una operación donde los pixeles vecinos toman el valor de los pixeles dentro de la matriz de kernel asignada (Figura 1-12). El resultado de esta operación está comprendida como conjunto de elementos, donde al menos un elemento de B es contenido en A (Ecuación 1-6).

$$A \oplus B = \{x / (B^x) \cap A\} \quad \text{Ecuación 1-6}$$



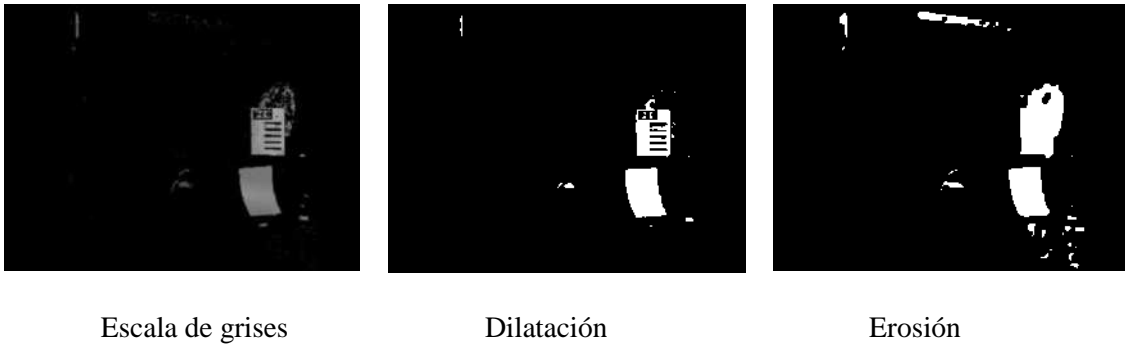
**Figura 12-3** Operación dilatación de imagen binarizada.

Fuente: <http://eia.udg.es/~rafa/docencia/morfologia.pdf>

### 1.2.14 Apertura y cierre

Al aplicar funciones de erosión y dilatación estas transformaciones son irreversibles una vez aplicada a una imagen no puede volverse a obtener los valores originales de la imagen, puede verse en la Figura 1-13, a este proceso de dilatación seguida de erosión se lo conoce como proceso de cierre de una imagen(x), por un elemento estructural (B) se denota como:

$$X \cdot B = (X \oplus B) \ominus B \quad \text{Ecuación 1-6}$$



**Figura 13-4** Proceso de cierre.

**Realizado por:** Saá H. 2018

La operación global de los objetos no se distorsiona, el cierre conecta objetos que están próximos entre sí, rellena pequeños huecos y suaviza el contorno del objeto relleno los pequeños valles mientras que la apertura produce un efecto contrario (Low 1991). (Pajares y De la Cruz, 2007, p. 255)

### 1.3 OpenCV (Open Source Computer Vision)

El uso de OpenCv facilita interacción, del programador con esta librería, gracias a que la compatibilidad que mantiene con el lenguaje Python (que está basado en C++) permite su desarrollo. Además de que Python es un lenguaje conocido y de fácil entendimiento para el programador, permitiendo diseñar y desarrollar el sistema, basado en sus conocimientos y a estar relacionado con el lenguaje de programación.

OpenCV, conjunto de librerías open source diseñadas por Intel, que contiene variados módulos con finalidad a trabajos de visión computarizada. Las librerías se diseñaron con el fin de beneficiarse de las características computacionales para aplicación a tiempo real, que requiere de una exigencia de todos los recursos para el desenvolvimiento de los procesos que conlleva.

### **1.3.1 *OpenCV y Python***

Python es un lenguaje de programación de alto nivel, fácil de comprender su codificación y sintaxis, evitando errores comunes de programación y manteniendo una estructura ordenada.

En este proyecto se escogió trabajar con este lenguaje de programación por la experiencia del programador con su estructura y sintaxis, permitiendo desarrollar el sistema sin mayor complicación.

Tomando en cuenta que todos los algoritmos usados en OpenCV están escritos bajo el lenguaje C++, pueden vincularse con Python, gracias a los interpretes dentro del programa que permiten comunicar los dos lenguajes, lo que permite trabajar con funciones de lenguaje C por medio de Python y poder resolver problemas, generar procedimientos así como algoritmos en tiempo real para visión artificial.

## **1.4 Medición de niveles de ríos**

Al igual que la medición de la lluvia, la medición de los niveles o caudales de un río se realiza por medio de equipos o referencias sencillas, como reglas limnimétricas. La observación y monitoreo de algunos SATI (Sistema de Alerta Temprana para Inundación) comunitarios han implementado medidores por medio del pintado de puntos fijos cercanos al cauce del río.

Idealmente se requiere una escala en cada río y quebrada, así como en sus tributarios. Las escalas hidrométricas no son otra cosa que reglas con las cuales se lee el nivel o altura de las aguas de los ríos y quebradas. Esta regla debe ser lo suficientemente larga para poder medir el nivel cuando el río esté muy alto. Se puede construir dos tipos de escala:

- Escala hidrométrica sujeta a estructuras fijas y
- Escala hidrométrica en serie

### 1.4.1 Lectura de Escala fija

En estos sistemas de nivel se usan estructuras fijas tal como el pilar de un puente, de una estructura o bien un árbol macizo Figura 1-14. El uso de reglas limnimétricas es una opción de bajo costo. Estas reglas se colocan como un sistema de escalas (las reglas son de tipo metálico y numeración en centímetros). (Americanos, 2015)



**Figura 14-5** Escalas para registro de niveles de río

Fuente: <http://copeco.gob.hn/documents/Manual-SATI-21-07-2015.pdf>

## 1.5 Raspberry Pi3.

Computador de placa reducida de bajo costo, libre, diseñado por la fundación Raspberry Pi, trabaja bajo el software, de código abierto que ha desarrollado la empresa denominado raspbian, a su vez brindando la posibilidad de trabajar con otros sistemas operativos también.

Se ha optado por trabajar con esta placa por bajos costes que representa y por la experiencia de interactuar con la placa, con esto saber que se puede desarrollar este sistema.

A medida que ha pasado el tiempo y la placa ha mostrado auge en el mercado, su última versión la Raspberry Pi 3 modelo B mostrada en la Figura 1-15.



**Figura 15-1** Placa de desarrollo, raspberry Pi 3 b

**Fuente:** <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>



### 1.5.1 Hardware placa Raspberry RPi3

Esta versión de placa raspberry se presenta como la tercera generación manteniendo características que se describirán en la siguiente Tabla 1-1:

**Tabla 1-1** Características Placa Raspberry Pi3

<b>Raspberry Pi 3 Modelo B</b>	
<b>UNIDAD CENTRAL DE PROCESO</b>	<ul style="list-style-type: none"><li>· 1.2 GHz</li><li>· Arquitectura de 64 bits</li><li>· Cuatro núcleos ARMv8</li></ul>
<b>JUEGO DE INSTRUCCIONES</b>	<ul style="list-style-type: none"><li>· RISC de 32 bits</li></ul>
<b>MEMORIA (SD RAM)</b>	<ul style="list-style-type: none"><li>· 1 Giga [bite]</li></ul>
<b>PUERTOS USB 2.0</b>	<ul style="list-style-type: none"><li>· 4 Terminales</li></ul>
<b>ENTRADAS DE VIDEO</b>	<ul style="list-style-type: none"><li>· Conector MIPI CSI, que permite alojar el módulo de cámara</li></ul>
<b>SALIDAS DE VIDEO</b>	<ul style="list-style-type: none"><li>· Conector RCA</li><li>· High Definition Multimedia Interface</li></ul>
<b>ALMACENAMIENTO</b>	<ul style="list-style-type: none"><li>· A través de tarjeta Micro SD</li></ul>
<b>CONECTIVIDAD A LA RED</b>	<ul style="list-style-type: none"><li>· Ethernet RJ45</li><li>· Wi-Fi 802.11n</li></ul>
<b>PERIFÉRICOS DE BAJO NIVEL</b>	<ul style="list-style-type: none"><li>· 40 x GPIO</li></ul>
<b>CONSUMO ENERGÉTICO</b>	<ul style="list-style-type: none"><li>· 800 m [A]</li><li>· 4 [W]</li></ul>
<b>FUENTE DE ALIMENTACIÓN</b>	<ul style="list-style-type: none"><li>· 5 [V] Puerto micro USB</li><li>· GPIO Header</li></ul>
<b>DIMENSIONES</b>	<ul style="list-style-type: none"><li>· 85.60 mm x 53.98 mm</li></ul>

Realizado por: Saá H. 2018.

### 1.5.2 Software, Sistemas Operativos de la Placa raspberry pi3

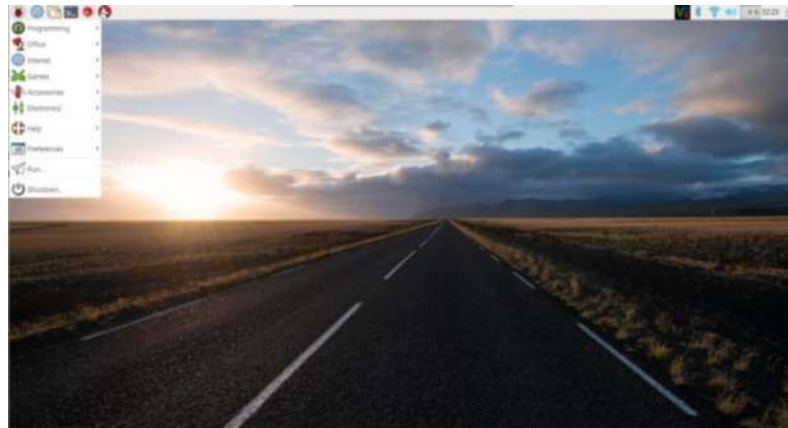
El software usado en la placa raspberry se basa por lo general, en sistemas diseñados con bases en Linux. Algunos de los sistemas con los que trabaja la placa son:

- Raspbian Stretch
  
- Raspbian Stretch Lite

- Windows 10
- Ubuntu Mate
- Linux

### 1.5.3 Raspbian Stretch v2\_4\_5

Entorno visual del escritorio del sistema operativo Raspbian Stretch Figura 1-16:



**Figura 16-6** Raspbian Stretch v2\_4\_5

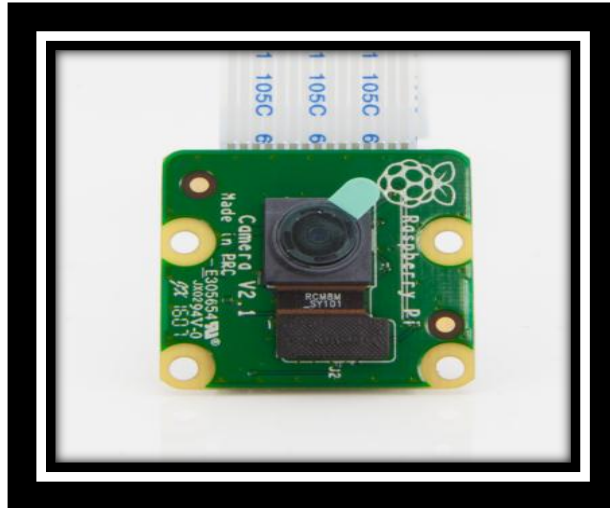
Realizado por: Saá H, 2018.

Raspbian Stretch basa su funcionamiento estructura Debian, la prestación de sus programas básicos son capaces de controlar y a su vez optimizar los recursos de hardware de la placa raspberry pi3.

Al levantarse el SO (sistema operativo) el escritorio gráfico está predispuesto para el ambiente de trabajo, mostrando mayores prestaciones para interactuar con el sistema, raspbian stretch incluye la posibilidad de ingresar a las opciones de configuración del sistema operativo por medio de una interfaz gráfica, la que permite habilitar y configurar preferencias como: habilitación del puerto para el módulo de cámara, comunicación SSH, y demás requerimientos.

### 1.5.4 Módulo de cámara (Pi\_camera )

Módulo de captura cámara pi construido para raspberry Figura 1-17:



**Figura 17-1** Módulo de cámara Raspberry Pi 3 8 Mega\_pixels.

**Fuente:** <https://www.raspberrypi.org/blog/new-8-megapixel-camera-board-sale-25/>

Este dispositivo lanzado por Raspberry, cámara con definición de 8 Mega [píxeles] con alta resolución y calidad en las capturas. Gracias a su compatibilidad con la placa, se conecta al controlador usando el slot CSI permitiendo integración entre los dispositivos. Características en la siguiente Tabla 1-2:

**Tabla 2-1** Características de módulo de cámara

<b>CARACTERISTICAS</b>	<b>DETALLES</b>
<b>RESOLUCIÓN</b>	3280 x 2464 [ píxeles]
<b>CAPTURA DE VIDEO</b>	1080p/30, 720p/60 y 640p x 480p/90
<b>SENSOR DE IMAGEN</b>	Sony IMX219 de 8Mp
<b>DIMENSIONES</b>	25 x 23 x 9 mm
<b>MARCA</b>	Raspberry Pi
<b>PESO</b>	18 gramos

Realizado por: Saá H. 2018.

## **1.6 Internet de las cosas (Internet Of Things)**

Trata de un medio de comunicación entre dispositivos usando el servicio de la web, dándoles un grado de procesamiento microcontroladores, transmisores y receptores para comunicarlos permitiendo interactuar entre ellos. Aprovechando el acceso a la web, la interacción de dispositivos presenta una amplia gama de posibilidades para crear infinidad de aplicaciones, que pueden mejorar o a su vez cubrir variadas necesidades que de alguna manera antes eran complicadas resolver.

Al tratar sobre el tema del internet de las cosas, además de establecer un fácil acceso e interacción con diferentes dispositivos como: cámaras, electrodomésticos, actuadores, sensores y demás, nos permite imaginar y crear nuevas aplicaciones dando solución a variadas necesidades, ya que podemos tener respuesta de los sistemas en menor tiempo y eficiencia.

Debemos tener en cuenta que la interacción de dispositivos mediante la red, representa una nueva evolución de internet, que será un enorme salto en su capacidad para reunir, analizar y distribuir datos que podemos convertir en información, conocimiento y en última instancia, sabiduría. (Evans, 2011, p.01)

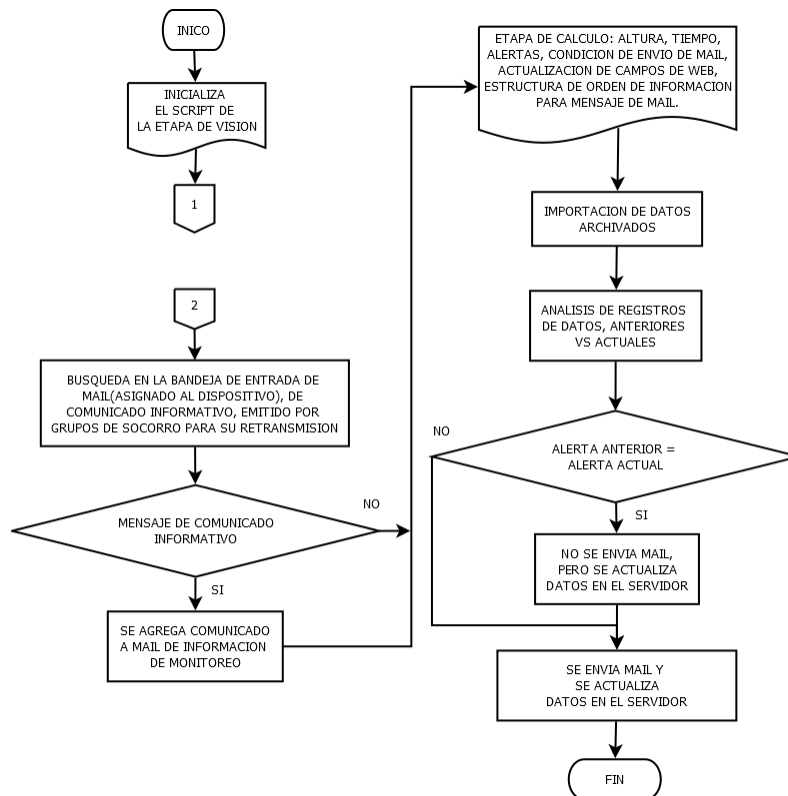
La accesibilidad al Internet permite que dispositivos y sensores puedan enviar grandes cantidades de datos a través de redes fijas e inalámbricas, para luego ser analizadas o usadas para cual sea su fin.

## CAPITULO II

### 2 MARCO METODOLÓGICO

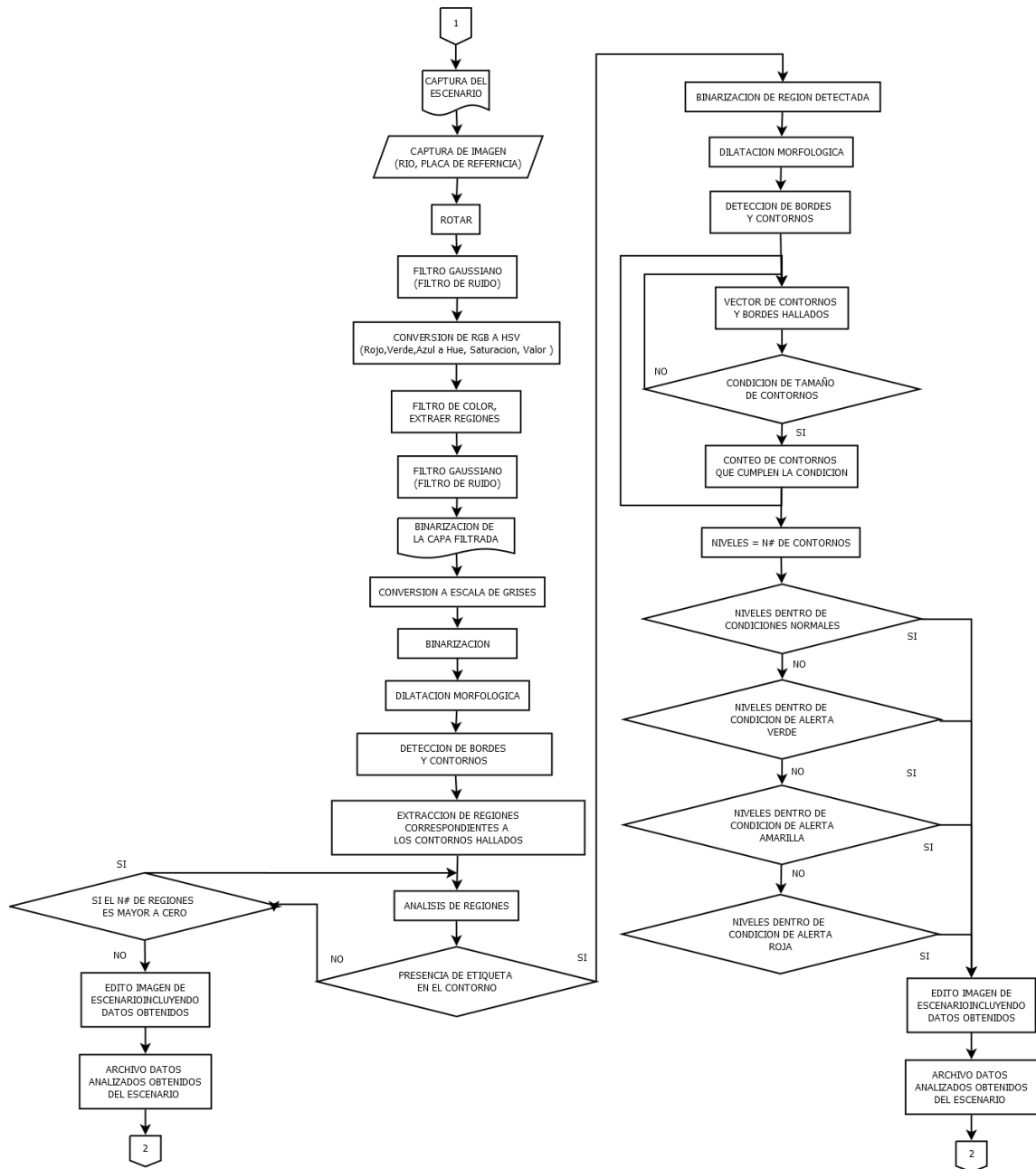
#### 2.1 Secuencia de procesos que realiza el programa.

