

FACULTAD 4



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

**Título: Módulo Servidor de Streaming de video desde cámaras IP para el  
Sistema de Medición y Adquisición de datos AREX**

**Autor: Rosalena Hasty Rojas**

**Tutores:**

**MSc. Tatiana Leyva Estrada**

**Ing. Manuel Fórnés Martínez**

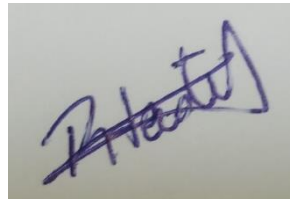
**Ing. Yaritza Bárbara González Ramírez**

**Ing. Jordanis Viltres Chávez**

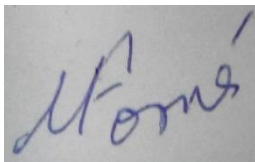
**La Habana, 1º de diciembre del 2022**

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 1<sup>os</sup> días del mes de diciembre del año 2022.



Firma del Autor  
Rosalena Hasty Rojas



Firma tutor  
Ing. Manuel Fórnes Martínez



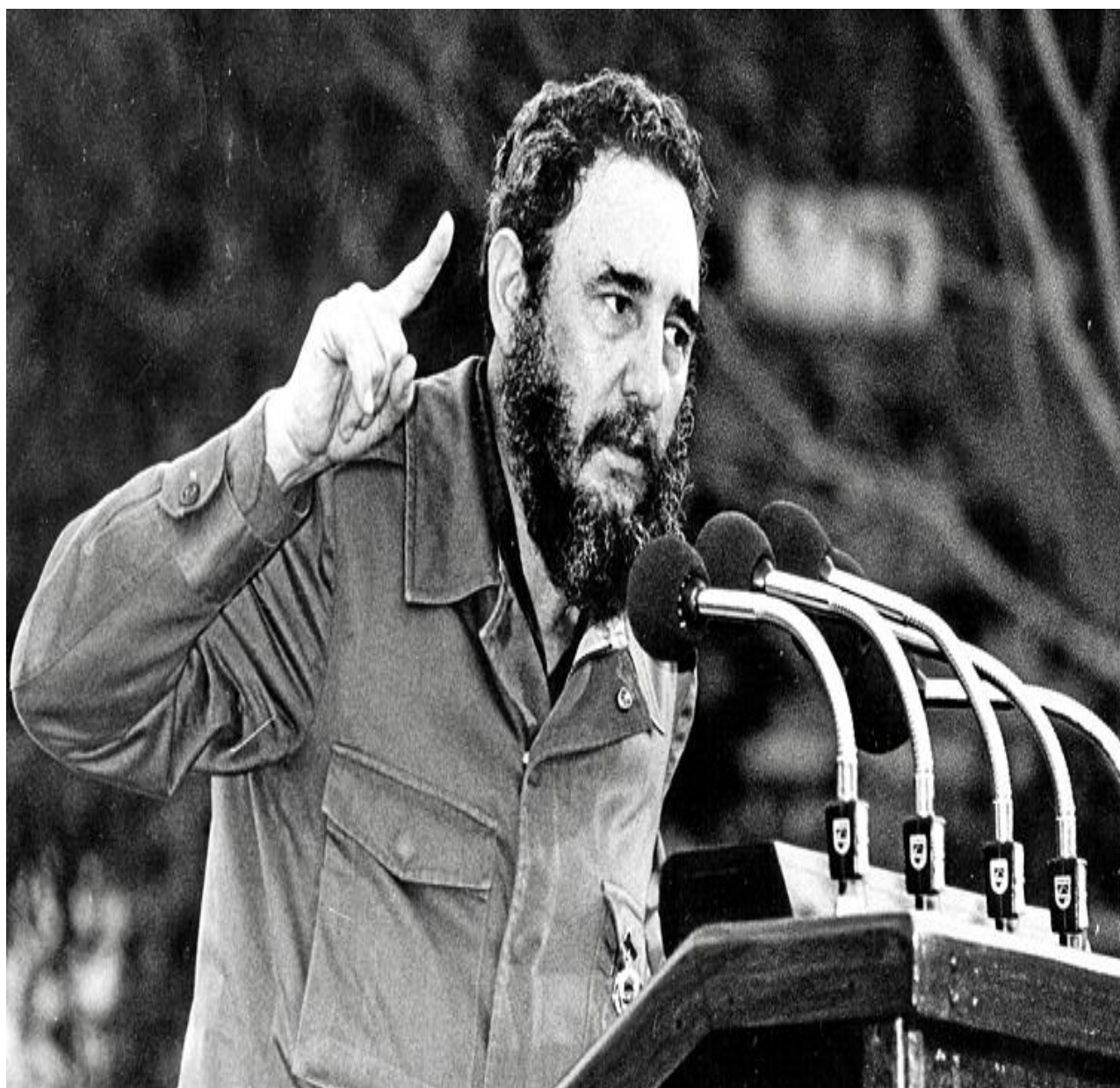
Firma tutor  
MSc. Tatiana Leyva Estrada

Firma tutor  
Ing. Jordanis Viltres Chávez

Firma tutor  
Ing. Yaritza Barbara Gonzáles Ramírez

“Un arquitecto, un ingeniero de cualquier especialidad que conozca la computación, puede multiplicar muchas veces su productividad y hacer las cosas con más exactitud”.

