



“Módulo de recolección de variables mediante cámaras para el Sistema de Medición y Adquisición Arex.”

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor: Arlet Galguera López.

Tutor: Ing. Jordanis Viltres Chávez.

Tutor: Ing. Hurshel Norberto Desouza Noguera.

La Habana, 2019

“Año 61 del triunfo de la revolución”

Declaración de autoría:

Declaro ser el autor del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo el presente a los ____ días del mes de _____ del año _____.

Firma del Autor
Arlet Galguera López.

Firma del Tutor
Ing. Jordanis Viltres Chávez

Firma del Tutor
Ing. Hurshel N. Desouza Noguera.

Datos de contacto:

Tutor: Ing. Jordanis Viltres Chávez

Título de la especialidad de graduado: Ingeniero Informático.

Instituto donde se graduó: Universidad Tecnológica de La Habana (CUJAE)

Correo electrónico: jviltres@uci.cu

Tutor: Ing. Hurshel N. Desouza Noguera.

Título de la especialidad de graduado: Ingeniero en Ciencias Informáticas

Instituto donde se graduó: Universidad de las Ciencias Informáticas (UCI)

Correo electrónico: hurshel@uci.cu

Dedicatoria:

Quiero dedicar este trabajo de diploma a mi abuela que está en el cielo y debe estar muy orgullosa y a mi bebé que viene en camino.

Agradecimientos:

Quisiera agradecer a mi mamá Yurisma López Labaut que es la responsable de que se hayan hecho realidad todos mis sueños gracias a su apoyo incondicional y sus esfuerzos. Gracias mami por ser la mejor madre del mundo y permitirme cumplir todas mis metas y sueños.

Quisiera agradecer a mi prima Zulismary y mis tios Antonia y Felipe por su dedicación y su labor de padres y hermana, gracias por todo su cariño, consejos y apoyo para poder lograr todo lo que he logrado a lo largo de mi vida.

Quisiera agradecer al amor de mi vida Ronaldo Flanes por ayudarme y apoyarme en todas mis decisiones para poder graduarme. Gracias mi amor por darme tantos consejos para convertirme en una mejor persona.

Jambièn quisiera agradecerte que me hayas regalado la mayor bendición de una mujer, el derecho de ser madre y sobre todo apoyar una decisión tan difícil e importante para los dos. Gracias mi amor por todo, eres muy importante para mi.

Quisiera agradecer a Maday Ortega Fernández que más que una amiga se ha convertido en una hermana a lo largo de estos 5 años de mi carrera, gracias por hacer los trabajos conmigo y prestarme tu laptop para poder estudiar y lograr graduarme, gracias por aguantarme y apoyarme.

Quisiera agradecer a mi hermana Niuris y mi abuela Atida por preocuparse por mí y mi evolución como persona.

Quisiera agradecer a todas esas amistades que compartieron gran parte de mi vida y me dedicaron un espacio en sus corazones. Gracias Arianna, Lysbeth, Maria, Suley por ser tan buenas compañeras de apartamento y por pasar tantas noches estudiando juntas y aprendiendo de cada una un poco.

Gracias a mi tutor por aguantarme y resolver todas mis dudas e inquietudes.

Gracias a la Universidad de Ciencias Informáticas por darme la oportunidad de estudiar, superarme y lograr ser una ingeniera en ciencias informáticas.

Gracias a todas aquellas personas que de una forma u otra dieron paso a que yo pudiera cumplir todas mis metas y sueños y que lograra convertirme en lo que soy ahora.

Gracias a todos los integrantes del centro $CFDIN$ por aclararme todas mis dudas y ayudarme a lograr una buena tesis.

Resumen:

La Universidad de Ciencias Informáticas cuenta con varios centros de investigación y desarrollo de software, entre ellos se encuentra el Centro de Informática Industrial (CEDIN) que tiene como objetivo principal desarrollar productos y servicios informáticos para la automatización de procesos industriales, con un alto valor agregado y que cumplan las necesidades y expectativas de los clientes, potenciando la formación especializada y la investigación. El centro mismo contiene varias líneas de investigación como por ejemplo Sistemas Embebidos. Además, contiene un Sistema de Medición y Adquisición llamado Arex que permite el monitoreo y control de procesos de mediana y pequeña complejidad. El objetivo de la investigación es desarrollar un módulo de recolección de variables mediante cámaras para el Sistema de Medición y Adquisición Arex.

En el siguiente trabajo se propone realizar un análisis de los conceptos más importantes relacionados con el tema propuesto. Se determinan las tecnologías y herramientas que se deben utilizar para la implementación y diseño del módulo luego de una extensa búsqueda y análisis, entre las mismas se encuentran el lenguaje de programación C++, el framework de desarrollo Qt, como lenguaje de modelado UML, la herramienta Visual Paradigm para modelar los diagramas del sistema, se utiliza además la biblioteca OpenCV para generar las variables necesarias para el desarrollo del módulo y se selecciona los protocolos de comunicación HTTP y RTSP. Se expone la solución propuesta, definiendo la arquitectura del sistema. Se definen los requisitos funcionales y no funcionales que poseerá la aplicación. Por último, se realizarán las pruebas al sistema para garantizar la calidad, eficiencia y el correcto funcionamiento del software.

Palabras clave: módulo, variables.

Abstract:

The University of Computer Science has several research and software development centers, including the Center for Industrial Informatics (CEDIN) whose main objective is to develop computer products and services for the automation of industrial processes, with a high added value. and that meet the needs and expectations of customers, enhancing specialized training and research. The center itself contains several lines of research such as Embedded Systems. In addition, it contains a Measurement and Acquisition System called Arex that allows the monitoring and control of medium and small complexity processes. The objective of the research is to develop a variable collection module through cameras for the Arex Measurement and Acquisition System.

In the following work we propose to carry out an analysis of the most important concepts related to the proposed topic. The technologies and tools that should be used for the implementation and design of the module are determined after an extensive search and analysis, among which are the C ++ programming language, the Qt development framework, as a UML modeling language, the tool Visual Paradigm to model the diagrams of the system, the OpenCV library is also used to generate the variables necessary for the development of the module and the HTTP and RTSP communication protocols are selected. The proposed solution is exposed, defining the architecture of the system. The functional and non-functional requirements that the application will possess are defined. Finally, tests will be carried out on the system to guarantee the quality, efficiency and correct operation of the software.

Keywords: module, variables.

Índice

Introducción	13
Capítulo 1. Fundamentación teórica	15
1.1 Introducción	15
1.2 Conceptos	15
1.2.1 Domótica	15
1.2.2 Streaming	15
1.2.3 Scripting.....	16
1.2.4 Cámaras IP.....	16
1.2.5 Reconocimiento de Patrones	16
1.2.6 Visión por computadora.....	17
1.2.7 Sistema de Medición y Adquisición Arex	17
1.3 Cámaras IP.....	17
1.3.1 Aplicaciones para VideoVigilancia IP	20
1.3.2 Modelos de cámaras IP.....	21
1.3.2.1 Cámaras de red fijas	21
1.3.2.2 Cámaras de red domo fijas	21
1.3.2.3 Cámaras de red PTZ.....	22
1.3.2.4 Cámaras de red Domo PTZ	23
1.3.3 Instalar y configurar una cámara IP	24
1.3.4 ¿Cómo acceder a una cámara IP?	24
1.3.5 Seguridad de las cámaras IP.	25
1.3.6 ¿Cuándo graban las cámaras? Cámaras con detección de movimiento.	25
1.3.7 Avisos y cámaras IP con alarma.....	25
1.3.8 ¿Dónde graban las cámaras IP?.....	25
1.3.9 Grabar sólo imagen, ¿O también audio?	25
1.3.10 Calidad de imagen tienen las cámaras IP. Cámaras con visión nocturna.....	26
1.3.11 Ángulo de visión de las cámaras IP	26
1.3.12 ¿Tiene que estar conectada la cámara a la red eléctrica?	26

1.3.13	¿Puedo conectarme desde varios dispositivos diferentes a la cámara?	26
1.3.14	Diferencia entre una cámara IP y una cámara wifi	26
1.3.15	Diferencias entre una cámara IP y una cámara CCTV	27
1.4	Soluciones existentes	27
1.5	Análisis y selección de las herramientas y tecnologías	28
1.5.1	Lenguajes de programación	28
1.5.1.1	Lenguajes de programación C++	28
1.5.2	Protocolos de comunicación	29
1.5.2.1	Protocolo RTSP	29
1.5.2.2	Protocolo HTTP	30
1.5.3	Framework	31
1.5.4	IDE de desarrollo	31
1.5.5	Biblioteca	32
1.5.6	Lenguaje de modelado	32
1.5.7	Herramienta de modelado	33
1.5.8	Metodología de desarrollo del software	33
1.5.8.1	Proceso Unificado Ágil (AUP) con el modelo CMMI-DEV v 1.3	33
	Conclusiones parciales	34
	Capítulo 2. Diseño de la solución propuesta	35
2.1	Introducción	35
2.2	Propuesta de solución	35
2.3	Arquitectura	36
2.4	Requisitos Funcionales	38
2.4.1	Historias de usuarios	38
2.5	Requisitos no funcionales	40
2.5.1	Requisitos no funcionales de software	40
2.5.2	Requisitos no funcionales de hardware	40
2.6	Diagrama de clases	40
2.7	Patrones de Diseño	41

2.7.1	Patrones GRASP	42
2.7.2	Patrones GoF	44
	Conclusiones Parciales	44
Capítulo 3. Implementación y pruebas del sistema		45
3.1	Introducción	45
3.2	Diagrama de despliegue del sistema	45
3.3	Pruebas del sistema:	46
3.3.1	Pruebas unitarias.....	46
3.3.2	Pruebas de aceptación.....	48
	Conclusiones Parciales.....	49
Conclusiones		50
Recomendaciones.....		51
Bibliografía.....		52
Glosario de Términos		55

Índice de tablas

Tabla 1. HU1: Cargar el archivo de configuración de Arex, donde vengan especificados los datos para conectarse a cámaras ip.	38	
Tabla 2. HU2: Conectarse a cámaras configuradas y recibir el streaming de video.	39	
Tabla 3. HU3: Generar información en forma de variables procesando el streaming de video de cada cámara.	39	
Tabla 4. HU4: Atender solicitudes de clientes para acceder a las variables generadas.	40	
Tabla 5. Prueba de caja negra: Cargar el archivo de configuración de Arex, donde vengan especificados los datos para conectarse a cámaras ip.	47	
Tabla 6. Prueba de caja negra: Conectarse a cámaras configuradas y recibir el streaming de video.	47	
Tabla 7. Prueba de caja negra: Generar información en forma de variables procesando el streaming de video de cada cámara.....	48	
Tabla 8. Prueba de caja negra: Descripción	Atender solicitudes de clientes para acceder a las variables generadas.	48

Índice de figuras

Figura 1 Ejemplo de aplicación para la videovigilancia IP	21
Figura 2 Ejemplo de cámara de red fija	21
Figura 3 Ejemplo de cámara de red domo Fija	22
Figura 4 Ejemplo de cámara de red PTZ	23
Figura 5 Ejemplo de cámara de red domo PTZ	24
Figura 6 Propuesta de Solución.....	36
Figura 7 Arquitectura del Sistema.....	37
Figura 8 Arquitectura del Sistema.....	37
Figura 9 Diagrama de Clases del Sistema.....	41
Figura 10 Patrón de Diseño Experto.....	43
Figura 11 Patrón de Diseño Creador	43
Figura 12 Patrón de Diseño Singleton	44
Figura 13 Diagrama de Despliegue del Sistema.....	45
Figura 14 Pruebas de Aceptación.....	49

Introducción

Con el auge de las tecnologías han surgido soluciones que tributan al control automatizado de variables en hogares o recintos, a estos sistemas se le pueden integrar cámaras de vigilancia. Un Sistema Domótico es capaz de recoger información proveniente de sensores o entradas, procesarla y emitir órdenes a través de actuadores o salidas. Estos sistemas contribuyen a una mejor calidad de vida de los usuarios facilitando el ahorro energético, fomentando la accesibilidad, convirtiendo el hogar en un lugar más confortable, garantizando las comunicaciones mediante el control y supervisión remoto a través del teléfono, entre otros, pero también aportando seguridad mediante la vigilancia automática de personas, animales y bienes, así como de incidencias y averías.

La Universidad de las Ciencias Informáticas (UCI), cuenta con el Centro de Desarrollo de Informática Industrial (CEDIN) donde se han realizado componentes y aplicaciones para la captura y visualización de streaming de video mediante cámaras. Además, se cuenta con un Sistema de Medición y Adquisición llamado Arex, para el monitoreo de procesos de pequeña y mediana complejidad. Dicho sistema recolecta variables a través de sensores y ejecuta cambios en los procesos mediante actuadores. La observación de procesos utilizando cámaras ayuda a la toma de decisiones por operadores, sin embargo, Arex no cuenta con mecanismos automáticos para la toma de decisiones basándose en la captura de video. En el centro de desarrollo CEDIN se ha propuesto incorporarle al Sistema Arex un módulo para poder generar variables a partir del streaming de video recibido desde cámaras, que permita al módulo Recolector del Sistema Arex aplicar el mecanismo de scripting para ejecutar cambios en el sistema según los valores que toman las variables generadas antes mencionadas. Se propone estas variables: presencia de rostro (variable booleana), presencia de ojos (variable booleana), presencia de carros (variable booleana), presencia de movimiento (variable booleana), cantidad de rostro (variable entera), cantidad de ojos (variable entera), cantidad de carros (variable entera).