En siguiente esquema (Flujograma), se describe el funcionamiento del sistema y las etapas que contempla en la primera Figura 2-1, un esquema general de su funcionalidad, seguido por un diagrama de flujo que explica la etapa de visión Figura 2-2.



**Figura 1-2** Diagrama de flujo que describe el sistema

Realizado por: Saá H, 2018.



**Figura 2-1** Flujo grama describe el algoritmo de visión implementado

Realizado por: Saá H, 2018.

## **2.2 Adquisición de imágenes**

El sistema requiere para su análisis de imágenes del lugar para evaluar la situación que pueda estarse generando, sobre la cual se realizaran diversos procesos para obtener información del escenario monitoreado, usando como elemento principal una cámara para captura de imágenes del escenario que requerirá el sistema de procesamiento.

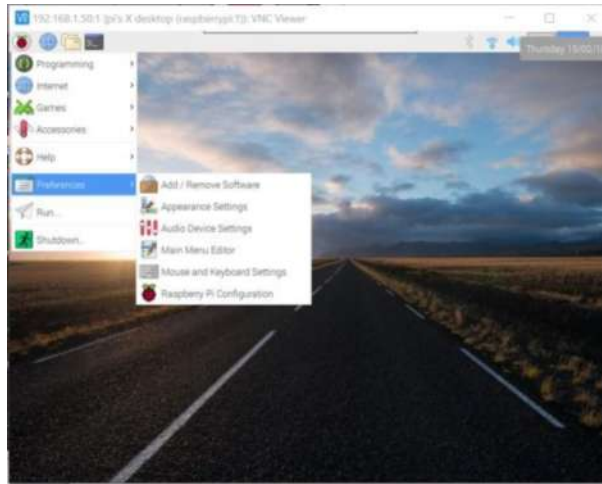
### **2.2.1 Selección del dispositivo de captura (Cámara)**

El sistema, al usar una placa raspberry Pi para la cual el fabricante ha diseñado, una cámara que facilita, integración entre los dos dispositivos. Dejando de lado el posible uso de otro tipo de cámara que a su vez podría comprometer el espacio que se busca priorizar para el dispositivo, así como la fidelidad de comunicación con la placa y sus programas como es en este caso OpenCV. Debido a estas razones se ha decidido trabajar con la cámara que provee el mismo fabricante de la raspberry pi3.

### **2.2.2 Captura de Escenario (Imagen)**

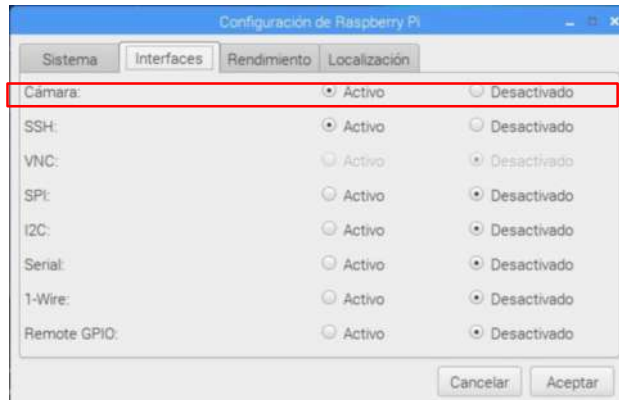
Al haber conectado el módulo de la cámara a través del puerto SCI, se debe activar el dispositivo mediante pocos pasos:

En la figura se muestra las opciones por las cuales navegar, para activar la cámara, la ventana de configuración para el dispositivo en el sistema operativo de raspbian, es bastante sencilla. Se debe dirigir hacia configuración del sistema (Preferences) Figura 2-3, luego a la configuración de la placa (Raspberry Pi Configuration) y activar la opción (cámara) de la interfaz Figura 2-4.



**Figura 3-2** Ventana para activar recurso de cámara

Realizado por: Saá H, 2018.



**Figura 4-2** Ventana para activar recurso de cámara

Realizado por: Saá H, 2018.

El módulo ofrece la posibilidad de modificar características dependiendo de la necesidad del proyecto, características como: resolución, brillo, contraste, etc.

La resolución que se usada en el proyecto, para capturar imágenes, fue de 3280 x 2464 con una relación de aspecto de 4:3 la máxima que la cámara permite, buscando así obtener una captura con la mejor definición para procesarla. Recalcar que el mismo echo de usar la mejor resolución de imagen las mejores características, se traduce en que el archivo de imagen generado, tendrá gran espacio en memoria, aumentando el tiempo para procesar la captura y demás algoritmos.

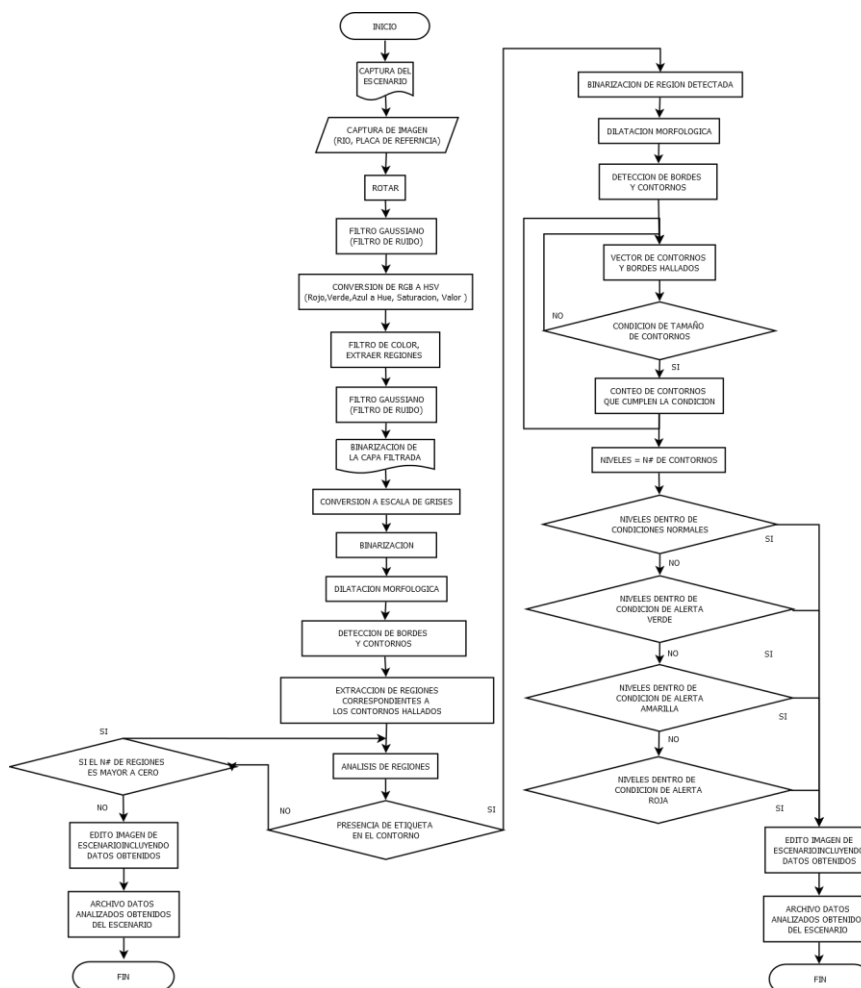
Los comandos `start_preview` y `stop_preview`, son usados para inicializar la cámara y cerrar el modulo, luego de inicializada la cámara se usa el comando `.capture()`, para archivar la imagen



capturada, guardando el archivo con una extensión asignada a la imagen. Eligiendo para el proyecto el formato jpg (Joint Photographic Experts Group), este formato de imagen nos permite obtener las mejores prestaciones de definiciones de imagen, que es lo que buscan, sistemas destinados a trabajar con visión artificial.

### 2.3 Etapa para detección de placa de referencia

Luego del proceso para obtener la imagen, el sistema continua con varios procesos para determinar el lugar en donde se encuentra la placa, la región con la cual se va a trabajar para extraer datos para los procesos que intervienen en el sistema continuación en la Figura 2-5 se presenta un flujo grama que explica de mejor manera el desarrollo del sistema.



**Figura 5-2** Flujo grama detección de placa de referencia

Realizado por: Saá H. 2018.

Este diagrama muestra paso a paso, la secuencia que mantiene el algoritmo del sistema implementado, para extraer datos desde la imagen de la captura del escenario.

### **2.3.1 Pre-procesamiento de imagen (Escenario)**

Se aplica un subproceso para identificar el contorno que contiene la placa, la región que será posteriormente analizada con el fin de determinar la placa de referencia que contiene los niveles.

### **2.3.2 Lectura de imagen.**

Importo la imagen que fue capturada del escenario, usando el comando `imread`, asignándole un nombre al archivo importado (`img1`), seguido del comando que da lectura a la imagen almacenada en memoria especificando nombre y extensión (`cv2.imread ('img_fnt. extensión')`), código:

```
cv2.imread ('01_img_ambt.jpg')
```

```
img1 = cv2.imread ('01_img_ambt.jpg')
```

### **2.3.3 Giro de la imagen capturada (Escenario)**

En este punto debo recalcar que depende de cómo fue instalado el dispositivo en el lugar desde el cual monitoreara el escenario, para este caso el dispositivo necesita rotar la imagen original 90 grados, usando la función, `rows, cols = img1.shape[:2]`

### **2.3.4 Filtro de Gauss (Filtro para eliminación de ruido)**

En esta parte del proceso se busca eliminar el ruido que se encuentre en la imagen modificando muchas veces la iluminación de la captura. Se usó un filtro Gaussiano, que aplica una distribución gaussiana en relación a los píxeles vecinos. La ecuación 1-2 muestra la transformada de Gauss considerando la media igual a cero.

$$T(x, y) = e^{-\frac{(x^2+y^2)}{2\pi\delta^2}} \quad \text{Ecuación 1-2}$$

Aplicar suavizado a una imagen dependerá del tamaño de la máscara que se use para este filtrado, el tamaño de la máscara es de N x M, al aplicar una máscara con valores de 3 x 3, generando un efecto de suavizado y eliminación de ruido, poco perceptible ante nuestra mirada. Como puede verse en la figura 2-6 donde a la imagen original se le aplicó un filtro Gaussiano.

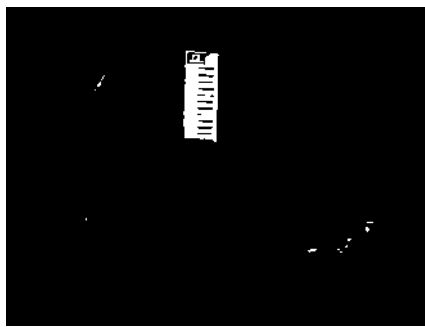
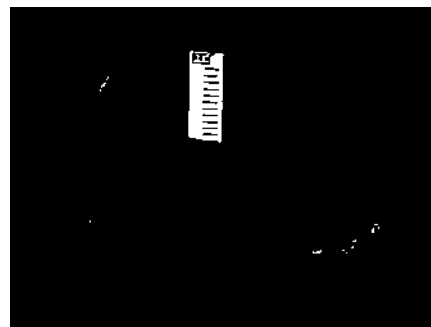


Imagen Original



Filtro Gaussiano

**Figura 6-2** Imagen supresión de ruido, filtro de gauss.

Realizado por: Saá H. 2018.

Para aplicar este filtro se hizo uso del comando: `cv2.GaussianBlur(img_fnt,(m x n),0)`. Para esta función los argumento que requiere son la imagen, el tamaño de la matriz de transformación y la desviación estándar que por general toma un valor de cero, con el valor de cero se permite al sistema usar el valor que optimice el filtrado. # Aplicar suavizado Gaussiano #-Aplico filtro para eliminar ruido con un kernel de 5\*5(impar), o(para que el sistema halle la media)

```
frame_mask_HSV = cv2.GaussianBlur(imagen_filtrada1, (5,5), 0)
```

### 2.3.5 Conversión de RGB a HSV

Convierte de una combinación de tres capas RGB (rojo, verde, azul), a un valor de pixel que describe una combinación HSV (Hue/tonalidad, saturation/saturación, intensity/intensidad).

### 2.3.6 Filtro de color

El proceso que se ejecuta en este punto usa la función “bitwise and”, que compara cada pixel con el rango de color requerido, de cumplir estar dentro del rango de color (rojo HSV), el bit se mantiene y no es modificado, de no estar dentro del rango el pixel toma el valor de 0 (ausencia de color), que describe un pixel de color negro, se puede evidenciar de mejor manera en la Figura 2-7:

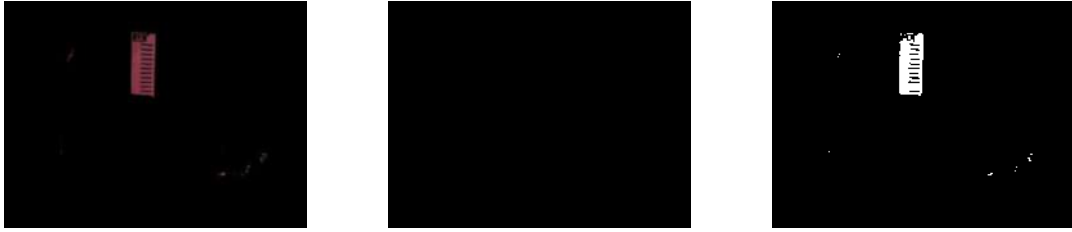


**Figura 7-2** Filtro de color rojo

Realizado por: Saá H. 2018.

### 2.3.7 Binarizado

Se debe hacer variadas pruebas para identificar el valor de umbral de binarización, que mejor define los rasgos de interés de placa para posteriores procesos, tomar un valor cual sea provocaría errores en la etapa de segmentación, afectado por los valores de brillo, contrastes, definición de los contornos, el uso del algoritmo de THRESH\_BINARY deja de lado este problema porque intrínsecamente el algoritmo elige el valor que mejor se ajuste para un umbral óptimo basado, en los valores de gris que presentan los pixeles. Estos resultados se pueden distinguir en la Figura 2-8.



**Figura 8-2** Binarización

**Realizado por:** Saá H. 2018.

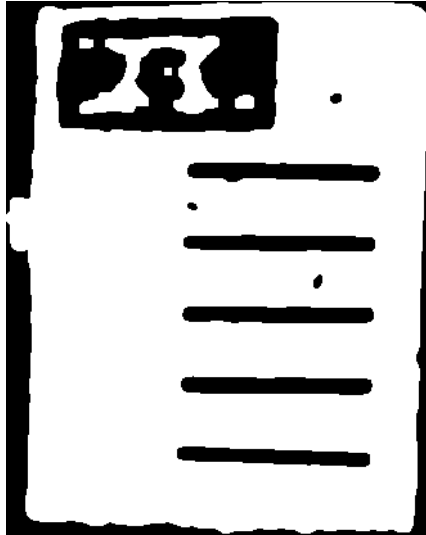
El uso del comando `cv2.threshold("Imagen")`, asigna valores de blanco (255) o negro (0) comparando cada píxel con los rangos entre 0 y 255. Usando el código de binarización, `dst0a=cv2.threshold(imagen_gray_ama, 10, 255, cv2.THRESH_BINARY)`.

### 2.3.8 *Morfología*

Las operaciones morfológicas se encargan del realce de la geometría de los píxeles, fundamentándose en la teoría de conjuntos tomando a una imagen binaria.

### 2.3.9 *Dilatación de la morfología, closing(cerrada)*

En esta operación, expongo la imagen a una dilatación con el código, `cv2.morphologyEx(img_gray_gri_plac, cv2.MORPH_CLOSE, kernel)`, permitiendo que el kernel tenga un valor de 5x5 que quiere decir que tomara una matriz de 5 por 5 de los píxeles de la imagen y analiza los valores que contemplan los píxeles en esa matriz y luego evalúa un valor promedio en relación a los píxeles vecinos y modifica los valores de los píxeles (Figura 2-9), una vez usado este código, no se puede restaurar a valores anteriores los píxeles.



**Figura 9-2** Dilatación morfológica CLOSING

Realizado por: Saá H. 2018.

### ***2.3.10 Detección de bordes y contornos Canny***

Las prestaciones que brinda el algoritmo Canny para detectar bordes o contornos (objetos). Permiten hallar los contornos presentes en la imagen a la vez que el tamaño de cada contorno.

