Universidad de las Ciencias Informáticas

Facultad 6

Trabajo de Diploma para optar por el Título de

Ingeniero en Ciencias Informáticas

Título: *Componente video-sensor de detección de fuego para el Sistema de Video-Vigilancia Xilema Suria.*

Autor:

Alejandro del Rey Abad

Tutores:

Ing. Reinier Pupo Ruiz

Msc. Rafael Cardero Álvarez

Junio, 2015

"Año 57 de la Revolución"

Declaración de autoría

Yo, Alejandro del Rey Abad, con carné de identidad 91012328106, declaro ser autor de la presente tesis que tiene por título “Componente video-sensor de detección de fuego para el Sistema de Video-Vigilancia Xilema Suria” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ___ días del mes de _____ del año 2015.

Autor: Alejandro del Rey Abad

Tutor: Ing.Reinier Pupo Ruiz

Co-Tutor: Msc. Rafael Cardero Álvarez

Datos de contacto

Ing. Reinier Pupo Ruiz

Graduado de la Universidad de las Ciencias Informáticas (UCI) en el año 2009. Pertenece al Departamento de Desarrollo de Componentes del Centro de Geoinformática y Señales Digitales (GEYSED) de la Facultad 6. Pertenece al proyecto de Video-Vigilancia Xilema SURIA, donde es el responsable del módulo Video-Sensores.

Co-Tutor: Msc. Rafael Cardero Álvarez.

Ingeniero en Ciencias Informáticas (2008). Máster en Informática Aplicada (2011). Trabaja en lo referido al procesamiento de imágenes digitales y el reconocimiento de patrones en general. Ha participado en varios eventos afines a su campo de interés.

Agradecimientos

Quiero agradecer en primer lugar a mis padres por ser padres ejemplares en todos los sentidos, los quiero muchísimo. Gracias por criarme, por ser mis guías, por todos estos años de sacrificio, por brindarme amor y comprensión.

A mi abuelo del alma por siempre estar a mi lado y apoyarme en todas mis locuras.
A mi familia por siempre estar preocupada por mi futuro.
A mi ESPOSA Diana por acompañarme en estos 5 años y siempre estar cuando la necesité.
A Rolo y Guille por ser especiales conmigo y darme la oportunidad de tenerlos como mis hermanos.
A mis amigos de la Universidad, en especial a Yosiel por brindarme su apoyo cuando nadie lo hizo.
A ForeverinCuba por demostrarme que con esfuerzo y sacrificio todo es posible.
A mi grupo de Reggaetón favorito (“SUPERNIGGERS”) por darme la oportunidad de formar parte de ellos.
A Todos los que de alguna forma hayan compartido su tiempo conmigo, infinitas gracias.
A mis profesores, en especial a Odiel y a Vilmavis, gracias por apoyarme cuando los necesité.
En ESPECIAL, quiero agradecer a mis dos tutores, Leodan y Pupo, que gracias a ellos este sueño es posible. Gracias por siempre estar cuando los necesité, Gracias por guiarme por el camino correcto, Gracias por TODO.

Dedicatoria

A mis padres Remigio y Amneris.

A mi abuelo Aurelio.

A mi familia.

A mi esposa Diana.

A mis hermanos Rolo y Guille.

A la Vida.

Resumen

En la Universidad de las Ciencias Informáticas se desarrolló el Sistema de Video-Vigilancia Xilema Suria, el cual utiliza video-sensores para detectar situaciones. Dicho sistema actualmente no es capaz de detectar la existencia de fuego, por lo cual se requiere desarrollar un video-sensor con este objetivo. En esta investigación se aborda el desarrollo de un video-sensor de detección de fuego para el sistema Xilema Suria. Como parte de ello, se estudiaron los algoritmos existentes para la detección de fuego y se diseñó uno nuevo, el cual es implementado en el video-sensor propuesto. El componente video-sensor fue desarrollado utilizando C++ como lenguaje de programación, Qt como marco de trabajo, QtCreator como Entorno de Desarrollo Integrado y OpenCV como biblioteca para el procesamiento de imágenes. Se utilizó la metodología XP para guiar el desarrollo. Se probó el componente y se verificó que está apto para ser utilizado.

Palabras claves: componente, detección de fuego, video-vigilancia, video-sensor.

Abstract

Xilema Suria is video surveillance software developed by University of Informatics Sciences. It is based on video-sensors to detect events. Currently Xilema Suria is not able to detect fire, therefore the development of a video-sensor to achieve such a goaQl is required. This document describes the development of a video-sensor to detect fire as a part of Xilema Suria. Existing algorithms to detect fire on video sequences were reviewed and a new one is proposed in this thesis. Such algorithm is implemented in the proposed component. This video-sensor was developed using C++ as programing language, Qt as framework, QtCreator as Integrated Development Environment and OpenCV as a useful tool to perform digital image processing tasks. eXtreming Programming was the methodology used to drive software development. The component was tested and obtained result evidenced that it is able to be used.

Keywords: component, fire detection, video-surveillance, video-sensor.

Índice

Índice de Figuras

Índice de Tablas

Introducción

El fuego, aunque precursor del desarrollo de la humanidad, es también la causa de accidentes en distintos escenarios de la sociedad. Conocido como un elemento de la naturaleza, es ampliamente utilizado para el trabajo con metales, la cocción de alimentos, entre otros. A pesar de sus múltiples ventajas se reconoce por tener efectos devastadores. Tanto es así que la ocurrencia de incendios ha sido la causa de la pérdida de recursos económicos y de vidas humanas en todas las épocas.

En Cuba “en el período 2010-2011 las pérdidas económicas por los incendios ascendieron a 22 millones de pesos. En la etapa de enero-mayo ocurrieron 695 incendios, que afectaron 24 044 hectáreas, lo cual puede estar provocado por la presencia de un evento de sequía más duradero e intenso” (Juventud Rebelde, 2011).

“El primer teniente Pedro José Rodríguez, de la Jefatura de Bomberos en Cuba, subrayó que 1 586 incendios tuvieron lugar en la vía pública y 1 317 en el sector residencial, durante el año 2013. Esto representó el 32,39 por ciento del total de incendios registrados, focalizados en su mayoría en la capital, Santiago de Cuba, Villa Clara y Matanzas” (Juventud Rebelde, 2013). Esto ha provocado que los incendios sean motivo de atención en lo que a la seguridad respecta.

A lo largo de la historia se han empleado distintos mecanismos para vigilar los objetos de interés. Dichos mecanismos han estado en correspondencia con el desarrollo existente en el momento de utilizarlos.

Una de las opciones que se tiene en cuenta para detectar posibles incendios, es la instalación de sensores de calor o sensores de humo. Esta alternativa, incluso cuando está probada, podría ser más costosa desde el punto de vista económico, pues el área que controla cada dispositivo es reducida y en consecuencia se debe instalar una gran cantidad de ellos.

En la era de la tecnología, estos mecanismos se han visto asociados fundamentalmente al desarrollo de sistemas de video-vigilancia. La video-vigilancia “es un método de supervisión por imágenes para detección y registros de ilícitos en tiempo real que pueden ser utilizadas en casos legales o de auditoría” (Greenberg, 2010). Según Francisco Klauser “la video-vigilancia es ‘conservadora’ y ‘protectora’; conservadora porque se encarga de la conservación del orden público y de la prevención de la conducta antisocial, y protectora pues protege diferentes áreas del peligro” (Klauser, 2004). Un sistema de video-vigilancia sería entonces, aquel que permite detectar las irregularidades en el comportamiento de factores que podrían afectar la seguridad del área vigilada.

El avance tecnológico en las cámaras de seguridad y en los sistemas de video-vigilancia, se apoya en el

desarrollo de video-sensores. Estos son conocidos como un concepto innovador de control y vigilancia, que complementa de forma eficaz y rigurosa el servicio ofrecido por los tradicionales sistemas de trasmisión de video. De esta forma, los vigilantes pueden controlar un número mayor de áreas con una mayor precisión.

En Cuba, la utilización de este tipo de sistema es un hecho tangible. Al igual que en el resto del mundo, las empresas, comercios e instituciones de distintas áreas, optan por la vigilancia de sus medios a través de cámaras de vigilancia. La utilización de sistemas de seguridad avanzados y de video-sensores de distintos tipos, ha impulsado a empresas especializadas en el desarrollo de software a elaborar productos de este tipo para el país.

La Universidad de las Ciencias Informáticas, además de su vital accionar como entidad educativa de alto nivel, vincula la enseñanza con la producción de software. Esto se logra con la constitución organizada de centros de desarrollo, especializados en proyectos con perfiles específicos, que aportan a los productos desarrollados mayor calidad y profesionalidad. Entre estos centros se encuentra “Geoinformática y Señales Digitales”, conocido por sus siglas como GEYSED.

> *“GEYSED tiene como misión desarrollar productos, servicios y soluciones informáticas en el campo del procesamiento de las Señales Digitales y la Geoinformática, contribuyendo a la formación integral de profesionales que responden a las necesidades del progreso científico técnico y socioeconómico, permitiendo del posicionamiento en el mercado nacional e internacional”* (UCI, 2006).

Específicamente el Departamento de Desarrollo de Componentes se especializa en el desarrollo de software, ya sean nuevos sistemas o módulos que incrementen el funcionamiento de los productos existentes. En tal caso se encuentra el Sistema de Video-Vigilancia Xilema Suria, el cual “brinda una plataforma para la video-vigilancia de cualquier entorno que se desee proteger. Este sistema contiene varios módulos que operan en conjunto, teniendo cada cual una tarea en específico” (Hernàndez de Arma, y otros, 2014). Destaca por su interés para la investigación el módulo de Análisis, que se encarga de acoplar un conjunto de video-sensores que faciliten la vigilancia, aportándole mayor valor agregado al sistema.

El avance tecnológico de los sistemas de video-vigilancia en distintas aristas de la seguridad se encuentra en incremento, por lo que la competencia en el mercado aumenta en la medida que estos son fortalecidos. La importancia de mantener el nivel comercial del sistema Xilema Suria, hace necesaria la búsqueda de nuevas funcionalidades que le brinden valor agregado frente a la competencia.

Durante encuentros con desarrolladores del sistema Xilema Suria, se pudo conocer que, a pesar de

implementar un conjunto de funcionalidades encaminadas fundamentalmente a la detección de diferentes comportamientos, ninguna permite determinar la presencia de fuego de forma directa. Lo anterior afecta el nivel competitivo de dicho sistema.

Partiendo de la problemática descrita se plantea como **problema de la investigación**: ¿Cómo contribuir a la detección de fuego utilizando el Sistema de Video-Vigilancia Xilema Suria?

Teniendo como **Objeto de estudio:** Los sistemas de video-vigilancia.

Definiendo como **Campo de acción:** La detección de fuego en los sistemas de video-vigilancia.

Objetivo general: Implementar un componente video-sensor de detección de fuego para el Sistema de Video-Vigilancia Xilema Suria.

Para la presente investigación se plantean como **Preguntas científicas**:

1. ¿Cuáles son los fundamentos teórico-metodológicos asociados a los sistemas de video-vigilancia y la detección de fuego?
2. ¿Cómo desarrollar un componente video-sensor de detección de fuego para el Sistema de Video-Vigilancia Xilema Suria?
3. ¿Cómo validar y verificar el componente video-sensor implementado para el Sistema de Video-Vigilancia Xilema Suria?

Para el cumplimiento del objetivo general especificado se plantean como **Tareas de la investigación**:

1. Determinar los fundamentos teórico-metodológicos asociados a los sistemas de video-vigilancia y la detección de fuego.
2. Describir las herramientas y tecnologías a utilizar en la implementación del componente.
3. Elaborar los artefactos generados por la metodología de desarrollo de software escogida para la implementación del componente.
4. Realizar pruebas de software que permitan validar y verificar el componente implementado.

En la presente investigación se utilizaron los siguientes **Métodos Científicos:**

Métodos teóricos

- **Analítico – Sintético:** Se utilizó para estudiar las características de los video-sensores y de los algoritmos asociados a la detección de fuego. Además su utilización se evidenció al estudiar bibliografía de diferentes autores, para sintetizar las definiciones asociadas a la investigación.
- **Modelación:** Se utilizó para modelar el diagrama conceptual asociado al desarrollo del componente video-sensor de detección de fuego.

Técnica de recopilación de información:

- **Entrevista:** Se utilizó para conocer elementos del Sistema de Video-Vigilancia Xilema Suria significativos para la investigación, aportados por una parte de los especialistas que conforman el proyecto (Ver Anexo #1).

El documento está conformado por tres capítulos, estructurados de la siguiente forma:

- **Capítulo 1:** Se definen los elementos teóricos que sustentan la investigación. Se hace referencia a los conceptos fundamentales que posibilitarán el entendimiento de los sistemas de video-vigilancia y la detección de fuego. Se realiza el estudio de productos existentes que puedan aportar de alguna forma a la solución del problema de la investigación. Además se selecciona la metodología de desarrollo a utilizar y se describen las herramientas y las tecnologías.
- **Capítulo 2:** Se especifican los requisitos funcionales y no funcionales definidos, según las necesidades del cliente y el modelo conceptual elaborado. A partir de estos requisitos el componente es diseñado siguiendo las fases de la metodología de desarrollo seleccionada. Además se describen los patrones de diseño presentes en la implementación y el estándar de código utilizado.
- **Capítulo 3:** Se muestran los elementos utilizados para la implementación de la solución, a través de la descripción de la solución desarrollada. Se documenta el proceso de pruebas realizadas al video-sensor, describiendo las técnicas utilizadas y los resultados obtenidos.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se describen los principales conceptos que definen a los sistemas de video-vigilancia y a los video-sensores de manera general. Se estudian video-sensores y algoritmos dedicados a la detección de fuego con el fin de conocer su funcionamiento. Se analizan las características de la solución propuesta y del equipo de desarrollo utilizando la Estrella de Boehm y Turner para definir la metodología de desarrollo a utilizar en la construcción del componente. Además, se describen las características de las herramientas y tecnologías que intervienen en los procesos de diseño e implementación del video-sensor para la detección de fuego.

1.1 Conceptos fundamentales

Un paso importante, antes de comenzar el desarrollo de una investigación, es conocer los principales conceptos asociados a ella. El estudio de las definiciones que describen los elementos del objeto de estudio y el campo de acción, posibilita comprender la solución propuesta y sentar sus bases teóricas.

Video-vigilancia

La video-vigilancia es la captación y el tratamiento de imágenes con fines de vigilancia. Esta técnica es una práctica extendida en la sociedad moderna. Generalmente tiene como objetivo garantizar la seguridad de los bienes y las personas, o se utiliza en entornos empresariales con la finalidad de verificar el cumplimiento de los trabajadores (Videovigilancia, 2014).

Basándose en el contexto analizado, el autor de la presente investigación coincide con León Hempel y Eric Töpfer al considerar la video-vigilancia como “un sistema de tecnología de vigilancia por cámaras, que puede ser configurado y utilizado por las autoridades en lugares públicos, para la prevención de la delincuencia y para llevar el control de diferentes peligros como incendios, accidentes y crímenes” (Hempel, et al., 2002)

Más allá de la video-vigilancia como medida de control, su utilización puede abarcar aspectos tan importantes como la prevención de fuego. La utilización de tecnología permite convertir las cámaras de seguridad en un objeto mediante el cual se puede detectar cambios de patrones utilizando el desarrollo de video-sensores.

Video-sensor

Los sensores de manera general son “dispositivos que funcionan como captadores de información de un proceso, por lo que también se denominan captadores o detectores. En general, transforman una magnitud

física en una señal eléctrica[1] de baja potencia o en una señal óptica[2]" (Salazar Guerrero, 2009).

Este concepto llevado al mundo del video hace que se defina un video-sensor como "una herramienta de análisis de video digital que ofrece información significativa proveniente de una secuencia de video. El desarrollo de video-sensores tiene su base en el procesamiento digital de imágenes" (Colomer, 2003).

Fuego

El fuego se puede definir como "un proceso de combustión caracterizado por una reacción química de oxidación (desde el punto de vista del combustible) de suficiente intensidad para emitir la llama y en muchos casos luz y calor" (Galán Esmeralda, et al., 2011).

Detección de fuego

Se entiende por detección de fuego el hecho de descubrir y avisar que hay un incendio en un determinado lugar. La detección de fuego se puede realizar mediante (Muñoz, 1990):

- Detección humana.
- Una instalación de detección automática.
- Sistemas mixtos.