La situación problemática anteriormente planteada permite formular el siguiente **problema científico**: ¿Cómo realizar la recolección de variables a partir de streaming de video en el Sistema de Medición y Adquisición Arex del CEDIN?

A partir de la situación problemática que se presenta se define como **objeto de estudio**: la recolección de variables a partir de streaming de video, específicamente en el **campo de acción**: la recolección de variables a partir de streaming de video para Sistemas Domótico.

Para darle solución al problema científico se toma como **objetivo general**: Desarrollar un módulo de recolección de variables mediante cámaras para el Sistema de Medición y Adquisición Arex.

Para dar cumplimiento al objetivo general definido, se elaboran las siguientes **tareas de investigación**:

- Elaborar el marco teórico de la investigación a partir del estudio del estado del arte sobre el tema propuesto.

- Identificar las herramientas, metodologías y artefactos a utilizar durante el desarrollo del trabajo.
- Describir la propuesta de solución.
- Caracterizar las tecnologías actuales para streaming de video.
- Identificar los requisitos funcionales y no funcionales para el desarrollo de la solución propuesta.
- Generar los artefactos relacionados con el análisis y diseño de la solución propuesta.
- Diseñar e implementar la solución propuesta.
- Diseñar y aplicar las pruebas a la aplicación implementada para verificar su correcto funcionamiento.
- Elaboración del documento del Trabajo de Diploma.

A partir de las tareas de investigación descritas anteriormente, se emplean varios **métodos científicos**. Entre los que se encuentran los métodos teóricos y los métodos empíricos.

Como **métodos teóricos** se tiene:

- **Analítico-Sintético:** se utilizó para identificar los principales fundamentos teóricos relacionados con la domótica, a través del estudio de la documentación recopilada.
- **Modelación:** mediante este método se realizarán los modelos para especificar las características que se van a desarrollar en el módulo y esto permitirá una mejor comprensión a los desarrolladores para su posterior implementación.

Y como **métodos empíricos**:

- **Experimental:** se empleó en la realización de pruebas, donde se crean las condiciones para verificar el correcto funcionamiento del módulo desarrollado en la presente investigación.
- **Entrevista:** aplicada a especialistas con experiencia y conocimiento en el sistema Arex para poder desarrollar el módulo propuesto, con el objetivo de definir las funcionalidades del mismo.

El presente documento está estructurado en 3 capítulos:

En el **Capítulo 1** se describen las herramientas y las tecnologías utilizadas, y se realiza un estudio acerca de la metodología empleada en el desarrollo del trabajo. Además, se analiza la arquitectura que posee actualmente el sistema Arex.

En el **Capítulo 2** se describen las principales funcionalidades que posee el sistema, para comprender con una mayor claridad lo que se pretende desarrollar. Se enuncian los requisitos no funcionales. Se detalla la propuesta de solución y las principales clases que la van a componer. Además, se analizan los patrones de diseño que se desean emplear en el proyecto.

En el **Capítulo 3** se describe la implementación del módulo a desarrollar, se realiza el diagrama de despliegue y al final, las pruebas pertinentes de software al sistema para garantizar su buen funcionamiento y por último la integración al sistema.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se describen los principales conceptos asociados al tema de la investigación para lograr una mejor comprensión y familiarización del mismo, se caracterizan tanto las tecnologías como las herramientas a utilizar. Además, se realiza un estudio acerca de la metodología escogida para el desarrollo del trabajo.

1.2 Conceptos

Se presentan un conjunto de conceptos relacionados con la investigación para lograr una mejor comprensión y familiarización con el tema.

1.2.1 Domótica

Conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda. La vivienda domótica nace para facilitar la vida a los ciudadanos, haciéndola más cómoda, más segura, y con mayores posibilidades de comunicación y ahorro energético. Algunos de los aspectos relacionados con la domótica no son exclusivos del hogar, sino que también pueden ser aplicados en otros lugares, como, por ejemplo, oficinas. (Huidobro, 2007)

Con la implementación del Sistema Doméstico de Video Vigilancia con cámaras IP por internet va a beneficiar a las personas que dispongan del servicio de internet, ya que permite monitorear remotamente sectores considerados por el usuario como críticos en los lugares de residencia, preservando de esta manera sus bienes y colaborando con la prevención del cometimiento del delito en sectores aledaños. Además, las personas optimizan el uso del internet que poseen en sus domicilios al dedicarle a la trasmisión de video cuando se encuentran ausentes de sus domicilios.

1.2.2 Streaming

El término streaming se refiere a la tecnología de transmisión de datos por Internet que pueden ser accedidos por los usuarios sin la necesidad de descarga previa. (Santos, 2018)

La carga del contenido publicado en ese formato se realiza mientras el archivo está siendo accedido, lo que disminuye el tiempo de espera de los usuarios, permitiendo así el acceso prácticamente instantáneo al material de audio o vídeo. (Santos, 2018)

Con el streaming, es posible transmitir contenidos en vivo o grabados anteriormente, sin que los datos tengan que ser archivados en computadora, celular, Tablet o cualquier otro dispositivo móvil. (Santos, 2018)

Uno de los protocolos estándar que permite la implementación de esta tecnología es RTSP (del inglés Real Time Streaming Protocol), sin embargo dado que la mayoría de servidores de aplicaciones y servicios en internet están soportados en peticiones HTTP, es poco el trabajo que se ha realizado en cuanto a la generación de herramientas, para realizar pruebas de estrés sobre servidores de streaming. (Golondrino, et al., 2015)

La transferencia de streaming de video se realiza preferiblemente con los protocolos de comunicación RTSP y HTTP. El cliente puede interactuar con el servidor de transmisión de video utilizando el protocolo RTSP y los enlaces de transmisión de audio suelen ser con el protocolo HTTP.

1.2.3 Scripting

El scripting es un código que se usa para automatizar procesos, de lo contrario, un desarrollador web debería realizarlo paso a paso. Cuando se usan diferentes tipos de lenguajes de codificación (como los lenguajes de marcado, como HTML y CSS) para mostrar a las computadoras cómo mostrar un sitio web, se usan lenguajes de script (incluidos PHP, Ruby y JavaScript) para dar instrucciones a los programas (como los sitios web) que se ejecutan en una computadora. En otras palabras, no utiliza scripts o lenguajes de scripting para programar funciones estáticas, como la apariencia general o el diseño de un sitio web o una aplicación web. Utiliza un lenguaje de scripting para decirle a un sitio web estático "algo que hacer", lo que hace que el contenido estático sea dinámico. (Hoogenraad, 2018)

1.2.4 Cámaras IP

Son cámaras de seguridad pensadas para ser visualizadas mediante internet o desde una red local. Por ejemplo, si tengo una red de computadoras interconectadas entre sí mediante un switch o router, estas cámaras se conectarían como si fuera una computadora más. Las más clásicas y profesionales poseen un puerto ethernet con terminal Rj45 y se conectan al switch o router mediante cable utp. También hay cámaras IP inalámbricas, las cuales se conectan a nuestra red WIFI por una antena. (Sosio, 2013)

Estas últimas cámaras si bien pueden ser efectivas para algunos casos, no entran dentro de la categoría de cámaras de seguridad profesionales debido a que la gran mayoría son de escasa calidad y los enlaces inalámbricos WIFI son muy fáciles de bloquear con un simple jammer.

En resumidas cuentas, una cámara Ip es una cámara analógica que en su salida tiene un conversor que hace digital la imagen. (Sosio, 2013)

1.2.5 Reconocimiento de Patrones

El término de reconocimiento de patrones se refiere a un procesamiento de información que tiene una gran importancia práctica que da solución a un amplio rango de problemas. Algunos de estos problemas son resueltos por los humanos sin mucho esfuerzo. Sin embargo, en muchos casos, la solución a estos problemas, usando computadoras, se vuelve extremadamente difícil. (Bishop, 1995)

En términos más concretos, el Reconocimiento de Patrones como una disciplina científica tiene el objetivo de clasificar objetos en un número específico de categorías o clases. Dependiendo de la aplicación, estos objetos pueden ser imágenes, sonidos, olores, en general, señales producto de mediciones que deben ser clasificadas. Estos objetos se denotan con el término genérico de patrones. (Theodoridis, y otros, 1999)

¿Cómo utiliza Google el reconocimiento de patrones para dar sentido a las imágenes?

Los ordenadores no "ven" las fotos y los vídeos de la misma forma que lo hacen las personas. Al observar una foto, puedes ver a tu mejor amiga de pie delante de su casa. Desde la perspectiva de un ordenador, esa misma imagen es simplemente un conjunto de datos que se puede interpretar como formas e información sobre los valores de los colores. Aunque un ordenador no reaccionará igual que tú al ver esa foto, puede entrenarse para reconocer determinados patrones de colores y formas. Por ejemplo, se puede entrenar un ordenador para que reconozca los patrones comunes de las formas y de los colores que componen la imagen digital de un paisaje como una playa, o de un objeto como un coche. Esta tecnología hace posible que Google Fotos organice las fotos y permite que los usuarios encuentren cualquier foto a través de una sencilla búsqueda.

1.2.6 Visión por computadora

La visión por computador o visión artificial es el conjunto de herramientas y métodos que permiten obtener, procesar y analizar imágenes del mundo real con la finalidad de que puedan ser tratadas por un ordenador. Esto permite automatizar una amplia gama de tareas al aportar a las máquinas la información que necesitan para la toma de decisiones correctas en cada una de las tareas en las que han sido asignadas. (Ortiz, 2018)

1.2.7 Sistema de Medición y Adquisición Arex

Es un sistema que permite desde una estación central medir procesos de baja y mediana complejidad, a partir de la adquisición y procesamiento de datos en tiempo real asociados a las variables del proceso. Está diseñado para ser ejecutado en dispositivos con recursos de hardware limitados. Permite representar gráficamente los datos y las variables del proceso y monitorear las mismas por medio de alarmas. Es un sistema genérico que puede ser configurado para la automatización de procesos en diferentes entornos, por ejemplo, en la industria, en sistemas de confort y seguridad, en la agricultura, en sistemas hidráulicos, estaciones meteorológicas, instalaciones eléctricas, entre otros.

1.3 Cámaras IP

Son cámaras de seguridad pensadas para ser visualizadas mediante internet o desde una red local. Las más clásicas y profesionales poseen un puerto ethernet con terminal rj45 y se conectan al switch o router mediante cable utp. En resumidas cuentas, una cámara ip es una cámara analógica que en su salida tiene un conversor que hace digital la imagen. (Fernandez, 2016)

Las cámaras ip o cámaras por red se pueden supervisar en tiempo real desde cualquier navegador web estándar o utilizando un software de gestión de video, conectados a la red LAN y WAN (internet), estas cámaras emite las imágenes directamente a la intranet o internet sin necesidad de un ordenador. el sensor de imagen convierte la imagen, que está compuesta por información lumínica, en señales eléctricas. la señal de video digital en principio transporta la misma información que la analógica, pero mediante números: unos y ceros. estas señales eléctricas se encuentran ya en un formato que puede ser comprimido y transferido a través de redes. ellas tienen incorporadas su

propio miniordenador, lo que le permite emitir vídeo por sí misma. además de comprimir el vídeo y enviarlo, puede tener una gran variedad de funciones adicionales. (Fernandez, 2016)

Una cámara ip (también conocida como cámara de red) es un dispositivo que contiene: (Fernandez, 2016)

- Una cámara de vídeo de gran calidad, que capta las imágenes
- Un chip de compresión que prepara las imágenes para ser transmitidas por internet, y
- Un ordenador que se conecta por sí mismo a internet

La calidad de imagen de las cámaras ip es de alta definición (high definition = hd) y su resolución se mide en megapíxeles, la velocidad de cuadros por segundos (FPS) y funciones como auto IRIS, BLC, WDR, 3DNR, AWB, AGC y otros. (Fernandez, 2016)

Con las cámaras ip el usuario se conecta a través de internet a una dirección ip que tienen sus cámaras. Las cámaras ip permiten al usuario tener la cámara en una localización y ver desde cualquier otro lugar qué está pasando en este preciso momento a través de internet por medio de ip públicas fijas o sistemas DDNS. (Fernandez, 2016)

El acceso a estas imágenes está totalmente restringido/cifrado, sólo las personas autorizadas pueden verlas. también se puede ofrecer acceso libre y abierto si el vídeo en directo se desea incorporar al website de una compañía para que todos los visitantes tengan acceso. Las cámaras ip son la solución a la migración a sistemas abiertos y el distanciamiento respecto a los dvr (videograbadoras), combinado con las ventajas de la conexión en red, la captura y manipulación de imágenes digitales y la inteligencia de las cámaras. (Fernandez, 2016)

Las tres principales categorías de cámaras digitales están definidas por sus estándares de cableado: el estándar paralelo, el estándar camera link, y el estándar ieee 1394. La mayoría de las cámaras digitales usan la interface de estándar paralelo, el cual es un estándar bien establecido que provee un amplio rango de velocidades de adquisición, tamaño de imágenes y profundidad en píxeles. Las cámaras paralelas requieren frecuentemente que los usuarios personalicen su cableado y conectores para ajustarse a su videograbador. (Fernandez, 2016)

Las principales ventajas de las cámaras ip son: (Fernandez, 2016)

- Alta velocidad
- Alta resolución
- Imágenes más grandes
- Opciones y funcionalidades fáciles de configurar
- Menos ruido en la imagen

Otras ventajas de la tecnología ip son: (Fernandez, 2016)

- **Calidad de video:** Las cámaras ip ofrecen una calidad de video superior a las cámaras analógicas e híbridas. ofrecen más posibilidades de visualización, tales como ampliar o estrechar el campo de visión y mejor capacidad de zoom, debido a que las transmisiones de

las señales son realmente digitales, ofrecen más detalles y hacen que sean mucho mejor para el reconocimiento facial o la detección de matrículas.