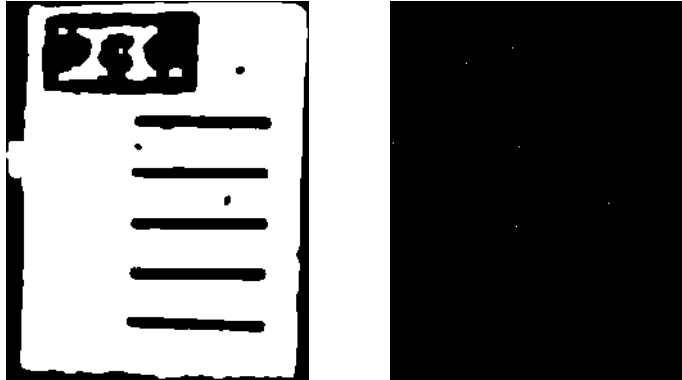
Este algoritmo entre los métodos de detección de contornos es el más usado, basa su funcionamiento en máscaras de convolución y la primera derivada. Los puntos que definen los contornos, son las zonas de píxeles en las que se presentan cambios drásticos de nivel. Para el tratamiento de imágenes, se trabaja con áreas de píxeles, en un ambiente de desarrollo discreto, es ahí que el algoritmo Canny usa máscaras, las cuales representan aproximaciones en diferencias finitas. (Valverde, 2011, p.01)

Desde la imagen extraída del filtro de color después de haber sido binarizada, se aplica el comando:

```
# Detectamos los bordes con Canny
```

```
canny = cv2.Canny(frame_canny, 50, 150)
```

Devolviendo los contornos en que no varía la intensidad de píxeles, se puede ver en las imágenes (Figura 2-10).



**Figura 10-2** Ventana de activación de la cámara

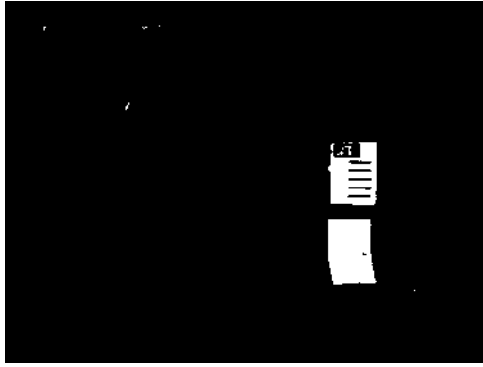
Realizado por: Saá H. 2018.

### ***2.3.11 Recortar imagen***

Con la función Canny aplicada anterior mente, en un vector discreto se mantiene archivado el número de contornos así como también sus dimensiones en píxeles, que me servirán para halla el contorno más grande para analizar que sea la placa de referencia que es de nuestro interés.

### ***2.3.12 Selección del contorno más grande***

Las comparaciones consecutivas entre el tamaño de cada contorno hallado, nos permite indirectamente organizar, para identificar posteriormente el contorno más grande (Figura 2-11), que será de interés para su análisis.



Contornos hallados (Canny)

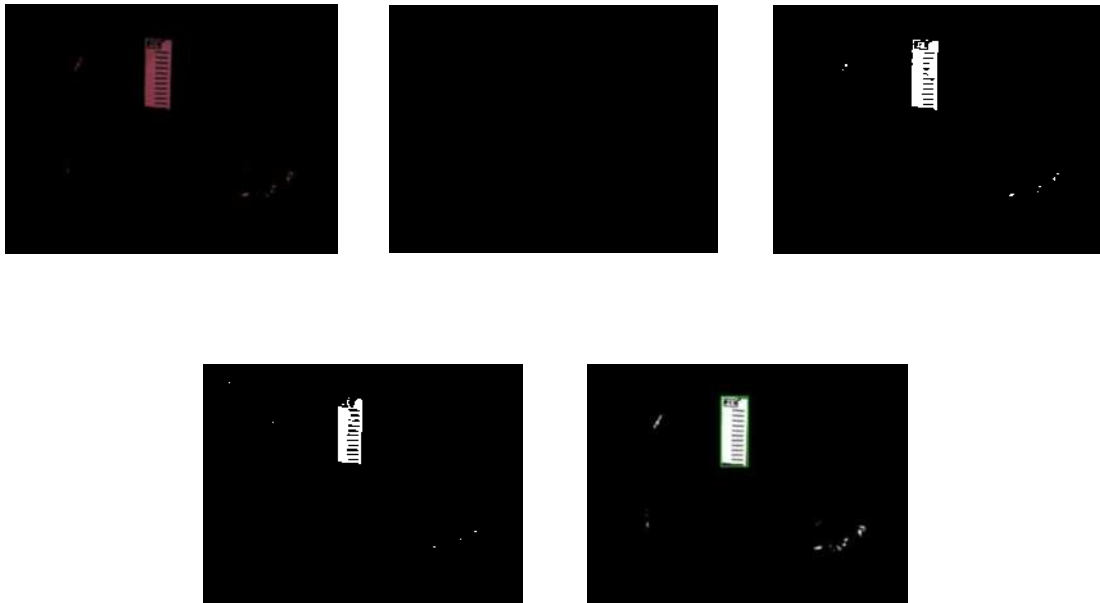


Detección de los contornos mas grandes

**Figura 11-2** Detección del contorno más grande.

Realizado por: Saá H. 2018.

Definir, limitando el lugar donde se ubica la placa de referencia como se muestra en la Figura 5-2 en la zona delimitada en verde (Figura 2-12).



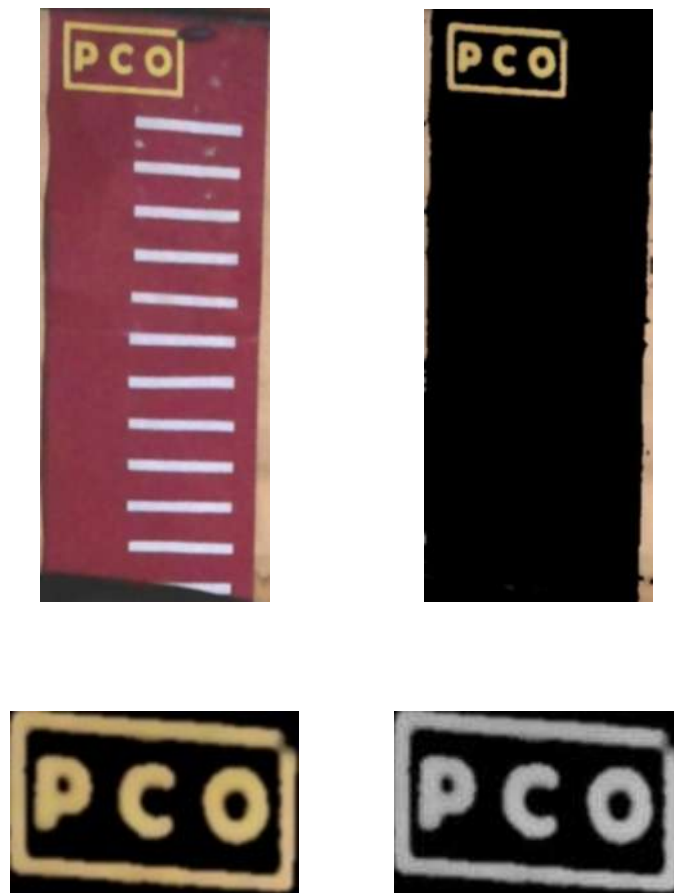
**Figura 12-2** Detección del contorno más grande.

Realizado por: Saá H. 2018.



### 2.3.13 Análisis de etiqueta

En este punto al haber determinado los contornos más grandes el contorno, nos permite recortar de la imagen original la región que dicho contorno representa, para analizar, la presencia o ausencia de la etiqueta amarilla, que determina que en verdad es la placa de referencia (Figura 2-13), si la región que fue seleccionada primero, no contiene la etiqueta, se analiza el siguiente contorno.



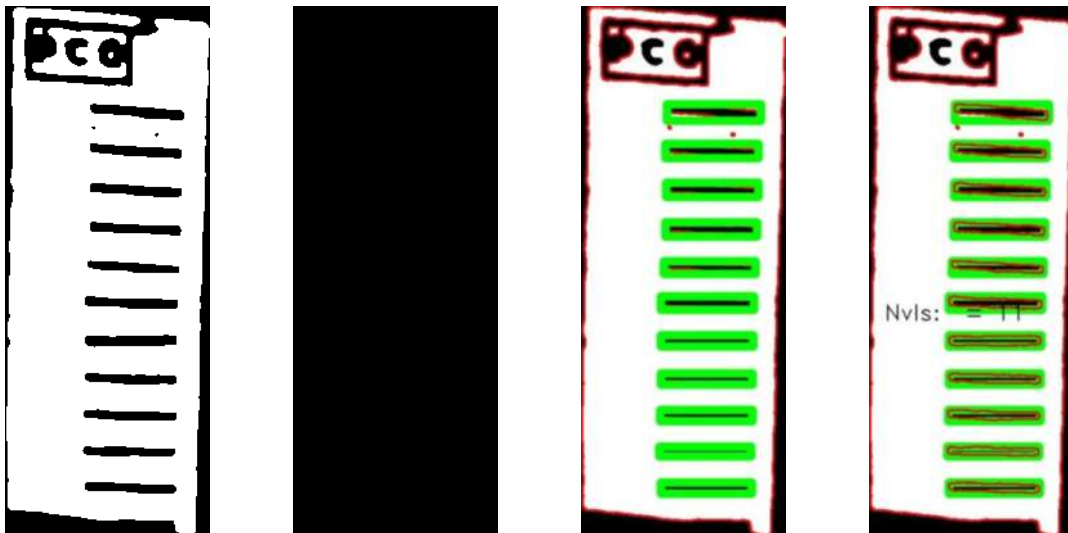
**Figura 13-2** Análisis de etiqueta

Realizado por: Saá H. 2018.

Ya una vez identificado el contorno que contiene la etiqueta, este contorno representa el área de interés para el conteo de niveles que nos permitirán identificar la actividad fluvial que se esté presentando.

### 2.3.14 Conteo de niveles

De la región de interés hallada, que fue detectada en la etapa de binarización, la someto al algoritmo de conteo y reconocimiento de contornos y bordes (Canny), y en función al tamaño que determine para los niveles que podrán identificados, el sistema genera un conteo, al comparar cada región hallada con el tamaño asignado dentro del programa (Figura 2-14).



**Figura 14-2** Conteo de niveles a analizar

Realizado por: Saá H. 2018.

### 2.3.15 Respaldo de datos identificados

De las características halladas anterior mente como, presencia de placa (contorno que contiene la etiqueta), niveles hallados, con estos datos analizo la situación que se ha presentado.

### 2.3.16 Análisis de datos

Para determinar las alertas a generar en una situación de inundación, gestión de riesgos determina a que altura se describiría cada alerta tomando en cuenta que cada altura que describa el rio cubrirá ciertas áreas que determinan un interés de riesgo, he ahí la necesidad de la referencia de la altura que va tomando el rio e ira cubriendo cada nivel, pueden presentarse cinco situaciones que describan lo que se puede interpretar del monitoreo (Tabla 2-1):

**Tabla 1-2** Referencias de alertas, rango de niveles.

SITUACIONES	NIVELES	ALERTAS	CONDICION
1	17-15	Niveles Normales	0
2	14-10	Alerta Verde	1
3	9-5	Alerta Amarilla	2
4	4-0	Alerta Roja	3
5	Ausencia de placa	Error de Placa	4

**Realizado por:** Saá H. 2018.

Al haber identificado, la situación que presenta, saber el número de niveles que se pueden identificar en la placa, que tipo de alerta describe la situación, así como el tipo de condición que me ayudara, para cálculos en los siguientes procesos del programa, continuo con el análisis de la situación.

En un principio antes de correr el programa, registro, la distancia entre niveles, así como el número de niveles que contemplara toda la placa, la altura hasta al marco inferior de la placa, como la altura que cubre hasta el marco superior de la placa (Tabla 2-2).

**Tabla 2-1** Datos de referencia para calculo

<b>Parámetro</b>	<b>Valor</b>
Distancia entre nivel	4 centímetros
Altura marco inferior	1,24 metros
Altura marco superior	2 metros
Límite superior Alerta verde	10
Límite superior Alerta amarilla	5
Límite superior Alerta Roja	0

Realizado por: Saá H. 2018.

### 2.3.17 *Altura de nivel de rio*

En este punto calculo la altura que ha alcanzado el nivel del rio:

$$H \text{ nivel de rio} = ((\text{Distancia entre niveles}) * (\# \text{ numero de niveles identificados}) + (\text{Altura descrita hasta el marco superior de la placa}))/100 \text{ [metros]}$$

Ejemplo:

$$H \text{ nivel de rio} = \frac{(4) * (14) - (200)}{100} \text{ [ metros]}$$

$$H \text{ nivel de rio} = \frac{(4) * (14) - (200)}{100} \text{ [ metros]}$$

La altura en la que se encuentra:

$$H \text{ nivel de rio} = 1.44 \text{ [ metros]}$$

### 2.3.18 Situación Actual de Rio

Luego para determinar si ha crecido o está bajando el nivel del rio:

$$\text{Situacion de Actividad de Rio} = (H \text{ nivel de rio anterior}) - (H \text{ nivel de rio actual})$$

(*H nivel de rio anterior*): Este datos se mantiene archivado en un documento, es un respaldo de la situación anterior, del monitoreo del ciclo anterior, a este nuevo ciclo de análisis.

Condiciones de situación:

*dif H*: Diferencia de altura.

Si  $dif H > 0$  Entonces: La actividad está disminuyendo.

Si  $dif H < 0$  Entonces: La actividad está aumentando.

Si  $dif H = 0$  Entonces: La actividad se mantiene.

Ejemplo:

$$\text{Situacion de Actividad de Rio (dif H)} = (1[m]) - (1.44[m])$$

$$dif H = -0.44[m] \quad \text{El nivel de rio esta aumentando.}$$

### 2.3.19 Taza de cambio de nivel de rio

El sistema se ejecuta automáticamente cada minuto, entonces todos los datos que procese y adquiero de la etapa de visión artificial, serán adquiridos cada minuto entonces puedo definir, cada cambio que registre por minuto, [datos/minuto], así como cada altura identificada, [altura/minuto], con esto poder predecir que si de mantenerse a esa tasa de cambio, cuando aumente, calcular en que tiempo la situación podría evolucionar hacia la siguiente alerta de emergencia.

**Tabla 3-2** Límites superiores de rangos de nivel

Parámetro	Valor
Límite superior Alerta verde	10
Límite superior Alerta amarilla	5
Límite superior Alerta Roja	0

Realizado por: Saá H. 2018.

**Tabla 4-2** Datos de referencias, para cálculo de información

Parámetro	Valor
Distancia entre nivel	4 centímetros
Altura marco inferior	1,24 metros
Altura marco superior	2 metros
Límite inferior Alerta verde	14
Límite inferior Alerta amarilla	9
Límite inferior Alerta Roja	4

Realizado por: Saá H. 2018.

**Tabla 5-2** Datos de referencia de alertas

SITUACIONES	NIVELES	ALERTAS	CONDICION
1	17-15	Niveles Normales	0
2	14-10	Alerta Verde	1
3	9-5	Alerta Amarilla	2
4	4-0	Alerta Roja	3
5	Ausencia de placa	Error de Placa	4

Realizado por: Saá H. 2018.

Ejemplo:

El sistema ha identificado 14 niveles.

$$H \text{ nivel de rio} = \frac{(4) * (14) - (200)}{100} \text{ [ metros]}$$

$$(dif H) = (1[m]) - (1.44[m])$$

$$dif H = -0.44[m] \quad \text{El nivel de rio esta aumentando.}$$

Se ha identificado que el cambio en relación a la situación anterior registra una tasa de cambio de 0.44 [mt], 14 niveles, en una situación de alerta verde.

La siguiente alerta de riesgo a la que podría evolucionar la situación, es hacia alerta amarilla, que será considerada cuando cubra los 9 niveles de la placa de referencia.

Ejemplo:

- La distancia que faltaría por cubrir para llegar a la próxima emergencia:

$$\text{Distancia hacia la siguiente alerta} = \left| \left( \frac{(4) * (9) - (200)}{100} \right) \right| - \left| \left( \frac{(4) * (14) - (200)}{100} \right) \right| \text{ [ metros]}$$

$$\text{Distancia hacia la siguiente alerta} = |-1.64| - |-1.44| \text{ [ metros]}$$

$$\text{Distancia hacia la siguiente alerta} = |-1.64| - |-1.44| \text{ [ metros]}$$

$$\text{Distancia hacia la siguiente alerta} = |0.2| \text{ [ metros]}$$

Si la tasa de cambio se definió en: 44 [ctm/min] en relación al análisis del ciclo anterior.

Y la diferencia hacia el próximo nivel de alerta esta en 20 centímetros.

$$\text{Tiempo estimado: } \frac{(20 \text{ centímetros}) * (60 \text{ segundos})}{(44 \text{ centímetros tasa de cambio})} [\text{segundos}]$$

$$\text{Tiempo estimado: } 27.27 [\text{segundos}]$$

De seguir cambiando la actividad del rio a una tasa de 44 centímetros por minuto, la situación podrá evolucionar a la siguiente alerta en 27,27 segundos.

### **2.3.20 Envío de información**

La información extraída de la situación de monitoreo, es reenviada, retransmitida, para mantener informada a la población así como a los diferentes grupos de socorro para que puedan tomar decisiones para atender a los posibles afectados, a más de emitir comunicados de información, para que la gente pueda estar prevenida.

### **2.3.21 Información vía Mail**

El monitoreo se realiza cada minuto y cada minuto tengo datos referentes a la situación que se esté generando, pero la información no puede ser enviada en cada monitoreo, debido a que más que informar sería una molestia recibir mail de información cada minuto, tomando en cuenta esto, se diseñó el sistema para enviar información mail, solo dos casos:



Primero en el caso de que la situación, cambie de alerta, la Tabla 2-7 servirá de referencia para el ejemplo:

**Tabla 6-2** Alertas y rangos

SITUACIONES	NIVELES	ALERTAS	CONDICION
1	17-15	Niveles Normales	0
2	14-10	Alerta Verde	1
3	9-5	Alerta Amarilla	2
4	4-0	Alerta Roja	3
5	Ausencia de placa	Error de Placa	4

Realizado por: Saá H. 2018.