Según David Martín "para contrarrestar el fuego es necesario seguir tres acciones fundamentales: prevención, detección y extinción" (Domènech, 2012). El presente trabajo se enmarca en la detección de fuego mediante el procesamiento de imágenes en la video-vigilancia.

1.2 Características del fuego

Uno de los métodos que se utiliza con frecuencia para detectar fuego se basa en el análisis del modelo de color RGB (Gonzalez, et al., 2007). Para ello se definen umbrales, que pueden ser individuales o múltiples, para cada componente del modelo. Los píxeles que posean valores dentro del subconjunto delimitado por el umbral, son seleccionados como píxeles de fuego.

[1] Tipo de señal generada por algún fenómeno electromagnético. Estas señales pueden ser de dos tipos: analógicas o digitales.
[2] Forma de comunicación que utiliza la luz como medio de transmisión.

"El modelo RGB se basa en que cualquier color puede ser representado como una combinación de los tres colores primarios: rojo, verde y azul. El modelo RGB es un modelo aditivo, en el que la combinación de los tres colores forman el color blanco y la ausencia de los tres el negro" (Ibraheem, et al., 2012). A continuación se muestra el diagrama de representación de los colores primarios y sus combinaciones básicas en el modelo RGB:

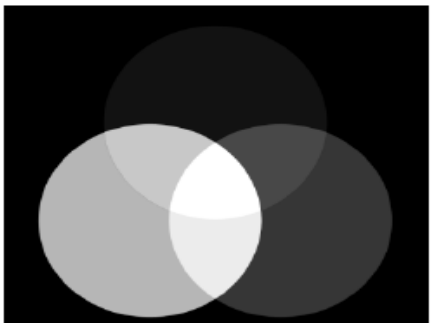

Figura 1. Colores del modelo RGB.

El fuego tiene colores que varían entre el rojo, el naranja y el amarillo (Liu, 2004). El espacio RGB del fuego tiene valores predominantes de rojo, seguidos de verde y por último, menos de azul. Esta caracterización tan clara del color permite poder separarlo con facilidad de otros elementos en la imagen. En imágenes de 8 bits de profundidad, los valores de rojo, verde y azul están en el rango [0,255].

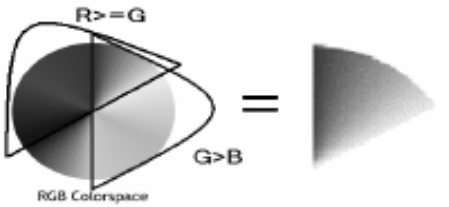

Figura 2. Combinación de los colores RGB que representan al fuego.

En la edición 17 de la Conferencia Internacional de Reconocimiento de Patrones, en el año 2004, se presentó un método para la detección de fuego según su color. La utilización de este método no se centra sólo en los valores del modelo de color RGB, sino también en las propiedades de saturación y de intensidad del fuego. Para esto se utiliza el modelo de color HSL (Liu, 2004).

HSL[3] es una combinación representativa de coordenadas cilíndricas para los colores, más flexible que RGB. El modelo HSL consiste en descomponer el color según criterios fisiológicos (Ibraheem, et al., 2012):

[3] Significa *Hue* (Matiz), *Saturation* (Saturación) *y Lightness* (Brillo).

- **Matiz:** corresponde a la percepción del color.
- **Saturación:** describe la pureza del color.
- **Luminosidad:** indica la cantidad de luz del color.

En el espacio de color HSL el elemento **H** se encuentra de 0 a 360 grados , **S** y **L** del 0 al 100%. El gráfico siguiente es una representación del modelo HSL, donde el color se representa mediante un círculo cromático, y la luminosidad y la saturación se representan mediante dos ejes:

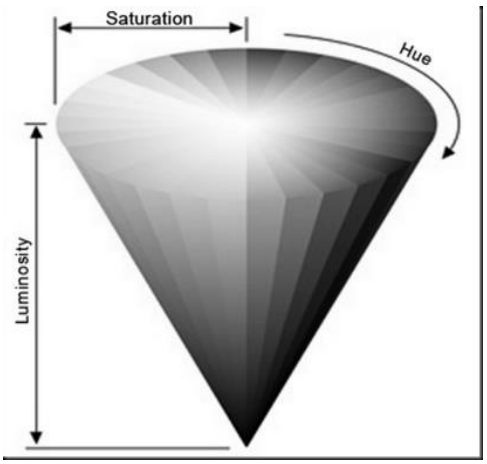

Figura 3. Colores del modelo HSL.

Sin embargo el color no es la única característica para identificar el fuego. Hay algunos objetos con el mismo color del fuego. La principal diferencia entre el fuego y estos objetos es la naturaleza de su movimiento, ya que debido a la circulación de aire y a la quema de materiales, el tamaño y la forma de la llama pueden variar.

Figura 4. Movimientos de las llamas del fuego.

1.3 Soluciones existentes

La prevención de incendios en el mundo no es un tema novedoso. En grandes empresas, entidades y viviendas las personas buscan continuamente alternativas para evitar este tipo de accidentes. Esto ha provocado que la industria de la seguridad brinde alternativas para combatir la ocurrencia de fuego desde distintos puntos de vista, ajustándose a las necesidades existentes. Es por ello que decidir desarrollar una funcionalidad para detectar fuego en el Sistema de Video-Vigilancia Xilema Suria, precisa que se realice el estudio de sistemas homólogos. Este análisis permitirá al autor de la investigación conocer las principales características a tener en cuenta en el desarrollo del componente video-sensor de detección de fuego.

Proyecto Comets

Los UAVs[4] son vehículos aéreos que pueden ser controlados de forma remota o son capaces de realizar tareas de forma autónoma. Se diferencian de los vehículos tradicionales de control remoto, en que como mínimo contienen funcionalidades para la adquisición y transmisión de datos. Algunos UAVs están equipados con instrumentación suficiente para realizar operaciones fuera del campo de vista. Otros son completamente autónomos y pueden realizar tareas tales como seguimiento de trayectorias, aterrizaje, despegue y detección automática de eventos (Ollero, et al., 2007).

El proyecto *COMETS* está concebido para la detección temprana de fuego desde el espacio aéreo, utilizando sensores a bordo y UAVs dedicados a la protección del personal encargado de controlar los incendios. La utilización de varios UAVs aumenta la complejidad del sistema, pero proporciona la ventaja de utilizar las funciones específicas de cada vehículo y obtener información desde distintos puntos de vista.

En este proyecto la vigilancia se realiza repartiendo el área de vigilancia entre los UAVs disponibles, considerando sus capacidades (tiempo y distancia de vuelo), maniobrabilidad y video-sensores a bordo. Durante la vigilancia cada UAV aplica técnicas automáticas de detección de incendios utilizando los video-sensores disponibles. La conclusión de resultados efectivos está relacionada con la comunicación que se establece entre ellos y el sistema central que los controla (Ollero, et al., 2007).

A pesar de detectar fuego, este proyecto no es considerado como solución a la problemática planteada, pues está estructurado para monitorizar el espacio áereo e implica la adquisición de varios UAVs, cuyo precio es muy elevado en el mercado internacional.

[4] *Unmanned Aircraft* Vehicle: Vehículo aéreo no tripulado.

FireWatch

> *"Propone una arquitectura distribuida, con cámaras situadas junto a ordenadores con tareas de pre-procesado de imagen. Después, un ordenador central procesa lo captado por cada cámara y proporciona avisos a los operarios"* (García Fernández, et al., 2009).

Una forma en la que esta arquitectura podría verse simplificada es aglutinando las funciones de preprocesado junto con las de adquisición de datos en las cámaras IP, desventaja que presenta el sistema comercial. Otra desventaja es la utilización de tecnologías propietarias, como *Limax*[5] para apoyar la transmisión de los datos a la estación central de forma inalámbrica, lo que supone problemas en la escalabilidad[6] del sistema (García Fernández, et al., 2009).

Este sistema posee dificultades en la sincronización de los procesos previos a la emisión de alertas pues el sistema se encuentra distribuido en diferentes módulos que transmiten la información a un servidor central, en el cual es analizada. Esta característica muestra diferencias notables con las del sistema Xilema Suria en cuanto a la arquitectura, por lo que sería preciso unificar todas las funcionalidades como un módulo. Teniendo en cuenta que sería necesario reprogramar el sistema como un componente para poder añadirlo a Xilema Suria y que se auxilia de una tecnología propietaria para medir la distancia entre los diferentes puntos de transmisión de información, se decidió descartar *FireWatch* como solución a la problemática planteada.

ForestWatch

Emplea los mismos principios que el sistema *FireWatch*, aunque en este caso centraliza la inteligencia en la estación central, lo que implica una alta cantidad de datos fluyendo desde las estaciones remotas (García Fernández, et al., 2009). Su similitud implica las mismas desventajas que *FireWatch*, agregando que solamente está enfocado para áreas forestales. El Sistema de Video-Vigilancia Xilema Suria comprende, además, las áreas urbanas, por lo que sería necesario ampliar las funcionalidades del sistema. Debido a estas desventajas, dicho sistema fue descartado como una posible solución.

SigniFire

SigniFire de la empresa *Fike* es una solución de video integral para detección de llamas, humo e intrusos. La tecnología *SigniFire* emplea cámaras que permiten detectar visualmente la presencia de un incendio o

[5] Tecnología creada para medir distancia. Su propietario es Ergo-Electronic, compañía especializada en la creación de componentes para medir distancias.

[6] Propiedad de un sistema para manejar su crecimiento continuo y hacerse más grande sin perder calidad.

de humo en el punto de origen, independientemente del flujo de aire en la zona afectada. *SigniFire* tiene como ventaja la alarma temprana en la detección de incendios. Sus principales características son (Fike Corporation, 2012):

- Detecta llamas en segundos.
- Provee video en vivo a locaciones remotas.
- Activa sistemas de alarma contra incendios.
- Ofrece historial en video como evidencia forense para futuras investigaciones sobre incendios.
- Posee capacidad para video vigilancia.

La tecnología puede considerarse como una de las más completas en la actualidad. Es flexible, pues puede configurarse para distintos tipos de cámaras y posee certificado UL/FM[7,8]. Como ventaja adicional se puede mencionar que, las cámaras IP *SigniFire* pueden conectarse con un panel de control que agiliza el proceso de detección y por tanto la emisión de alarmas, cuando existe la presencia de fuego.

A pesar de proyectarse como una solución ideal se debe tener en cuenta que la tecnología no esta libre de pago. El monto total para obtenerla no se encuentra disponible en la red, por lo que es necesario comunicarse con alguna de las empresas autorizadas a comercializarlo. Estas empresas se encuentran en varios países del mundo, dentro de los cuales están Puerto Rico y Venezuela.

Partiendo de que, debido a la necesidad de detectar fuego con la mayor precisión posible, se decida adquirir la tecnología, se debe tener en cuenta que *SigniFire* no es un sistema para instalar en las computadoras, sino una tecnología especial de cámaras IP. En tal caso, además del pago por su obtención, se debe contactar a la empresa que la distribuye cada vez que la infraestructura de las cámaras cambie, para configurar la tecnología de acuerdo a las características que estas poseen. Esta particularidad haría al sistema Xilema Suria dependiente de la tecnología y aumentaría el costo por concepto de mantenimiento. Destacar que esto iría en contra de las políticas de soberanía tecnológica impulsada actualmente por nuestro país.

[7] *Underwriters Laboratories Inc*. (UL) es una organización de certificación de productos sin fines de lucro que efectúa pruebas en los productos y emite certificaciones de validez.

[8] *Factory Mutual Research Corporation* certifica que los productos cumplan con los requisitos especiales que los hacen aptos para trabajar bajo condiciones extremas y en ambientes considerados peligrosos.

Video-Sensor para la Detección de Humo

Este video-sensor enfoca su desarrollo a la detección de fuego mediante la identificación de humo en el alcance de la cámara de vigilancia y se desarrolló para el sistema Xilema Suria. El nivel de compatibilidad entre este componente y la arquitectura del sistema se encuentra probada, además de poseer la documentación necesaria para comprender su funcionamiento (Vázquez Suárez, 2014).

A pesar de esto, no se consideró como solución a la problemática, pues el humo puede extenderse desde otras áreas hasta la zona de la cámara o estar asociado a restos de humo de un fuego ya eliminado. Esto crearía falsas alarmas, confirmando la necesidad de crear un video-sensor más preciso que detecte el fuego directamente, auxiliándose de las caraterísticas propias de las llamas. Aun así por sus ventajas y similitud con la propuesta de solución, es considerado como una guía indispensable para el desarrollo del componente video-sensor de detección de fuego propuesto en esta investigación.

1.4 Video-sensores para la detección de fuego

La detección de fuego en la video-vigilancia está relacionada con la utilización de video-sensores desarrollados con este fin. La forma de detectar fuego puede realizarse utilizando diferentes técnicas, que pueden variar de acuerdo a los tipos de cámaras, el área de vigilancia donde se aplica, o las tecnologías del sistema mediante el cual se hará la detección. Por este motivo, es necesario realizar el estudio de distintos video-sensores para conocer cuáles son las principales características que poseen y qué aspectos son comúnmente analizados en ellos para detectar fuego.

1.4.1 Video-sensor de infrarrojos

Los video-sensores infrarrojos detectan la radiación emitida por los materiales calientes y la transforman en una señal eléctrica. Se puede definir como “un dispositivo electrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión” (Alcántara Silva, 2010).

La mayoría de los cuerpos emiten radiación en forma de puntos infrarrojos. Específicamente el fuego “emite la radiación en forma de bandas, similar a como lo hace un cuerpo negro[9] y varía según la temperatura” (Domènech, 2012).

El uso de bandas infrarrojas para la detección de fuego es uno de los métodos más usados. Las cámaras infrarrojas se identifican según tres bandas de radiación distintas (Domènech, 2012):

[9] Objeto teórico o ideal que absorbe toda la luz y toda la energía radiante que incide sobre él.

- Infrarrojo cercano (0,78 μm – 1,1 μm).
- Infrarrojo medio o térmico (1.1 μm – 15 μm).
- Infrarrojo lejano (15 μm – 100 μm).

Este método a pesar de ser muy utilizado es variable en dependencia de distintas condiciones atmosféricas. En caso de encontrarse en la banda media puede interferir la cantidad de infrarrojo del sol, sin embargo a este nivel no interfiere la tierra. En otras bandas se pueden mencionar afectaciones tales como el tipo de vegetación en latitudes específicas. Estas irregularidades han sido trabajadas con el fin de disminuir las falsas alarmas.

En el año 2007, en la Universidad de Ankara, se desarrolló un sistema que solo es aceptable para distancias menores de 30 metros. El mismo está basado en dos fases (Ugur Töreyin, 2007):

1. Búsqueda de objetos en movimiento, brillantes con bordes abruptos.
2. Análisis de la modificación temporal del fuego.

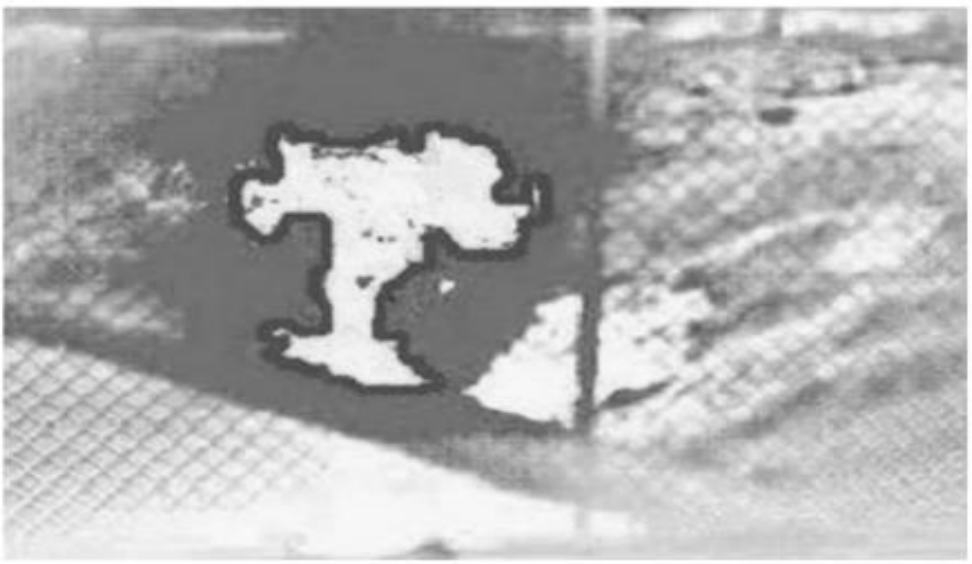

Figura 5. Un fuego a menos de 30 metros desde una cámara infrarroja.