- **Mejor resolución:** Generalmente, las cámaras digitales ofrecen resoluciones 6 o 20 veces superiores a las cámaras analógicas, ofrecen resoluciones que van desde 1.3 megapíxeles hasta 5 megapíxeles (2560 x 1920) comprimidas y codificadas. esto le permitirá cubrir un área mucho más amplia de observación y cuando realice un zoom, obtener imágenes más detalladas con frecuencia de imagen máxima de 25/30 o 50/60 imágenes por segundo y un espectro de colores más amplio que en televisores estándar.
- **Wireless:** Las conexiones wireless en las cámaras ip pueden ser la solución ideal en áreas donde es muy complicado o muy caro instalar cable. el wireless también puede utilizarse en edificios donde resulten poco práctico o imposible instalar cable, tales como en edificios históricos.
- **Mayor distancia:** Las cámaras analógicas e híbridas pueden enviar video sobre cable de par trenzado hasta 1,5 kilómetros de distancia y hasta 300 metros sobre cable coaxial, pero las transmisiones analógicas pierden calidad en distancias largas o cuando la señal es convertida en formatos diferentes. las cámaras ip pueden enviar video digital a 100 metros por cable ethernet de par trenzado y a distancias ilimitadas mediante redes ip.
- **Inteligencia y gestión:** Las cámaras ip disponen de inteligencia de red y gestión remota. pueden transmitir imágenes y diferentes partes de imágenes, a diversos destinatarios simultáneamente. Además, pueden realizar tareas adicionales tales como el envío de un mensaje cuando detectan movimiento, reconocimiento de matrículas, conteo de personas y el seguimiento de objetos, entre otros. la inteligencia al nivel de la cámara habilita un medio de vigilancia mucho más productivo y efectivo.
- **Fácil instalación:** Las cámaras analógicas requieren más cableado que las cámaras ip. por ejemplo, se requiere un cable adicional para controlar las funciones de giro, inclinación y zoom. si hay audio, se requiere otro cable. una cámara analógica puede requerir tres cables separados: poder, audio, vídeo. las cámaras ip pueden recibir la alimentación, vídeo, audio, control PTZ y señales de control a través de un solo cable conocido como POE, lo cual permite un ahorro en el costo de instalación y cableado, eliminando preocupaciones sobre realizar tendidos de cables eléctricos. asimismo, también permite disponer de sistemas de refrigeración o calefacción integrados sin la necesidad de cables adicionales.
- **Seguridad:** Las cámaras ip hacen que los datos sean difíciles de interceptar. las cámaras ip cifran y comprimen los datos antes de transportarlo a través de internet a su servidor y son compatibles con VPN.
- **Comunicación segura:** Con una cámara analógica, la señal de vídeo se transporta por un cable coaxial sin ningún cifrado ni autenticación. de esta forma, cualquiera puede interceptar el vídeo o, lo que aún es peor, cambiar la señal de una cámara por otra señal de vídeo (piratear

la señal). en el caso del vídeo en red digital, la cámara puede cifrar el vídeo que se envía a través de la red para asegurarse de que no pueda visualizarse ni interferirse.

- **Capacidad de Ampliación:** Las cámaras ip ofrecen mayor capacidad de ampliación y escalabilidad que las cámaras analógicas porque sus requisitos de cableado son menos complejos. al migrar a cámaras ip, es posible aprovechar la infraestructura de cableado existente mediante el uso de convertidores y extensores.
- **Control PTZ y de entrada/salida integrado:** Con una cámara PTZ analógica, la comunicación en serie que controla el movimiento PTZ requiere un cableado independiente de la señal de vídeo, lo que resulta caro y engorroso. la tecnología digital en cámaras de red permite el control PTZ a través de la misma red que transporta el vídeo. además, pueden integrar señales de entrada y salida como las alarmas y los bloqueos de control. todo esto equivale a menos cable y menos dinero y al aumento de la funcionalidad y la integración potencial.
- **Audio integrado:** Una cámara de red digital captura el audio en la cámara, sincronizándolo con el vídeo o incluso integrándolo en el mismo flujo de vídeo, y devolviéndolo después para la supervisión y/o grabación a través de la red.

1.3.1 Aplicaciones para VideoVigilancia IP

El video en red se puede utilizar en un numero casi ilimitado de aplicaciones; sin embargo, la mayoría de sus usos quedan dentro del ámbito de la VideoVigilancia y seguridad o la supervisión remota de personas, lugares, propiedades y operaciones. A continuación, se muestra algunas posibilidades de aplicación habituales en sectores industriales clave: (Mata, 2011)

- Reconocimiento de matrículas.
- Monitorización de tráfico rodado
- Monitorización de procesos industriales como Fundiciones o cadenas de Montaje.
- Vigilancia en condiciones de absoluta oscuridad, utilizando luz infrarroja.
- Vigilancia en vehículos de transporte público.
- Vigilancia del comportamiento de empleados.
- Vigilancia de los niños en el hogar, en la escuela, parques, guarderías.
- Vigilancia de espacios públicos: aeropuertos, calles, estaciones, estadios.
- Vigilancia de espacios privados y negocios: hogares, comercios, hoteles, banca.
- Análisis facial para identificación de crímenes en áreas públicas.
- Aplicaciones educativas.
- Teleasistencia.



Figura 1 Ejemplo de aplicación para la videovigilancia IP

1.3.2 Modelos de cámaras IP

1.3.2.1 Cámaras de red fijas

Una cámara de red fija, que puede entregarse con un objetivo fijo o varifocal, es una cámara que dispone con un campo de vista fijo (normal/telefoto/gran angular) una vez montada. Este tipo de cámara es la mejor opción en aplicaciones en las que resulta útil que la cámara este bien visible. Normalmente, las cámaras fijas permiten que se cambien sus objetivos. Pueden instalarse en carcassas diseñadas para su uso en instalaciones interiores o exteriores. (Mata, 2011)



Figura 2 Ejemplo de cámara de red fija

1.3.2.2 Cámaras de red domo fijas

Una cámara de red domo fija, también conocida como mini domo, consta básicamente de una cámara fija preinstalada en una pequeña carcassa domo. La cámara puede enfocar el punto seleccionado en cualquier dirección. La ventaja principal radica en su discreto y disimulado diseño, así como la dificultad de ver hacia qué dirección apunta la cámara. Asimismo, es resistente a las manipulaciones. (Mata, 2011)

Uno de los inconvenientes de las cámaras domo fijas es que normalmente no disponen de objetivos intercambiables, y si pueden intercambiarse, la selección de objetos está limitada por el espacio

dentro de la carcasa domo. Para compensarlo, a menudo se proporciona un objetivo varifocal que permite realizar ajustes en el campo de visión de la cámara. (Mata, 2011)

Las cámaras domo fijas están diseñadas con diferentes tipos de cerramientos, a prueba de vandalismo y/o con clasificación de protección IP66 cuyo valor significa, IP índice de protección, el primer dígito 6 protección completa contra personas y entrada de polvo, el segundo dígito 6 protección contra fuertes chorros de agua de todas direcciones, incluido olas. Generalmente, las cámaras domo fijas se instalan en la pared o en el techo. (Mata, 2011)



Figura 3 Ejemplo de cámara de red domo Fija

1.3.2.3 Cámaras de red PTZ

Las cámaras PTZ (Pan-Tilt-Zoom) son cámaras que pueden moverse horizontalmente y verticalmente y disponen de un zoom ajustable dentro de un área, de forma tanto manual como automática. Son ideales para su uso en ubicaciones en áreas muy espaciosas (parkings, terminales de aeropuertos, estadios deportivos) y pueden ser usadas remotamente mediante un PC remoto. (Mata, 2011)

Las cámaras PTZ pueden incorporar las siguientes funciones: (Mata, 2011)

Estabilización electrónica de imagen (EIS): En instalaciones exteriores, las cámaras domo PTZ con factores de zoom superiores a los 20x son sensibles a las vibraciones y al movimiento causados por el tráfico o el viento. La EIS ayuda a reducir el efecto de la vibración en un video. Además de obtener videos más útiles, EIS reducirá el tamaño del archivo de la imagen comprimida, de modo que se ahorrará un valioso espacio de almacenamiento.

Máscara de privacidad: La máscara de privacidad permite bloquear o enmascarar determinadas áreas de la escena frente a visualización o grabación para que en esa área no aparezca en el video solo una franja blanca. (Mata, 2011)

Posiciones predefinidas: Muchas cámaras PTZ permiten programar posiciones predefinidas, normalmente entre 20 y 100 posiciones. Una vez las posiciones predefinidas se han configurado en la cámara, el operador puede cambiar de una posición a la otra de forma muy rápida. (Mata, 2011)

E-flip: En caso de que una cámara PTZ se monte en el techo y se utilice para realizar el seguimiento de una persona, por ejemplo, en unos grandes almacenes, se producirá situación en las que el individuo en cuestión pasara justo por debajo de la cámara. Sin la funcionalidad E-flip, las imágenes de dicho seguimiento se verían del revés. En este caso, E-flip gira las imágenes 180 grados de forma automática. Dicha operación se realiza automáticamente y no será advertida por el operador. (Mata, 2011)

Auto-flip: generalmente, las cámaras PTZ, a diferencia de las cámaras domo PTZ, no disponen de un movimiento vertical completo de 360 grados debido a una parada mecánica que evita que las cámaras hagan un movimiento circulas continuo. Sin embargo, gracias a la función Auto-flip, una cámara de red PTZ puede girar el instante 180 grados su cabezal y seguir realizando el movimiento horizontal más allá de su punto cero. De este modo, la cámara puede continuar siguiendo el objeto o la persona en cualquier dirección. (Mata, 2011)

Autoseguimiento: El Autoseguimiento es una función de video inteligente que detecta automáticamente el movimiento de una persona o vehículo y lo sigue dentro de una zona de cobertura de la cámara. Esta función resulta especialmente útil en situaciones de videovigilancia no controlada humanamente en las que la presencia ocasional de personas o vehículos requiere especial atención. La funcionalidad recorta notablemente el coste de un sistema de supervisión, puesto que se necesitan menos cámaras para cubrir una escena. Asimismo, aumenta la efectividad de la solución debido a que permite que las cámaras PTZ graben áreas de una escena en actividad. (Mata, 2011)



Figura 4 Ejemplo de cámara de red PTZ

1.3.2.4 Cámaras de red Domo PTZ

Las cámaras de red domo PTZ pueden cubrir una amplia área al permitir una mayor flexibilidad en las funciones de movimiento horizontal, vertical y zoom. Asimismo, permiten un movimiento horizontal continuo de 360 grados y un movimiento vertical de normalmente 180 grados. Debido a su diseño, montaje y dificultad de identificación del ángulo de visión de la cámara (el cristal de las cubiertas de la cúpula puede ser transparente o ahumado), las cámaras de red domo PTZ resultan idóneas para su uso en instalaciones discretas. (Mata, 2011)

El zoom óptico de las cámaras domo PTZ se mueve, generalmente, entre valores de 10x y 35x. Las cámaras domo PTZ se utilizan con frecuencia en situaciones en las que se emplea un operador. En

caso de que se utilice en interiores, este tipo de cámara se instala en el techo o en un poste o esquina para instalaciones exteriores. (Mata, 2011)



Figura 5 Ejemplo de cámara de red domo PTZ

1.3.3 Instalar y configurar una cámara IP

La instalación de una cámara IP es muy sencillo, pero debemos saber que, para su funcionamiento correcto vamos a necesitar una conexión ADSL.

- En un primer momento, debemos conectar nuestra cámara al router mediante cable ethernet, para que comience a funcionar dentro de nuestra red local. Es idéntico a añadir un ordenador portátil, pero para evitar cualquier tipo de error, lo mejor es que la primera conexión la realicemos a través del cable ethernet.
- Después nos vamos a nuestro teléfono móvil y descargamos la app del fabricante, para lo que necesitaremos un correo.
- Una vez abierta la app, buscamos las cámaras que están disponibles en nuestra red local. Como ya hemos conectado la cámara a la red mediante el cable ethernet, la cámara nos aparecerá listada.
- Tendremos la opción de conectarnos a la cámara, para lo que nos pedirá una clave que viene con la cámara. Actualmente muchas cámaras ya vienen con código QR, de manera que lo pasamos por el lector y la cámara se conecta automáticamente.
- Una vez que la cámara ya está accesible desde el móvil, ya podremos desconectarla del router, y tendremos acceso remoto desde el móvil.
- Las apps también están disponibles para el ordenador o Tablet, de manera que te podrás conectar a la cámara desde cualquier dispositivo con acceso a la red y administrar todas las opciones: grabación, detección de movimientos, alarmas, avisos, etc.

1.3.4 ¿Cómo acceder a una cámara IP?