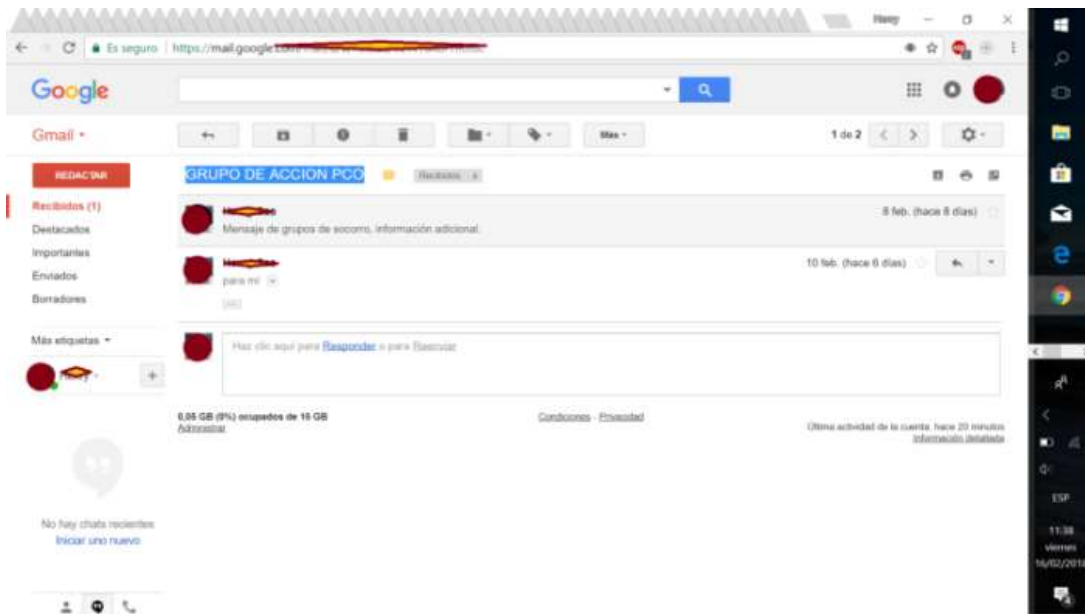
Ejemplo:

Si la situación del ciclo anterior determino que se encontraba en una alerta verde, en este ciclo la situación evoluciono hacia una alerta amarilla se envía mail de información y se actualiza.

Segundo en el caso de que se haya emitido cualquier comunicado de los grupos de socorro, se envía información vía mail, junto con toda la información que se haya generado en ese momento, en esa situación de monitoreo.

### **2.3.22 Retransmisión de comunicados**

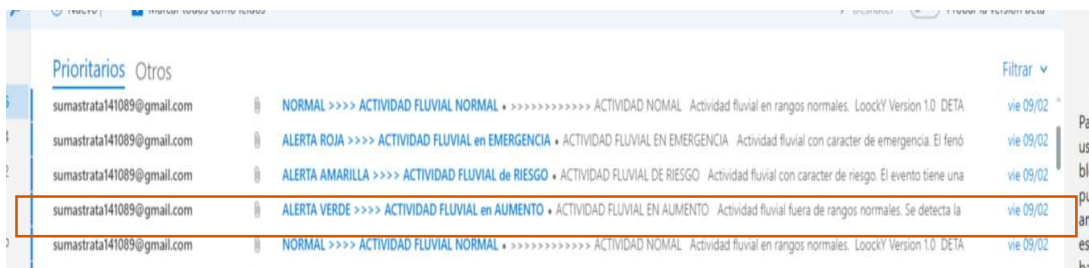
El sistema se enlaza a la cuenta de mail asignada para el dispositivo, analizando la bandeja de entrada, y verifica si existe algún comunicado procedente de los grupos de socorro filtrando con búsqueda entre las etiquetas de los mensajes en la bandeja, al hallar una etiqueta similar a “GRUPO DE ACCION PCO” (Figura 2-15), y lo estructura en texto, para posterior retransmitirlo hacia los usuarios, así como hacia el servidor web.



**Figura 15-2** Bandeja de entrada mail, del dispositivo.

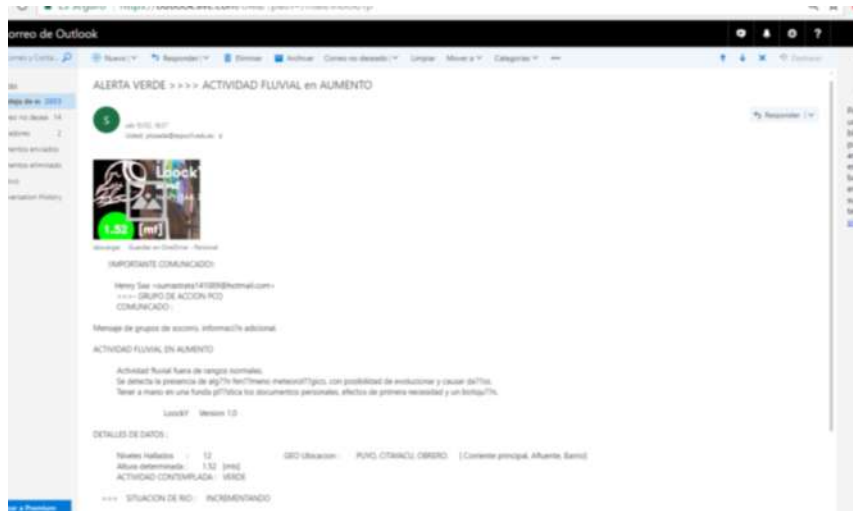
Realizado por: Saá H. 2018.

El mail (Figura 2-16) consta de información así como también imagen del río monitoreado, y de haber un comunicado también estará adicionado en el cuerpo del mail (Figura 2-17).



**Figura 16-2** Mail, bandeja de entrada (Usuarios)

Realizado por: Saá H. 2018.



**Figura 17-2** Mensaje informativo de mail enviado por el dispositivo

Realizado por: Saá H. 2018.

### 2.3.23 Información a Servidor WEB

En un host, en este caso para el desarrollo del proyecto de tesis use un host local, en el cual se sube toda la información que ha generado el análisis del monitoreo, vease en la Figura 2-18.



**Figura 18-2** Pagina WEB de monitoreo

Realizado por: Saá H. 2018.

El fin de poder informar mediante una página, es brindar una alternativa para aquellos que deseen conocer la situación que se esté presentando en dicho sector monitoreado, pudiendo así informar a gente que aunque no viva en el lugar sepa sobre la situación.

## CAPITULO III

### 3 DISCUSIÓN Y ANÁLISIS DE RESULTADOS.

Se mostrara los resultados, que se obtenidos de este prototipo de sistema de alerta temprana, proyecto de titulación, respaldando el funcionamiento óptimo complementado con análisis que los respalde.

Brindar y mantener información de cualquier situación, es importante, porque podemos planificar cualquier tipo de decisión, en relación a datos interés, y tener una alternativa a la eventualidad que pueda darse.

Mediante este proyecto, se busca el desarrollo y vinculación de campos tecnológicos para hallar soluciones a problemas sociales de nuestro medio figura 3-1.



**Figura 1-3** Disposición de dispositivos

Realizado por: Saá H. 2018.

### **3.1 Pruebas de funcionamiento del prototipo**

Las pruebas del prototipo se subdividen en tres partes:

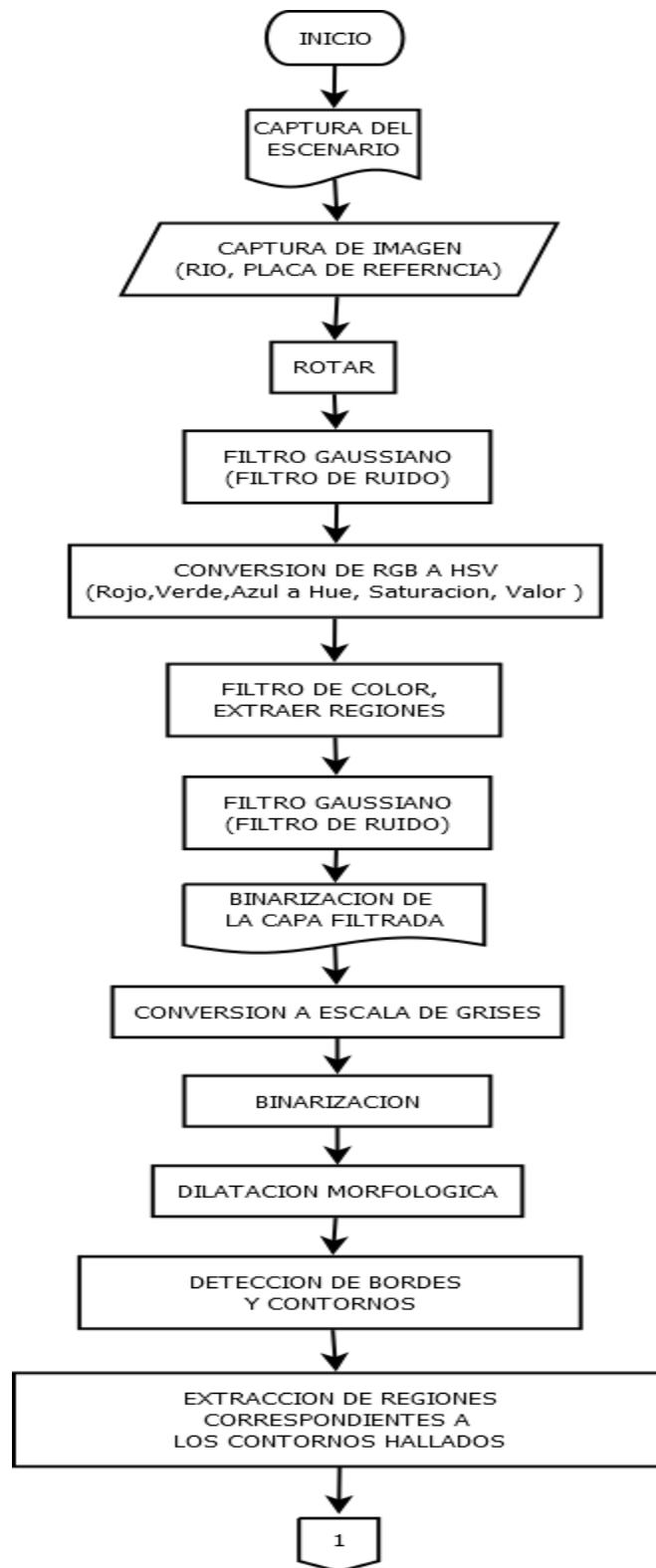
- **Pruebas de detección de placa de referencia**
- **Pruebas de conteo de niveles**
- **Pruebas de envío de informes (Mails)**
- **Pruebas de actualización de datos en la página web**

#### ***3.1.1 Pruebas de detección de placa de referencia***

Para identificar la presencia o ausencia de la placa de referencia, en la imagen capturada del escenario, el sistema realiza el proceso descrito en la (Figura 3-1 Algoritmo de detección de placa parte a) y (Figura 3-2 Algoritmo de detección de placa parte b).

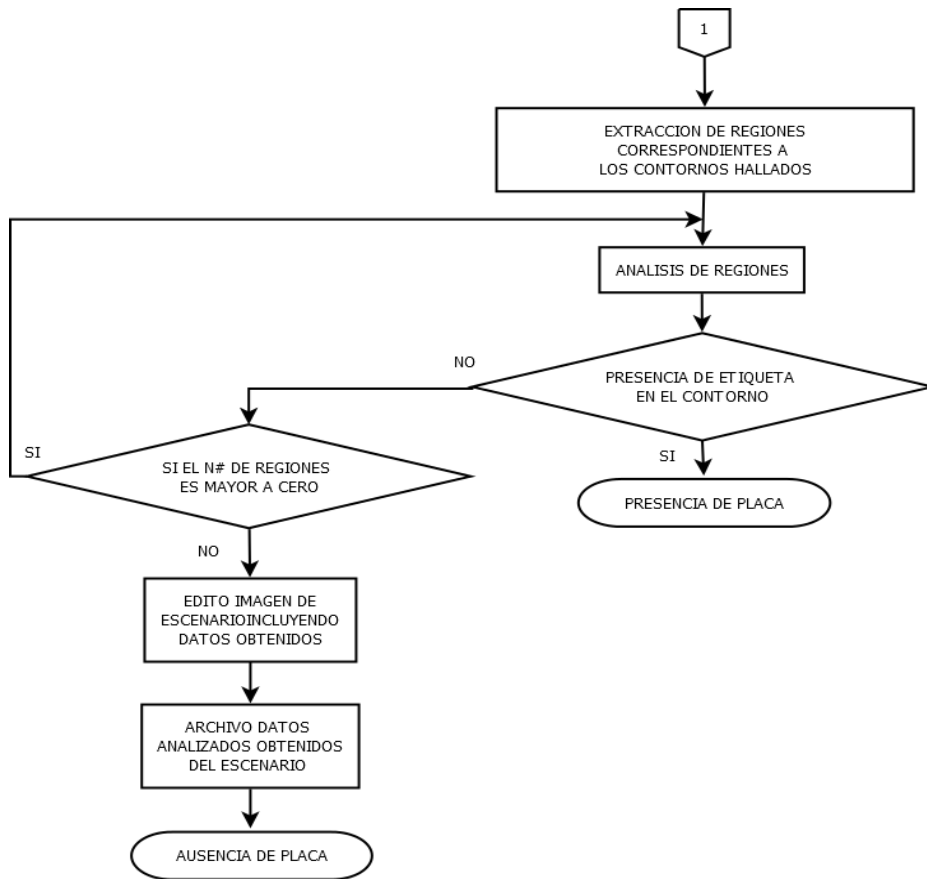
El sistema en principio, captura la imagen del escenario a ser monitoreado, luego de varios procesos como rotacion, filtro de ruido (Gaussiano), conversion de RGB a HSV, extracion de capa filtrada por color, binarizacion de la capa, dilatacion, el proceso de conteo y deteccion de contornos me permite entre las regiones halladas compararlas para ordenarlas desde la region con mayor tamaño hacia la region de menor tamaño.

Para luego analizar cada region e identificar si contiene la etiqueta que distingue la placa, si la region que es analizada no muestra contener la etiqueta, se analizara la siguiente region, este proceso se ejecutara consecutivamente hasta hallar la region que representa la placa como puede notarce en las figuras , de no detectarse una region que corresponda a la placa, se registrara que la placa esta ausente (Figura 3- 3 Ausencia de placa).



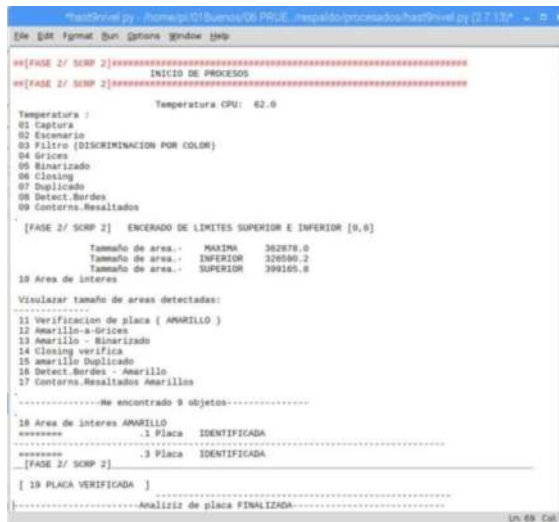
**Figura 2-3** Algoritmo de detección de placa parte a

Realizado por: Saá H. 2018.



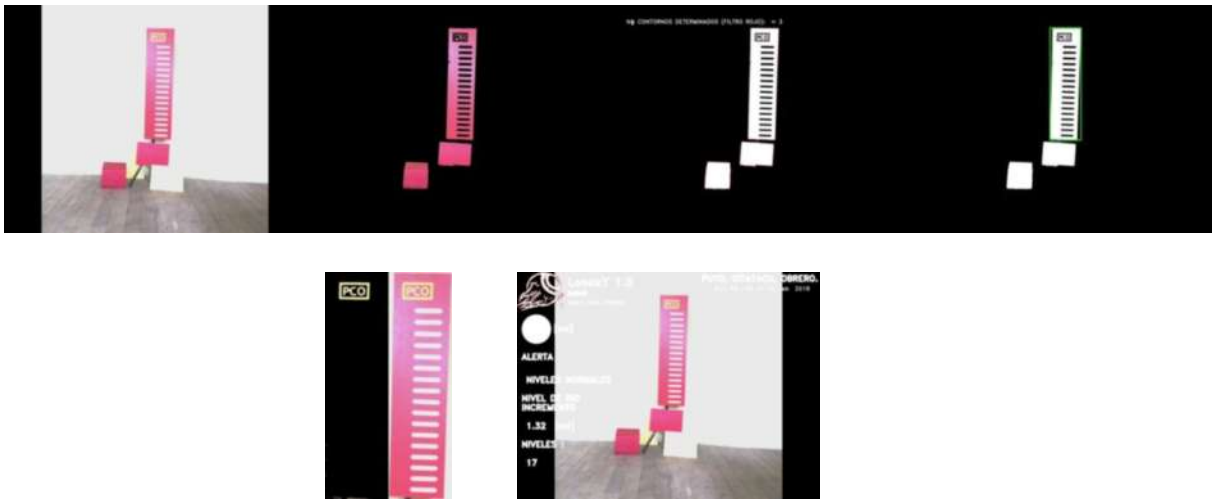
**Figura 3-1** Algoritmo de detección de placa parte b

Realizado por: Saá H. 2018.



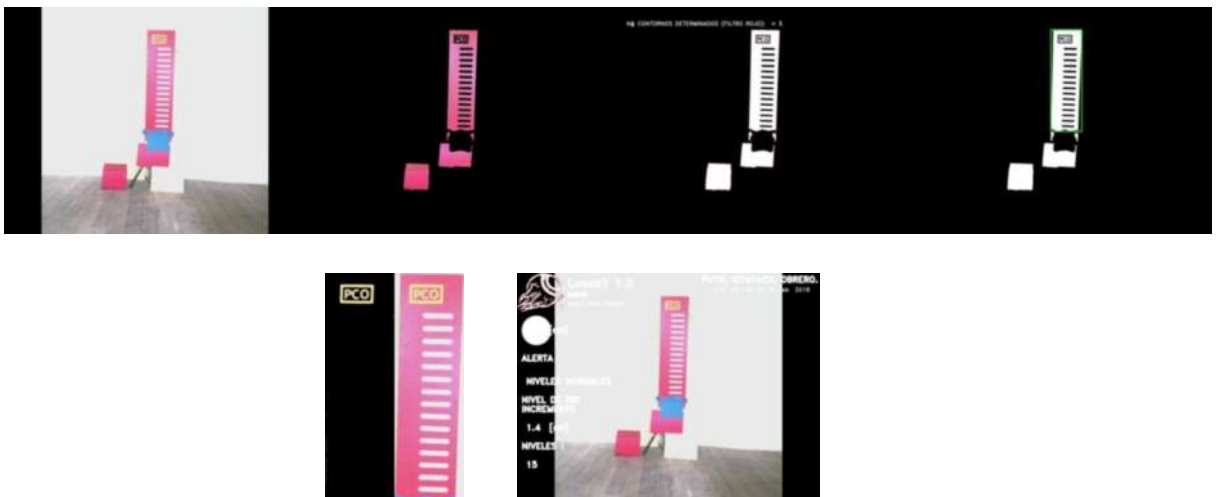
**Figura 4-3** Ventana de monitoreo de procesos del programa

Realizado por: Saá H. 2018.