Este tipo de video-sensor no puede ser utilizado como solución pues depende de la utilización de cámaras con tecnología infrarroja para analizar los colores. Por tanto, de ser utilizado, el alcance del sistema Xilema Suria se reduciría, pues solamente sería útil para entidades que cuenten con el tipo de cámaras especificadas.

1.4.2 Video-sensores de espectro visible

Los video-sensores desarrollados para el análisis de espectros visibles[10] se utilizan en distintos modelos[11]. Particularmente, el modelo para la detección de fuego busca parámetros típicos del fuego (color y variación temporal) sobre una imagen en movimiento. En este modelo se propone un sistema de reconocimiento del fuego en proximidad basado en dos fases (Walter, y otros, 2002):

- Se promedian los colores de un pixel[12] a lo largo del tiempo y el color obtenido se compara con los colores de una base de datos entrenada.
- Se observa su variación temporal.

En caso de que en la primera fase los resultados no sean correctos, la segunda se encarga de corregirlos. Aun así, los propios autores referenciados recomiendan utilizar la segunda fase en caso de ser un fuego próximo pues, de estar alejado, se efectuarán las verificaciones para la detección de humo.

Otro método que se utiliza en esta área, es el basado en contornos brillantes que después se comparan con la forma común del fuego. Para determinar si las regiones potenciales de fuego lo son realmente, se realiza un doble análisis sobre ellas (Liu, 2004):

1. **Análisis espacial:** se análizan los coeficientes de Fourier[13] para determinar el contorno del fuego.
2. **Análisis temporal:** se crea un modelo autorregresivo[14] y se buscan los parámetros de la región que se está analizando.

Con los resultado obtenidos de cada uno de los análisis (espacial y temporal), se realiza una clasificación que diferencia fuego de no-fuego.

Este video-sensor realiza el análisis para la detección de fuego de forma profunda y bien estructurada. A pesar de ello no se utilizó porque, para su correcto funcionamiento, precisa de una base de datos con información precisa sobre los valores y rangos adecuados para el fuego. La obtención de esta base de datos se encuentra limitada por lo que la utilización de este video-sensor no aportaría los resultados esperados.

Ninguno de los video-sensores descritos anteriormente resuelve la problemática planteada en la investigación. Sin embargo, la oportunidad de estudiarlos posibilitó reconocer las principales técnicas que

[10] Es el espectro de radiación electromagnética que es visible para el **ojo humano**.
[11] Refiere al **Modelo científico** como representación conceptual o gráfica de fenómenos, sistemas o procesos.
[12] Menor unidad homogénea en color que forma parte de una imagen digital.
[13] Representan las variables a_n y b_n de la serie de Fourier.
[14] **Modelo autorregresivo:** representación de un tipo de proceso aleatorio, que como tal, describe ciertos procesos variables en el tiempo ya sea en la naturaleza, la economía, entre otros.

se utilizan para detectar fuego. Tomando en cuenta los recursos disponibles para llevar a cabo la investigación y las características de la problemática, se seleccionaron para la solución propuesta el análisis de movimiento y de color, bajo un esquema de video-sensor de espectro visible.

1.5 Algoritmos para la detección de fuego

Durante el análisis de los video-sensores se pudo apreciar la utilización del movimiento y el color como rasgos útiles para detectar fuego. Partiendo de esta conclusión, se decidió realizar el análisis de algoritmos que incluyan estos rasgos como solución para la detección de fuego.

1.5.1 Método para la detección de fuego basado en los colores del video

Es un método basado en la visión por computadora para detectar automáticamente la presencia de fuego en secuencias estables de video. El algoritmo se basa no sólo en el color y el movimiento del fuego, sino que analiza también la variación temporal en la intensidad, la variación espacial de color y la tendencia del fuego a agruparse en torno a un punto central. Este método es efectivo en la detección de todos los tipos de fuegos no controlados, en varias situaciones, condiciones de iluminación y entornos (Ebert, et al., 2012).

Este algoritmo está organizado en 5 pasos (Ebert, et al., 2012):

1. Detectar áreas con una luminancia de alta frecuencia de parpadeo utilizando una matriz derivada del tiempo acumulado.
2. Analizar el color de cada fotograma cuyo acumulado en la matriz se encuentra en el rango de color del fuego.
3. Dilatar las regiones móviles con los colores del fuego para determinar si están situadas cerca.
4. Analizar la variación temporal.
5. Analizar la variación espacial.

A continuación se describen los pasos enunciados tomando como referencia el artículo del las autoras Jessica Ebert y Jennie Shipley (Ebert, et al., 2012).

Paso 1: Detectar parpadeo

Una de las propiedades más destacadas del fuego, que lo distingue de otros objetos, es su tendencia a oscilar con una frecuencia entre 1-10 Hz[15]. Las regiones con una luminancia de parpadeo de alta frecuencia son detectadas mediante el uso de un método que deriva el tiempo acumulado, representado en una matriz

[15] **Hercio** o ***Hertz***, unidad de frecuencia del sistema internacional de unidades.

que se compone de las posiciones verticales y horizontales de los píxeles. Esto valores se hallan calculando la intensidad acumulada y el tiempo derivado.

Con el fin de eliminar falsos valores se crea una matriz de luminancia con el mismo tamaño de la matriz de tiempo acumulado y se multiplican. Si el valor de la matriz de tiempo acumulado es mayor o igual a una constante establecida entre la luminancia media y la alta luminancia, se toma el valor de la matriz. De lo contrario el valor del pixel en esa posición será igual a cero. Una vez que todos los píxeles son evaluados se eliminan de la escena aquellos que tienen valor cero o se encuentra cercano a cero.

Paso 2: Analizar color

El color del fuego se encuentra en el rango de colores del rojo al amarillo. La forma de detectar este rango en el fotograma es seleccionando aquellos píxeles donde el valor del color rojo, en el formato RGB, es mayor o igual que el verde y este último debe ser mayor que el color azul. Las posición de los píxeles con coincidencias dentro del rango de color establecido es reflejada en una matriz de color.

Además de la utilización del formato RGB se utiliza el formato HSV[16] que permite representar la saturación y la intensidad en la imagen. En el caso de la saturación, se crea una matriz que contiene los niveles de rojo. Este valor puede variar dependiendo si el fuego es la fuente de luz principal en el video o no, siendo mayor o menor la saturación. La intensidad, por su parte, se calcula definiendo un umbral experimental que debe ser mayor que 0.5. Aquellos píxeles cuyo valor resulte mayor que el umbral definido se multiplican. De esta manera se determina que los píxeles que se asocien por tener una alta intensidad y saturación, sean considerados como candidatos para representar al fuego.

Finalmente se eliminan los píxeles con menor promedio de intensidad y saturación. Además se suprimen aquellos que no poseen una representación fuerte dentro de los colores RGB definidos para el fuego. El resultado obtenido genera una nueva matriz que contiene las regiones que pueden considerarse, por los píxeles que las representan, como áreas de posible presencia de fuego.

[16] Modelo de color que comprende los componentes de matiz, saturación y brillo.

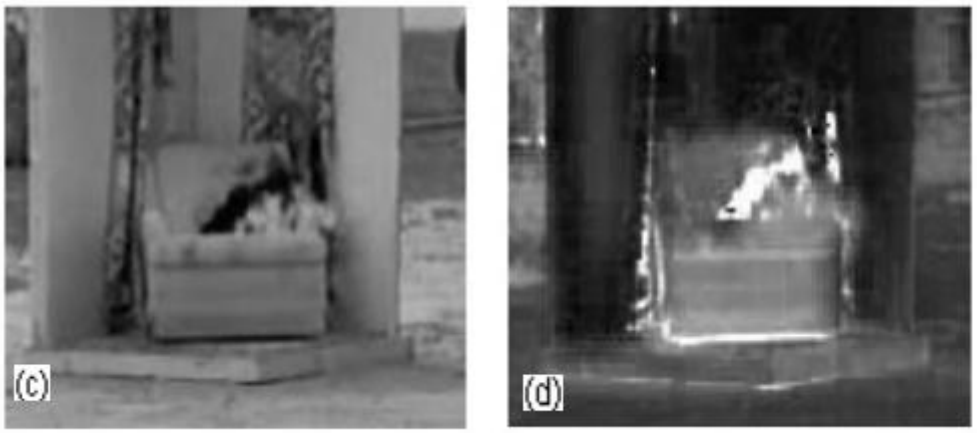

Figura 6. Ejemplo de fuego (c) y con valores de saturación (d).

Paso 3: Regiones en movimiento

Aunque durante el análisis de color se obtuvieron las regiones con probabilidad de presentar fuego, es necesario refinar el resultado para evitar falsas alarmas. Para ello los fotogramas son escalados, permitiendo reconocer aquellas regiones compuestas por solo 1 o 2 píxeles, que deben ser eliminadas (Fig 7). Una vez que se desechan estas regiones, se restaura el fotograma a su tamaño original.

Figura 7. Ejemplo de regiones en movimiento.

Paso 4: Calcular variación temporal de color

El cálculo de variación temporal de color se realiza analizando los cambios, a lo largo del fotograma, que ocurre en los píxeles con los colores del fuego. En tal caso es necesario adicionar los valores R, G y B del pixel y se resta al anterior. Si el valor absoluto de la diferencia entre el pixel, en los dos momentos diferentes,

es mayor que un umbral definido de forma experimental, se considera como un cambio, que se toma como variación. Una vez que todo el fotograma es recorrido, se eliminan aquellos píxeles cuyo valor se encuentra debajo del umbral establecido, quedando solamente aquellos con cambios notables en la imagen.

Paso 5: Variación espacial del color

La eliminación definitiva de los elementos no-fuego se hace analizando la variación espacial de color entre los píxeles con el color del fuego, seleccionados en el paso anterior. El fuego es único, ya que no mantiene un color estable sino que se compone de varios colores que varían dentro de un área pequeña. Esto ayuda a eliminar objetos con el color de las llamas tales como camisas con iluminación consistente.

Para hallar la variación espacial se transforma el video en una máscara binaria que resulta en una nueva matriz, que luego será unida a la original para confirmar la veracidad de los colores. Se establece un rango de ± 3 para cada pixel, que representa el valor máximo y el mínimo de los colores rojo, verde y azul del fotograma. Aquellos valores que representen una gran zona verde se consideran como píxeles de fuego. Esto es porque el valor rojo del fuego tiene un rango limitado de cambio, mientras que el valor verde varía en gran medida (Fig 8).

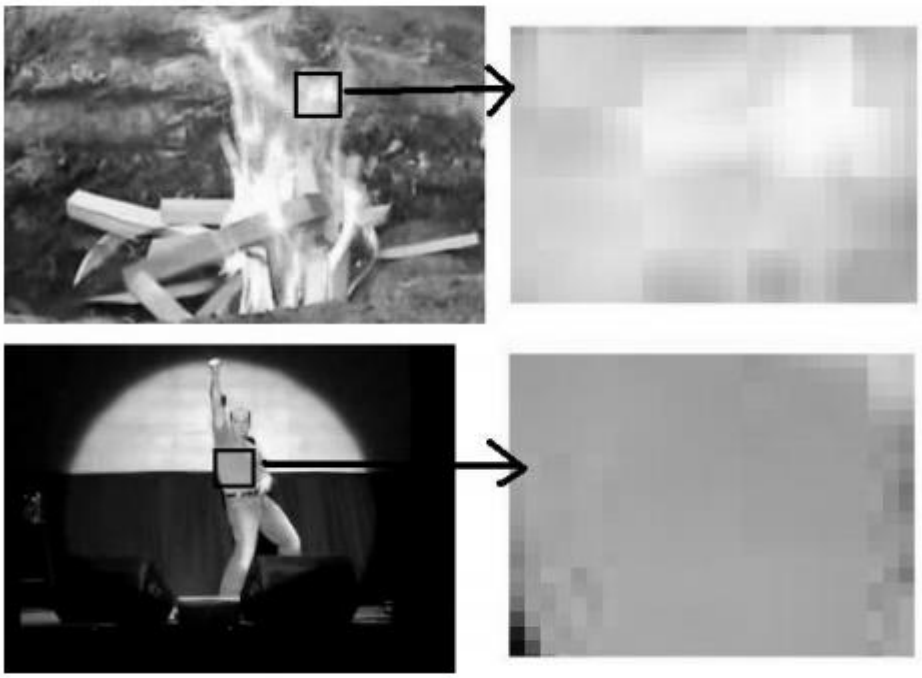

Figura 8. Ejemplo de variación espacial entre el fuego y los objetos con los colores del fuego.

Resultado final

La presencia de fuego se determina finalmente por la relación entre la cantidad de píxeles fuego y no fuego. Si la relación es mayor que un umbral determinado, entonces se considera que hay fuego presente y se emite una alarma.

1.5.2 Modelo de color RGB como base para el algoritmo de detección de incendios en secuencias de video en Wireless Sensor Network

Este algoritmo procesa la información visual, adquirida a través de una cámara estática, que permite incorporar el algoritmo a un sistema de vigilancia de circuito cerrado de televisión. Este método de detección de incendios está basado en la información del movimiento en cualquier imagen de entrada. Para ello se comparan los elementos del fondo con el fin de identificar un primer plano, utilizando el modelo estadístico de mezclas gaussianas[17]. Para analizar el primer plano en las secuencias de video se utilizó el espacio de color RGB. El procedimiento del algoritmo elimina todos los objetos que no cumplan los requisitos de color similares al fuego (Kim, et al., 2013).

Este algoritmo consta de tres pasos: 1) detección de píxeles o regiones en movimiento en el fotograma, 2) detección del color de los píxeles en movimiento y 3) el análisis de las regiones en movimiento del video. El método propuesto se muestra en la siguiente figura (Kim, et al., 2013):

[17] Consiste en un modelo para aproximar la distribución estadística de una señal.

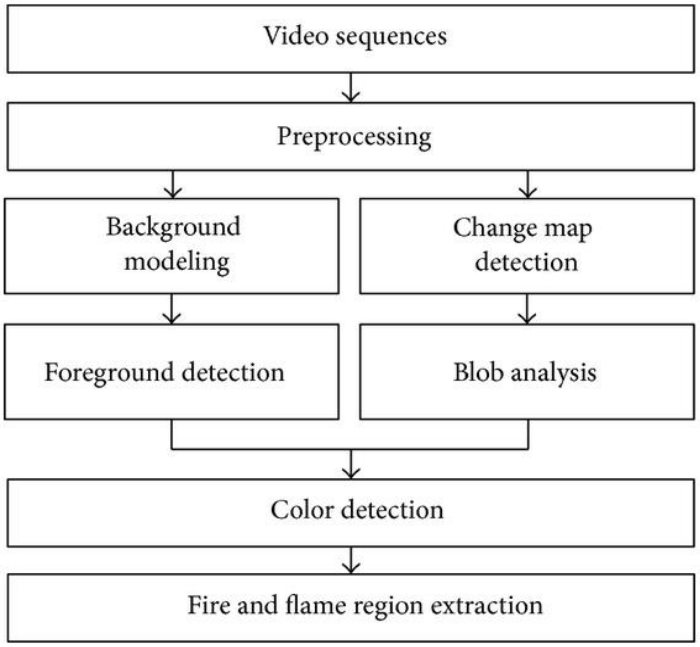

Figura 9. Representación del algoritmo de detección de fuego para el *Wireless Sensor Network*.

1.5.3 Detección de llamas utilizando técnicas de procesamiento de imágenes

El algoritmo propuesto utiliza el enfoque combinado de detección de color, detección de movimiento y la dispersión en el área para detectar fuego en videos. En primer lugar, el algoritmo localiza el color deseado en las regiones de los fotogramas del video. A continuación, se determina la región en el video donde existe movimiento, y en el último paso se calcula el área de píxeles de la trama. El cumplimiento de cada una de estas etapas se basa en la utilización de los modelos de color RGB y HSL para la calcular la intensidad (Patel, et al., 2012).