Luego de que un instalador de cámaras de seguridad idóneo realice la instalación de la cámara IP es muy sencillo. Si estamos en una red local, accederemos a la cámara escribiendo en nuestro navegador de Internet la dirección Ip de la cámara. (Sosio, 2013)

Las cámaras IP poseen internamente un soft que se comporta como si fuera una página web, entonces al abrir la dirección ip de la cámara de seguridad, esa página que estamos accediendo nos mostrara la imagen de la cámara. Por norma general, los profesionales de la instalación de cámaras de seguridad, sugerimos utilizar cámaras IP con resoluciones mayores al Mega Pixel, ya que por debajo de esa resolución es conveniente utilizar cámaras analógicas por una cuestión de costos. (Sosio, 2013)

1.3.5 Seguridad de las cámaras IP.

Las cámaras IP son mucho más seguras que las antiguas cámaras CCTV (), ya que las antiguas conexiones por cable coaxial eran fácilmente manipulables. Las cámaras IP emiten los datos en una red protegida por usuario y password, y sólo se comunican con el rúter al que están conectadas. Actualmente es la tecnología más segura. (Diaz, 2019)

Ancho de Banda

1.3.6 ¿Cuándo graban las cámaras? Cámaras con detección de movimiento.

Existen cámaras de todo tipo, desde las que poseen grabación continua, las que graban determinadas franjas horarias, hasta las que sólo **graban cuando detectan un movimiento**. Estas últimas son las más habituales, ya que si no la capacidad de la memoria se agotaría rápidamente. En todo caso, antes de comprar verifica que opciones ofrece la cámara en cuestión. (Diaz, 2019)

Obviamente, la sensibilidad al movimiento se puede programar, así como la zona que quieres que sea la “sensible” al movimiento. Dependiendo de los modelos, algunos también se pueden poner en marcha cuando detectan sonido. (Diaz, 2019)

1.3.7 Avisos y cámaras IP con alarma

Las apps suelen ofrecer la posibilidad de que la cámara lance un aviso, que recibirás en tu teléfono móvil cuando la cámara detecte un movimiento. Se pueden configurar de múltiples formas, dependiendo del modelo, de manera que puedas recibir una imagen o un vídeo en tu correo electrónico. Otras cámaras, no sólo te avisan, también pueden poner en marcha sistemas disuasorios, como es que se ponga en marcha una alarma acústica y/o visual. También se pueden conectar con otros sistemas externos, como es una alarma independiente de la cámara o un detector de movimiento que no tiene por qué ser el de la cámara. (Diaz, 2019)

1.3.8 ¿Dónde graban las cámaras IP?

Las cámaras IP independientes vienen preparadas con una ranura para añadir **una tarjeta SD**, de manera que la grabación se quedaría en la cámara. Las cámaras CCTV IP, que son las de circuito cerrado, mandan las imágenes a un NVR (un grabador). Hay una opción, pero que no está disponible para todas las cámaras, que es mandar las imágenes a la nube o un servidor Ftp, de manera que las imágenes no se almacenan en la cámara, están en un servicio externo. (Diaz, 2019)

1.3.9 Graban sólo imagen, ¿O también audio?

Para que la cámara detecte y grabe ruido debe poseer detector de audio. Algunas van más allá y poseen comunicación de voz con la cámara, de manera que puedes hablar a través de la aplicación

y que si alguien está cerca de la cámara te oiga. Es una forma de disuadir ante posibles intrusos. (Diaz, 2019)

1.3.10 Calidad de imagen tienen las cámaras IP. Cámaras con visión nocturna

Una de las mejoras que se ha conseguido con las cámaras IP es mejorar la calidad de imagen. La calidad de imagen de las cámaras IP se mide en píxel o megapíxel. Las más habituales son las de 1080P, que ya ofrecen una calidad de imagen muy alta, otras tienen una calidad inferior 720P, pero sin embargo las tienes con HD (High Definition) con varios MP. A medida que los píxeles aumentan la calidad de la imagen también, la nitidez de los objetos y el alcance del objetivo de la cámara. Todas las cámaras IP poseen visión nocturna y la cantidad de píxeles o megapíxeles que tiene la cámara, se hacen especialmente evidentes en situaciones de baja iluminación. A la hora de grabar, podremos rebajar la calidad de la imagen, para que el espacio que ocupan en la SD, disco duro, nube, etc, sea menor. Un detalle que tenemos que tener en cuenta es que, a mayor calidad de la imagen, mayor consumo de ancho de banda. Si no tenemos una buena conexión a Internet podemos tener problemas para ver correctamente las imágenes. (Diaz, 2019)

1.3.11 Ángulo de visión de las cámaras IP

Hay cámaras con diferentes ángulos de visión, diseñadas para todo tipo de situaciones. Las cámaras con menor ángulo de visión son las cámaras bala, que sólo poseen 90°, después hay un buen número de cámaras con hasta unos 110°, pero ya hay disponibles en el mercado cámaras con 180°, con gran angular, visión HD, etc. (Diaz, 2019)

1.3.12 ¿Tiene que estar conectada la cámara a la red eléctrica?

La gran mayoría de las cámaras están diseñadas para enchufarlas a la red, por lo tanto, necesitamos un punto eléctrico. Sin embargo, en el mercado hay múltiples tipos de cámaras, desde las que funcionan a batería recargable hasta las que funcionan con energía solar, especialmente diseñadas para instalarlas en el exterior en lugares remotos. (Diaz, 2019)

1.3.13 ¿Puedo conectarme desde varios dispositivos diferentes a la cámara?

Se puede acceder a las cámaras desde diferentes dispositivos a la vez, móviles, ordenador...lo único que tienen que tener es el mismo correo y clave. También se pueden tener varias cámaras instaladas en una misma aplicación, siempre que sean de la misma marca. Lo que no puedes es acceder a varias cámaras a la vez desde un dispositivo, primero accederás a una y después a otra. (Diaz, 2019)

1.3.14 Diferencia entre una cámara IP y una cámara wifi

Una cámara IP o de protocolo de Internet, está diseñada para conectarse a una red local mediante cable ethernet. La red local la forma nuestro rúter y los dispositivos que tenemos conectados a internet a través del mismo. Una cámara wifi también es una cámara IP, pero la conexión a la red local la realiza a través de señal inalámbrica wifi. Actualmente todas las cámaras IP son wifi, no nos tenemos que preocupar en exceso, pero entiendo que surjan dudas entre unas cámaras y otras. (Diaz, 2019)

1.3.15 Diferencias entre una cámara IP y una cámara CCTV

Las cámaras CCTV o de “circuito cerrado de televisión”, son cámaras analógicas que se conectan a un grabador mediante cable coaxial. Estas imágenes se quedan almacenadas, y se podría acceder a ellas a través de Internet. (Diaz, 2019)

Los sistemas CCTV requieren obligatoriamente la instalación de un sistema de procesado y grabación de las imágenes. (Diaz, 2019)

Las cámaras IP no necesitan de un sistema de almacenamiento externo de las imágenes, las pueden grabar en la propia cámara o no (si no queremos) y en todo momento tenemos acceso a ellas a través de Internet. (Diaz, 2019)

La calidad de las imágenes de las cámaras IP es mucho mayor que las de las cámaras “analógicas” CCTV. (Diaz, 2019)

1.4 Soluciones existentes

La Universidad de las Ciencias Informáticas (UCI) cuenta con un centro de Video Vigilancia ubicado en la Facultad 6, que ha desarrollado una aplicación que permite detectar personas, detectar autos y detectar movimiento, la cual está desarrollada bajo el sistema operativo de Windows, en el lenguaje de programación C++, el framework visual estudio, la biblioteca OpenCV. Esta aplicación utiliza los haarcascade entrenados por ellos mismos, el método de detectar autos lo realizan con una base de datos de imágenes de los carros de la UCI y el método de detectar movimiento lo realizan con otra biblioteca diferente a OpenCV. El módulo que se propone realizar estará bajo el sistema operativo Linux, con el framework Qt, en el lenguaje de programación C++, se utilizarán los haarcascade predefinido por la biblioteca de OpenCV, el método detectar autos se realizará no solo para autos de la UCI, ya que no contiene una base de datos solo para esos autos, sino una base de autos amplia para cualquier auto, el método detectar movimiento se realizará con la biblioteca de OpenCV. Además, este módulo permitirá leer un archivo de configuración del Sistema Arex.

También existe una empresa de las Fuerzas Armadas Revolucionarias llamada XETID, la cual tiene varios proyectos relacionados con la Video Vigilancia y que utilizan OpenCV, pero por sus políticas de seguridad no pueden brindar más información.

Existen algunas herramientas con características similar es al tema de la investigación las cuales se mencionarán a continuación

- ❖ **NudityDetectioni2v**: Este algoritmo detecta la desnudez en las imágenes utilizando el algoritmo del etiquetador de ilustraciones para determinar si una imagen contiene desnudos o no. Utiliza un vector de salida de etiqueta de Illustration Tagger para ponderar las diferentes etiquetas de acuerdo con la forma en que coinciden con la desnudez en las imágenes.
- ❖ **InceptionNet**: extrae características de imágenes y genera etiqueta. Este algoritmo es una implementación directa de InceptionNet de Google, que se capacitó en el conjunto de datos de ImageNet 2015. Se implementa utilizando los enlaces de Python Tensorflow de Google.

- ❖ **CarMakeandModelRecognition:** Este algoritmo proporciona una solución para clasificar los automóviles por su marca, modelo, estilo de carrocería y año del modelo a partir de las imágenes. Las capacidades proporcionadas hasta ahora son las siguientes: Reconociendo más de 2400 modelos de autos, Detección desde múltiples ángulos y la Cobertura de una gran parte de las marcas y modelos occidentales, desde 2000 (y más) hasta el presente.

Todas estas soluciones cumplen con el objetivo trazado al iniciar esta investigación, pero son privadas y para su utilización es necesario tener que pagar y realizar contratos con las empresas que administran esos sistemas que son de origen extranjero. Nuestro país posee condiciones que nos permite que se puedan utilizar estos algoritmos para resolver problemas. Por todo lo mencionado anteriormente se decide realizar este proyecto.

1.5 Análisis y selección de las herramientas y tecnologías

Se realiza un análisis y estudio de las herramientas y tecnologías existentes, resumiendo sus características y beneficios. Luego se seleccionaron las adecuadas y las que brindan solución al problema propuesto para la investigación.

1.5.1 Lenguajes de programación

Se escogen los dos lenguajes de programación que se mencionan a continuación para describir sus características ya que presentan una serie de requisitos que facilitan el trabajo posterior.

1.5.1.1 Lenguajes de programación C++

La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido. Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos). Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma. Tiene un conjunto completo de instrucciones de control. Permite la agrupación de instrucciones. Incluye el concepto de puntero (variable que contiene la dirección de otra variable). Los argumentos de las funciones se transfieren por su valor. (Stroustrup, 2007)

Es un lenguaje imperativo (las instrucciones se ejecutan unas tras otras, de manera secuencial salvo cuando se encuentran estructuras de control o bucles) orientado a objetos derivado de C. En realidad, un superconjunto de C, que nació para añadirle cualidades y características de las que carecía. Este lenguaje tiene una alta potencia para la programación a bajo nivel. Se le han añadido elementos que le permite un estilo de programación con alto nivel de abstracción. Entre las grandes ventajas de C++ se pueden mencionar la variedad tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacios de nombres, sobrecarga de operadores, referencias y operadores para manejo de memoria persistente. Es orientado a objetos, lo que permite estructurar mejor las ideas y encapsular los conceptos de la realidad en clases. (Stroustrup, 2007)

Las entradas y salidas no forman parte del lenguaje, sino que se proporciona a través de una biblioteca de funciones. Permite la separación de un programa en módulos que admiten compilación independiente. Programación de bajo nivel (nivel bit). (Stroustrup, 2007)

Se escoge este lenguaje de programación por las características mencionadas anteriormente, además porque se desea trabajar con la librería OpenCV y con el IDE Qt Creator, y los mismos admiten este lenguaje. Además, el centro trabaja con el mismo y facilita la inclusión del módulo al sistema.

1.5.2 Protocolos de comunicación

Un protocolo es un método estándar que permite la comunicación entre procesos (que potencialmente se ejecutan en diferentes equipos), es decir, es un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red. Existen diversos protocolos de acuerdo a cómo se espera que sea la comunicación. Algunos protocolos, por ejemplo, se especializarán en el intercambio de archivos (FTP); otros pueden utilizarse simplemente para administrar el estado de la transmisión y los errores (como es el caso de ICMP), etc. (Pillou, 2018)

En Internet, los protocolos utilizados pertenecen a una sucesión de protocolos o a un conjunto de protocolos relacionados entre sí. Este conjunto de protocolos se denomina TCP/IP. Entre otros, contiene los siguientes protocolos: HTTP, FTP, ARP, ICMP, IP, TCP, UDP, SMTP, Telnet, NNTP. (Pillou, 2018)

Los protocolos de comunicación seleccionados para el desarrollo de la investigación fueron el protocolo de flujo en tiempo real (RTSP) y el protocolo de transferencia de hipertexto (HTTP) dado que pueden ser utilizados para transmitir videos, pero trabajan sobre distintas capas de transporte, lo que implica comportamientos diferentes.