**Figura 5-3** Placa de escenario en condiciones normales

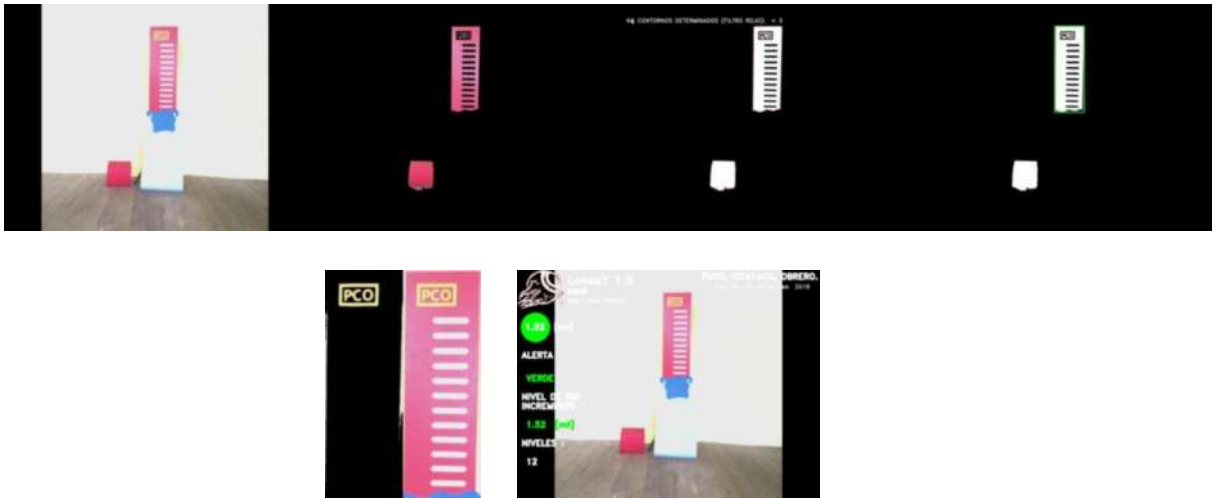
Realizado por: Saá H. 2018.



**Figura 6-3** Placa de escenario en condiciones normales

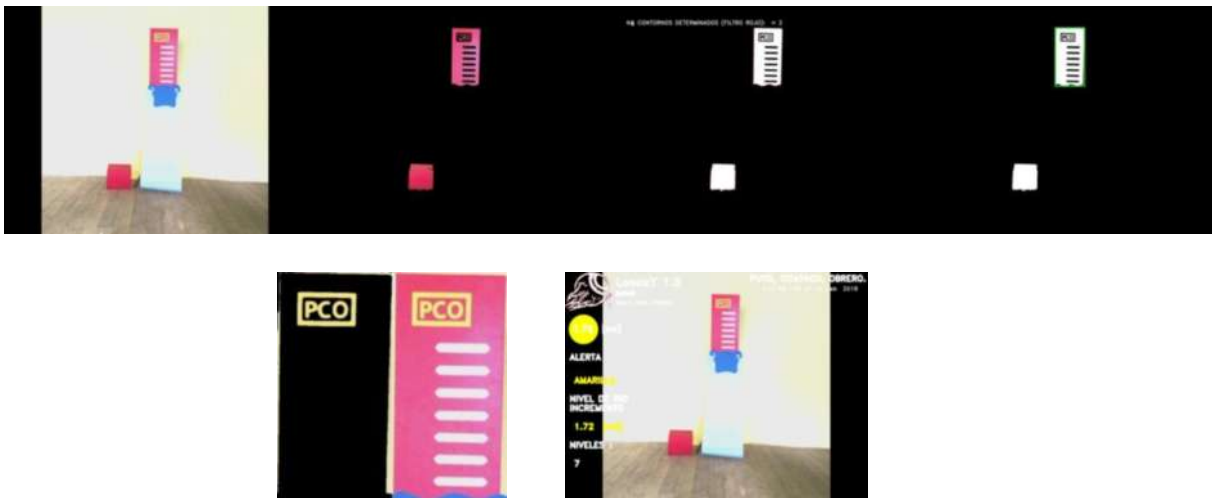
Realizado por: Saá H. 2018.





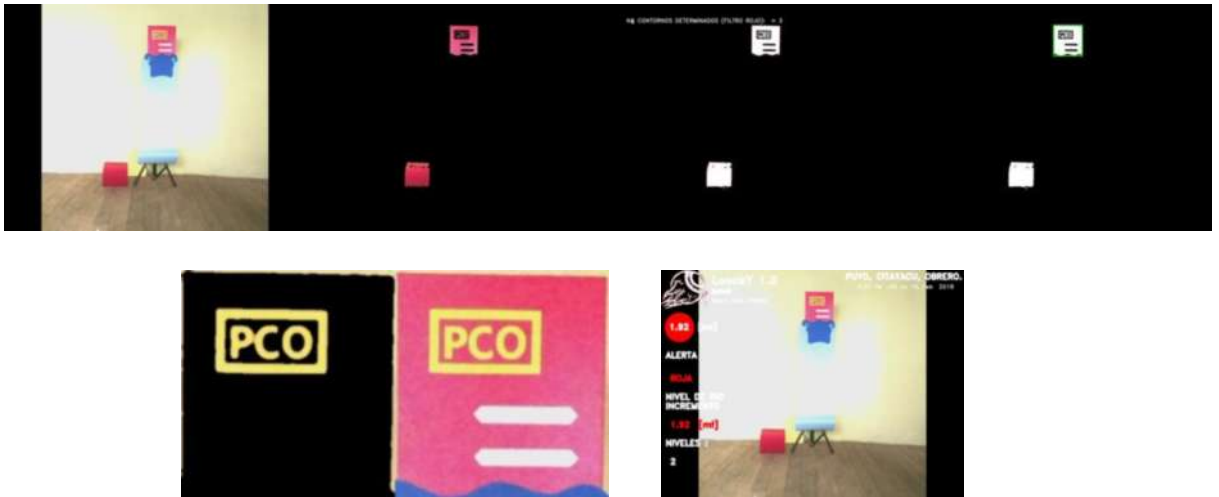
**Figura 7-3** Placa de escenario en condición de alerta verde

Realizado por: Saá H. 2018.



**Figura 8-3** Placa de escenario en condición de alerta amarilla

Realizado por: Saá H. 2018.



**Figura 9-3** Placa de escenario en condición de alerta roja

Realizado por: Saá H. 2018.

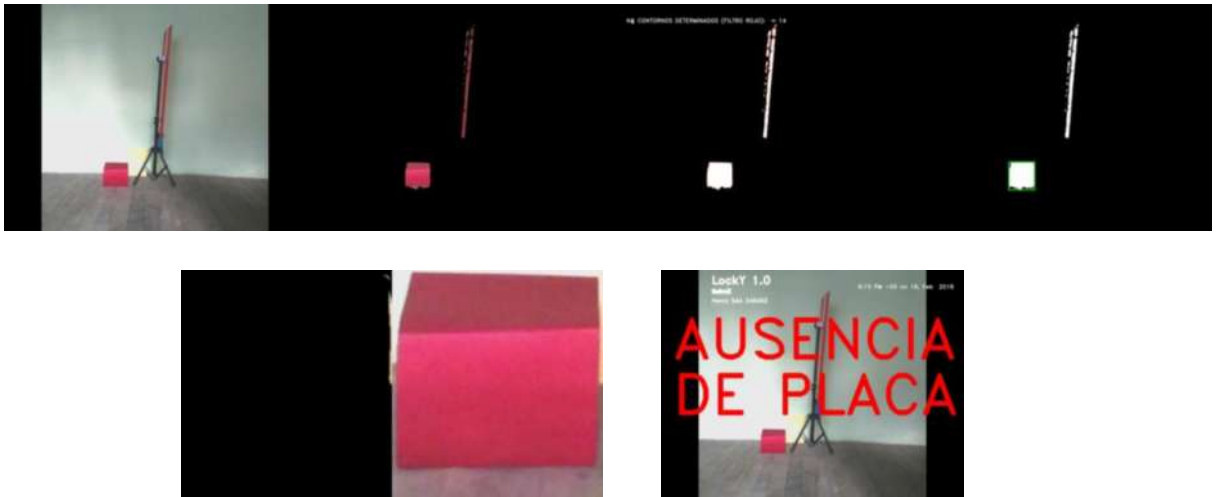
```

Python 2.7 Shell: C:\Users\saah\Documents\PRUEBAS\escena\procesado\escena.py [2.7.13]
File Edit Format Run Options Window Help
**[FASE 2/ SCRP 2]**=====
INICIO DE PROCESOS
**[FASE 2/ SCRP 2]**=====
Temperatura CPU: 59.0
Temperatura :
01 Captura
02 Escenario
03 Filtro (DISCRIMINACION POR COLOR)
04 Grises
05 Binarizado
06 Closing
07 Duplicado
08 Detect.Bordes
09 Contornos.Resultados
- [FASE 2/ SCRP 2] -----No encontrado 14 objetos-----
[FASE 2/ SCRP 2] ENCERADO DE LIMITES SUPERIOR E INFERIOR [0,0]
Tamaño de area.- MAXIMA 74585.0
Tamaño de area.- INFERIOR 67054.0
Tamaño de area.- SUPERIOR 81955.5
10 Area de interes
Visualizar tamaño de areas detectadas:
-----
11 Verificacion de placa ( AMARILLO )
12 Amarillo-a-Grises
13 Amarillo - Binarizado
14 Closing verifica
15 amarillo Duplicado
16 Detect.Bordes - Amarillo
17 Contornos.Resultados Amarillos
-----No encontrado 2 objetos-----
18 Area de interes AMARILLO
[FASE 2/ SCRP 2] AGENCIA DE PLACA
---DATOS PARA WALL--- COLOR IDENTIFICADO: AGENCIA DE PLACA
SE HA IDENTIFICADO : No hay placa
[FASE 2/ SCRP 2]
[ 18 PLACA VERIFICADA ]
-----Analisis de placa FINALIZADA-----
-----
Placa hallada : 0.0
-----
NO HAY PLACA
-----
(Amarillo) No hay placa
-----
Ln: 85 Col: 0

```

**Figura 10-3** Ventana procesos del programa, ausencia de placa

Realizado por: Saá H. 2018.



**Figura 11-3** Ausencia de placa

**Realizado por:** Saá H. 2018.

Esta etapa del algoritmo, de detección de la placa, muestra un funcionamiento satisfactorio detectando y localizando en cada escena la placa de referencia, tomando en cuenta que para las pruebas del algoritmo, en el escenario de monitoreo, se introdujo varios objetos de color de la placa para someter, el programa a evaluación de entornos de prueba, forzando al programa a discernir entre los entornos hallados, en búsqueda de la placa, lo que se tradujo en mayor tiempo, en esta etapa de proceso puede verse los tiempos resultado de estos procesos en la tabla 3-1 donde se hace un análisis con dos objetos en el escenario, placa de referencias y un objeto extraño de color rojo.

En esta tabla 3-1 se evidencia, el tiempo que conlleva el análisis de los contornos hallados mediante la función Canny. Contornos que después fueron sometidos a un ciclo que compara entre los contornos hallados, para generar un vector donde se archive en forma descendente (desde el contorno más grande, hasta el mas pequeño) el tamaño y la posición de cada contorno. Para luego ser analizados, en busca de detectar la región de etiqueta que identificaría, al contorno como la placa de referencia, si la región identificada dentro del contorno cumple con ser la etiqueta.

**Tabla 1-3** Tiempos de detección de placa

CASO #	NUMERO DE OBJETOS EN LA ESCENA	# NUMERO DE NIVELES	TIEMPO DE EJECUCION [ Seg ]	ALTURA DETERMINADA [ mt ]
1	2	12	15,1898695305	1.52
2	2	11	15,1898709884	1.56
3	2	10	15,1898746887	1.6
4	2	9	15,1898768951	1.64
5	2	8	15,1898815434	1.68
6	2	7	15,1898836671	1.72
7	2	6	15,1898870624	1.76
8	2	5	15,1898896675	1.80
9	2	4	15,1898923417	1.84
10	2	3	15,199929211	1.88
11	2	2	16,11299318822	1.92
12	2	1	16,12499334955	1.96
13	2	0	16,1923423466	2.0

Realizado por: Saá H. 2018.

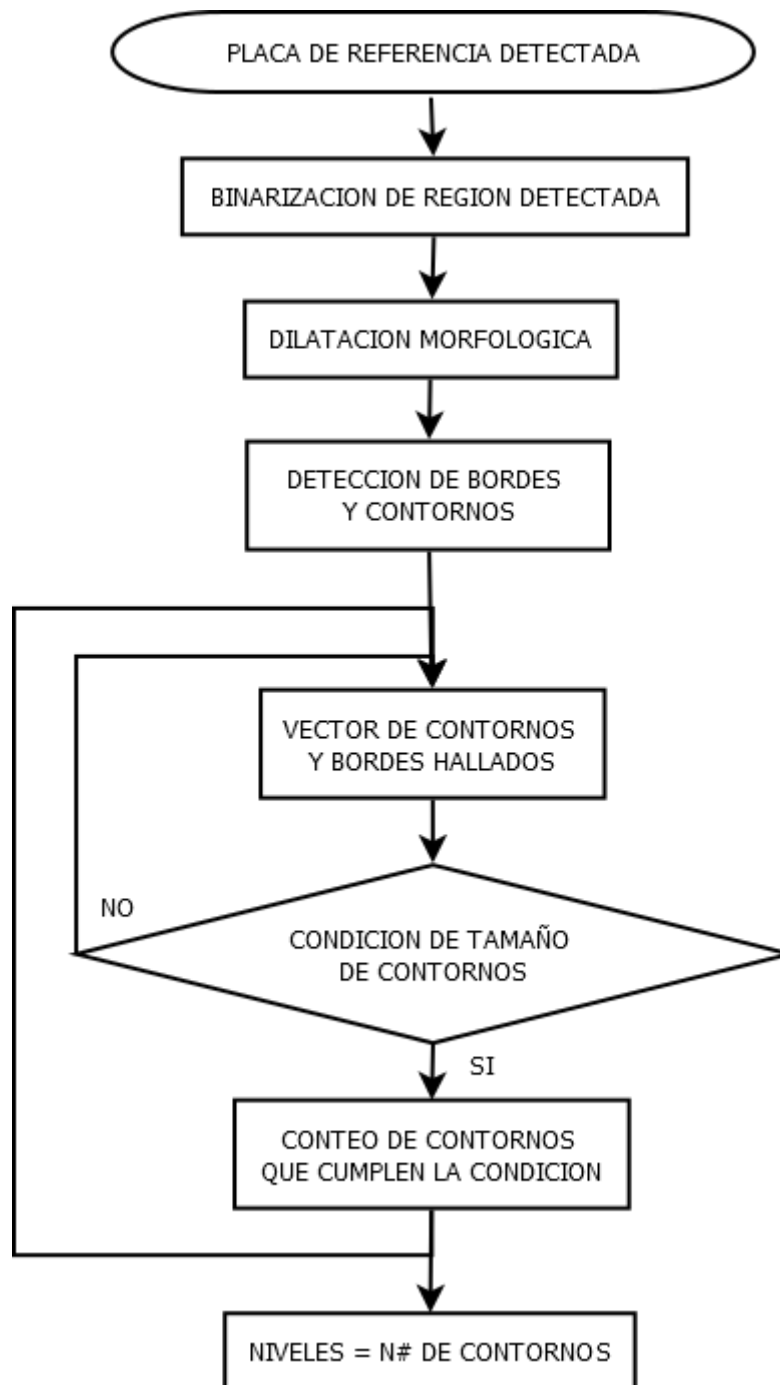
Ante el análisis de la tabla 3-1, concluyo que debido a la función implementada para ordenar las regiones para luego su posterior análisis. Donde empezando el análisis desde la región más grande, a medida de que la región de la placa de referencia, en el escenario empieza reducirse, lo que se entendería que el nivel de río ha cubierto gran parte de la placa y además en el escenario se encuentre otros objetos de color similar a la placa, si el objeto introducido al escenario en ese momento presenta una región con mayor tamaño a la de la placa, por el proceso que con lleva su análisis para determinar si esa región es la placa y de no ser, continua por analizar la siguiente región guardada en el vector, hasta identificar la placa de referencia, con lleva más tiempo, mostrándose como un tiempo que retarda los siguientes procesos.

### **3.1.2 Pruebas de conteo de niveles**

Para identificar y hacer un conteo de los niveles presentes en la region que se ha detectado (Placa de referencia), el sistema recortara de la imagen original, esta region que correspondiente a la placa, luego esa imagen se binariza, para aplicar procesos de morfologia y dilatacion, luego ejecutar el proceso de deteccion y conteo de contornos.

Luego el contador interno que mantiene el programa, se incrementa mientras los contornos hallados en la region de la placa, cumplen con la condicion de tamaño que fue ingresada, para que sean identificados como niveles, si cumple con la condicion, el contador interno del programa incrementara en uno, de no cumplir con la condicion, analiza el siguiente contorno hallado, este proceso se ejecutara sucesivamente, hasta terminar el analisis con todos los contornos que fueron hallados.

Luego se archiva todos los datos recabados por el programa, para que los siguientes procesos continuen su ejecucion.