En la detección de fuego, el modelo RGB se utiliza para determinar la cantidad de color rojo en la imagen. Una imagen de fuego puede ser descrita mediante el uso de sus propiedades de color asociando los tres elementos: R(rojo), G(verde) y B(azul). En el caso del color rojo(R), por ser el dominante, se define un umbral experimental. Una vez que se definen estos aspectos se debe cumplir que R>umbral y R>G>B. El análisis de color, apoyado en la determinación de las regiones en movimiento y las áreas de variación, permite detectar la presencia de fuego. A continuación se describe el algoritmo gráficamente (Patel, et al., 2012):

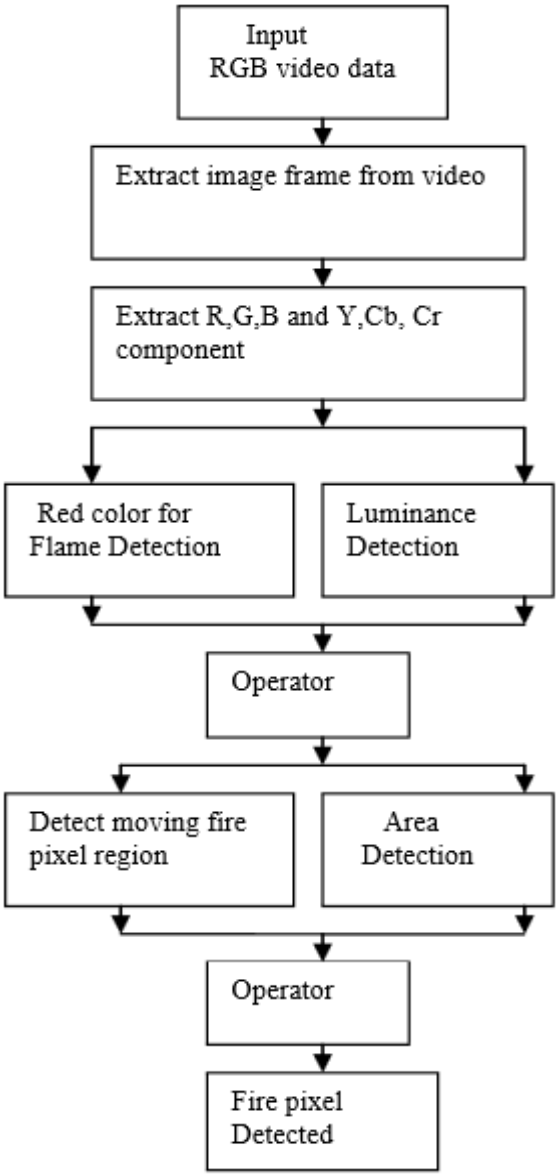

Figura 10. Representación del algoritmo de detección de llamas utilizando procesamiento de imágenes.

1.5.4 Un método rápido y eficiente para detectar fuego utilizando el procesamiento de imágenes

El algoritmo de detección de incendios propuesto consta de dos partes principales: 1) modelado del color del fuego y 2) detección de movimiento. El modelado del color se desarrolla utilizando el espacio de color

en CIE (L*a*b*)[18] para identificar los píxeles de fuego. Además permite identificar diferentes tipos de fuego. El desarrollo del algoritmo tiene como condición que la imagen adquirida del dispositivo sea en el formato RGB. A continuación se describe el algoritmo gráficamente (Celik, 2012):

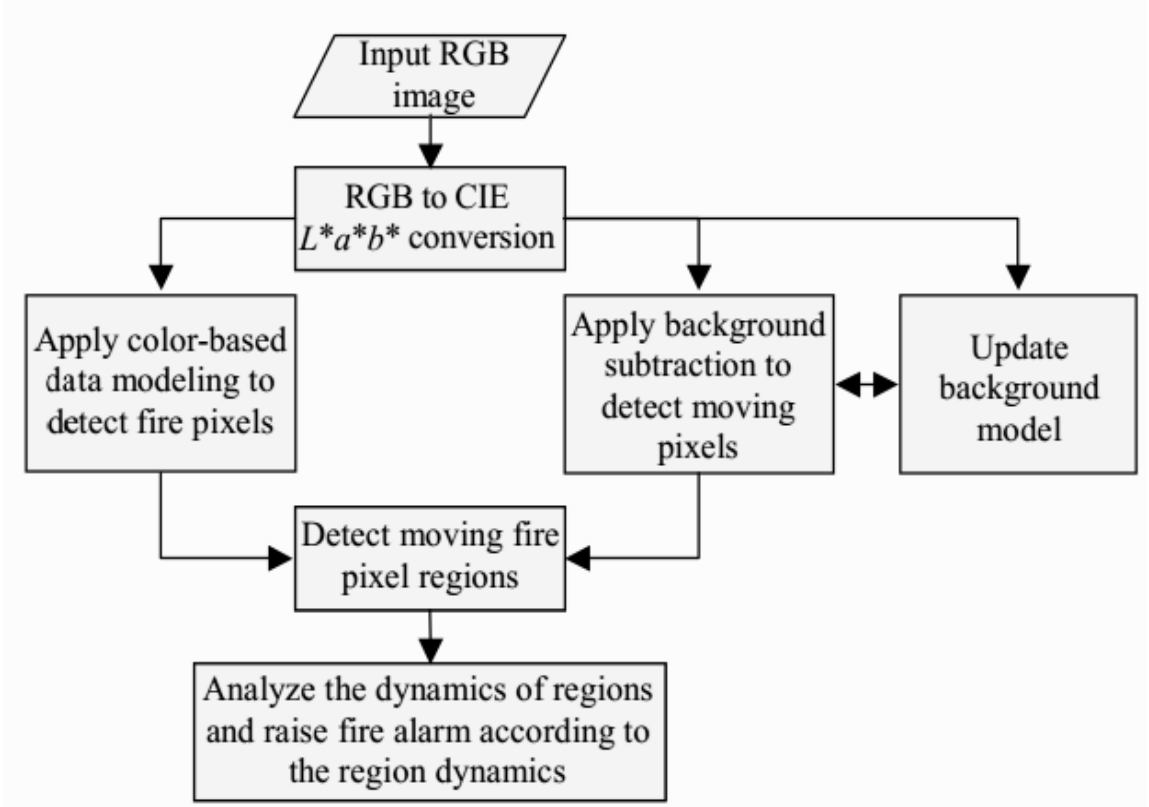

Figura 11. Representación del algoritmo.

Paso 1: Convertir espacio de color RGB a CIE (L*a*b)

Algunas cámaras de video generan una salida RGB pero es necesario que pueda utilizarce otro espacio de color. CIE (L*a*b) es el espacio de color seleccionado en el algoritmo bajo análisis.

Paso 2: Modelado de color para la detección de fuego

Cuando se obtiene la imagen en el espacio de color CIE (L*a*b) se procede a modelar el color según las características del fuego. El rango de color del fuego se encuentra entre el rojo y el amarillo, y generalmente se encuentra cercano al rojo y con mayor iluminación. Para determinar los colores de los píxeles se aplican métricas que devuelven la luminosidad **L** del color y los valores **a** y **b** respectivamente,

[18] Modelo cromático usado normalmente para describir todos los colores que puede percibir el ojo humano.

teniendo en cuenta el total de píxeles y la ubicación de cada uno de ellos en la imagen.

Utilizando la característica del fuego, de que su color está cercano al rojo, lo valores obtenidos se normalizan en 0 y 1, descartando aquellos píxeles que sean igual a 0. Una vez que se obtiene la máscara binaria con los píxeles de fuego se evaluan 3 planos distintos dentro del espacio de color, siendo estos (L*-a*), (L*-b*), y (a*-b*). Para evaluar los planos se establecen un conjunto de niveles de evaluación, donde se estima un valor que se prueba para cada posible resultado del vector <L,a,b>. Este valor es normalizado en el rango de [0,1] para cada conjunto de datos, determinándose que mientras mayor sea el valor del pixel en cuanto al valor estimado, mayor será la probabilidad de que sea un pixel de fuego.

Paso 3: Detección de píxeles en movimiento

El primer paso es calcular la diferencia en el mapa del histograma binario entre dos entradas consecutivas. Al mismo tiempo, la diferencia del mapa binario de fondo se genera mediante la comparación de la trama de entrada actual con el histograma de fondo, almacenado en el *buffer*. Este mapa se utiliza como información fundamental para la detección de píxeles en movimiento.

En el segundo paso, de acuerdo con el mapa de diferencia de los últimos histogramas, se analizan los cambios en el tiempo de los píxeles de la imagen. Aquellos píxeles que no se mueven, se consideran como un registro poco fiable. En este paso se mantiene almacenado en *buffer* el fondo actualizado, así como un registro de disponibilidad de un pixel.

En el tercer y último paso, el mapa binario de diferencias de fondo y el mapa binario con las diferencias de los histogramas, se utilizan para crear un mapa binario de píxeles en movimiento. Si el mapa de registro de antecedentes indica que la información de fondo de un pixel está disponible, el mapa de diferencias de fondo se utiliza como el mapa binario de píxeles en movimiento. De lo contrario, se realiza una copia del mapa binario de píxeles en movimiento.

1.5.5 Resultado del análisis de algoritmos

Durante el estudio de algoritmos para la detección de fuego, se comprobó que la mayoría de ellos utilizan el modelo de color RGB para evaluar los colores del fuego, siendo el rojo el punto de atención. Además se denotó la reiterada selección de regiones en movimiento para estrechar el área de análisis en el video.

De esta manera se propone, para el desarrollo de la solución, realizar la selección de las regiones en movimiento mediante el modelo de mezclas gaussianas propuesto en (Kim, et al., 2013). Además se decidió utilizar las comparaciones entre los modelos RGB y HSL, que se proponen en (Patel, et al., 2012).

1.6 Metodología de desarrollo de software

Las metodologías de desarrollo de software permiten determinar qué hacer en cada fase de trabajo en un proyecto. Con su utilización se evitan contradicciones entre los miembros del equipo de desarrollo y se facilita la interacción con los clientes del producto final (Boeras Velázquez, et al., 2012).

Se pueden clasificar en tradicionales o ágiles, y su uso está determinado por las características de cada proyecto en particular. Si el producto a desarrollar cuenta con recursos monetarios, humanos y de tiempo suficientes para su construcción, es recomendable la utilización de una metodología tradicional, pues se ajusta a estas condiciones. Por otra parte si cuenta con un equipo de trabajo pequeño, con pocos recursos y escaso margen de tiempo, es entonces factible utilizar metodologías ágiles.

Para la selección de la metodología a utilizar se empleó el Método de Boehm y Turner, que permite caracterizar el proyecto de software a partir de 5 criterios y estimar cuán ágil o prescriptivo podría ser el enfoque a utilizar. Los criterios son (Boeras Velázquez, et al., 2012):

1. **Tamaño**: Se utiliza para representar el número de personas involucradas en el proyecto.
2. **Criticidad**: Se utiliza para evaluar la naturaleza del daño ocasionado por defectos que no hayan sido detectados al producto.
3. **Dinamismo**: Representa la rapidez con la que pueden estar cambiando los requerimientos del proyecto.
4. **Personal**: Representa la proporción del personal con experiencia alta, media y baja.
5. **Cultura**: Las organizaciones y las personas que relaciona el proyecto pueden depender de la confianza o de la relación contractual.

La figura 8 muestra la representación de los criterios analizados en la Estrella de Boehm y Turner. La forma obtenida sugiere con claridad un enfoque ágil (Fig 12), teniendo en cuenta que:

- Los vértices que representan los valores de Tamaño y Cultura, se ubican en territorio ágil.
- El Dinamismo, el Personal y la Criticidad se muestran en un área de incertidumbre donde la agilidad y el formalismo se encuentran balanceados.

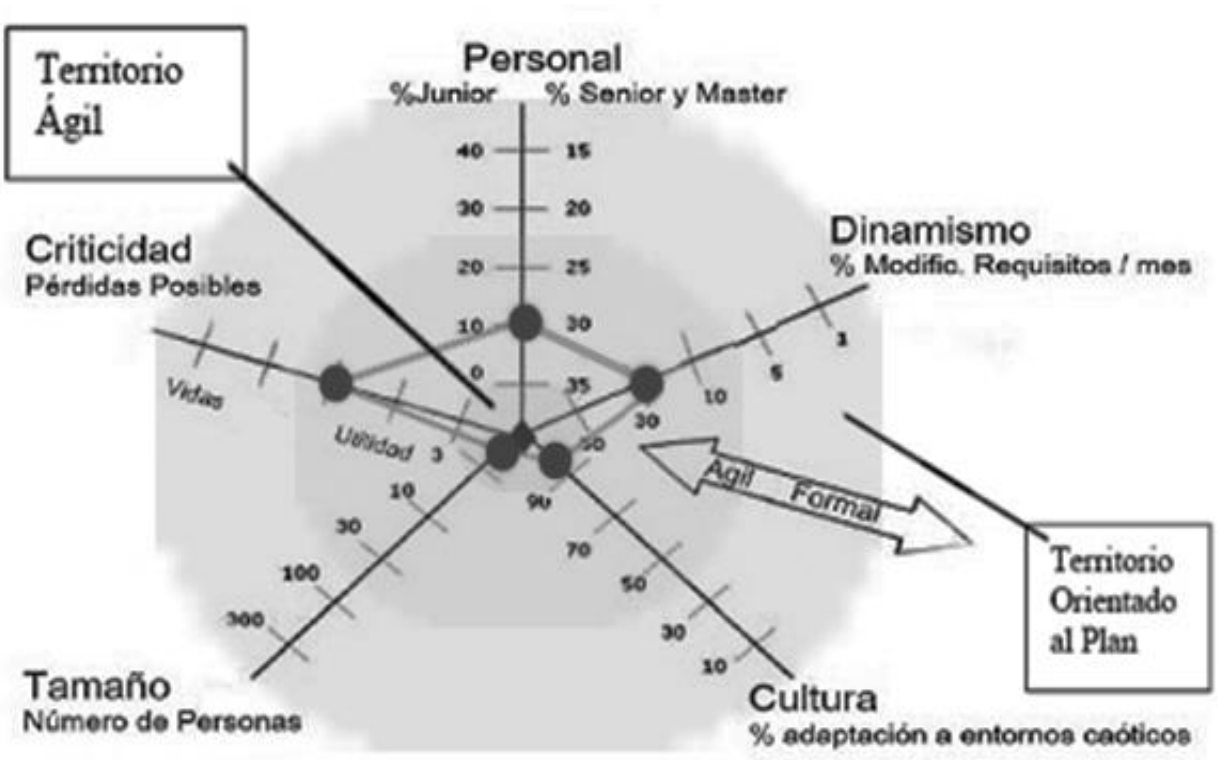

Figura 12. Estrella de Boehm y Turner para el video-sensor de detección de fuego.

En el proyecto donde se desarrolla el Sistema de Video-Vigilancia Xilema Suria, el nivel de formalismo en la entrega de la documentación, el control y las comunicaciones, es bajo. Por tanto, existe alta libertad en el desarrollo del componente, ubicándolo al 95% en el eje de Cultura. El equipo de desarrollo está formado por 1 persona, característica que permite clasificar el equipo de desarrollo como pequeño y se determinó que los requisitos podrían cambiar hasta un máximo del 25%, pues se conoce el proceso a automatizar.

El producto a desarrollar tiene gran impacto social, aunque no necesariamente está relacionado con la protección de vidas humanas. Teniendo en cuenta que este factor depende de la utilización que tenga el componente posteriormente, el valor de criticidad dentro de la estrella se establece como medio. Por otra parte, aunque el desarrollador no se considera experto en la tecnología que se utiliza, la conoce lo suficiente para enfrentar el desarrollo del componente. En tal caso, el punto de evaluación estará en gran medida cerca del centro en el eje de coordenadas. Partiendo del análisis de criterios realizado y los resultados obtenidos mediante su ubicación en la Estrella de Boehm y Turner, se encaminará el desarrollo del video-sensor hacia el uso de las metodologías con enfoques ágiles.

1.5.1 XP (eXtreme Programming)

"Más que una metodología, XP se considera una disciplina, la cual está sostenida por valores y principios propios de las metodologías ágiles. Existen 4 valores que sirven como pilares en el desarrollo de esta metodología" (Echeverry Tobón, y otros, 2007).

- **La comunicación:** en XP la relación con el cliente es tan estrecha, que es considerado parte del equipo de desarrollo.
- **La simplicidad:** no utiliza recursos para la realización de actividades complejas, solo se desarrolla lo que el cliente demanda, de la forma más sencilla.
- **La retroalimentación:** está presente desde el principio del proyecto, ayuda a encaminarlo y darle forma.
- **El coraje:** el equipo de desarrollo debe estar preparado para enfrentarse a los continuos cambios que se presentarán en el transcurso del proyecto.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Define como fases fundamentales: exploración, planificación, diseño, implementación y pruebas. Es una metodología que se basa en la reutilización del código desarrollado (Wells, 2013).