1.5.2.1 Protocolo RTSP

Protocolo de flujo de datos en tiempo real no orientado a conexión que se utiliza para definir cómo se hará el envío de información entre el cliente y el servidor. Este protocolo trabaja a nivel de aplicación y controla que la entrega de datos se realice correctamente, pues el tipo de contenido con el que se trabaja normalmente al hacer streaming es muy sensible a la sincronía temporal (o a la falta de ella). Así pues, se podría considerar que el RTSP (del inglés Real Time Streaming Protocol) actúa como si de una especie de mando a distancia de red para servidores multimedia se tratase. RTSP define diferentes tipos de conexión y diferentes conjuntos de requisitos, para intentar conseguir siempre un envío de flujo de datos a través de redes IP lo más eficiente posible. Además, establece y controla uno o más flujos sincronizados de datos como audio y video. A tal fin se definió el uso de sesiones, mediante identificador único, en este protocolo. Es independiente del protocolo de transporte y puede funcionar tanto sobre UDP, RDP o TCP. Durante una sesión, un cliente puede abrir y cerrar conexiones fiables de transporte con el servidor mediante peticiones RTSP. Los flujos controlados por RTSP pueden usar RTP como hemos visto, pero el modo de operación de RTSP es

independiente del mecanismo de transporte usado para transmitir el continuo flujo de datos. Sin embargo, en la mayoría de casos se utiliza TCP para el control del reproductor y UDP para la transmisión de datos con RTP. (Costilla, et al., 2015)

Se decide escoger el protocolo RTSP por sus amplios beneficios que ofrece anteriormente descritos, además de sus propiedades como son las siguientes:

- **Extensible:** nuevos métodos y parámetros pueden ser añadidos a RTSP fácilmente.
- **Seguro:** RTSP usa los mecanismos de seguridad de la web, cualquiera a nivel de transporte. Se pueden aplicar directamente los mecanismos de autenticación de http.
- **Capacidad multi-servidor:** Cada flujo de contenido perteneciente a una misma presentación puede residir en diferentes servidores.
- **Separación del control del stream y la iniciación de la conferencia:** El control de stream no tiene nada que ver con las invitaciones de conferencia a un servidor. En particular para estos casos se suele usar SIP.
- **Neutral respecto de la descripción de las presentaciones:** no impone ninguna descripción particular o formato concreto de metarchivo.
- **Muy similar a http:** cuando es posible RTSP reutiliza los conceptos de http, por lo cual puede usar la infraestructura ya establecida.
- **Capacidad de negociación:** si las características básicas están desactivadas, hay un mecanismo para que el cliente pueda determinar qué métodos van a ser implementados. Esto permite a los clientes usar la interfaz más apropiada. (Costilla, et al., 2015)

En una sesión RTSP son comunes los intercambios que se producen un ejemplo de esto es el siguiente: el cliente envía una petición RTSP, con formato HTTP al servidor del streaming. Cuando se establece la conexión con el mismo, normalmente vía TCP, se manda una petición pidiendo los datos al servidor para configurar una sesión. Una vez hecho esto, el cliente está preparado para pedir una descripción de la presentación concreta a reproducir, a la cual el servidor responde con todos los valores de inicialización necesarios para dicha presentación. La siguiente petición será para cada flujo de datos que se quiera reproducir, obteniendo los protocolos aceptados para el transporte. Con todos los datos conocidos ya podemos pedir que comience el envío de los datos por parte del servidor. (Costilla, et al., 2015)

1.5.2.2 Protocolo HTTP

Desde 1990, el protocolo HTTP (Hypertext Transfer Protocol, Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet. La versión 0.9 solo tenía la finalidad de transferir los datos a través de Internet (en particular páginas web escritas en HTML). La versión 1.0 del protocolo (la más utilizada) permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME. (Pillou, 2018)
El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato

HTML). Entre un navegador (el cliente) y un servidor web (denominado, entre otros, http en equipos UNIX) localizado mediante una cadena de caracteres denominada dirección URL. (Pillou, 2018)

El protocolo HTTP (HyperText Transfer Protocol) es quizá el protocolo más importante que debemos conocer como desarrolladores Web y la razón por la que muchos tenemos hoy un empleo ;). Fue diseñado para transmitir HTML (HyperText Markup Language) pero hoy en día se utiliza para transmitir todo tipo de documentos (imágenes, audio, video, PDF, etc.) y para crear aplicaciones Web. (Escobar, 2017)

HTTP es un protocolo cliente-servidor, lo que significa que el cliente envía una petición al servidor y espera un mensaje de respuesta del servidor. Es un protocolo sin estado, lo que significa que el servidor no guarda información del cliente, cada petición es independiente de las demás. (Escobar, 2017)

1.5.3 Framework

Qt es una framework multiplataforma en C++ de desarrollo de aplicaciones. Se utiliza fundamentalmente para desarrollar aplicaciones con interfaz gráfica, gracias al conjunto de controles independientes de la plataforma que ofrece, aunque también es usado para crear herramientas de línea de comando o consolas de gestión para servicios. (Torres, 2013)

Qt está disponible para sistemas tipo UNIX (Linux, BSD, UNIX, etc.) con servidor gráfico X Windows System, Apple Mac OS X, Microsoft Windows y sistemas Linux embebidos. Además se puede hacer uso de la librería desde lenguajes diferentes a C++ gracias al empleo de bindings: Python, Java, Ruby, Ada, Pascal, Perl, PHP, Haskell, Lua, D, .NET, etc. (Torres, 2013)

Se decide trabajar con este framework ya que incluye las clases de C++, que fue el lenguaje de programación escogido para el desarrollo del módulo.

1.5.4 IDE de desarrollo

Qt es una biblioteca de software que desarrolla Nokia para crear interfaces gráficas de usuario. Para los desarrolladores, Nokia ofrece Qt Creator, un entorno de desarrollo (IDE) multiplataforma muy completo. (Cambiaso, 2018)

Principales características de Qt Creator (Cambiaso, 2018):

- Posee un avanzado editor de código C++.
- Soporta los lenguajes: C#/.NET Languages (Mono), Python: PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby.
- Posee también una GUI integrada y diseñador de formularios.
- Herramienta para proyectos y administración.
- Ayuda sensible al contexto integrado.
- Depurador visual.
- Resaltado y auto-completado de código.
- Soporte para refactorización de código.
- Y mucho más.

Qt Creator es distribuido bajo tres tipos de licencias: Qt Commercial Developer License, Qt GNU LGPL v. 2.1, Qt GNU GPL v. 3.0 y está disponible para las plataformas: Linux, Mac OSX; Windows, Windows CE, Symbian y Maemo. (Cambiaso, 2018)

Se escoge este IDE de desarrollo ya que se trabaja con el framework Qt y además porque posee un avanzado editor de código C++ y el mismo fue el lenguaje escogido.

1.5.5 Biblioteca

OpenCV (Open Source Computer Vision) es una biblioteca open-source de visión artificial y machine learning. (Gracia, 2013)

OpenCV provee una infraestructura para aplicaciones de visión artificial. (Gracia, 2013)

La biblioteca tiene más de 2500 algoritmos, que incluye algoritmos de machine learning y de visión artificial para usar. (Gracia, 2013)

Estos algoritmos permiten identificar objetos, caras, clasificar acciones humanas en vídeo, hacer tracking de movimientos de objetos, extraer modelos 3D, encontrar imágenes similares, eliminar ojos rojos, seguir el movimiento de los ojos, reconocer escenarios. (Gracia, 2013)

Se usa en aplicaciones como la detección de intrusos en vídeos, monitorización de equipamientos, ayuda a navegación de robots, inspeccionar etiquetas en productos. (Gracia, 2013)

OpenCV está escrito en C++, tiene interfaces en C++, C, Python, Java y MATLAB interfaces y funciona en Windows, Linux, Android y Mac OS. O. (Gracia, 2013)

Se escoge esta biblioteca ya que las aplicaciones incluyen, la robótica, análisis y procesamiento de imágenes o vídeos, seguimiento y detección de objetos, detección y reconocimiento de rostros, reconocimiento de placas de vehículos, análisis de formas, reconstrucción 3D, realidad aumentada, y mucho más.

1.5.6 Lenguaje de modelado

UML (Unified Modelling Language) es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. UML fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para modelar aplicaciones de muy diversos dominios de aplicación (telecomunicaciones, comercio, sanidad, etc.) y plataformas de objetos distribuidos. (Fuentes, et al., 2015)

El hecho de que UML sea un lenguaje de propósito general proporciona una gran flexibilidad y expresividad a la hora de modelar sistemas. Sin embargo, hay numerosas ocasiones en las que es mejor contar con algún lenguaje más específico para modelar y representar los conceptos de ciertos dominios particulares. Esto sucede, por ejemplo, cuando la sintaxis o la semántica de UML no permiten expresar los conceptos específicos del dominio, o cuando se desea restringir y especializar los constructores propios de UML, que suelen ser demasiado genéricos y numerosos. (Fuentes, et al., 2015)

1.5.7 Herramienta de modelado

Visual Paradigm for UML es una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería de Software. (Fuentes, et al., 2015)

Se decide utilizar esta herramienta ya que se escoge como lenguaje de modelado el UML (Unified Modelling Language) y esta herramienta soporta este lenguaje. Además, propicia un conjunto de ayudas para el desarrollo de software, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente y la documentación.

1.5.8 Metodología de desarrollo del software

1.5.8.1 Proceso Unificado Ágil (AUP) con el modelo CMMI-DEV v 1.3

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. (Molina Romero, et al., 2015)

El Proceso Unificado Ágil consta de cuatro fases que el proyecto atraviesa de forma secuencial. Dichas fases son, al igual que en el Proceso Unificado de Rational (Torrecilla, 2012):

- **Iniciación:** El objetivo de esta fase es identificar el alcance inicial del proyecto, una arquitectura potencial para el sistema y obtener, si procede, financiación para el proyecto y la aceptación por parte de los promotores del sistema.
- **Elaboración:** Mediante esta fase se pretende identificar y validar la arquitectura del sistema.
- **Construcción:** El objetivo de esta fase consiste en construir software desde un punto de vista incremental basado en las prioridades de los participantes.
- **Transición:** En esta fase se valida y despliega el sistema en el entorno de producción.

Se decide escoger una metodología ágil ya que se necesita obtener un módulo para el sistema Arex en poco tiempo y que a la vez sea confiable, y con un número de desarrolladores mínimo. AUP utiliza las técnicas ágiles, y es seleccionada la metodología AUP en su variante UCI, ya que es flexible, no requiere de una gran cantidad de desarrolladores para favorecer a la estandarización de una metodología en la universidad.

Conclusiones parciales

En este capítulo se realizó un análisis de los principales conceptos asociados a la investigación para una mejor comprensión del tema propuesto. Además, se analizaron y escogieron las principales tecnologías y herramientas para poder desarrollar de una mejor forma un módulo para poder generar variables a partir del streaming de video recibido desde cámaras IP. Se escogió la biblioteca OpenCV ya que permiten identificar objetos, caras, encontrar imágenes similares, entre otros, además, está escrito en C++ que fue el lenguaje de programación escogido para la implementación del módulo. Se decide seleccionar Qt Creator como entorno de desarrollo integrado, UML como lenguaje de modelado, Visual Paradigm como herramienta de modelado, y AUP en su variante UCI como metodología de desarrollo del software. Los protocolos de comunicación que se escogieron fueron HTTP Y RTSP, ya que el primero permite la transferencia de archivos entre un navegador y un servidor web localizado mediante una dirección URL, y el segundo que nos permite el flujo de video en tiempo real.

Capítulo 2. Diseño de la solución propuesta

2.1 Introducción

En el presente capítulo se describe la propuesta solución. Se enuncian los requisitos funcionales y no funcionales que posee el módulo, para comprender con una mayor claridad lo que se pretende desarrollar. Se detalla las principales clases que va a poseer el módulo. Además, se analizan los patrones de diseño que se desean emplear en el proyecto.

2.2 Propuesta de solución

Para darle respuesta a la situación problemática se propone desarrollar un módulo para poder generar variables a partir del streaming de video recibido desde cámaras IP para el Sistema de Medición y Adquisición Arex del CEDIN. La solución estará diseñada para permitir al módulo Recolector del Sistema Arex aplicar el mecanismo de scripting para ejecutar cambios en el sistema según los valores que toman las variables generadas. La solución propuesta se describe e ilustran a continuación:

Recolector: es el encargado de aplicar el mecanismo de scripting para ejecutar cambios en el sistema según los valores que toman las variables que se generan.

Despachador de Clientes: es el encargado de atender las peticiones de los clientes y del recolector

Manejador de Cámaras: es el encargado de controlar todo lo que tiene que ver con las cámaras ip.

Biblioteca de cámara: permite ejecutar acciones sobre las cámaras, entre ellas solicitar el streaming de video.

Biblioteca OpenCV: recibe el flujo de video generado por las cámaras IP. Además, es la biblioteca encargada de generar las variables propuestas para la investigación, las cuales son: presencia de rostro (variable booleana), presencia de ojos (variable booleana), presencia de carros (variable booleana), presencia de movimiento (variable booleana), cantidad de rostro (variable entera), cantidad de ojos (variable entera), cantidad de carros (variable entera).

Cámaras IP: son las encargadas de enviar los streaming de videos generados.