**Figura 12-3** Proceso de detección de niveles

Realizado por: Saá H. 2018.

```

"fast9nivel.py" /home/pi/01Buenos/06 PRUE.../respaldo/procesados/fast9nivel.py (2.7.13)
File Edit Format Run Options Window Help
-----Analiz de placa FINALIZADA-----
Placa hallada : 1.0
20 Analisis de nivel, (conteo)
21_Niveles.Resultados
-[FASE 2/ SCRIP 2] -----He identificado 22 niveles-----
22_Niveles.Resultados
--[FASE 2/ SCRIP 2]-----
Niveles encontrados:      17
ALTURA CRECIDO :
[mt]  1.32
-----
23_Señales de emergencia, CONDICION DE NIVELES
NIVELES NORMALES
---DATOS PARA MAIL=      COLOR IDENTIFICADO:      NIVELES NORMALES
-----
24_MONITOREO RIO
25_Señales de emergencia, CONDICION DE NIVELES
Niveles Normales
--[FASE 2/ SCRIP 2]-----Fin de procesos-----
--[FASE 2/ SCRIP 2]-----Etapa de Visualizadores-----
k imagen de Niveles Hallados - Monitoreados
[FASE 2/ SCRIP 2] ##### Funcion Define (FINAL) #####
Tiempos 1518984704.78
Conteo desde que corrio el programa
TIEMPO DE EJECUCION [Seg]:
23.6944629609
[FASE 4/ SCRIP 1] TIEMPO cronometrado [Seg]:    24.0
[FASE 2/ SCRIP 2]
Ln 112 Col 0

```

**Figura 13-2** Ventana de procesos de sistema

Realizado por: Saá H. 2018.



**Figura 14-3** Resultado de procesos de niveles

Realizado por: Saá H. 2018.

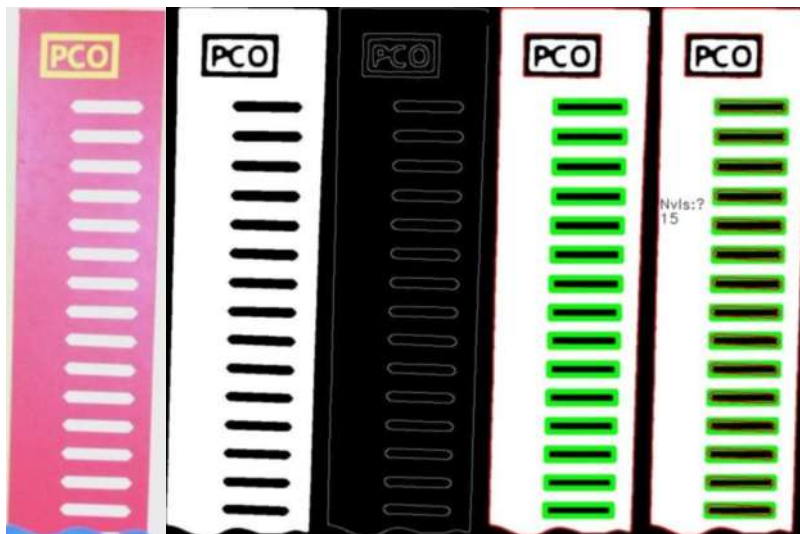
```

*fastSnivel.py - /home/pi/01Buenos/06 PRUE. /respaldo/procesados/fastSnivel.py [2.7.13]
Ede Edit Fgmat Bun Options Window Help
-----Analizir de placa FINALIZADA-----
Placa hallada : 1.0
20 Analisis de nivel. (conteo)
21_Niveles.Resultados
-[FASE 2/ SCRIP 2] -----He identificado 17 niveles-----
22 Niveles.Resultados
--[FASE 2/ SCRIP 2]-----
Niveles encontrados:      15
ALTURA CRECIDO :
[mt] 1.4
-----
23_Señales de emergencia, CONDICION DE NIVELES
NIVELES NORMALES
---DATOS PARA MAIL=      COLOR IDENTIFICADO:      NIVELES NORMALES
-----
24_MONITOREO RIO
25_Señales de emergencia, CONDICION DE NIVELES
Niveles Normales
--[FASE 2/ SCRIP 2]-----Fin de procesos-----
--[FASE 2/ SCRIP 2]-----Etapas de Visualizadores-----
k imagen de Niveles Hallados - Monitoreados
[FASE 2/ SCRIP 2] ##### Funcion Define (FINAL) #####
Tiempo 1518985833.99
Conteo desde que corrio el programa
TIEMPO DE EJECUCION [Seg]:
24.339933157
[FASE 4/ SCRIP 1] TIEMPO cronometrado [Seg]:      24.0
[FASE 2/ SCRIP 2]
Ln: 833 Col: 1

```

**Figura 15-3** Ventana de procesos de sistema

Realizado por: Saá H. 2018.



**Figura 16-3** Resultado de procesos de niveles

Realizado por: Saá H. 2018.



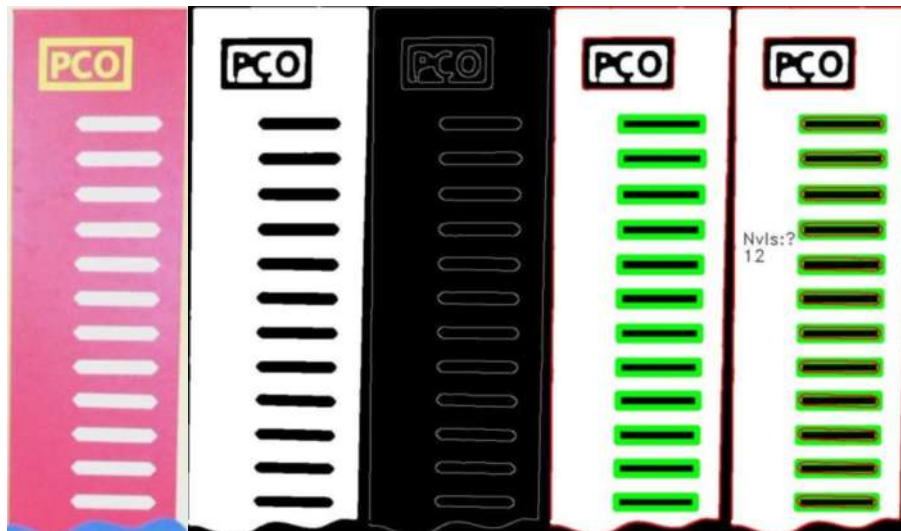
```

fast9nivel.py - /home/jp/01/Buenos/06 PRUE./respaldo/procesados/fast9nivel.py (2.7.13)
Ede Edit Format Run Options Window Help
-----Analisis de placa FINALIZADA-----
Placa hallada : 1.0
20 Analisis de nivel, (conteo)
21 Niveles.Resultados
--[FASE 2/ SCRP 2] -----He identificado 15 niveles-----
22 Niveles.Resultados
--[FASE 2/ SCRP 2]-----
Niveles encontrados:      12
ALTURA CRECIDO :
[mt] 1.52
-----
23_Señales de emergencia, CONDICION DE NIVELES
VERDE
----DATOS PARA MAIL=      COLOR IDENTIFICADO: VERDE
-----
24_MONITOREO RIO
25_Señales de emergencia, CONDICION DE NIVELES
VERDE
--[FASE 2/ SCRP 2]-----Fin de procesos-----
--[FASE 2/ SCRP 2]-----Etapa de Visualizadores-----
* imagen de Niveles Hallados - Monitoreados
[FASE 2/ SCRP 2] ##### Funcion Define (FINAL) #####
tiempos 151000953.05
Conteo desde que corrio el programa
TIEMPO DE EJECUCION [Seg]:
23.4043958187
[FASE 4/ SCRP 1] TIEMPO cronometrado [Seg]: 23.0
[FASE 2/ SCRP 2]
ln:1949 Col:0

```

**Figura 17-3** Ventana de procesos de sistema

Realizado por: Saá H. 2018.



**Figura 18-3** Resultado de procesos de niveles

Realizado por: Saá H. 2018.

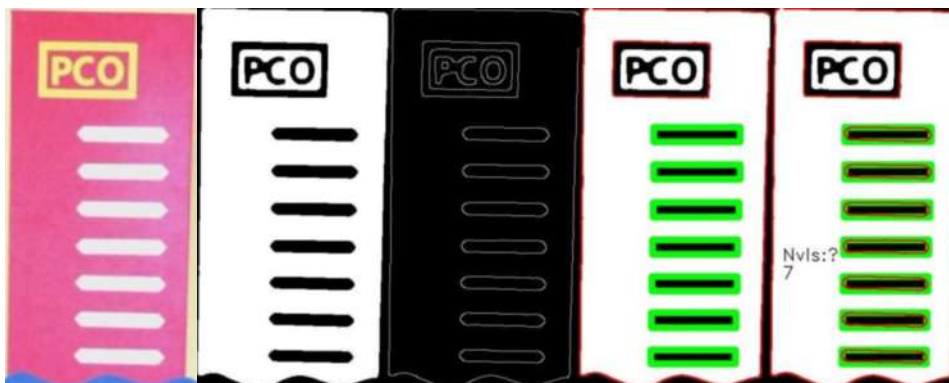
```

4nivel.py - /home/pl01Buenos/06 PRUEBAS/respaldo/procesados/4nivel.py [2.7.13]
File Edit Format Run Options Window Help
-----Analizis de placa FINALIZADA-----
Placa hallada : 1,0
20 Analisis de nivel. (conteo)
21 Niveles.Resultados
-[FASE 2/ SCRIP 2] -----He identificado 11 niveles-----
22 Niveles.Resultados
--[FASE 2/ SCRIP 2]-----
Niveles encontrados:      7
ALTURA CRECIDO :
[et] 1.72
-----
23_Señales de emergencia, CONDICION DE NIVELES
AMARILLO
---DATOS PARA MAIL=      COLOR IDENTIFICADO:      AMARILLA
-----
24_MONITOREO RIO
25_Señales de emergencia, CONDICION DE NIVELES
AMARILLO
--[FASE 2/ SCRIP 2]-----Fin de procesos-----
--[FASE 2/ SCRIP 2]-----Etapa de Visualizadores-----
k imagen de Niveles Hallados - Monitoreados
[FASE 2/ SCRIP 2] ***** Funcion Define (FINAL) *****
Tiempo 151998366.71
Conteo desde que corrio el programa
TIEMPO DE EJECUCION [Seg]:
23.3713328838
[FASE 4/ SCRIP 1] TIEMPO cronometrado [Seg]:      23.0
|[FASE 2/ SCRIP 2]
Ln: 3831 Col: 0

```

**Figura 19-3** Ventana de procesos de sistema

Realizado por: Saá H. 2018.



**Figura 20-3** Resultado de procesos de niveles

Realizado por: Saá H. 2018.

```

Inozz.py - /home/pcp/01/Buenos/06 PRUEBAS/respaldo/Inozz.py [2.7.13]
File Edit Format Run Options Window Help
[FASE 2/ SCRIP 2] -----He identificado 6 niveles-----
22 Niveles.Resultados
--[FASE 2/ SCRIP 2]-----
Niveles encontrados:      2
ALTIMA CRECIDA :
[mc]  1.82
-----
23 Señales de emergencia, CONDICION DE NIVELES
ROJO
---DATOS PARA MAIL=      ROJO      COLOR IDENTIFICADO:      ROJO
-----
24 MONITOREO RIO
25 Señales de emergencia, CONDICION DE NIVELES
ROJO
--[FASE 2/ SCRIP 2]-----Fin de procesos-----
--[FASE 2/ SCRIP 2]-----Etapa de Visualizadores-----
k Imagen de Niveles Hallados - Monitoreados
[FASE 2/ SCRIP 2] ***** Funcion Define (FINAL) *****
Tiempo 151809218.22
Conteo desde que corrio el programa
TIEMPO DE EJECUCION [Seg]:
23.181859844
[FASE 4/ SCRIP 1] TIEMPO cronometrado [Seg]:      23.8
[FASE 2/ SCRIP 2] -----
Temperatura :      Temperatura CPU: 71.0
Ln 2269 Col 0

```

**Figura 21-3** Ventana de procesos de sistema

Realizado por: Saá H. 2018.



**Figura 22-3** Resultado de procesos de niveles

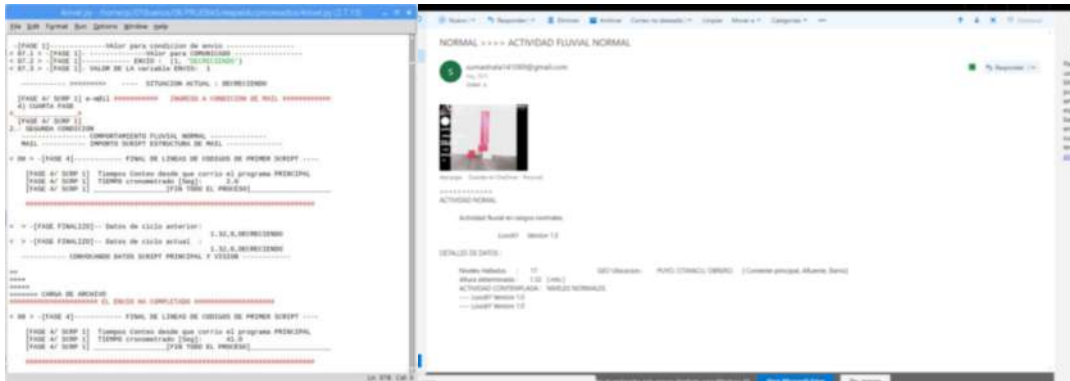
Realizado por: Saá H. 2018.

El sistema muestra un desempeño óptimo para el fin que fue diseñado, que es contar los niveles presentes en la placa de referencia para cálculos en relación a estos datos.

### 3.1.3 Pruebas de envío de informes (Mails)

Una vez recabado datos de la actividad del río, estos datos sirven para los cálculos correspondientes, sobre nivel, alertas, tiempos de predicción, y tomar decisión de emitir mensajes vía mail, en la Figura 3-23.

El mail consta de un archivo adjunto, que es la imagen del escenario, información sobre el tipo de alerta, datos extraídos de la imagen, datos de predicción de tiempos y nuevos niveles que podría tomar el río.



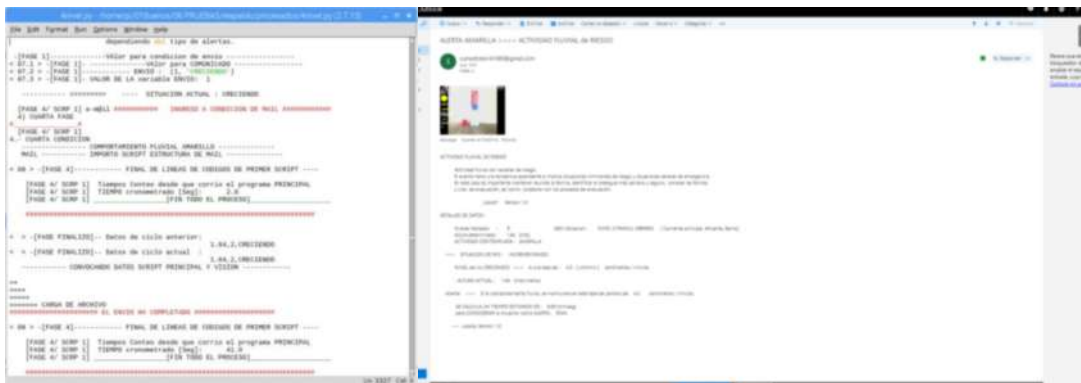
**Figura 23-3** Envío de informe de actividad “Normal”

Realizado por: Saá H. 2018.



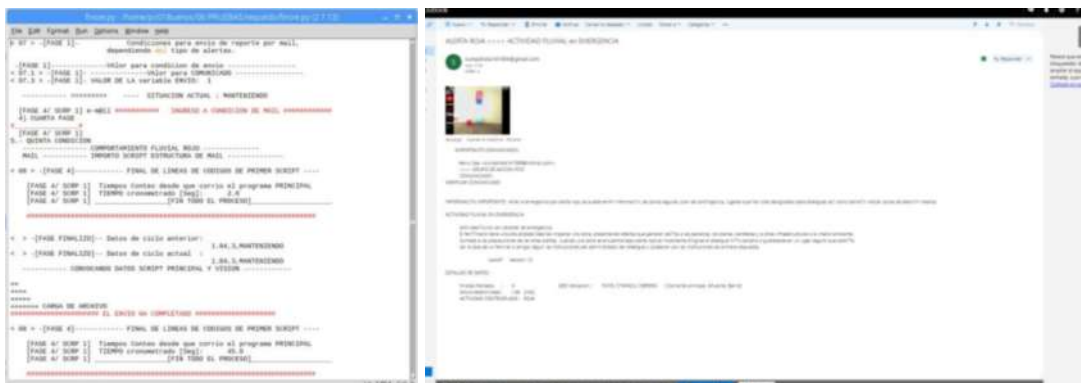
**Figura 24-3** Envío de informe de actividad “Verde”

Realizado por: Saá H. 2018.



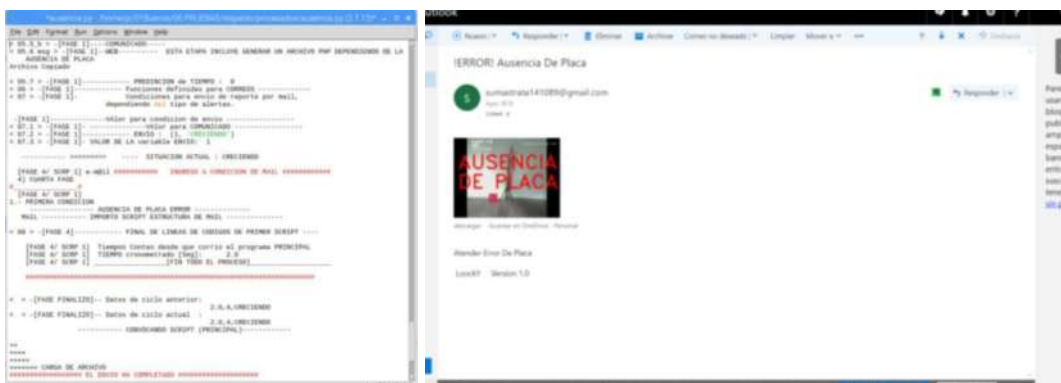
**Figura 25-3** Envió de informe de actividad “Amarilla”

Realizado por: Saá H. 2018.



**Figura 26-3** Envió de informe de actividad “Roja”

Realizado por: Saá H. 2018.