Se utilizó XP porque propone un conjunto de etapas básicas, que pueden servir como guía durante el desarrollo de la solución propuesta. Es ideal para equipos de desarrollo pequeños y permite generar documentación precisa que contenga los aspectos técnicos, permitiendo la entrega del producto a corto plazo. En resumen, porque existe una gran correspondencia entre sus características, las del producto a desarrollar y las del equipo de desarrollo.

1.7 Herramientas y tecnologías

Teniendo en cuenta que el componente video-sensor de detección de fuego debe ser integrado al Sistema de Video-Vigilancia Xilema Suria, se decidió utilizar las mismas herramientas y tecnologías con que este fue desarrollado. Es por este motivo que no se realizaron las comparaciones correspondientes, sino que se enumeran sus principales características.

1.6.1 Lenguaje de modelado 2.0

"El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para

> *especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales"* (Rumbaugh, y otros, 1998).

La utilización del lenguaje UML posibilita la creación del modelo conceptual. Brinda estereotipos formales para representar aspectos relevantes en la investigación.

1.6.2 Herramienta de modelado. Visual Paradigm 8.0

Visual Paradigm es una herramienta UML profesional, que soporta el ciclo de vida completo del desarrollo de software (Quintana Rondon, y otros, 2011). Fue diseñado para facilitar el proceso de análisis y diseño de todo tipo de aplicaciones, soporta los estándares de la notación UML y puede ser usada por los usuarios en la construcción de software con un enfoque orientado a objetos. Además presenta como ventaja que agiliza la construcción de aplicaciones con calidad y a un menor costo.

1.6.3 Lenguaje de programación.

Los lenguajes de programación son idiomas artificiales compuestos por reglas semánticas y sintácticas, escritas mediante símbolos que facilitan la conversión del lenguaje natural al lenguaje de máquina. (Wilson, Leslie B., 1993).

Lenguaje de programación C++

C++ surge como una extensión del lenguaje C. Es por ello que incluye gran parte del lenguaje original, agregando características sofisticadas como el paradigma de Programación Orientada a Objetos, excepciones, sobrecarga de operadores y plantillas. Es un lenguaje potente, con una curva de aprendizaje más alta que C, de mayor entendimiento y multiplataforma (Bravo, 2006).

La utilización de este lenguaje para la implementación del componente, además de su correspondencia con la arquitectura del sistema Xilema Suria, posibilta el acceso a funcionalidades importantes de procesamiento de imágenes que se encuentran escritas en C++. La característica de poseer una comunidad de desarrollo amplia y con experiencia, permite que el desarrollador pueda obtener fácilmente documentación para adquirir los conocimientos sobre dicho lenguaje.

1.6.4 Framework

Los *frameworks* o marcos de trabajo, por su significado en español, se refieren a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que se le pueden añadir las últimas piezas para construir una aplicación concreta" (Carvajal Catagña, y otros, 2010).

Los objetivos principales que persigue un *framework* son (Carvajal Catagña, y otros, 2010):

- Acelerar el proceso de desarrollo.
- Reutilizar el código ya existente.
- Promover buenas prácticas de desarrollo como el uso de patrones.

Framework Qt

Qt es un *framework* de desarrollo de aplicaciones escrito en C++. Compuesto por bibliotecas, Qt proporciona también módulos que facilitan tareas comunes, como acceso a bases de datos, manejo de XML, entre otras. Además posee su propia documentación y sus herramientas de desarrollo (Mínquez Pérez, et al., 2005). Permite reutilizar un conjunto de funcionalidades ya desarrolladas en C++ y contenidas en este marco de trabajo. Apoya la creación de objetos nativos de C++ y le brinda al componente mayor organización desde el punto de vista estructural.

1.6.5 Biblioteca OpenCV 2.4.6.0

OpenCV (*Open Source Computer Vision Library*) es una biblioteca de funciones en C y C++ desarrollada por Intel, que proporciona funciones para tareas de procesamiento de imágenes y visión artificial. Es de código abierto, gratuita, multiplataforma (disponible para entornos MS Windows, Mac OS y Linux) y está desarrollada bajo la licencia BSD[19]. Además es rápida, de fácil uso y se encuentra en continuo desarrollo. (Corporation, 2014).

Las principales funcionalidades utilizadas fueron:

- **resize:** permite redimensionar las imágenes.
- **cvtColor:** permite convertir una imagen de un espacio de color a otro.
- **Mat:** es una matriz para almacenar imágenes. Las imágenes pueden tener 1, 2, 3 ó 4 canales, y distintos tipos de profundidad.

[19] Distribución de Software Berkeley.

- **VideoCapture:** clase que permite trabajar con flujos de video.
- **read:** función de la clase *VideoCapture* que permite leer fotogramas.
- **open:** función de la clase *VideoCapture* que permite capturar videos.
- **BackgroundSubtractorMOG:** clase que permite utilizar el modelo de sustracción de fondo.
- **cvLabel:** pertenece a la librería *cvblob*. Permite la detección y etiquetado de los componentes conexos.
- **cvFilterByArea:** pertenece a la librería *cvblob* y permite filtrar los componentes conexos, según un tamaño definido.

1.6.6 Entorno de Desarrollo Integrado (IDE)

Un Entorno de Desarrollo Integrado (IDE) es un programa que está compuesto por un conjunto de herramientas para un programador. Facilita un marco de trabajo amigable para una gran cantidad de lenguajes de programación tales como C++, Java y C#, logrando utilizarse en el mismo uno o varios lenguajes de programación (Veranes Rodríguez, 2012).

QtCreator 5.3.0

QtCreator es una excelente herramienta para la elaboración de aplicaciones de escritorio y da soporte al lenguaje de programación C++ (seleccionado para el desarrollo). Ofrece gran compatibilidad con varios sistemas operativos, posee riquezas en sus funciones y librerías, que permiten el desarrollo de interfaces sencillas. Posee gran documentación y una comunidad de desarrollo que lo respalda. Además, ha sido adaptado a las características de los entornos de desarrollo actuales y está pensado para sistemas operativos recientes (Corporation, 2014).

Le brinda al desarrollador la herramienta necesaria para implementar las funcionalidades que componen el video-sensor propuesto.

1.8 Conclusiones del capítulo

La descripción de los conceptos asociados al objeto de estudio, sirvió de base para comprender teóricamente la problemática planteada. El análisis de sistemas similares a la solución propuesta ratificó la necesidad de crear un componente video-sensor de detección de fuego, que se ajuste a las características del Sistema de Video-Vigilancia Xilema Suria. El estudio asociado a la detección de fuego, permitió conocer que el análisis de movimiento y color en las secuencias de video, son elementos básicos que pueden servir de punto de partida en el desarrollo del componente propuesto.

La metodología de software XP permitirá organizar el desarrollo del componente. La elección de las mismas herramientas y tecnologías que se usan en el proyecto Xilema Suria, posibilitará aprovechar el conocimiento ya establecido al respecto.

Capítulo 2: Características del componente video-sensor

Una parte importante en el desarrollo del software es el análisis de las características que este debe cumplir. En este Capítulo se describe la solución propuesta, se organiza el equipo de desarrollo de acuerdo a los roles identificados y se especifican los requisitos del componente según las necesidades del cliente y el modelo conceptual elaborado. En la etapa del diseño se elaboran las Historias de Usuarios y las Tarjetas CRC, con el fin de sentar las bases necesarias para la implementación del componente. Además se identifican los patrones de diseño y se define el estándar de codificación a utilizar.

2.1 Solución propuesta

El Sistema de Video-Vigilancia Xilema Suria se compone de módulos que se especializan en distintas funciones, entre los cuales se encuentra el módulo *Analytics*. Este módulo se encarga de agrupar los video-sensores que se desarrollan para el sistema, donde cada uno de ellos está orientado a identificar algún aspecto específico en las imágenes captadas.

El objetivo de la solución que se propone es detectar tempranamente la ocurrencia de fuego, evitando de esta manera, pérdidas materiales y humanas. El componente está dirigido a detectar fuego basado en dos factores fundamentales:

1. **Detección de movimiento:** se analiza la imagen captada por el video-sensor y se identifican todas las regiones en movimiento.
2. **Evaluación de colores:** de las regiones en movimiento, se evalúan los colores de los píxeles y se comparan con los valores, que según los modelos RGB y HSL, deben poseer para ser identificados como fuego.

Los factores expuestos anteriormente fueron seleccionados por ser los más utilizados para detectar fuego en los video-sensores y algoritmos estudiados. Su combinación aumenta la precisión del componente a la hora de detectar fuego.

2.1.1 Modelo conceptual

Un paso importante para diseñar e implementar la solución propuesta es conocer la problemática planteada. Para ello se elaboró el modelo conceptual que la representa. Este modelo se describe a continuación:

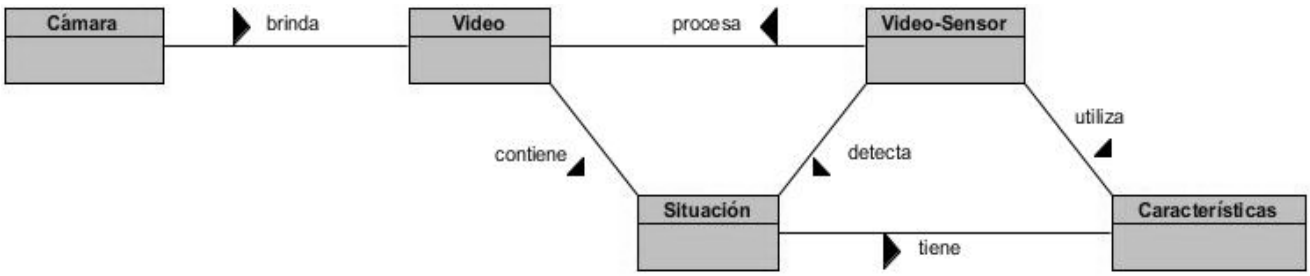

Figura 13. Modelo conceptual de la problemática.

Glosario de términos del modelo conceptual

- **Cámara:** cámara IP para la captura de los flujos de video utilizada en el sistema Xilema Suria.
- **Video:** secuencia de imágenes, transmitida por la cámara IP, en la que puede existir o no presencia de fuego.
- **Situación:** sucesos en el video que pueden ser captados por el video-sensor.
- **Características:** especificidades que diferencian una situación de otra.
- **Video-sensor:** componente que procesa el flujo de video para detectar una situación.

2.2 Concepción del Sistema

Cuando se detecta una problemática y su solución se asocia con el desarrollo de un producto informático, es necesario organizar algunos aspectos importantes para obtener mejores resultados. De acuerdo con lo propuesto por la metodología XP, seleccionada como guía para el desarrollo, uno de estos aspectos es el equipo de desarrollo. Definir los roles de cada miembro del equipo en relación con las tareas que realizan, aumenta la calidad del software y apoya el cumplimiento de las iteraciones que se planifiquen. Además, permite involucrar al cliente en la elaboración del producto y agiliza la entrega de las iteraciones que estén previstas en el desarrollo del componente.

Teniendo en cuenta estos aspectos, se declara durante la concepción del sistema la visión y el alcance del producto a desarrollar. Además se describen los roles de cada participante dentro del equipo de desarrollo, organizando las tareas según los perfiles definidos.

2.2.1 Visión y alcance del componente

El componente video-sensor a desarrollar cuenta con tres fases importantes:

1. Capturar el flujo de video.
2. Detectar la existencia de fuego.

3. Emitir alarma.

En cada una de estas fases se definen elementos específicos que aseguren los resultados esperados, dependiendo una de otra en forma consecutiva.

2.2.2 Planificación del proyecto por roles

Teniendo en cuenta las etapas en las que se desarrolla un software, es necesario contar con tareas bien definidas y el personal calificado para asumirlas. En un primer momento, es importante definir roles e indicarle a cada miembro del equipo el perfil que posee de acuerdo al rol especificado. De esta manera, el proyecto a realizar se encontrará debidamente organizado y las responsabilidades correctamente identificadas.

En el desarrollo del componente video-sensor de detección de fuego se establecieron 4 roles principales, teniendo en cuenta que se trata de un equipo de trabajo pequeño.

- **Programador:** responsable de diseñar y desarrollar las funcionalidades del componente. Debe tener en cuenta los patrones de diseño previstos para el desarrollo, así como definir el estándar de codificación a utilizar. Debe tener dominio de las herramientas y tecnologías definidas. Además debe asegurar la entrega de un producto final que cumpla los requisitos funcionales y no funcionales definidos.
- **Encargado de pruebas:** debe conocer los tipos de pruebas recomendadas por XP, así como aquellas que por particularidades del producto se definan hacer. Es el responsable de elaborar cada caso de prueba y entregar la documentación vinculada a esta fase. Es el rol que declara al software libre de no conformidades y posibilita la entrega del componente como un producto funcional.
- **Consultor:** es la persona con la debida experiencia, tanto del dominio del problema como de las herramientas y tecnologías.
- **Cliente:** en el caso de la metodología de desarrollo XP, la persona o entidad que solicita el producto es un miembro del equipo de desarrollo. En tal caso la responsabilidad del cliente es participar en cada fase del desarrollo del software, emitiendo sus criterios y valoraciones a partir de los cuales fluctúan las decisiones que se tomen. Es el responsable de verificar que se cumpla con los requisitos propuestos y, en caso de ser necesario, modifica alguno de ellos. Participa de forma activa en la elaboración de las pruebas de aceptación que se le realizan al software.

A continuación se asignan estos roles a los miembros del equipo de desarrollo:

Tabla 1. Planificación del proyecto por roles.

Rol	Responsable
Programador	Alejandro del Rey Abad
Encargado de pruebas	Alejandro del Rey Abad Reinier Pupo Ruiz
Consultor	Reinier Pupo Ruiz
Cliente	Reinier Pupo Ruiz

2.3 Fase de Exploración

Durante esta fase se determinan los aspectos previos necesarios para el diseño y desarrollo de la solución.

> "Los clientes plantean a grandes rasgos las Historias de Usuarios que son de interés para la entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizan en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema" (Letelier, et al., 2007).

2.3.1 Especificación de requisitos

"En la ingeniería del software y el desarrollo de sistemas, un requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Los requerimientos son declaraciones que identifican atributos, capacidades, características y cualidades que necesita cumplir un sistema" (Alegsa, 2015). Los requisitos se dividen en dos categorías: funcionales y no funcionales. Comprender cada uno de estos requisitos, es importante para lograr un producto estable y que cumpla con los objetivos planteados.

Los requisitos funcionales se caracterizan por ser parte fundamental para la creación de un software, pues se especifican a partir de las necesidades planteadas por el cliente. De ellos se desglosan las funcionalidades que se implementarán en el sistema. Los requisitos no funcionales, por su parte, describen las características que debe tener el sistema en cuanto a seguridad, confiabilidad, elementos de rendimiento, así como algunos aspectos de hardware que deben tenerse en cuenta para poder instalar y utilizar el software (Alegsa, 2015).

Lista de Reserva del Producto

La metodología XP propone la Lista de Reserva del Producto para especificar los elementos fundamentales en lo que a requisitos se refiere. Este artefacto describe el tiempo estimado en el que se desarrollará la funcionalidad asociada a cada requisito funcional y la persona encargada de realizar la estimación. Además, describe los requisitos no funcionales que debe cumplir el sistema.

En el caso de la solución que se describe, se estimó un tiempo general de desarrollo de 90 días, valorado en alrededor de 12 semanas. En base a esta estimación se le asignó a cada requisito funcional días de desarrollo, teniendo en cuenta su prioridad e importancia.

Tabla 2. Lista de Reserva del Producto.

Prioridad	Ítem *	Descripción	Estimación	Estimado por
Requisitos Funcionales				
Muy Alta				
	RF 1	Capturar flujo de video.	7 días	Programador
Alta				
	RF 2	Extraer fotogramas.	7 días	Programador
	RF 3	Obtener regiones en movimiento.	42 días	Programador
	RF 4	Convertir formato de color.	10,5 días	Programador
	RF5	Evaluar color de las regiones en movimiento.	14 días	Programador
	RF6	Emitir alarma.	7 días	Programador
Media				
Baja				
Requisitos No Funcionales				
	Rendimiento	El procesamiento del flujo de video de la cámara debe ser en tiempo real bajo una tasa de 30fps.		
	Restricciones del diseño y la implementación	Lenguaje: El video-sensor debe ser implementado utilizando el lenguaje C++. Utiliza el *framework* QT.		
	Hardware	Procesador Core i3 3.30GHz o superior. Espacio en disco duro de 512MB como mínimo. 2GB como mínimo de memoria RAM.		