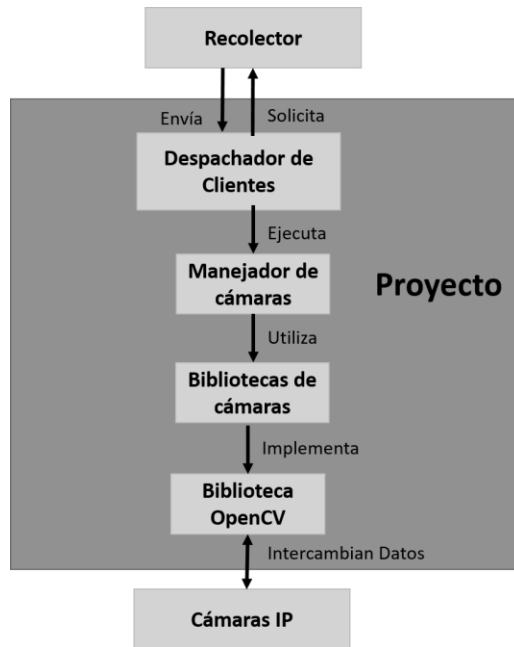


Figura 6 Propuesta de Solución

2.3 Arquitectura

Arquitectura cliente- servidor:

La arquitectura cliente -servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. (Linarte, et al., 2016)

Una característica importante son las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre cliente y servidores una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. (Linarte, et al., 2016)

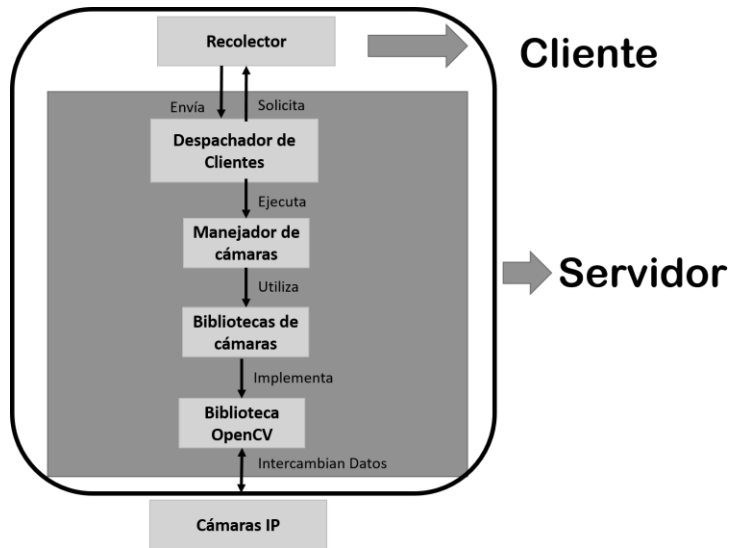


Figura 7 Arquitectura del Sistema

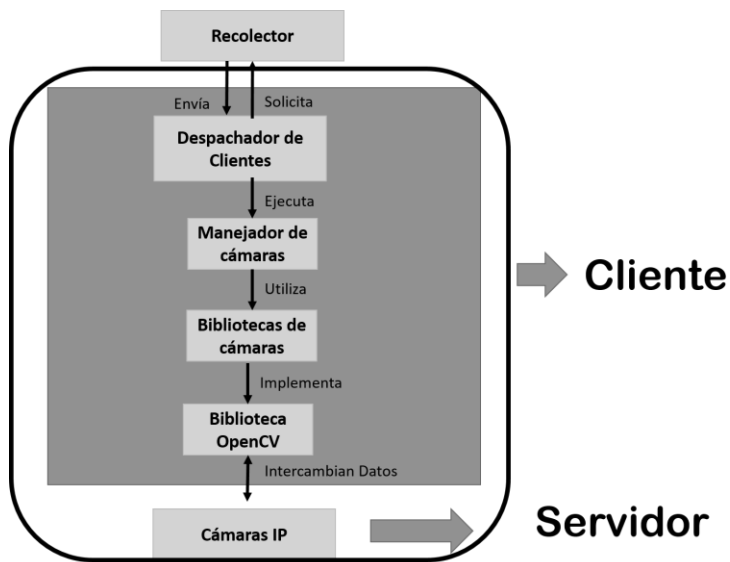


Figura 8 Arquitectura del Sistema

Las figuras muestran la arquitectura del sistema propuesto compuesta por el Recolector de Arex, el despachador de clientes, el manejador de cámaras, las bibliotecas de cámaras, la biblioteca OpenCV y por último las cámaras IP conectadas. La arquitectura Cliente/Servidor en un primer momento que el Recolector de Arex le realiza peticiones al despachador de cliente, por tanto, el Recolector de Arex hace función de cliente y el despachador de clientes hace función de servidor, y en un segundo instante el módulo implementado es denominado cliente y las cámaras conectadas serían servidores. Por esta razón existe una comunicación en un entorno Cliente/Servidor.

2.4 Requisitos Funcionales

Los requisitos funcionales del sistema definen las funciones que presentan cada uno de sus componentes y establecen como debería ser el comportamiento del sistema.

Requisitos funcionales que presenta el sistema:

- ❖ **RF1.** Cargar el archivo de configuración de Arex, donde vengan especificados los datos para conectarse a cámaras ip.
- ❖ **RF2.** Conectarse a cámaras configuradas y recibir el streaming de video.
- ❖ **RF3.** Generar información en forma de variables procesando el streaming de video de cada cámara.
- ❖ **RF4.** Atender solicitudes de clientes para acceder a las variables generadas.

2.4.1 Historias de usuarios

Las historias de usuario son descripciones, siempre muy cortas y esquemáticas, que resumen la necesidad concreta de un usuario al utilizar un producto o servicio, así como la solución que la satisface. Como muchas otras herramientas Ágiles, las historias de usuario surgieron como una respuesta orientada al sector de desarrollo de software, aunque con el tiempo se están aplicando a otros tipos de negocio. Su función principal es identificar problemas percibidos, proponer soluciones y estimar el esfuerzo que requieren implementar las ideas propuestas. (2017)

Descripción de Requisitos de Software:

Número: RF.1	Nombre del requisito: Cargar el archivo de configuración de Arex, donde vengan especificados los datos para conectarse a cámaras ip.
Programador: Arlet Galguera López.	Iteración Asignada: V 1.0
Prioridad: Alta	Tiempo Estimado: 3 semanas
Riesgo en Desarrollo: El programador puede presentar problemas de salud.	Tiempo Real: 2 semanas
Descripción: El sistema explora el archivo de configuración del Sistema de Mediciones AREX, reconoce el identificador correspondiente a cada una de las cámaras de video, carga la configuración de cada cámara.	

Tabla 1. HU1: Cargar el archivo de configuración de Arex, donde vengan especificados los datos para conectarse a cámaras ip.

Número: RF.2	Nombre del requisito: Conectarse a cámaras configuradas y recibir el streaming de video.
Programador: Arlet Galguera López.	Iteración Asignada: V 1.0
Prioridad: Alta	Tiempo Estimado: 1 semana
Riesgo en Desarrollo: El programador puede presentar problemas de salud.	Tiempo Real: 1 semana
Descripción: A partir de los parámetros de conexión a las cámaras IP se realiza la conexión con cada una cuyo objetivo es obtener el streaming de video.	

Tabla 2. HU2: Conectarse a cámaras configuradas y recibir el streaming de video.

Número: RF.3	Nombre del requisito: Generar información en forma de variables procesando el streaming de video de cada cámara.
Programador: Arlet Galguera López.	Iteración Asignada: V 1.0
Prioridad: Alta	Tiempo Estimado: 5 semanas
Riesgo en Desarrollo: El programador puede presentar problemas de salud.	Tiempo Real: 7 semanas
Descripción: Se genera información en forma variables a partir del procesamiento del streaming de video recibido de cada cámara ip. Las variables son: presencia de rostro (variable booleana), presencia de ojos (variable booleana), presencia de carros (variable booleana), presencia de movimiento (variable booleana), cantidad de rostro (variable entera), cantidad de ojos (variable entera), cantidad de carros (variable entera).	

Tabla 3. HU3: Generar información en forma de variables procesando el streaming de video de cada cámara.

Número: RF.4	Nombre del requisito: Atender solicitudes de clientes para acceder a las variables generadas.
Programador: Arlet Galguera López.	Iteración Asignada: V 1.0
Prioridad: Alta	Tiempo Estimado: 3 semanas
Riesgo en Desarrollo: El programador puede presentar problemas de salud.	Tiempo Real: 2 semanas

Descripción: Permite al cliente que realice peticiones para obtener la lista de variables generadas. Las variables son: presencia de rostro (variable booleana), presencia de ojos (variable booleana), presencia de carros (variable booleana), presencia de movimiento (variable booleana), cantidad de rostro (variable entera), cantidad de ojos (variable entera), cantidad de carros (variable entera).

Tabla 4. HU4: Atender solicitudes de clientes para acceder a las variables generadas.

2.5 Requisitos no funcionales

Los requisitos no funcionales del sistema son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Se puede decir que los requisitos no funcionales constituyen la forma en la que debe actuar el sistema para que funcione de manera eficaz, atendiendo aspectos tales como la disponibilidad, flexibilidad, seguridad y facilidad de uso. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto debido a que forman una parte significativa de la especificación. (Londoño, 2005)

2.5.1 Requisitos no funcionales de software

- ❖ Framework QT.
- ❖ PC con Sistema Operativo Linux.

2.5.2 Requisitos no funcionales de hardware

- ❖ PC con microprocesador Pentium 4 o superior.
- ❖ PC con memoria RAM mínima de 256 MB.

2.6 Diagrama de clases

El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura y el comportamiento de cada uno de los objetos del sistema y sus relaciones con los demás objetos, pero no muestra información temporal. (Cillero, 2009)

Una clase describe un conjunto de objetos con propiedades (atributos) similares y un comportamiento común. Los objetos son instancias de las clases. (Cillero, 2009)

No existe un procedimiento inmediato que permita localizar las clases del diagrama de clases. Estas suelen corresponderse con sustantivos que hacen referencia al ámbito del sistema de información y que se encuentran en los documentos de las especificaciones de requisitos y los casos de uso. (Cillero, 2009)

Dentro de la estructura de una clase se definen los atributos y las operaciones o métodos:

- Los atributos de una clase representan los datos asociados a los objetos instanciados por esa clase.
- Las operaciones o métodos representan las funciones o procesos propios de los objetos de una clase, caracterizando a dichos objetos. (Cillero, 2009)

El diagrama de clases permite representar clases abstractas. Una *Clase abstracta* es una clase que no puede existir en la realidad, pero que es útil conceptualmente para el diseño del modelo orientado a objetos. Las clases abstractas no son instanciables directamente sino en sus descendientes. Una clase abstracta suele ser situada en la jerarquía de clases en una posición que le permita ser un depósito de métodos y atributos para ser compartidos o heredados por las subclases de nivel inferior. (Cillero, 2009)

En el diagrama la clase *CamerasManager* representa la clase principal y controladora de dicho módulo, además, utiliza las clases *XmlReader* y *ClientsDispatcher*.

La clase *ClientsDispatcher* utiliza las funciones y métodos de la clase *ClientConnection* para poder establecer una buena comunicación entre el cliente y el servidor.

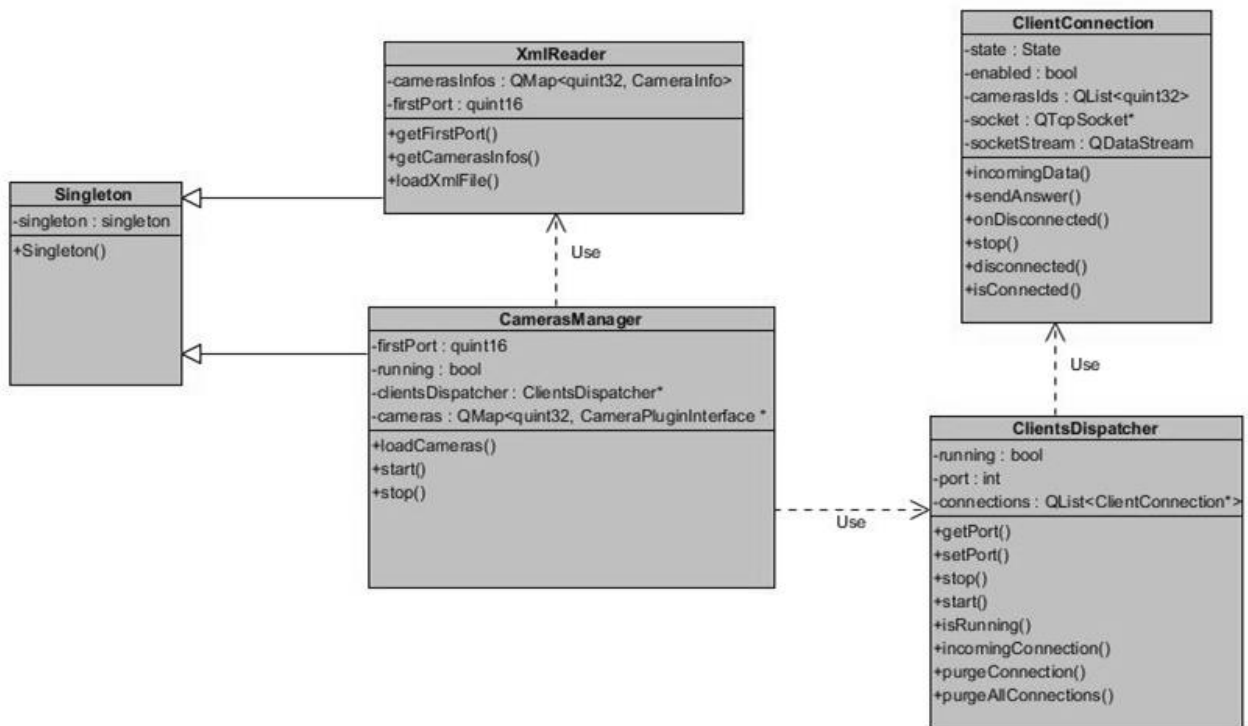


Figura 9 Diagrama de Clases del Sistema

2.7 Patrones de Diseño

Un patrón de diseño se caracteriza como “una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución” [Ale79]. Para el diseño de software, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución. (Pressman, 2010)

Un buen patrón de diseño incorpora el conocimiento de diseño pragmático, ganado con dificultad, en una forma que permite que otros lo reutilicen “un millón de veces sin elaborarla dos veces de la misma forma”. Un patrón de diseño evita “reinventar la rueda” o, peor aún, inventar una “nueva rueda” que sea un poco menos redonda, demasiado pequeña para el uso que se pretende y muy angosta para el terreno en el que rodará. Si se usan con eficacia, los patrones de diseño invariablemente harán del lector un buen diseñador de software. (Pressman, 2010)

2.7.1 Patrones GRASP

Los patrones GRASP constituyen un apoyo para la enseñanza que ayuda a uno a entender el diseño de objetos esencial, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades. (Larman, 2003)

Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Es importante entender y ser capaces de aplicar estos principios durante la creación de los diagramas de interacción porque un desarrollador de software con poca experiencia en la tecnología de objetos necesita dominar estos principios tan rápido como sea posible; constituyen la base de cómo se diseñará el sistema. GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para sugerir la importancia de aprehender (grasping en inglés) estos principios para diseñar con éxito el software orientado a objetos. (Larman, 2003)

Los patrones GRASP tienen nombres concisos como Experto en Información, Creador, Variaciones Protegidas. (Larman, 2003)

Luego de un análisis de los patrones de diseño GRASP, en el módulo se ponen de manifiesto dos patrones los cuales se expondrán sus características.