**Figura 27-3** Envió de informe de actividad “Error de placa”

Realizado por: Saá H. 2018.

El emitir información vía permitirá tener a los usuarios más relacionados a lo que esté sucediendo, evitando el desconocimiento total o zozobra que se puede genera en relación a estas situaciones así tomar acciones preventivas.

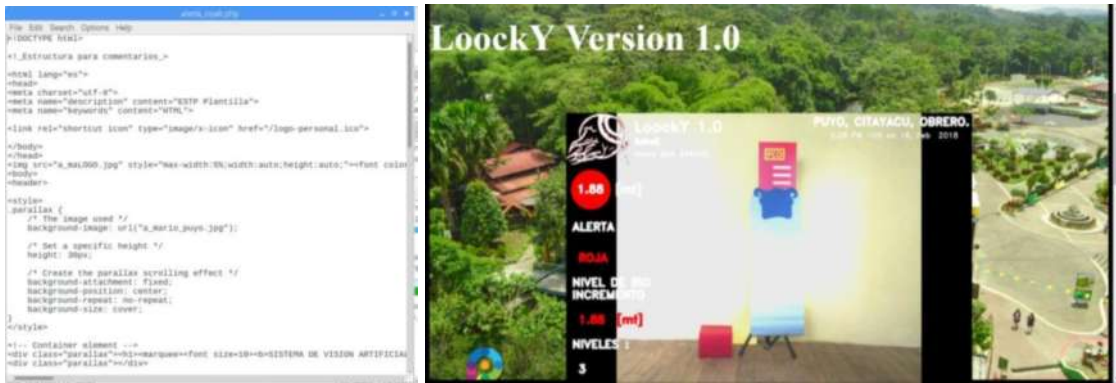
### **3.1.4 Pruebas de actualización de datos en la página web**

El sistema permite subir datos hacia un servidor web con el fin de informar a la población en general que se encuentre interesada por conocer la situación de ese punto, en la página se puede encontrar la imagen del escenario, información sobre algún comunicado emitido por los grupos de socorro, cálculos en relación a los datos adquiridos de la imagen figura 3-28.



**Figura 28-3** Página de información

Realizado por: Saá H. 2018.



**Figura 29-3** Imagen que muestra el sitio web

Realizado por: Saá H. 2018.



**Figura 30-4** Imagen de comunicado, publicado en web

Realizado por: Saá H. 2018.



**Figura 31-5** Imagen de datos publicados en el servidor web

Realizado por: Saá H. 2018.



**Figura 32-6** Información adicional

Realizado por: Saá H. 2018.

### 3.2 Análisis de resultados de pruebas obtenidas

#### 3.2.1 *¿Cuáles son las principales características de implementación y funcionamiento del sistema desarrollado?*

Las características del sistema desarrollado en este proyecto de titulación para su implementación son:

Tarjeta única de desarrollo (Raspberry pi3)

- Tarjeta única de desarrollo (Raspberry pi3)
- Cámara de bajo costo
- Tarjeta de procesamiento de bajo costo

Software libre

- Lenguaje de desarrollo Python
- Librerías de visión artificial Open CV
- Mensajería vía internet
- Servicio web
- Motores de datos





**Figura 33-7** Prototipo de SAT.

**Realizado por:** Saá H. 2018.

### ***3.2.2 ¿Como influyen las librerías y algoritmos, en conjunto con el desempeño de los procesos de visión artificial, en la funcionalidad de este sistema?***

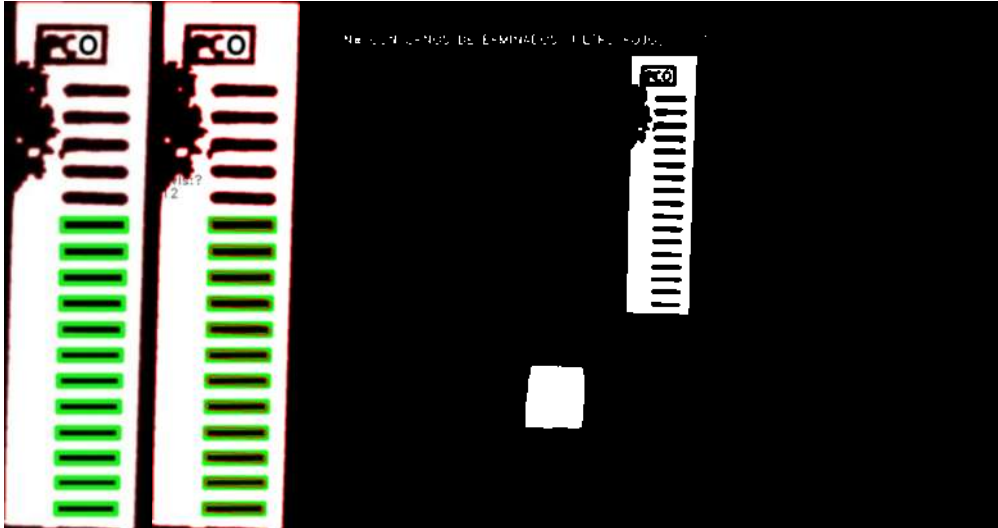
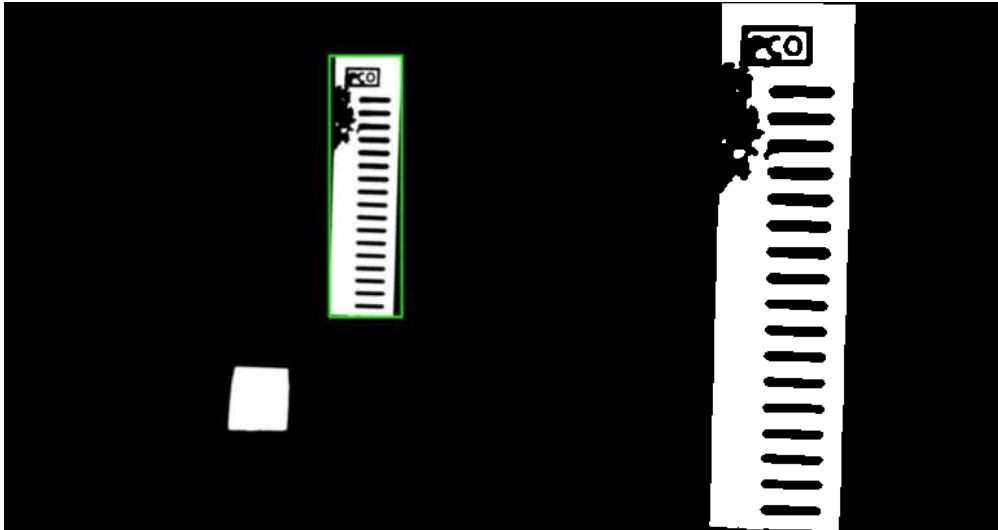
El poder integrar variadas librerías que servirán, para solucionar procesos contemplados en un algoritmo, permite proyectarse a dar mayores prestaciones y soluciones a diversos problemas que de otra manera sería complicado o en un extremo imposible dar solución.

Además, debido a la extensa investigación que demandaría cada proyecto de visión artificial al evitar el uso de librerías predefinidas, las mismas que cumplen una funcionalidad y que a medida que son usadas y experimentadas son modificadas buscando su optimización, el desarrollo se volvería muy complicado y de larga duración.

Por estas razones, en el desarrollo de este proyecto fue fundamental el uso y comprensión de las prestaciones que brinda cada librería destinada a procesar y extraer variadas características de

las imágenes permitiendo así cubrir el desarrollo y la necesidad que planteo este proyecto. Una de las librerías utilizadas fue la de definición de un filtro de suavizado Gaussiano presente en la compilación de librerías de OpenCV.

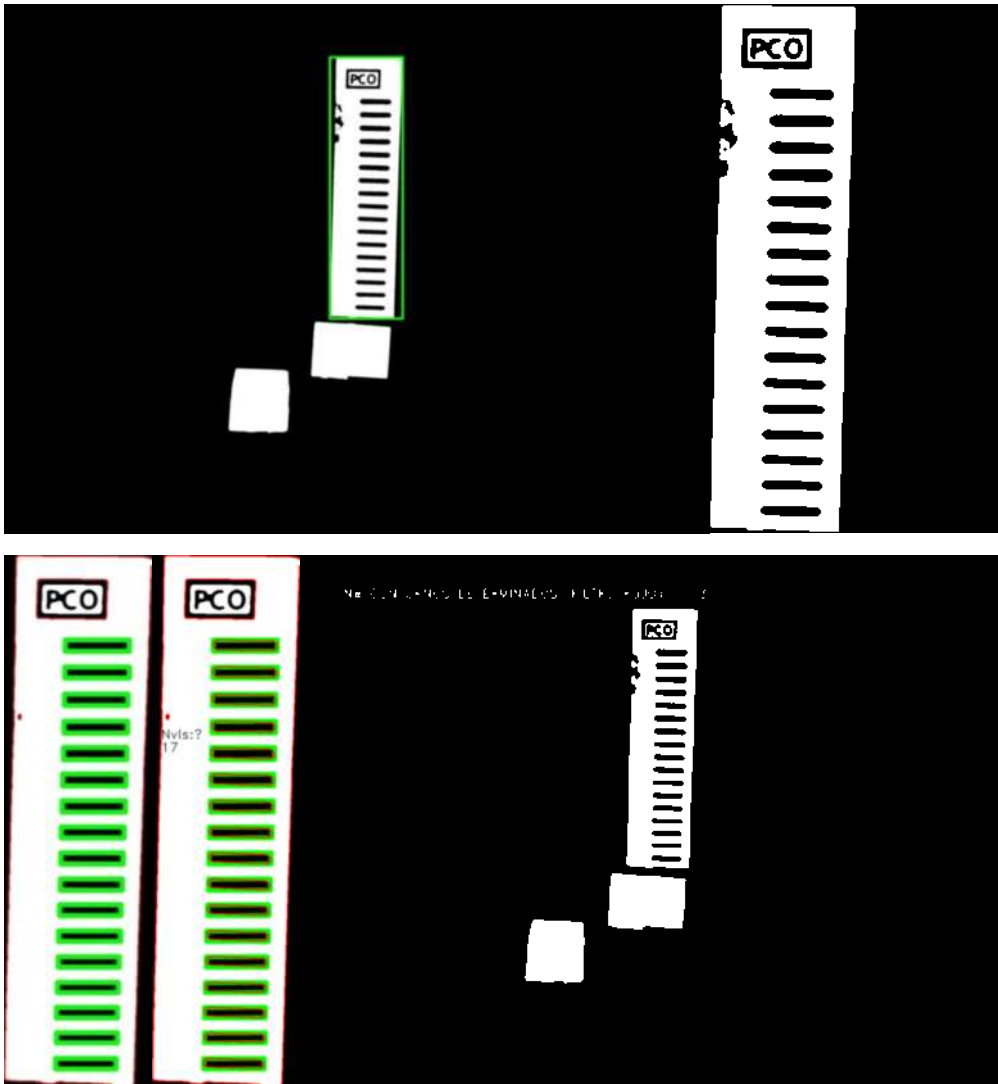
En la figura 33-3 se puede apreciar la imagen, que se destinaria a análisis sin uso de suavizado gaussiano, mostrando ciertas perturbaciones en relación al reflejo de luz.



**Figura 34-8** Imagen en ausencia de filtro Gaussiano

Realizado por: Saá H. 2018.

En la figura 34-3 se puede apreciar el uso de un filtro de suavizado Gaussiano:



**Figura 35-9** Imagen en presencia de filtro Gaussiano

**Realizado por:** Saá H. 2018.

Como es posible observar en las anteriores figuras, el uso de un filtro Gaussiano permitió eliminar en gran medida perturbaciones que afectaban el análisis de las imágenes capturadas.

Por otro lado, en la tabla 2-3, se puede notar la importancia que tuvo el uso de filtrado Gaussiano para esta etapa de visión, donde la presencia del filtro en el código del programa mejoró la efectividad en el reconocimiento de niveles, en relación a usar el código sin este filtro para la misma actividad.

**Tabla 2-3** Resumen de los resultados obtenidos con y sin Filtro Gaussiano durante el análisis de las imágenes.

FILTRO GAUSIANO	NIVELES DETECTADOS	TIEMPO DE VISION [Seg]	RESULTADO
PRESENCIA	17	21.6976048946	OPTIMO
AUSENCIA	12	16.468932587	ESCASO

**Realizado por:** Saá H. 2018.

Como se puede observar los tiempos implicados por el procesamiento ante la implementación del filtro, muestran un retardo aceptable ante el procesamiento en ausencia de dicho filtro.

El uso de este filtro, aunque genero tiempos adicionales por su procesamiento que no son trascendentes en el resto de procesos debido a que la actualización y monitorización del estado de caudal del rio se realiza cada minuto, permitió que la detección de niveles sea mejorada considerablemente.

### ***3.2.3 Beneficios en el uso de la red para información sobre situaciones de interés para evitar catástrofes.***

La determinación de los beneficios presentados por el sistema desarrollado se presenta desde varios puntos de vista, los mismos que son presentados a continuación:

#### ***Opinión Personal***

La evolución de los sistemas de comunicación, así como de dispositivos que facilitan interconectividad de sistemas y servicios, permiten a la colectividad mantenerse informada sobre situaciones de interés. Además, el servicio de acceso a la red está muy difundido y al alcance de un gran porcentaje de persona a nivel mundial, sin mencionar que sus ventajas de costos en relación a otros servicios de comunicación son relativamente reducidos. Los beneficios para emitir información de interés de cualquier índole, no tienen limitaciones que impidan mantener al tanto de cualquier situación de riesgo a personas que pueden ser afectadas,

así como informar a grupos y autoridades encargadas de precautelar la seguridad de las personas que se pudieran encontrar en peligro. De esta manera, entre sus cualidades más importantes de los sistemas de comunicación en red es el hecho de tener información a tiempo real de cualquier fenómeno que pudiera ser monitoreado para fines de precaución y gestión.

### ***Opinión de un Experto***

Se realizó una entrevista al MSc. Manuel Montenegro, experto en gestión de riesgos, master en seguridad industrial, y Presidente de la Cruz Roja de Chimborazo, el pasado viernes 09 de marzo de 2018 en horas de la tarde. Entre las actividades realizadas, mediante una exposición del sistema desarrollado y su funcionamiento, se procedió a solicitar su punto de vista como experto, de acuerdo a las siguientes preguntas:

- ¿Consideraría que la información sobre estado del nivel de caudales de ríos que puede emitir el dispositivo puede ser un aporte importante para la logística de grupos de socorro, porque?

*Desde el inicio está garantizado (prestaciones del dispositivo) la aplicabilidad de este proyecto, porque es un estudio que podría servir para garantizar la atención a las personas del lugar donde se registren inundaciones por desbordamientos de ríos. Cabe indicar que el Ecuador no tiene una aplicación a nivel de tecnología para la monitorización de este tipo de eventos.*

MSc. Manuel Montenegro.

- ¿Cómo consideraría usted, la aplicabilidad de este dispositivo en poblaciones de zonas de riesgo, por inundación y desbordamiento de ríos, para organizar grupos de socorro?

*Al fin y al cabo está justificado, porque el hecho de que se salve así sea una sola vida, significaría que este proyecto ha cumplido con su misión. Entonces es factible, viable, se debe tocar puertas, en Ecuador muchos municipios están interesados en contrarrestar este tipo de situaciones catastróficas (inundaciones).*

MSc. Manuel Montenegro.

- ¿Conoce de sistemas semejantes para el manejo de este tipo de eventos en el Ecuador?

*A nivel de ríos no está muy desarrollado porque son eventos (inundaciones) que se presentan mayormente, en etapas de invierno.*

*En el Cotopaxi (presencia de sistemas de alerta temprana) debido al número de personas que viven alrededor del volcán, estos sistemas activan sirenas, donde a la ciudadanía, se le indica los procesos de evacuación.*

*En caso de erupción del volcán Cotopaxi, el casquete de nieve que tiene el volcán (se derretiría), bajara por el río Cutuchi que cruza por la mitad o el centro de la ciudad de Latacunga dando un promedio de una hora en relación a los treinta metros (altura, desde la superficie de la nieve hasta su fondo asentado sobre el volcán) que se considera de lahares, donde sistemas de alerta temprana han sido implementados, junto con cámaras, equipos, sismógrafos para monitoreo.*

*En buena hora que se pueda aplicar a un río, conociendo que la crecida de un río es instantánea donde la gente no tiene tiempo para evacuar.*

MSc. Manuel Montenegro.

- ¿Cómo consideraría usted que el diseño de sistemas de monitoreo pueda ser impulsado de la mano de instituciones educativas para un desarrollo tecnológico?

*Bien, creo que sería ideal que la forma de presentar una tesis se convierta a futuro en un proyecto de implementación real en la sociedad. Esto es importante, más aun cuando hay muchas cosas que hacer en gestión de riesgos. El problema está en que no hay gente o recursos para que sean realizados, y en este caso esta investigación taja una línea de guía. Es el interés ciudadano, detrás de un monitoreo esta la vida de muchas personas.*

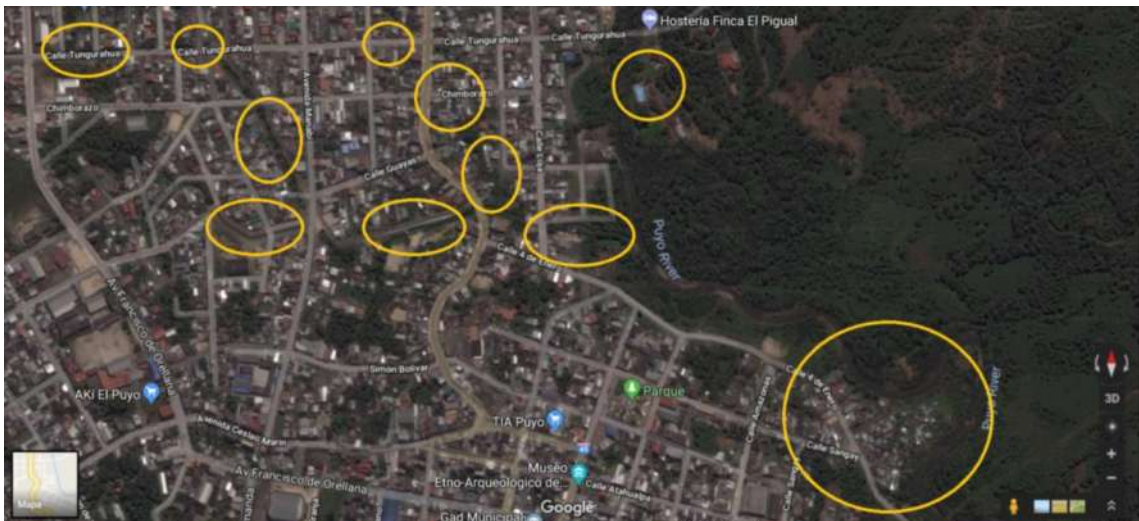
MSc. Manuel Montenegro.