		Monitor VGA o superior.		

2.3.2 Historias de Usuarios

Se utilizan para especificar los requisitos de las aplicaciones software en las metodologías ágiles. Las historias de usuario son tarjetas en donde el interesado describe brevemente (con el fin de que sean dinámicas y flexibles) las características que el sistema debe poseer, sean requisitos funcionales o no funcionales (Suaza, 2013). En lo adelante se encontrará este término como HU. Se realizaron un total de 8 HU de las cuales se muestran 4 en el presente epígrafe, el resto se pueden encontrar en la sección de los Anexos (Ver Anexo #2).

Uno de los beneficios de las Historias de Usuario es que pueden ser escritas en diferentes niveles de detalle. Es posible escribir historias que cubran múltiples funcionalidades. Estas historias más grandes son llamadas Épicas y dado que típicamente no pueden ser finalizadas en una iteración, se les divide en múltiples Historias de Usuario (Cohn, 2009). En tal caso se encuentra el requisito funcional Obtener regiones en movimiento, que por su complejidad, se dividió en tres HU que representan las funcionalidades básicas del requisito.

Tabla 3. Historia de Usuario “Sustraer fondo”.

Historia de Usuario	
Número: HU_3	**Nombre Historia de Usuario:** Sustraer fondo.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Alejandro del Rey	**Iteración Asignada:** 1
Prioridad en Negocio: Alta	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 2 semanas
Descripción: Se encarga de obtener el fondo de la escena del video con los píxeles de color negro, las sombras con los píxeles de color gris y los objetos en primer plano, representados por píxeles de color blanco.	
Observaciones: Para sustraer el fondo debe haberse capturado el flujo de video y extraído los fotogramas.	

Tabla 4. Historia de Usuario “Eliminar sombras”.

Historia de Usuario	
Número: HU_4	**Nombre Historia de Usuario:** Eliminar sombras.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Alejandro del Rey	**Iteración Asignada:** 2

Prioridad en Negocio: Alta	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 2 semanas
Descripción: Se encarga de eliminar los píxeles de color gris en la imagen.	
Observaciones: Para eliminar las sombras se debe haber realizado la sustracción del fondo.	

Tabla 5. Historia de Usuario "Seleccionar regiones en movimiento".

Historia de Usuario		
Número: HU_5	**Nombre Historia de Usuario:** Seleccionar regiones en movimiento.	
Modificación de Historia de Usuario Número: ninguna		
Usuario: Alejandro del Rey		**Iteración Asignada:** 2
Prioridad en Negocio: Alta		**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto		**Puntos Reales:** 2 semanas
Descripción: Se encarga de identificar los píxeles blancos que se encuentran conectados.		
Observaciones: Para seleccionar las regiones en movimiento se debe haber realizado la eliminación de sombras.		

Tabla 6. Historia de Usuario "Evaluar color de las regiones en movimiento".

Historia de Usuario		
Número: HU_7	**Nombre Historia de Usuario:** Evaluar color de las regiones en movimiento.	
Modificación de Historia de Usuario Número: ninguna		
Usuario: Alejandro del Rey		**Iteración Asignada:** 3
Prioridad en Negocio: Alta		**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto		**Puntos Reales:** 2 semanas
Descripción: Se encarga de evaluar los colores de las regiones en movimiento identificadas en los fotogramas. Esta funcionalidad debe comparar los colores característicos del fuego con los colores de dichas regiones para determinar la posible presencia de fuego. Además valora la intensidad y saturación del pixel para hacer más precisa la evaluación y por tanto la conclusión final. Una vez evaluados estos parámetros se decide la existencia de fuego si: ➢ **En el formato HSL:** la saturación y la intensidad se encuentran en el rango definido por un umbral. ➢ **En el formato RGB:** los valores del color rojo se encuentran en el rango definido por un umbral y el verde y azul son menores que este.		

➢ EL 50% o más del total de píxeles de la región evaluada, están contenidos en el rango delimitado por los umbrales establecidos.
Observaciones: Para realizar la evaluación de color debe previamente haberse capturado el flujo de video y detectado las regiones en movimiento. Además se deben establecer los umbrales de evaluación.

2.4 Fase de planificación

La fase de planificación se utiliza para establecer los tiempos estimados en las actividades que son necesarias para lograr la solución a desarrollar. Se define la prioridad de cada Historia de Usuario descrita y las iteraciones que se realizarán en consecuencia con las entregas definidas.

2.4.1 Estimación de esfuerzo

La estimación de esfuerzo está relacionada con las Historias de Usuarios descritas y el tiempo estimado que requiere el programador para desarrollar la funcionalidad asociada a cada una de ellas. La medida que generalmente se utiliza para describir el esfuerzo es el punto, por lo que 1 punto hará referencia a una semana promedio de desarrollo. En caso de que el tiempo estimado no sea exacto, se tomará el punto por mayoría, lo que se explica como sigue: media semana se representa en 0.5 semanas, lo que equivaldría a 1 punto de esfuerzo estimado.

Tabla 7. Plan de esfuerzo de las HU.

No.	Historia de Usuario	Punto de estimación
1.	Capturar flujo de video.	1
2.	Extraer fotogramas.	1
3.	Sustraer fondo.	2
4.	Eliminar sombras.	2
5.	Seleccionar regiones en movimiento	2
6.	Convertir formato de color.	2
7.	Evaluar color de las regiones en movimiento.	2

8.	Emitir alarma.	1

2.4.2 Plan de iteraciones

El desarrollo de las funcionalidades de un software es un proceso que debe realizarse sin apresurarse, evitando en gran medida posibles errores. La metodología de desarrollo XP, aunque ágil, permite que el desarrollo se organice de forma tal que, sin perder el tiempo, el desarrollador disponga del suficiente tiempo para entregar un producto confiable y útil. Una forma que se propone para organizar el momento en que se realiza cada funcionalidad es el Plan de iteraciones, donde se divide el tiempo total de desarrollo en bloques que agrupan una serie de HU. El tiempo de cada iteración estará en correspondencia con las entregas que se definan, concluyendo progresivamente la implementación del software.

Tabla 8. Plan de iteraciones.

Iteraciones	Historias de Usuario	Duración Total
Iteración 1	- Capturar flujo de video. - Extraer fotogramas. - Sustraer fondo.	4 semanas
Iteración 2	- Eliminar sombras. - Seleccionar regiones en movimiento.	4 semanas
Iteración 3	- Convertir formato de color. - Determinar color de las regiones en movimiento. - Emitir alarma.	4.5 semanas

2.4.3 Plan de entrega

Las iteraciones agrupan un conjunto de HU que representan las funcionalidades a desarrollar en cada una de ellas. Una vez que estos aspectos son organizados, es en el plan de entrega donde se especifica la fecha exacta en que se debe culminar la implementación de cada iteración, posibilitando que se tenga un tiempo estimado para la entrega del producto.

Tabla 9. Plan de entrega.

Artefacto	Iteración	Entrega
HU-1	1	1 de marzo de 2015
HU-2		

HU-3		
HU-4	2	29 de marzo de 2015
HU-5		
HU-6	3	30 de abril de 2015
HU-7		
HU-8		

2.5 Fase de Diseño

El correcto diseño de un software permite organizar mejor su implementación y por tanto aumentar la calidad de los resultados que se obtienen. La metodología XP dedica una de sus fases a la descripción de los elementos del diseño a tener en cuenta en el desarrollo del producto.

2.5.1 Tarjetas CRC

Una tarjeta CRC[20] es una ficha de papel o cartón que representa a una entidad del sistema, a las cuales asigna responsabilidades y colaboraciones. Luego, en la etapa avanzada del diseño o ya en la implementación del sistema, las tarjetas CRC se convierten en clases con métodos, atributos, relaciones de herencia, composición o dependencia. Desde un enfoque puramente Orientado a Objetos[21] cada tarjeta CRC es una clase (Casas, et al., 2009).

A continuación se describen las Tarjetas CRC diseñadas para el componente video-sensor de detección de fuego:

Tabla 10. Tarjeta CRC: Clase FireDetection.

Nombre de la Clase: FireDetection	
Responsabilidades	**Clases relacionadas**
Esta clase se encarga de trabajar con las regiones en movimiento del video y aplicarles los parámetros de comparación para determinar la presencia de fuego: ➢ **bool Process(Mat frame)**	BackgroundSustractor ShadowClean BlobRecognize

[20] En español significa, clases, relaciones y colaboraciones.

[21] Es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.

Tabla 11. Tarjeta CRC: Clase VideoProcessor.

Nombre de la Clase: VideoProcessor	
Responsabilidades	Clases relacionadas
Esta clase se encarga de iniciar el proceso de análisis del video. A través de él se captura el flujo de video y se controlan los datos de entrada y de salida: ➢ **VideoProcessor()** ➢ **Mat readNextFrameOpenCV()** ➢ **bool setInputOpenCV(QString filename)** ➢ **processImage(Mat image)** ➢ **void setROI(Rect ROI_rect)** ➢ **Rect getROI()** ➢ **void setSensorType(QString sensorTypeValue)** ➢ **void setDelay(int d)** ➢ **void stopIt()** ➢ **bool isStopped()** ➢ **bool isOpened()** ➢ **void run()** ➢ **QString getSensortype()** ➢ **void setCameraID(QString cameraIDvalue)** ➢ **QString getCameraID()** ➢ **void setShadowDetection(QString shadowValue)** ➢ **QString getsetShadowDetection()** ➢ **int getResizeFactor()** ➢ **void setResizefactor(int factor)** ➢ **void resizeFrame(Mat &frame, Mat &output, int divideFactor)** ➢ **void setFireObject(const firedetection &value)** ➢ **void setSensorPort(int port)** ➢ **int getSensorPort()**	Firedetection

Tabla 12. Tarjeta CRC: Clase BackgroundSustractor.

Nombre de la Clase: BackgroundSustractor.	
Responsabilidades	Clases relacionadas
Esta clase se encarga de modelar el fondo y detectar las sombras en una secuencia de video a través del uso de mezclas gaussianas**:** ➢ **BackgroundSustractor()** ➢ **void process(Mat &frame, Mat &output)** ➢ **Mat getBackground()** ➢ **void setLearningRate(float learningRate)**	

Tabla 13. Tarjeta CRC: Clase ShadowClean.

Nombre de la Clase: ShadowClean	
Responsabilidades	Clases relacionadas
Esta clase se encarga de la eliminación de las sombras producidas por los objetos: ➢ **ShadowClean ()** ➢ **bool getCleanState()** ➢ **void setCleanState(bool state)** ➢ **void process(Mat &frame, Mat &output)**	

Tabla 14. Tarjeta CRC: Clase BlobRecognize.

Nombre de la Clase: BlobRecognize	
Responsabilidades	Clases relacionadas
Esta clase se encarga de etiquetar las regiones en movimiento de la escena en una secuencia de video: ➢ **BlobDetect()** ➢ **void process(Mat &frame, CvBlobs &blobs)** ➢ **void setSensibility(int sensibilityValue)**	

2.5.2 Patrones de diseño

El desarrollo organizado y las relaciones entre los datos que se manejan en un producto de software, dependen de la implementación de un conjunto de patrones establecidos. A continuación se describen los patrones GRASP (*General Responsability Assignment Software Patterns*), los cuales son empleados en el diseño del componente:

Patrones GRASP

> *"Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas. Los patrones intentan codificar el conocimiento, las expresiones y los principios ya existentes. En consecuencia, los patrones GRASP[22] no introducen ideas novedosas; son una mera codificación de los principios básicos más usados"* (Lerman, 1999).

Experto: se encarga de "asignar una responsabilidad al experto en información: la clase que posee la información necesaria para cumplir con la responsabilidad. Si se hacen en forma adecuada, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y se presenta la oportunidad de reutilizar los componentes en futuras aplicaciones" (Lerman, 1999).

En el componente video-sensor de detección de fuego se manifiesta este patrón en la clase *FireDetection,* la cual maneja los datos necesarios para detectar la existencia de fuego según los parámetros establecidos.

Creador: plantea que se debe asignar a la clase B la responsabilidad de crear una instancia de la clase A (Lerman, 1999).

Este patrón se manifiesta en la clase *FireDetection,* la cual maneja objetos de otras clases para realizar la detección de fuego.

Alta Cohesión: en la perspectiva del diseño orientado a objetos, la cohesión (o, más exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realizan un trabajo enorme (Lerman, 1999).

La presencia de este patrón se manifiesta entre las clases *FireDetection* y *BackgroundSustractor*, donde

[22] Hace referencia a los patrones generales de software para asignación de responsabilidades.

cada una se encarga de realizar una parte del trabajo, incluso cuando la relación entre ellas es importante.

Bajo Acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras (Lerman, 1999).

Este patrón se manifiesta en la relación de la clase *ShadowClean* con las otras clases declaradas en el componente, pues en esta no se utilizan instancias de las demás y una modificación en ellas no afecta su comportamiento.

2.5.3 Estándar de codificación

Los estándares de codificación garantizan la comunicación fluida y directa entre los programadores, permitiendo la reutilización y mantenimiento de los sistemas. En el desarrollo del componente video-sensor se utiliza el estándar *CamelCase*. En la descripción de las clases se utiliza *UpperCamelCase*, que establece que la primera letra de cada palabra sea escrita con mayúscula. En el caso de la declaración de variables y métodos, se utiliza *lowerCamelCase* que define que la primera palabra sea siempre en minúscula y la primera letra de las demás palabras, comiencen con mayúscula. Además el código se organiza utilizando cuatro espacios en blanco a partir de la línea anterior. A continuación se muestra un ejemplo:

```
FireDetection::FireDetection()
{
}

bool FireDetection::Process(cv::Mat frame)
{
    bool returnFire = false;

    bgSustract.extractBackground = true;
    bgSustract.setLearningRate(0.001);
    bgSustract.process(frame,foreground);

    if(shadow.getCleanState())
        shadow.process(foreground, foregroundNoShadow);
    else
        foreground.copyTo(foregroundNoShadow);

    BlobsR.process(foregroundNoShadow,blobs);
```

Figura 14. Representación del estándar de codificación.

2.6 Conclusiones parciales

Teniendo en cuenta los aspectos descritos para la solución propuesta y el modelo conceptual realizado, se obtuvieron los requisitos funcionales del componente video-sensor de detección de fuego. Utilizando la Lista de Reserva del Producto se estimó el tiempo de desarrollo para cada uno de ellos, posibilitando una mejor descripción en las Historias de Usuarios.

Se diseñaron las clases del sistema a través de la Tarjetas CRC, lo que permitió reflejar las relaciones y responsabilidades de cada una de ellas. El uso de los patrones GRASP y el estándar de codificación descrito, evidencian el uso de buenas prácticas durante el desarrollo del video-sensor.

Capítulo 3: Implementación y pruebas

El presente Capítulo describe las fases de implementación y pruebas del componente video-sensor de detección de fuego, siguiendo el paradigma que recomienda la metodología de desarrollo XP. De forma individual, como parte de la fase de implementación, se describe el desarrollo de la solución propuesta. La fase de pruebas aborda los elementos que se tuvieron en cuenta para realizar las Pruebas unitarias y los resultados que dichas pruebas arrojaron. De igual manera, se muestran los resultados de las Pruebas de eficacia y las Pruebas de aceptación realizadas.

3.1 Fase de Implementación

3.1.1 Descripción de la solución desarrollada

El componente video-sensor de detección de fuego inicia su funcionamiento cuando se ejecuta el Sistema de Video-Vigilancia Xilema Suria. A partir de ese momento comienzan una serie de pasos que concluyen con la detección o no de fuego en el área vigilada.

A continuación se describen los pasos del algoritmo:

1. **Capturar flujo de video.**

   El flujo de video se captura en tiempo real a través de las cámaras de vigilancia instaladas en un área específica. Una vez que se obtiene el flujo de video, se van leyendo y procesando cada uno de los fotogramas que este posee (Fig 15).

Figura 15. Flujo de video capturado.