✂ Experto en Información (o Experto)

El Experto en Información se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. El Experto no pretende ser una idea oscura o extravagante; expresa la “intuición” común de que los objetos hacen las cosas relacionadas con la información que tienen. Nótese que el cumplimiento de la responsabilidad a menudo requiere información que se encuentra dispersa por diferentes clases de objetos. Esto implica que hay muchos expertos en información “parcial” que colaborarán en la tarea. Por ejemplo, el problema del total de las ventas al final requiere la colaboración de tres clases de objetos. (Larman, 2003)

Este patrón se pone de manifiesto en las clases CamarasManager ya que es el principio básico de asignación de responsabilidades de cada clase. Además, es la clase donde recae toda la responsabilidad de la creación de un objeto o la implementación de un método y también conoce toda la información necesaria para crearlo.



Figura 10 Patrón de Diseño Experto

✦ Creador

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón Creador es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Eligiéndolo como el creador se favorece el bajo acoplamiento. El Agregado agrega Partes, el Contenedor contiene Contenido, y el Registro registra Registros, son todas ellas relaciones comunes entre las clases en un diagrama de clases. El Creador sugiere que la clase contenedor o registro es una buena candidata para asignarle la responsabilidad de crear lo que contiene o registra. (Larman, 2003)

Este patrón se pone de manifiesto en la clase CamarasManager ya que permite identificar quién debe ser el responsable de la creación de nuevos objetos o clases en el módulo.



Figura 11 Patrón de Diseño Creador

2.7.2 Patrones GoF

Dentro de los patrones clásicos tenemos los **GoF (Gang of Four)**, estudiados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en su mítico libro Design Patterns se contemplan 3 tipos de patrones: (Gracia, 2013)

- ✂ **Patrones de creación:** tratan de la inicialización y configuración de clases y objetos.
- ✂ **Patrones estructurales:** Tratan de desacoplar interfaz e implementación de clases y objetos.
- ✂ **Patrones de comportamiento:** tratan de las interacciones dinámicas entre sociedades de clases y objetos.

Dentro del grupo de **Patrones de creación** tenemos:

Singleton: Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. (Gracia, 2013)

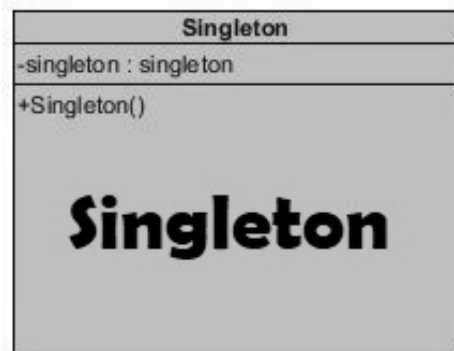


Figura 12 Patrón de Diseño Singleton

Conclusiones Parciales

En el presente capítulo se describió la propuesta de solución para el desarrollo del módulo de recolección de variables, así como elementos que lo componen y sus principales funciones. También se describe la arquitectura seleccionada para que el modulo posea una buena organización y que así el programador tenga mejor conocimiento a la hora de organizar las clases y sus funciones, se escoge la arquitectura cliente servidor ya que tiene ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Los requisitos funcionales nos permitieron definir las funciones que presentara el modulo. También se establecieron los requisitos no funcionales que son fundamentales en el éxito del producto debido a que forman una parte significativa de la especificación. Se crea el diagrama de clases que permite recoger las clases de objetos y sus asociaciones entre ellas y esto facilita trazar claramente la estructura del sistema.

Capítulo 3. Implementación y pruebas del sistema

3.1 Introducción

En este capítulo se mostrará el diagrama de despliegue del sistema, teniendo en cuenta las características que posee el mismo. Además, se expondrán las pruebas al sistema para verificar el comportamiento del módulo en un conjunto de casos de prueba. Las pruebas unitarias consisten en ver las funciones de cada clase por separado y las pruebas de aceptación no son más que determinar por parte del cliente la aceptación o rechazo del sistema que se desarrolla. Se eligen para realizar estas dos pruebas por las características que poseen, además, que nos permitirán comprobar la funcionalidad del módulo.

3.2 Diagrama de despliegue del sistema

El Diagrama de despliegue es un diagrama estructurado que muestra la arquitectura del sistema desde el punto de vista del despliegue (distribución) de los artefactos del software en los destinos de despliegue. (Sarmiento, 2013)

El diagrama de despliegue contiene instancias de nodos conectados por enlaces de comunicación. Las instancias de nodo pueden contener instancias de tiempo de ejecución, como instancias de componentes y objetos. Instancias de componentes y objetos también pueden contener otros objetos. El modelo puede mostrar las dependencias entre las instancias y sus interfaces, y también puede modelar la migración de entidades entre nodos u otros contenedores. (Sarmiento, 2013)

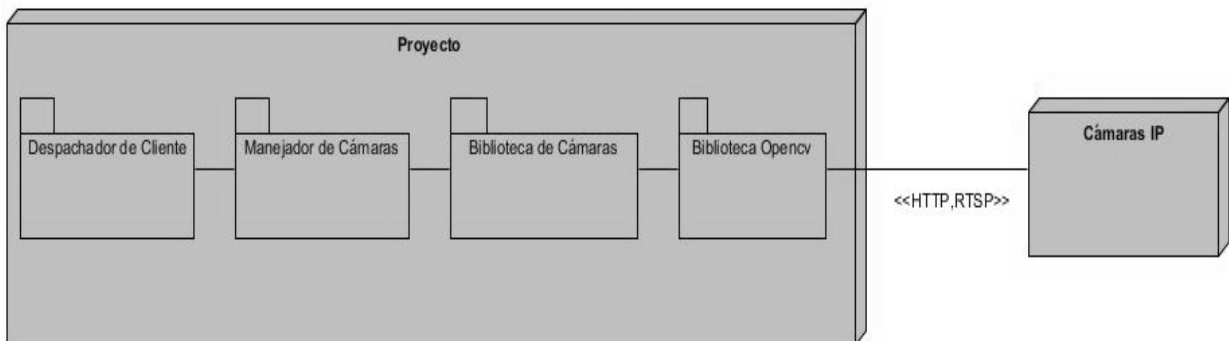


Figura 13 Diagrama de Despliegue del Sistema

En el diagrama de despliegue se muestra un nodo llamado Proyecto donde se ejecutará el módulo que se desea desarrollar.

Proyecto: es el nodo donde se ejecutará el módulo desarrollado.

Cámaras IP: es el nodo donde se encuentran las cámaras IP con las cuales se comunicará el módulo.

3.3 Pruebas del sistema:

Las pruebas del sistema tienen como objetivo ejercitar profundamente el sistema comprobando la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de sistemas de información con los que se comunica. (Cillero, 2009)

Para detectar los errores que puede llegar a presentar nuestro sistema se puede hacer uso de los métodos de prueba Caja Blanca y Caja Negra. Las pruebas de caja blanca requieren del conocimiento de la estructura interna del programa y las pruebas de caja negra se centran principalmente en los requisitos funcionales del software permitiendo así encontrar funciones incorrectas o ausentes. (Cillero, 2009)

3.3.1 Pruebas unitarias

Las pruebas unitarias son pruebas automatizadas que verifican la funcionalidad en el componente, clase, método o nivel de propiedad. (Quijano, 2018)

El objetivo principal de las pruebas unitarias es tomar la pieza más pequeña de software comprobable en la aplicación, aislarla del resto del código y determinar si se comporta exactamente como esperamos. Cada unidad se prueba por separado antes de integrarlas en los componentes para probar las interfaces entre las unidades. (Quijano, 2018)

Para la realización de esta prueba se utilizó la herramienta Visual Studio para conservar el estado del código, garantizar la cobertura del código y detectar errores y fallos antes de que lo hagan los clientes.

Escenario 1.1 Existe el archivo de configuración.	
Descripción	Cargar el archivo de configuración de Arex, donde vengán especificados los datos para conectarse a cámaras ip.
Variables	
Respuesta del sistema	Se carga el archivo de configuración del sistema.
Flujo Central	La aplicación se inicia automáticamente con el sistema operativo como proceso. / Se inicia o reinicia el proceso mediante comandos de consola siguientes: Detener: /etc/init.d/cedin-domotic-camerastreamer stop Iniciar: /etc/init.d/cedin-domotic-camerastreamer start Reiniciar: /etc/init.d/cedin-domotic-camerastreamer restart
Escenario 1.2 No existe el archivo de configuración.	

Descripción	Cargar el archivo de configuración de Arex, donde venga especificados los datos para conectarse a cámaras ip.
Variables	
Respuesta del sistema	No se puede establecer la comunicación con las cámaras, ya que no existe el archivo de configuración del sistema.
Flujo Central	La aplicación no se inicia automáticamente con el sistema operativo como proceso.

Tabla 5. Prueba de caja negra: Cargar el archivo de configuración de Arex, donde vengan especificados los datos para conectarse a cámaras ip.

Escenario 2.1 Se conecta a la cámara configurada y recibe el streaming de video.	
Descripción	Conectarse a cámaras configuradas y recibir el streaming de video.
Variables	
Respuesta del sistema	Se conecta a la cámara configurada y recibe el streaming de video.
Flujo Central	La aplicación se conecta a la cámara configurada por el sistema y recibe el flujo de video mediante un streaming.
Escenario 2.2 No se conecta a la cámara configurada y no recibe el streaming de video.	
Descripción	Conectarse a cámaras configuradas y recibir el streaming de video.
Variables	
Respuesta del sistema	No se puede conectar con la cámara configurada, por tanto, no recibe el streaming de video.
Flujo Central	La aplicación no se conecta con la cámara configurada, por tanto, no recibe el streaming de video.

Tabla 6. Prueba de caja negra: Conectarse a cámaras configuradas y recibir el streaming de video.

Escenario 3.1 Se genera información en forma de variables procesando el streaming de video de cada cámara.	
Descripción	Generar información en forma de variables procesando el streaming de video de cada cámara.
Variables	
Respuesta del sistema	Se genera información en forma de variables procesando el streaming de video de cada cámara.

Flujo Central	Se generan información en forma de variables luego de procesar el flujo de video, estas variables se actualizan cada 30 milisegundos.
----------------------	---

Tabla 7. Prueba de caja negra: Generar información en forma de variables procesando el streaming de video de cada cámara.

Escenario 4.1 Se atiende la solicitud del cliente para acceder a las variables generadas.	
Descripción	Atender solicitudes de clientes para acceder a las variables generadas.
Variables	
Respuesta del sistema	Se atiende la solicitud del cliente para acceder a las variables generadas.
Flujo Central	El cliente accede a la aplicación escribiendo la siguiente dirección en un navegador web: http://applp:commandsPort/1?variables=all, donde applp sería el ip donde se ejecuta el módulo, commandsPort el puerto de la cámara, el 1 el índice de la cámara y variables= all, el comando para acceder a las variables. Mediante esa petición se devuelven las variables obtenidas.
Escenario 4.2 La petición web está mal escrita.	
Descripción	La petición web está mal escrita.
Variables	
Respuesta del sistema	El sistema no devuelve las variables generadas para la cámara 1.
Flujo Central	El cliente accede a la aplicación escribiendo la siguiente dirección en un navegador web: http://applp:commandsPort/1?, no se escribe el comando variables= all por tanto no se pueden acceder a las variables. Esto es solo un ejemplo de una url mal escrita.

Tabla 8. Prueba de caja negra: Descripción Atender solicitudes de clientes para acceder a las variables generadas.