### ***Impacto en la comunidad***

Este trabajo se proyecta, a que a futuro se implemente este dispositivo en varios lugares de interés de monitoreo de los ríos con el fin de crear una red entre los dispositivos para analizar varios puntos, en varios tramos de los ríos.

Generando información en relación a tiempos de evacuación, logística de grupos de socorro, y estudios sobre lugares de afectación, pudiendo mantener información histórica sobre el comportamiento de los ríos para futuros análisis o estudios que se pretenda hacer.

Buscando implementar estos dispositivos como una red de monitoreo, se los ubicaría en varias zonas, como se muestra en la figura 35-3, donde se tomaría en cuenta las áreas remarcadas, correspondientes a los barrios el Obrero, el Chofer, la Isla, barrio Amazonas, sector Primero de Mayo, sector Sindicato de Choferes, sector Boayacu Puyo, sector Asilo, por donde cruzan ríos, y son áreas de constantes inundaciones, favoreciendo a la población de esos sectores.



**Figura 36-10** Zonas de interés de monitoreo.

Realizado por: Saá H. 2018.

### ***Impacto en la Investigación***

Finalmente podemos tener beneficios relacionados al levantamiento de datos para estudios estadísticos, que nos permitan realizar estudios predictivos sobre el estado de ríos. Generando ciertas ventanas de tiempos para alertar tempranamente y de mejor manera a la población y poder lograr mejores planes de contingencia.

## CONCLUSIONES

Las prestaciones de la placa usada, cubren las necesidades que se planteó solucionar, con este proyecto, en conjunto con la herramienta OpenCV diseñada para trabajar en el campo de visión artificial con archivos de imagen, generando así un sistema de monitoreo a tiempo real.

El algoritmo de detección de contornos y bordes, en conjunto con filtros de color nos permite ubicar de una forma más sencilla los elementos de interés que requerimos para interpretar la situación que puede estar generándose en el escenario monitoreado.

La funcionalidad y los beneficios que puede brindar de este sistema permiten proyectarse como una alternativa relevante ante los procedimientos que se usan de manera tradicional, para este tipo de situaciones emergentes, a más de que los usuarios que opten por los datos que puede generar este sistema podrían estar informados, mucho antes de encontrarse en situaciones de peligro o constatar por si mismos la actividad que se esté dando en ese momento.

La placa raspberry presenta limitaciones en tiempos de procesamiento, retardos por procesos y subprocesos, aun así depende del propósito al que será aplicado, tomando en cuenta la importancia o que tan crítico es el tiempo en que necesitemos los datos procesados, casos como estos si bien son situaciones de importancia, el tiempo en el que esta programado su ejecución automática, es aceptable, para tomar decisiones.

Tomando en cuenta el escenario donde se destinaria su uso, la placa de referencia debe usar colores de fondo, que contrarresten con el ambiente para fines prácticos se usó el color rojo, que podría contrarrestar con la vegetación de las zonas a monitorear.

El sistema se proyecta como una alternativa viable para solucionar esta problemática, ya que el sistema se basa en monitoreo, detección de niveles de referencia y retransmisión de datos, permitiendo que cualquier usuario este informado o pueda enlazarse al sitio web.



## RECOMENDACIONES

Trabajar con dispositivos que presenten las mejores características para captura de imágenes a fin de obtener capturas de alta calidad para mejorar la calidad de los procesos a los que serán sometidos.

Informarse sobre las características que va tomando cada nueva versión, de sistema operativo con los que se vaya a trabajar, con el fin de conocer sus limitaciones y ventajas sobre los programas que se usaran así como en relación a los dispositivos que se conectaran.

Usar dispositivos que de mejor manera eviten perturbaciones que afecten, las capturas de imagen de los escenarios a ser monitoreos, evitando la incidencia de luz directa hacia el lente, desde cualquier fuente.

Para crear placas de referencias de nivel usar para materiales, que eviten reflejar luz hacia el lente de la cámara a usarse para capturar imágenes de la escena, debido a que esta radiación afecta la imagen que se puede tomar del ambiente, y la captura presentaría errores que pueden afectar las etapas de procesamiento.

## BIBLIOGRAFÍA

**Organización de Estados Americanos.** *Manual para el Diseño, Instalación, Operación y Mantenimiento de Sistemas Comunitarios de Alerta Temprana ante Inundaciones*

. [En línea] 21 de Julio de 2015. [Citado el: 13 de Febrero de 2018.]. Disponible en:

<http://copeco.gob.hn/documents/Manual-SATI-21-07-2015.pdf>.

**EVANS, Dave.** *Internet de las cosas. Como la próxima evolución del Internet lo cambia todo.*

[En línea]. Cisco Systems Inc, San José – California. 2011, p. 02. [Consulta: 18 de abril de

2017]. Disponible en:

[http://www.cisco.com/c/dam/global/es\\_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf](http://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf)

**EZQUEDA ELIZONDO, José.** *Fundamentos de Procesamiento de Imágenes.* [En línea].

Universidad Autónoma de Baja California, Tijuana – México. 2002, p. 24 [Consulta: 02 de abril de 2017]. Disponible en:

[http://www.academia.edu/16801512/Fundamentos\\_de\\_procesamiento\\_de\\_im%C3%A1genes\\_digitales](http://www.academia.edu/16801512/Fundamentos_de_procesamiento_de_im%C3%A1genes_digitales)

**PAJARES MARTINSANZ, Gonzalo; & DE LA CRUZ GARCÍA, Jesús.** *Visión por computador, imágenes digitales y aplicaciones.* 2<sup>da</sup>. Ed. Madrid – España: Ra-Ma, 2007, pp. 01- 08.

**PEREZ, David.** *Sistemas embebidos y sistemas operativos embebidos.* [En línea]. Universidad Central de Venezuela. Caracas – Venezuela. 2009, pp. 04-11. [Consulta: 05 de mayo de 2017].

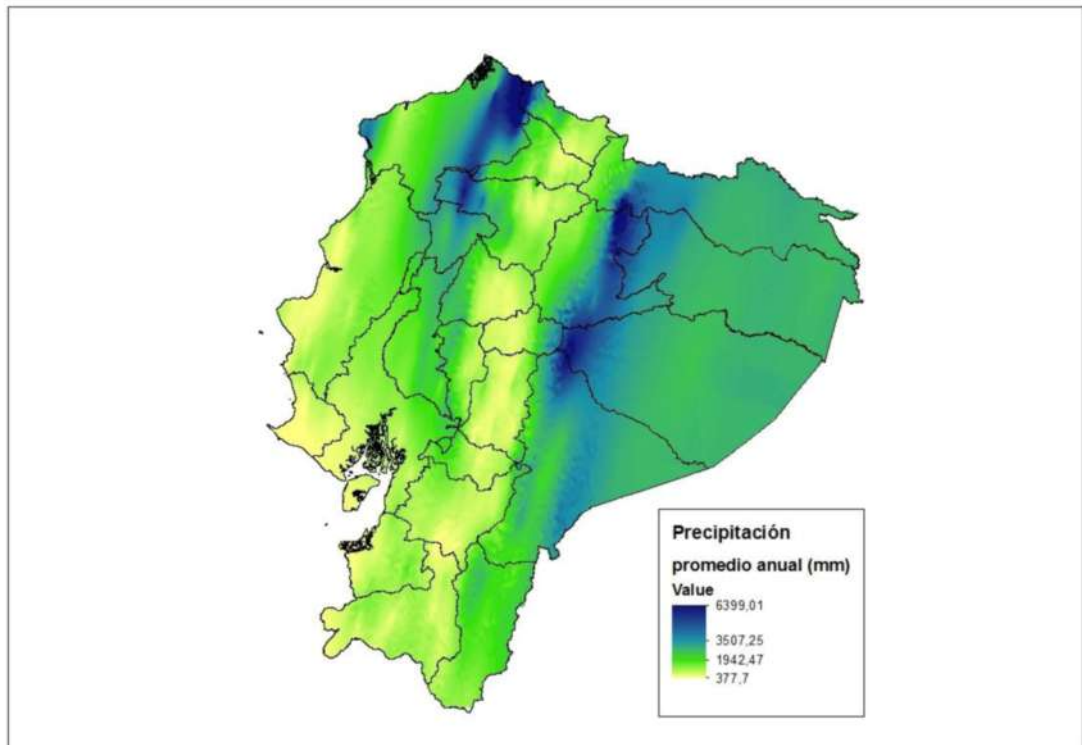
Disponible en: [http://www.academia.edu/16523506/Info\\_Sistemas\\_Embebidos](http://www.academia.edu/16523506/Info_Sistemas_Embebidos)

**UNCIENCIA.** Universidad pública. [En línea] [unciencia.unc.edu.ar](http://unciencia.unc.edu.ar), 16 de diciembre de 2015. [Citado el: 20 de febrero de 2018.]. Disponible en: <http://www.unciencia.unc.edu.ar/2015/diciembre/insus-ya-brinda-servicios-de-alerta-temprana-de-inundaciones-en-catamarca>.

**VÉLEZ, José; et al. :** *Visión por computador.* [En línea]. Universidad Rey Juan Carlos, Madrid – España. 2003, p. 128 [Consulta: 09 de abril de 2017]. Disponible en: <http://www.visionporcomputador.es/libroVision/VisionPorComputador.pdf>

## ANEXO A

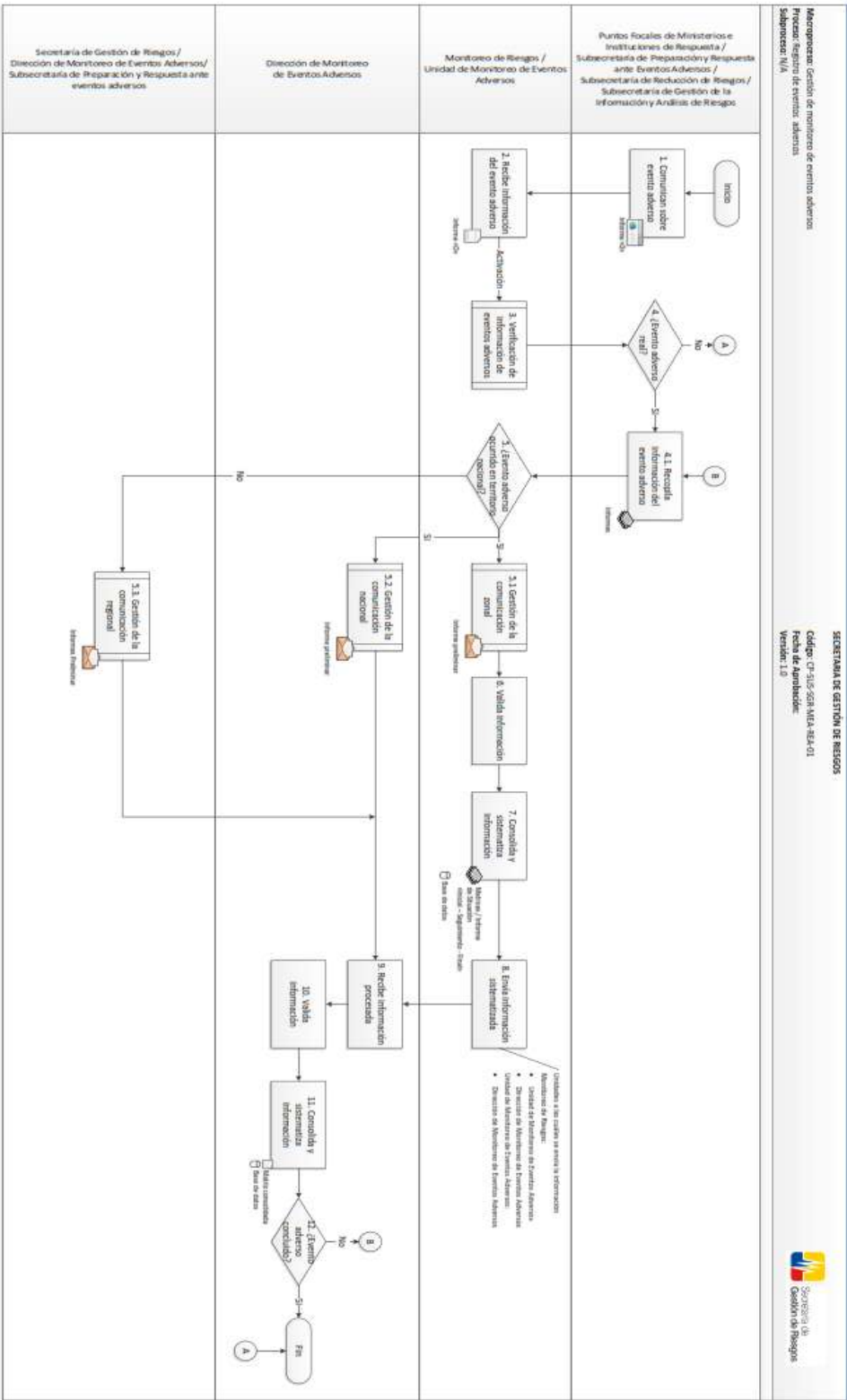
### PRECIPITACIÓN PROMEDIO ANUAL EN EL ECUADOR. FUENTE: MAE, 2013



## ANEXO B

### DIAGRAMA DE FLUJO DE PROCEDIMIENTOS PARA GESTION DE RIESGOS

#### DIAGRAMA DE FLUJO



## ANEXO C

### DIAGRAMA DE FLUJO DE PROCEDIMIENTOS PARA GESTIÓN DE RIESGOS

#### DIAGRAMA DE FLUJO

#### SECRETARÍA DE GESTIÓN DE RIESGOS

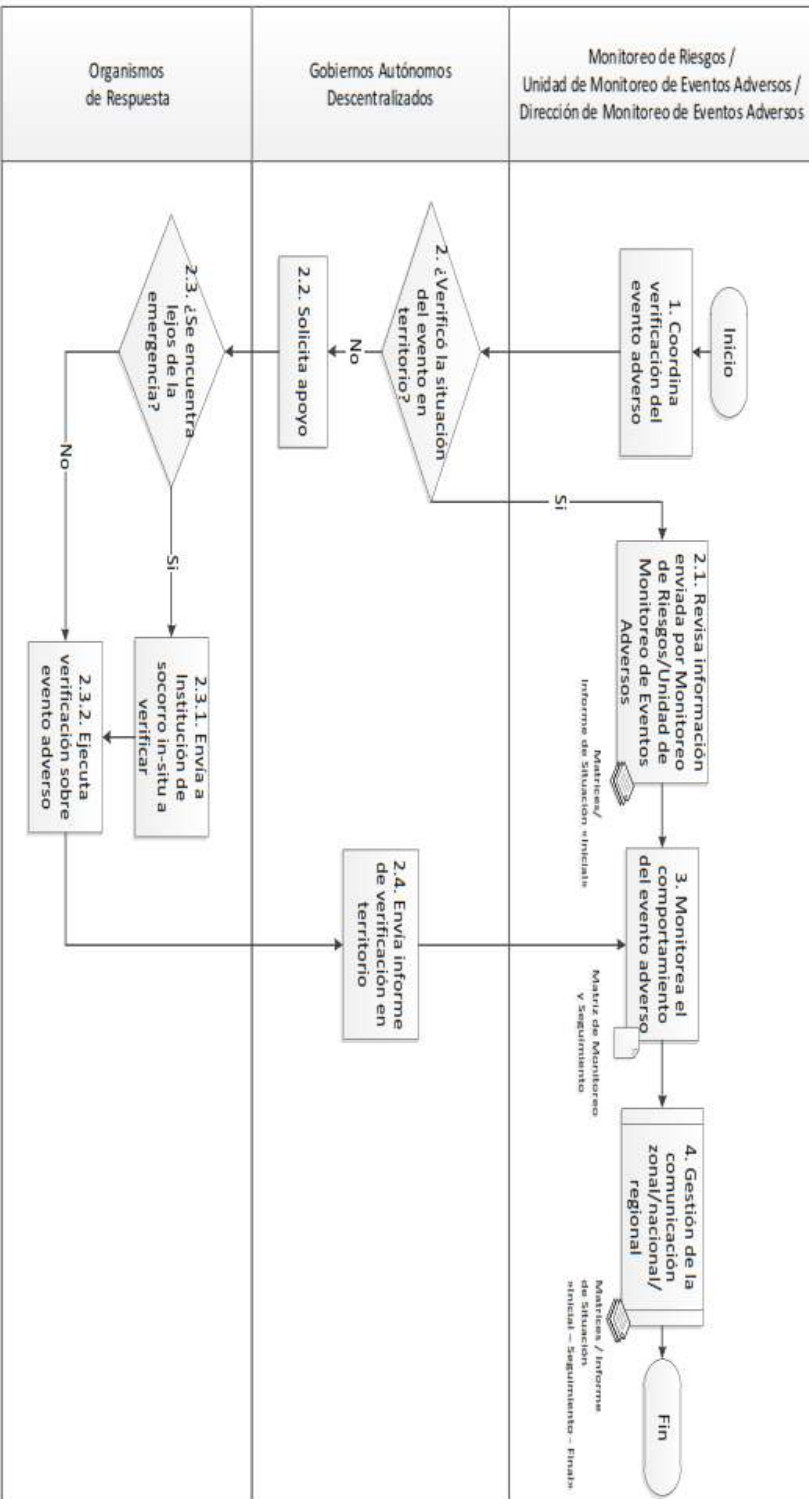
**Macroproceso:** Gestión de monitoreo de eventos adversos  
**Proceso:** Registro de eventos adversos  
**Subproceso:** Verificación de información de eventos adversos

**Código:** CP-SUS-SGR-MEA-VEA-01  
**Fecha de Aprobación:**  
**Versión:** 1.0



Secretaría de  
 Gestión de Riesgos

FAME



## ANEXO D

### RADAR SISTEMA DE ALERTA TEMPRANA DE INUNDACIONES