2. **Determinar regiones en movimiento.**

   Una vez que se recibe el fotograma, es necesario identificar las regiones en movimiento. Para realizar este paso se utiliza un método de sustracción de fondo que “se basa en el modelo *Mixture*

of Gauss (MoG)[23]. Este modelo permite el aprendizaje del fondo de la escena mediante la combinación de distribuciones de Gauss[24]. A partir de este aprendizaje, se separan del fondo los objetos en movimiento" (Pupo, 2012). De esta forma se obtiene (Fig 16):

- Una máscara de fondo con los píxeles identificados por el color negro.
- Los objetos en primer plano identificados por píxeles de color blanco.
- Las sombras relacionadas con los píxeles en tonos grises.

Figura 16. *Foreground* obtenido de la imagen.

Los píxeles de color negro pertenecen a los objetos estáticos de la imagen y los de color blanco a los objetos que están en movimiento. Con el fin de identificar mejor las regiones en movimiento es necesario eliminar las sombras. Para lograr esto último, se convierten los píxeles de tonos grises en píxeles de color negro. Como consecuencia se obtiene un *foreground*[25] sin sombras (Fig 17).

[23] Se refiere al término de Mezclas Gaussianas.
[24] Conocida también como distribución normal o gaussiana.
[25] Elementos del frente de la escena.

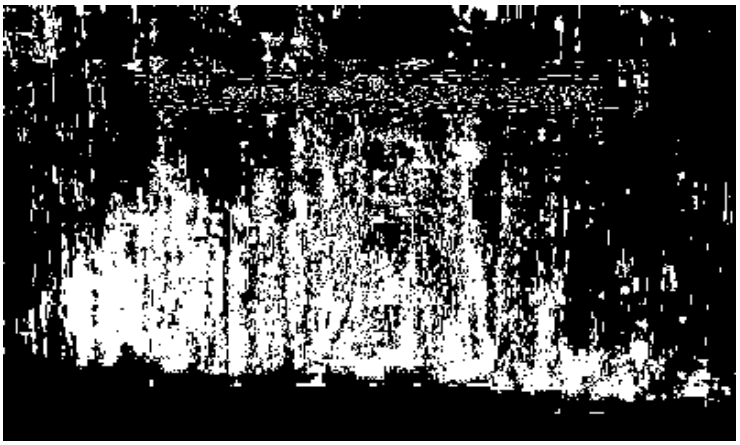

Figura 17. *Foreground* obtenido al eliminar las sombras de la imagen.

El conjunto de píxeles blancos que se encuentran unidos están asociados a los objetos en primer plano y conforman las regiones en movimiento. Teniendo en cuenta que un mismo fotograma puede poseer varias regiones en movimiento, es necesario identificarlas. Para esto se utiliza la librería *cvblob*, la cual devuelve un listado de las regiones en movimiento de la escena. Internamente, esta librería utiliza el algoritmo de componentes conexo[26] para obtener los objetos y algunos descriptores como área, perímetro y rectángulo envolvente.

Debido al ruido introducido en la máscara que contiene los objetos en movimiento, puede ocurrir que estos sean demasiado pequeños por lo que no es de interés evaluarlos. Este problema se soluciona eliminando las regiones en movimiento cuya área es muy pequeña.

Figura 18. Regiones en movimiento identificadas.

[26] Se le denomina a cada uno de los conjuntos conexos de un espacio topológico.

3. **Evaluar colores de las regiones en movimiento.**

   Para evaluar los colores de las regiones en movimiento, estas se recorren pixel a pixel y se evalúan los valores de los píxeles en formato RGB. Con el fin de obtener también los valores de los píxeles relacionados a la intensidad y la saturación, el fotograma es convertido al formato HSL y se realizan las siguientes evaluaciones:

   - Saturación > 20 e Intensidad ≥ 40.
   - Color Rojo > 120.
   - Color Verde ≤ Rojo.
   - Color Azul < Verde.

   Cuando todos los píxeles de las regiones en movimiento son evaluados, se verifica que más del 50% de la cantidad total de píxeles coincidan con los parámetros establecidos. De ser así, la región evaluada se identifica como fuego.

4. **Emitir alarma.**

   Una vez que todas las regiones en movimiento del fotograma son evaluadas, se verifica la existencia de fuego en las mismas. En caso afirmativo se emite una alarma indicando el evento ocurrido y se retornan las regiones asociadas al fuego (Fig 19).

Figura 19. Presencia de fuego identificada.

3.2 Fase de Pruebas

Cuando la fase de desarrollo del software termina, es necesario realizarle un conjunto de pruebas al software para prever y corregir posibles errores que puedan aparecer durante su utilización. La ejecución de las pruebas es un paso importante que depende de las características del producto, y la selección de los tipos de pruebas es responsabilidad del equipo de desarrollo.

3.2.1 Pruebas unitarias

Las pruebas unitarias se describen como "un procedimiento usado para probar que un módulo o método funciona apropiadamente y en forma independiente. A través de ellas se verifica si el bloque de código probado se ejecuta dentro de parámetros y especificaciones concretas. Permiten, además, detectar efectivamente la inyección de defectos durante fases sucesivas de desarrollo o mantenimiento" (Herrmann, 2012).

Estas pruebas se pueden realizar aplicando distintas técnicas dentro del método de Caja Blanca, que pueden ser ejecutadas de forma manual o mediante un programa automatizado. Al componente en cuestión se le aplica la técnica del camino básico y se hace de forma manual. Como parte de ello, se genera un grafo y se le calcula la complejidad ciclomática.

La ecuación que se utiliza para el cálculo de la complejidad ciclomática se plantea como $CC = A - V + 2$ donde:

- **CC:** es la complejidad ciclomática.
- **A:** el número de aristas del grafo.
- **V:** el número de vértices del grafo.

Los rangos de valores para evaluar la complejidad ciclomática calculada son (Martínez Salazar, 2012):

- **1** a **10:** programa simple, sin mucho riesgo.
- **11** a **20:** más complejo, riesgo moderado.
- **21** a **50:** complejo, programa de alto riesgo.
- **50** en adelante: programa de muy alto riesgo.

A continuación se muestra el proceso de pruebas realizado al método *process* de la clase *FireDetection*. Se seleccionó este método por ser el pilar del video-sensor.

En una primera iteración de pruebas, se detectó que el código fuente relacionado con el primer ciclo "*for*" resultaba indefinido por una declaración de variable errónea. Este error fue corregido obteniéndose el grafo de flujo que se muestra en la figura 20.

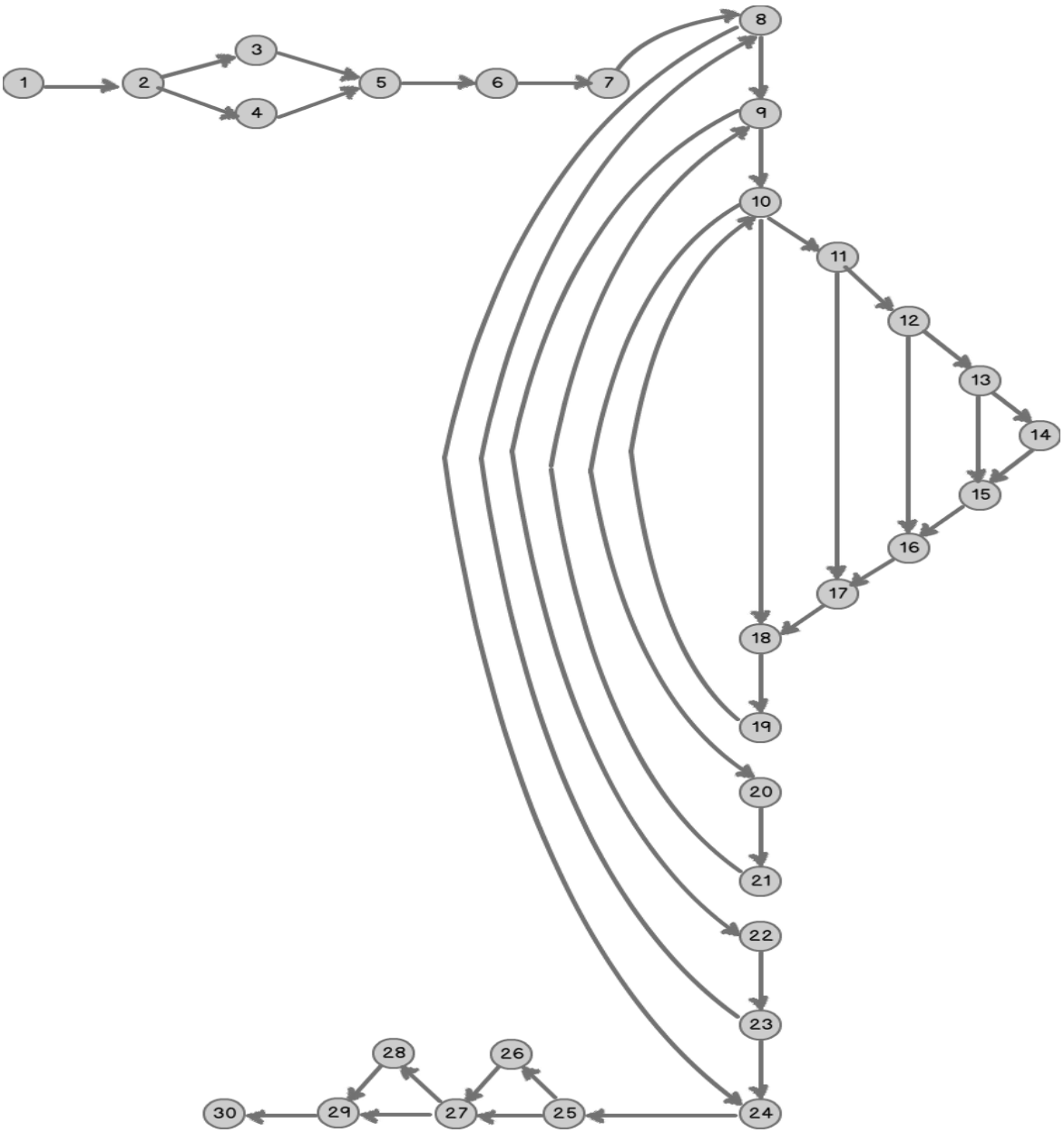

Figura 20. Grafo de flujo del método *process* de la clase *FireDetection*.

La complejidad ciclomática para este grafo es:

$CC = A\text{-}\mathrm{V} + 2$

$CC = 40 - 30 + 2$

$CC = 12$

Siendo 11 el inicio del segundo rango de evaluación de la complejidad ciclomática, se obtiene que el programa es de riesgo moderado.

A continuación se muestran los resultados obtenidos al hacer un caso de prueba para cada camino básico identificado:

Tabla 15. Tabla de resultado de los casos de pruebas para los caminos básicos identificados.

Camino Básico	**Entrada**	**Descripción**	**Resultado**
1,2,3,5,6,7,8,24,25,27,29,30	if(shadow.getRemovalState())	Guarda en una variable la imagen obtenida, con las sombras eliminadas.	Satisfactorio
1,2,4,5,6,7,8,24,25,27,29,30	else foreground.copyTo(foregroundNoShadow)	Copia la imagen obtenida en la variable de salida, porque no fue necesario eliminar las sombras.	Satisfactorio
1,2,3,5,6,7,8,24,25,26,27,29,30 **1,2,4,5,6,7,8,24,25,26,27,29,30**	if(firePixelCant != 0)	Determina si la cantidad de píxeles de fuego detectados es distinto de 0.	Satisfactorio
1,2,4,5,6,7,8,9,22,23,8,24,25,27,29,30 **1,2,4,5,6,7,8,9,22,23,8,24,25,26,27,28,29,30**	for (cvb::CvBlobs::const_iterator it=blobs.begin(); it!=blobs.end(); ++it)	Recibe las regiones en movimiento.	Satisfactorio

1,2,3,5,6,7,8,9,10,20,21,9,22,23,8,24,25,27,29,30	for (int j = rect.y; j < rect.y + rect.height; j++)	Recibe la posición en el eje Y de los píxeles de una región en movimiento.	Satisfactorio
1,2,3,5,6,7,8,9,10,18,19,20,21,9,22,23,8,24,25,27,29,30	for (int i=rect.x; i< rect.x + rect.width; i++)	Recibe la posición en el eje X de los píxeles de una región en movimiento.	Satisfactorio
1,2,4,5,6,7,8,9,10,11,17,18,19,10,20,21,9,22,23,8,24,25,27,29,30	if((pixelhsl[2]>20))	Compara la saturación del espacio de color HSL del pixel con un umbral establecido.	Satisfactorio
1,2,4,5,6,7,8,9,10,11,12,16,17,18,19,10,20,21,9,22,23,8,24,25,27,29,30	if((pixelhsl[1]>=40))	Compara la intensidad del espacio de color HSL del pixel con un umbral establecido.	Satisfactorio
1,2,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,10,20,21,9,22,23,8,24,25,27,29,30	if((pixelrgb[2]>120))	Compara el color rojo del espacio de color RGB del pixel con un umbral establecido.	Satisfactorio
1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,20,21,9,22,23,8,24,25,27,29,30	if((pixelrgb[2]>=pixelrgb[1])&&(pixelrgb[1]>pixelrgb[0])){	Compara los colores del espacio de color RGB del pixel a partir del umbral establecido para el color rojo.	Satisfactorio

3.2.2 Pruebas de eficacia: Precision (P) y *Recall* (R)

Los indicadores "*Precision*" y "*Recall*" se utilizan para medir la efectividad de un algoritmo de clasificación. (Makhoul, et al., 2007).

Estos son definidos por las siguientes ecuaciones:

$$Precision\ rate = \frac{DC}{DC+Fp} \qquad Recall\ rate = \frac{DC}{DC+Fn}$$

Donde:

- **DC:** cantidad de detecciones correctas.
- **Fp:** cantidad de falsos positivos.
- **Fn:** cantidad de falsos negativos.

Los valores de *Precision* y *Recall* se combinan en una métrica F, la cual indica el rendimiento del componente. Mientras mayor sea su valor, mejor se considera el rendimiento del componente. Esta métrica está determinada por la siguiente ecuación:

$$F = 2 * \frac{Precision\ rate * Recall\ rate}{Precision\ rate + Recall\ rate}$$

De la ecuación anterior se deriva un indicador de error, el cual se define como:

$$E = 1 - F$$

En el componente video-sensor de detección de fuego se realizaron 4 pruebas de eficacia. En tres de ellas se contemplaron valores distintos de los umbrales definidos para el color rojo, la intensidad y la saturación. Se realizó una cuarta prueba con los umbrales finales, seleccionados para medir de manera general la eficacia del componente en una mayor cantidad de videos. En todos los casos se tuvieron en cuenta videos con presencia de fuego en zonas urbanas y forestales.

El algoritmo implementado es capaz de procesar un fotograma individual de 320 x 240px[27] de tamaño en 31ms[28], lo cual asegura un procesamiento en tiempo real de videos con una tasa de 30fps, que es la requerida por el componente.

[27] Se refiere a la medida en píxeles.
[28] Se refiere a la medida en milisegundos.

Prueba de eficacia 1

Umbrales: 60< Rojo <90, Intensidad >= 80 y Saturación > 80, para 10 videos.

Tabla 16. Resultados del *Precision* y *Recall* para la primera prueba.

Parámetros	Valores
Detecciones correctas (DC)	5
Falsos Positivos (Fp)	2
Falsos Negativos (Fn)	3
Recall Rate	0.62
Precision Rate	0.71
Media F	0.66
Error E	0.34

Prueba de eficacia 2

Umbrales: 90< Rojo <120, Intensidad >= 60 y Saturación > 40, para 10 videos.

Tabla 17. Resultados del *Precision* y *Recall* para la segunda prueba.

Parámetros	Valores
Detecciones correctas (DC)	7
Falsos Positivos (Fp)	2
Falsos Negativos (Fn)	1
Recall Rate	0.87
Precision Rate	0.77
Media F	0.8
Error E	0.2

Prueba de eficacia 3

Umbrales: Rojo > 120, Intensidad >= 40 y Saturación > 20, para 10 videos.

Tabla 18. Resultados del *Precision* y *Recall* para la tercera prueba.

Parámetros	Valores
Detecciones correctas (DC)	9
Falsos Positivos (Fp)	1
Falsos Negativos (Fn)	0
Recall Rate	1
Precision Rate	0.9
Media F	0.94
Error E	0.06

Una vez que los resultados asociados a cada uno de los umbrales definidos fueron valorados, se determinó que el número de detecciones de mayor eficacia corresponde al umbral donde el color rojo > 120, la intensidad >= 40 y la saturación > 20. A continuación se describe la prueba realizada con estos umbrales para 30 videos.