3.3.2 Pruebas de aceptación

La prueba de aceptación es ejecutada antes de que la aplicación sea instalada dentro de un ambiente de producción. La prueba de aceptación es generalmente desarrollada y ejecutada por el cliente o un especialista de la aplicación y es conducida a determinar como el sistema satisface sus criterios de aceptación validando los requisitos que han sido levantados para el desarrollo, incluyendo a documentación y procesos de negocio. (Londoño, 2005)

Basado en esta prueba el cliente determina si acepta o rechaza el sistema. Estas pruebas están destinadas a probar que el producto está listo para el uso operativo. Suelen ser un subconjunto de las Pruebas de Sistema. (Londoño, 2005)

Sirve para que el usuario pueda validar si el producto final se ajusta a los requisitos fijados, es decir, si el producto está listo para ser implantado para el uso operativo en el entorno del usuario. Esta prueba es complementada por la prueba de estilo. (Londoño, 2005)

La gráfica muestra el resultado de las pruebas de aceptación realizadas al sistema. En la primera iteración fueron detectadas tres no conformidades de implementación, las cuales fueron solucionadas. En la segunda iteración fueron detectadas dos no conformidades de implementación, las cuales fueron solucionadas. En la tercera iteración fueron detectadas una no conformidad de implementación, la cual fue solucionada. En la cuarta iteración no se detectaron no conformidades.

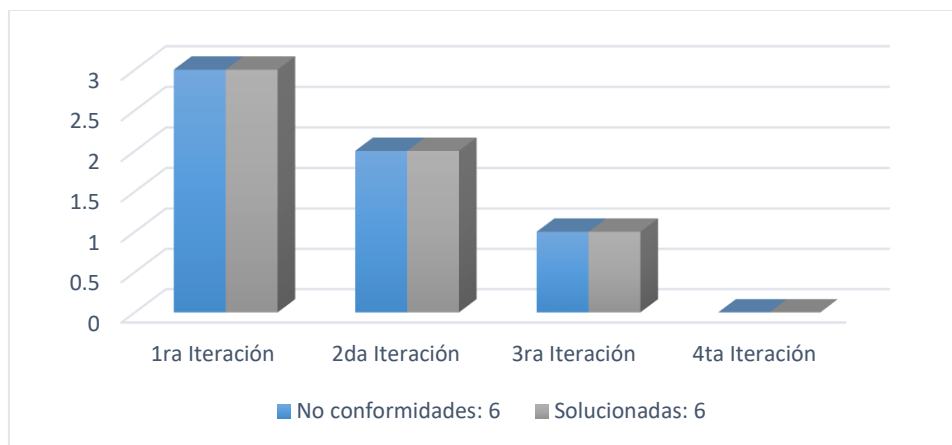


Figura 14 Pruebas de Aceptación

Conclusiones Parciales

En el presente capítulo se desarrolló el diagrama de despliegue del sistema que permite estructurar la arquitectura del sistema desde el punto de vista del despliegue de los artefactos del software. Se realizaron las pruebas pertinentes para garantizar la calidad, eficiencia y buen funcionamiento del módulo de las cuales se escogieron las pruebas de aceptación debido a que el cliente determina la aceptación o rechazo del sistema y las pruebas unitarias para verificar el funcionamiento de cada componente individualmente, además, tienen un gran beneficio en el desarrollo de aplicaciones por un lado mejoran la calidad del software, reducen los tiempos de depuración y evitan muchos errores de programación.

Conclusiones

En la presente investigación se definió como objetivo principal desarrollar un módulo de recolección de variables mediante cámaras para el Sistema de Medición y Adquisición Arex, al concluir la misma se dio solución al problema propuesto y se obtuvieron resultados como son los mencionados a continuación:

- ✂ Se realiza un buen estudio del arte que permite conocer que no existen soluciones que resuelvan totalmente el problema planteado.
- ✂ Se identificó las herramientas y tecnologías para poder desarrollar el módulo, como por ejemplo la biblioteca OpenCV que proporcionó múltiples facilidades a la hora de la generación de variables.
- ✂ Se realizan las pruebas pertinentes para comprobar la buena calidad y buen funcionamiento del módulo.
- ✂ Se realiza el módulo de recolección de variables mediante cámaras que puede ser incluido al Sistema Arex.

Recomendaciones

Se propone darle continuación a la investigación para ello se podrían generar más variables y realizar funciones a partir de las mismas.

Bibliografía

- Bishop, Josefo. 1995.** *Reconocimiento de patrones*. Mexico : s.n., 1995.
- Cambiaso, Diego. 2018.** PIXELCO Blog. [En línea] 06 de 03 de 2018. <https://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/>.
- Chazallet, Sebastien. 2016.** *Python 3 Los fundamentos del lenguaje*. Barcelona : ENI, 2016.
- Cillero, Mnuel. 2009.** manuel.cillero.es mi circunstancia digital. [En línea] 2 de Diciembre de 2009. <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-clases/>.
- Costilla, David Mateos y Montoro, Samuel Reaño. 2015.** *Streaming de Audio/Video. Protocolo RTSP*. 2015.
- Diaz, Julio Gutierrez. 2019.** Cámaras de Vigilancia. [En línea] Copyright 2019, 18 de Enero de 2019. https://camarasdevigilancia10.info/camaras-ip/#Que_es_una_camara_IP.
- Escobar, Germán. 2017.** Make it Real Camp. *Make it Real Camp*. [En línea] 09 de Agosto de 2017. <https://blog.makeitreal.camp/el-protocolo-http/>.
- Fernandez, Luciano Duran. 2016.** LUMIWEM. [En línea] 2016. [Citado el: 20 de Mayo de 2019.] <https://lumixen.com/que-son-las-camaras-ip-protocolo-de-internet/>.
- Flores, Eder Flores. 2017.** *METODOLOGÍA DE DESARROLLO DE SOFTWARE*. CHIMBOTE-PERU : s.n., 2017.
- Fuentes, Lidia y Vallecillo, Antonio. 2015.** *Una Introducción a los Perfiles UML*. Malaga : s.n., 2015.
- Golondrino, Gabriel E Chanchí, Ordoñez, Franco A Urbano y Muñoz, Wilmar Y Campo. 2015.** *Pruebas de estrés para servicios de videostreaming basados en el protocolo RTSP*. Colombia : s.n., 2015.
- Gracia, LuisMi. 2013.** Un poco de Java y +. [En línea] 9 de 10 de 2013. <https://unpocodejava.com/2013/10/09/que-es-opencv/>.
- . 2013. UN POCO DE JAVA Y + Otra forma de hablar de nuestro día a día... [En línea] 2 de Enero de 2013. <https://unpocodejava.com/2013/01/02/un-poco-de-patrones-de-diseno-gof-gang-of-four/>.
- . 2013. UN POCO DE JAVA Y + Otra forma de hablar de nuestro día a día... *UN POCO DE JAVA Y + Otra forma de hablar de nuestro día a día...* [En línea] 2 de Enero de 2013. <https://unpocodejava.com/2013/01/02/un-poco-de-patrones-de-diseno-gof-gang-of-four/>.
- Hidalgo. 1997.** 1997.
- Hoogenraad, Wim. 2018.** ITpedia. *ITpedia*. [En línea] 24 de Julio de 2018. <https://es.itpedia.nl/2018/07/24/scripttaal-is-iets-anders-dan-programmeertaal/>.
- Huidobro, José Manuel. 2007.** *La Domótica como Solución de Futuro*. Madrid : Thomson-Paraninfo, 2007.
- Larman, Craig. 2003.** *UML y Patrones UNA INTRODUCCIÓN AL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS Y AL PROCESO UNIFICAD*. Madrid : PEARSON EDUCACIÓN. S.A., 2003.

Linarte, Francisco J Granera, Rocha, Othniel A Reyes y Morales, Michael J Morales. 2016. *AUTOMATIZACIÓN DEL CONTROL DE LOS DIAGNÓSTICOS Y TRATAMIENTOS APLICADOS A LOS PACIENTES DEL CONSULTORIO DENTAL DE LA CLÍNICA "SANTA ANA" IMPLEMENTANDO LA ARQUITECTURA CLIENTE-SERVIDOR.* NICARAGUA : s.n., 2016.

Londoño, Jorge Hernán Abad. 2005. Ingeniería de Software . *Ingeniería de Software* . [En línea] 6 de Abril de 2005. <http://ing-sw.blogspot.com/2005/04/tipos-de-pruebas-de-software.html>.

Mata, Francisco Javier García. 2011. *Videovigilancia: CCTV usando vídeos IP.* s.l. : Vértice, 2011.

Molina Romero, Javier y Quishpi Betún, Luis. 2015. *Desarrollo de herramienta de gestión de proyectos RUP usando metodología Scrum + XP.* Madrid : s.n., 2015.

Ortiz, Eduardo Perez. 2018. INFAIMON Su solución en visión artificial . *INFAIMON Su solución en visión artificial* . [En línea] 18 de enero de 2018. <https://blog.infaimon.com/vision-computador-soluciones-permite/>.

Pearson. 2019. Lucidchart. [En línea] 22 de enero de 2019. <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml?a=0>.

Pillou, Jean-François. 2018. CCM. *CCM.* [En línea] 17 de enero de 2018. <https://es.ccm.net/contents/275-protocolo-de-comunicacion>.

Pressman, Roger S. 2010. *Ingeniería del software Un enfoque práctico.* Mexico : Miembro de la Cámara Nacional de la Industria Editorial Mexicana, Reg. Núm. 736, 2010.

Quijano, Juan. 2018. GENBETA. *GENBETA.* [En línea] 15 de Marzo de 2018. <https://www.genbeta.com/desarrollo/que-pruebas-debemos-hacerle-a-nuestro-software-y-para-que>.

Rodríguez, Daniel Santamaría. 2017. *EL STREAMING Y SUS IMPLICACIONES EN EL MARKETING.* . España : s.n., 2017.

Santos, Barbara. 2018. hotmart/Blog. [En línea] 09 de 04 de 2018. <https://blog.hotmart.com/es/que-es-streaming/>.

Sarmiento, Johana. 2013. UML: Diagrama de Despliegue . *UML: Diagrama de Despliegue* . [En línea] 12 de Abril de 2013. <http://umldiagramadespliegue.blogspot.com/>.

2017. solvingadhoc. [En línea] 18 de Diciembre de 2017. <https://solvingadhoc.com/las-historias-usuario-funcion-agilidad/>.

Sosio, Nicolás. 2013. S.O.S Seguridad. [En línea] 25 de Mayo de 2013. <http://www.seguridadsos.com.ar/camaras-de-seguridad-tipos-y-modelos/>.

Stroustrup, Bjarne. 2007. *El lenguaje de programación C++.* s.l. : Pearson Educación, 2002, 2007.

Theodoridis, Contrespo y Koutroumbas, Gregori. 1999. *Inteligencia artificial.* España : s.n., 1999.

Torrecilla, Pablo. 2012. El Proceso Unificado Ágil: fases y disciplinas. [En línea] 07 de 06 de 2012. <http://nosolopau.com/2012/06/07/mas-sobre-el-proceso-unificado-agil-fases-y-disciplinas/>.

Torres, Jesus. 2013. Medium. [En línea] 28 de 06 de 2013. <https://medium.com/jmtorres/proyecto-qt-framework-de-desarrollo-de-aplicaciones-2b2f895ac285>.

Villaroel, Javier. 2017. Porqueria. [En línea] 20 de Enero de 2017.
<https://ingeneriadesoftwareutmachala.wordpress.com/2017/01/20/requerimientos-funcionales-y-no-funcionales/>.

Glosario de Términos

Haarcascade: es un clasificador que se utiliza para detectar el objeto para el que ha sido entrenado, desde la fuente. Se entrena superponiendo la imagen positiva sobre un conjunto de imágenes negativas.

TensorFlow: es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, y desarrollado por Google para satisfacer sus necesidades de sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usado por humanos.

Tracking: es un término en inglés, traducible como seguimiento, que se utiliza para hablar de todas aquellas acciones con las que se siguen y se miden índices utilizados para cuantificar el comportamiento de determinadas variables en distintos momentos del tiempo.

RAM: Es una memoria de acceso aleatorio (Random Access Memory) se utiliza como memoria de trabajo de computadoras y otros dispositivos para el sistema operativo, los programas y la mayor parte del software. En la RAM se cargan todas las instrucciones que ejecuta la unidad central de procesamiento (Procesador) y otras unidades del computador, además de contener los datos que manipulan los distintos programas.

URL: es un localizador de recursos uniforme (más conocido por las siglas URL, del inglés Uniform Resource Locator) cuyos recursos referidos pueden cambiar, esto es, la dirección puede apuntar a recursos variables en el tiempo. Están formados por una secuencia de caracteres de acuerdo a un formato modélico y estándar que designa recursos en una red, como, por ejemplo, Internet.

Vigilancia: es el monitoreo del comportamiento. La vigilancia es la acción de observar las actividades de personas o grupos desde una posición de autoridad.

Framework: es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia. Para enfrentar y resolver problemas de índole similar. En el desarrollo de software, es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que pueden servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Sistema Domótico:

Sensor: es un dispositivo capaz de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas. Es todo aquello que tiene una propiedad sensible a una magnitud del medio, y al variar esta magnitud también varía con cierta intensidad la propiedad, es decir, manifiesta la presencia de dicha magnitud, y también su medida.

Actuador: es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un

elemento final de control, como por ejemplo una válvula. Son los elementos que influyen directamente en la señal de salida del automatismo, modificando su magnitud según las instrucciones que reciben de la unidad de control.

Variables: es una palabra que representa aquello que varía o que está sujeto a algún cambio. Se trata de algo que se caracteriza por ser inestable, inconstante y mudable. En otras palabras, una variable es un símbolo que permite identificar a un elemento no especificado dentro de un determinado grupo.

Pentium: es una gama de microprocesadores de arquitectura x86 desarrollado por Intel. Posee versiones de un único núcleo y de multinúcleo.