Prueba de eficacia 4

Umbrales: Rojo > 120, Intensidad >= 40 y Saturación > 20, para 30 videos.

Tabla 19. Resultados del *Precision* y *Recall* para la cuarta prueba.

Parámetros	Valores
Detecciones correctas (DC)	28
Falsos Positivos (Fp)	2
Falsos Negativos (Fn)	0
Recall Rate	1
Precision Rate	0.93
Media F	0.96
Error E	0.04

3.2.3 Pruebas de Aceptación

Las pruebas de aceptación, en lo adelante PA, se realizan cuando se implementa un software a la medida para un cliente, permitiendo que este compruebe el cumplimiento de los requisitos. La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado pruebas Alfa y Beta, para detectar errores que sólo el usuario final puede descubrir. La prueba Alfa se lleva a cabo por un cliente, con el desarrollador como observador del usuario, registrando los errores y los problemas de uso en un entorno controlado. La prueba Beta se lleva a cabo por los usuarios finales en un entorno de trabajo real. El usuario registra todos los problemas (reales o imaginarios) e informa al desarrollador cada cierto tiempo (Pressman, 2010). Se decidió realizar la prueba de aceptación de tipo Alfa al componente video-sensor de detección de fuego, en el laboratorio del proyecto Xilema Suria.

Se realizó 8 PA en correspondencia con el número de Historias de Usuarios. A continuación se muestran 4 de estas pruebas, el resto se encuentra en la sección de los Anexos (Ver Anexo #4).

Tabla 20. Prueba de aceptación “Sustraer fondo”.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU_3-1	**Nombre Historia de Usuario:** Sustraer fondo
Nombre de la persona que realiza la prueba: Alejandro del Rey Abad, Ing. Reinier Pupo Ruiz	
Descripción de la Prueba: Consiste en identificar las regiones que corresponden a los objetos en movimiento.	
Condiciones de Ejecución: Debe haberse capturado el flujo de video y extraído los fotogramas.	
Entrada / Pasos de ejecución: Para evaluar el requisito es necesario: ➢ Sustraer el fondo a través del modelo de Mezclas Gaussianas. ➢ Obtener la máscara con el fondo de la escena, los objetos en primer plano y las sombras.	
Resultado Esperado: Obtener el fondo de la escena del video con los píxeles de color negro, las sombras con los píxeles de color gris y los objetos en primer plano, representados por píxeles de color blanco.	
Evaluación de la Prueba: Satisfactoria	

Tabla 21. Prueba de aceptación “Eliminar sombras”.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU_4-1	**Nombre Historia de Usuario:** Eliminar sombras.
Nombre de la persona que realiza la prueba: Alejandro del Rey Abad, Ing. Reinier Pupo Ruiz	
Descripción de la Prueba: Consiste en eliminar las sombras.	
Condiciones de Ejecución: Debe haberse realizado la sustracción del fondo.	
Entrada / Pasos de ejecución: Para evaluar el requisito es necesario: ➢ Eliminar las sombras convirtiendo los píxeles grises en píxeles negros. ➢ Obtener la máscara con el fondo de la escena y los objetos en primer plano.	
Resultado Esperado: Obtener el fondo de la escena del video con los píxeles de color negro y los objetos en primer plano, representados por píxeles de color blanco.	
Evaluación de la Prueba: Satisfactoria	

Tabla 22. Prueba de aceptación “Seleccionar regiones en movimiento”.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU_5-1	Nombre Historia de Usuario: Seleccionar regiones en movimiento.
Nombre de la persona que realiza la prueba: Alejandro del Rey Abad, Ing. Reinier Pupo Ruiz	
Descripción de la Prueba: Consiste en identificar los píxeles blancos que se encuentran conectados.	
Condiciones de Ejecución: Debe haberse eliminado las sombras.	
Entrada / Pasos de ejecución: Para evaluar el requisito es necesario: ➢ Identificar las regiones en movimiento (*Blobs*). ➢ Etiquetar las regiones identificadas para diferenciarlas dentro del fotograma.	
Resultado Esperado: Obtener el listado de regiones en movimiento del fotograma.	
Evaluación de la Prueba: Satisfactoria	

Tabla 23. Prueba de aceptación “Evaluar color de las regiones en movimiento”.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU_7-1	**Nombre Historia de Usuario:** Evaluar color de las regiones en movimiento.
Nombre de la persona que realiza la prueba: Alejandro del Rey Abad, Ing. Reinier Pupo Ruiz	
Descripción de la Prueba: Consiste en evaluar el valor de los colores de los píxeles en el formato RGB y HSL.	
Condiciones de Ejecución: ➢ Identificar las regiones en movimiento. ➢ Obtener el fotograma en el formato HSL.	
Entrada / Pasos de ejecución: Para evaluar el requisito es necesario: ➢ Recorrer pixel a pixel las regiones en movimiento del fotograma. ➢ Guardar los valores de los píxeles del fotograma original y del fotograma HSL. ➢ Comparar los valores de los píxeles del fotograma original con los colores RGB. ➢ Comparar los valores de los píxeles del fotograma HSL con los valores establecidos para la saturación y la intensidad. ➢ Verificar que más del 50% de la cantidad total de píxeles coinciden con los elementos establecidos.	
Resultado Esperado: Determinar si el fotograma contiene al menos una región en movimiento que contenga los colores del fuego.	
Evaluación de la Prueba: Satisfactoria	

Resultados de las pruebas de aceptación

Las Pruebas de Aceptación se realizaron en correspondencia con el número de Historias de Usuario diseñadas, conformando un total de 8. Para la ejecución de las PA se establecieron los pasos necesarios para ejecutar cada funcionalidad de acuerdo a los criterios del cliente. Una vez seguidos estos pasos y detectadas las deficiencias de cada requisito, se finalizaron las pruebas en dos iteraciones.

En la primera iteración se evaluaron los 6 requisitos funcionales que debe cumplir el componente, detectándose un total de 4 no conformidades, representando el 66,67% del total evaluado. Las no conformidades se relacionaron principalmente con:

- Errores al capturar el flujo de video.

- Selección incorrecta de las regiones en movimiento.
- Definición incorrecta de posibles zonas de incendios al evaluar los parámetros de color. Esta funcionalidad resultó defectuosa por su relación con la selección de las regiones en movimiento.
- Error al emitir la alarma.

Las no conformidades se corrigieron en una segunda iteración, en la cual el 100% de las pruebas fueron satisfactorias.

3.3 Conclusiones parciales

Los resultados obtenidos al probar el algoritmo para detectar fuego son satisfactorios y las pruebas aplicadas al video-sensor evidenciaron que el mismo cumple a cabalidad con los requisitos planteados. Por tanto, el video-sensor puede ser utilizado en un ambiente real para la detección de fuego como parte del sistema Xilema Suria.

Conclusiones generales

Con el desarrollo de esta tesis se dio cumplimiento al objetivo general planteado al comienzo de la investigación. Se puede concluir que:

- Al estudiar el dominio del problema se evidenció que las soluciones existentes no resolvían la problemática planteada, por lo cual se necesitaba desarrollar un video-sensor de detección de fuego para el sistema Xilema Suria.
- El algoritmo propuesto para la detección de fuego se basa en los rasgos color y movimiento, lo cual está en correspondencia con las tendencias existentes al respecto actualmente.
- El lenguaje utilizado, las herramientas y la biblioteca incluida ofrecieron el soporte necesario para lograr un componente que cumpla con los requerimientos y las exigencias planteadas por el cliente.
- El video-sensor desarrollado le confiere un valor agregado al sistema Xilema Suria, lo cual aumenta su competitividad.
- Los resultados obtenidos al probar el componente evidencian que el mismo está apto para ser utilizado en un ambiente real como parte del sistema Xilema Suria para la detección de fuego.

Recomendaciones

- Implementar el algoritmo propuesto en paralelo, permitiendo el procesamiento en tiempo real de videos con mayor resolución y un mejor aprovechamiento del hardware disponible.

Referencias Bibliográficas

Alcántara Silva, Rogelio. 2010. *Análisis de señales y sistemas.* México : Universidad Nacional Autónoma de México., 2010.

Alegsa, Leandro. 2015. ALGENSA. *ALGENSA.* [En línea] DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA , 2015. http://www.alegsa.com.ar/Dic/requerimientos.php.

Boeras Velázquez, Mairelys , y otros. 2012. No. 6, La Habana. Cuba : Serie Científica de la Universidad de las Ciencias Informáticas., 2012, Vol. Vol. 5.

Bravo, Diego. 2006. *Ventajas y Desventajas: Comparación de los lenguajes C, C++ y Java.* Lima : s.n., 2006.

Carvajal Catagña, Edwin Aníbal y Loya Pachacama , Adrián. 2010. *Implementación de un Sistema de Gestión Empresarial en Código Abierto para PYMES.* Ecuador : s.n., 2010.

Casas, Sandra y Reinaga, Hector. 2009. *Aspectos Tempranos: un enfoque basado en Tarjetas CRC.* Colombia : Sociedad Colombiana de Computación, 2009.

Celik, Turgay . 2012. No. 6, Singapore : ETRI Journal, 2012, Vol. Vol. 32.

Cohn. 2009. *User Stories Applied: For Agile Software.* 2009.

Colomer, Antonio Albiol. 2003. *Seguimiento de objeto en secuencia de video.* España : s.n., 2003.

Corporation, OpenCV. 2014. OpenCV. *OpenCV.* [En línea] 2014. [Citado el: 16 de Enero de 2015.] http://opencv.org.

Corporation, QtCreator. 2014. Qt. [En línea] 2014. [Citado el: 16 de Enero de 2015.] http://doc.qt.io/qtcreator.

Domènech, David Martín Borregón. 2012. *Wildfire detection system using image processing tools.* Barcelona : s.n., 2012.

Ebert, Jessica y Shipley, Jennie . 2012. *Computer vision based method for fire detection in color videos.* Estados Unidos : Connecticut College, Utah State University, 2012.

Echeverry Tobón, Luis Manuel y Delgado Carmona, Luz Elena. 2007. *Caso práctico de la metodología ágil XP al desarrollo de software.* Pereira : Universidad Tecnológica de Pereira, 2007.

Fike Corporation. 2012. *SigniFire.* s.l. : Sistemas inteligentes de detección contra incendios., 2012. No. B9118SPA 0412.

Galán Esmeralda, David y Lunes Pino, Francesc-Xavier. 2011. *Comportamiento del fuego según tipologías de forjad.* España : Universidad politécnica de Catalunya., 2011.

García Fernández, Roberto, y otros. 2009. *V Congreso Iberoamericano de Telemática.* Asturias : CITA, 2009. ISBN-10: 978-84-613-2679-2.

Gonzalez, Rafael C. y Woods, Richard E. 2007. *Digital Image Processing.* 2007.

Greenberg, Ing. Enrique. 2010. 2010. [En línea] 09 de septiembre de 2010. [Citado el: 28 de octubre de 2013.]. [En línea] 09 de septiembre de 2010. www.seguridad-online.com.ar.htm..

Hempel, León y Töpfer, Eric. 2002. Urbaneye. *Urbaneye.* [En línea] enero de 2002. [Citado el: 15 de diciembre de 2014.] http://www.urbaneye.net/results/ue_wp1.pdf..

Hernàndez de Arma, Lorena y Bonet Sanabria, Mario Alfredo. 2014. *Componente de Reconocimiento Automàtico de Matrìcula en Vehìculos para Xilema Suria.* La Habana, Cuba : Universidad de las Ciencias Informàticas, 2014.

Herrmann, Elisa. 2012. *Pruebas de Software. Herramientas: Pruebas Unitarias. .* España : Universidad Rey Juan Carlos, 2012.

Ibraheem, Noor A. , y otros. 2012. No. 3, India : ARPN Journal of Science and Technology., 2012, Vol. Vol. 2. ISSN 2225-7217.

Juventud Rebelde. 2013. Cuba : Edición Digital, 2013.

—. 2011. Provocan los incendios 22 millones de pesos en pérdidas económicas. *Juventud Rebelde.* Edición Digital, 2011.

Kim, Yoon-Ho, Kim, Alla y Jeong, Hwa-Young. 2013. *RGB Color Model Based the Fire Detection Algorithm in Video Sequences on Wireless Sensor Network.* Republic of Korea : Humanitas College, Kyunghee University, 2013. 923609.

Klauser, Francisco. 2004. [En línea] 2004. http://www.surveillanceand-society.org/articles2%282%29/switzerland.pdf..

Lerman, Craig. 1999. *UML y Patrones.* México : s.n., 1999. 970-17-0261-1.

Letelier, Patricio y Pedanes, Maria Carmen. 2007. *Metodologías ágiles para el desarrollo del software: eXtreme Programming (XP).* Valencia : Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación., 2007.

Liu, Che-Bin, and Narendra Ahuja. 2004. Vision Based Fire Detection. *USA: IEEE Computer Society.* Proceedings of the Pattern Recognition, 17th International Conference., 2004, Vol. 04.

Makhoul, John, y otros. 2007. *Performance Measures for Information Extraction.* Cambridge : BBN Technologies, GTE Corp., 2007. MA 02138.

Martínez Salazar, Ing. Eduardo. 2012. *Propuesta de procedimiento para realizar pruebas de Caja Blanca*

a la aplicaciones que se desarrollan en el lenguaje Python. . Cuba : Facultad regional de Granma, 2012.

Mínquez Pérez, Eduardo y Garcinuño Enríquez, David. 2005. *Departamento de Informática y Automática.* Salamanca : s.n., 2005.

Muñoz, Ing. José Luis Villanueva. 1990. *Detección de incendios.* Espeña : INSHT, 1990.

Ollero, A. , y otros. 2007. *Sistema basado en el empleo de vehículos aéreos no tripulados para la lucha contra incendios forestales.* Sevilla : Wildfire, 2007.

Patel, Punam y Tiwari, Shamik. 2012. No. 18, s.l. : International Journal of Computer Applications, 2012, Vol. Vol. 58. 0975-8887.

Pressman, Roger. 2010. *Ingeniería del Software. Un enfoque práctico.* 2010.

Pupo, Reinier. 2012. *Video Sensor para el conteo de objetos.* La Habana : Universidad de las Ciencias Informáticas, 2012.

Quintana Rondon, y otros. **Medias, Diseño de la Base de Datos para Sistemas de Digitalización y Gestión. 2011.** No.15, La Habana : LIE-FI-UBA, 2011, Vol. Vol. 8. ISSN 1667-8338.

Rumbaugh, James , Jacobson, Ivar y Booch, Grady. 1998. *El Lenguaje Unificado de Modelado. Manual de Referencias.* s.l. : Series Editors, 1998.

Salazar Guerrero, Ing. René. 2009. *Fundamentos de Mecatrónica.* s.l. : Secretaría de Educación Pública, 2009.

Suaza, Katerine Villamizar. 2013. *Definición de equivalencias entre historias de usuario y especificaciones para el desarrollo ágil del software.* Medellín, Colombia. : Universidad Nacional de Colombia, 2013. 11631.

UCI. 2006. GESPRO. *GESPRO.* [En línea] Universidad de las Ciencias Informàticas, 2006. gespro.geysed.prod.uci.cu.

Ugur Töreyin, B, A Enis Çetin. 2007. *Wavelet based real-time smoke detection in video.* 2007.

Vázquez Suárez, Sergio . 2014. *Video Sensor para Detección de Humo.* La Habana : Universidad de las Ciencias Informáticas., 2014.

Veranes Rodríguez, Alejandro Miguel. 2012. *Módulo web de catalogación de materiales audiovisuales del sistema de captura y catalogación de medias.* La Habana : Universidad de las Ciencias Informáticas., 2012.

Videovigilancia, Guía de. 2014. Agencia española de Protección de Datos. *Agencia española de Protección de Datos.* [En línea] NILO Industria Gráfica, S.A., 2014. [Citado el: 15 de 12 de 2014.] www.aepd.es. NIPO: 052-08-007-8.

Walter, Phillips, Mubarak, Shah y Lobo, Niels da Vitoria. 2002. Flame recognition in video. 2002, Vol. 23, 1-3.

Wells, Don. 2013. eXtreme Programming. *eXtreme Programming.* [En línea] 2013. [Citado el: 14 de enero de 2014.] http://www.extremeprogramming.org.

Wilson, Leslie B. **Languages., Comparative Programming. 1993.** No. 75, s.l. : Addison-Wesley, 1993, Vol. 2da Edición. ISBN 0-201-56885-3.