**Fidel Castro Ruz**



## **Dedicatoria**

Esta tesis está dedicada a mi mamá y a mi papá que han sido mi motor impulsor, mis guías, siempre han estado conmigo y me han apoyado en los momentos más difíciles a lo largo de mi vida. Los amo.

También a mis abuelas Virgen y Rosita, que donde quiera que estén sé que están orgullosas de mí. A mi hermano que siempre me ha apoyado y a mi esposo que ha luchado estos 5 años junto a mí y no me ha dejado sola.

## **Agradecimientos**

La gratitud es un hermoso valor subjetivamente simple de mostrar, expresado totalmente del corazón y guardado para esas personas que acompañaron o estuvieron presentes en mi vida. Por lo que quiero agradecer a toda mi familia por el apoyo incondicional que me han brindado a lo largo de estos años, en especial a mi papá Mauricio Hasty y mi mamá Elena Rojas que ellos han dado todo para enseñarme hacer una mejor persona, gracias por estar siempre a mi lado en los momentos más difíciles de la carrera y de mi vida, a demostrarme que no existe obstáculo para lograr mis sueños. A mi hermano Yosbanys Rojas por apoyarme cuando lo necesitaba, a no dejarme sola, es el mejor hermano del mundo. A el amor de mi vida Walter Simón que durante estos 6 años ha estado conmigo luchando para que este sueño se haga realidad, gracias mi amor por tu amor incondicional, por enseñarme lo que es un amor verdadero y sin barreras.

A mis tíos Miriamnis y José que han sido mis segundos padres, a mi tía Marta que con sus peleas me enseñó a ser una mejor persona y a confiar en mí, a mi tía Margot que es como mi abuela mil gracias por tus consejos y tus oraciones.

También a mis suegros y a mi cuñado que estuvieron presente en estos años aconsejándome, gracias por la confianza que me han dado.

A mi familia UCI en especial a Andrés, a Dayana y a Jesús que desde el primer día de clases se convirtieron en mis hermanos, mis amigos incondicionales. A Made que me acogió en su casa como si fuera su familia, gracias por enseñarme a cocinar, a lavar, por apoyarme y ayudarme cuando la necesitaba. A Vilmita que con sus locuras ha estado presente en todo momento difíciles y de fiestas sacando la mejor sonrisa. A mi amigo Yoelvis que a pesar de la distancia me transmitió todas sus vibras para cumplir mis metas. A Viví que después que repetimos se convirtió en mi amiga incondicional y juntas logramos llegar a la meta, gracias por todo. A mi amigo Leudis que gracias a él pude aprobar inglés. A la profe Yadira Ramírez mi jefa de año que gracias a sus exigencias aprendí a no dejar las cosas para último y que, esforzándome más lograba lo que me propusiera. Gracias a la Universidad de las Ciencias Informáticas y a todos los profesores que me impartieron clases durante estos años.

Y de último, pero no menos importante mil gracias a mis tutores en especial a Yaritza y Jordanis Viltres que desde el primer día me ayudaron en todo, sino fuera por ellos esto no sería posible, no tengo como agradecer el esfuerzo y la dedicación, gracias.

Gracias a todas las personas que de una forma u otra dedicaron parte de su tiempo a obtener esta victoria.

## Resumen

La Universidad de Ciencias Informáticas cuenta con varios centros de investigación y desarrollo de software, entre los que se encuentra el Centro de Tecnologías Interactivas (VERTEX) que tiene como objetivo principal desarrollar productos y servicios informáticos asociados a los Entornos Interactivos 3D. El Sistema de Medición y Adquisición (AREX) es un ejemplo de los productos desarrollados en VERTEX, que permite el monitoreo y control de procesos de mediana y pequeña complejidad. Este sistema no cuenta con mecanismos que soporte el trabajo multimedia, por lo que provoca que no se pueda realizar un manejo adecuado de la información proveniente de fuentes de adquisición como cámaras IP, que requiera un tratamiento en tiempo real. El objetivo de la investigación es desarrollar un módulo servidor de streaming de video desde cámaras IP para AREX, que permita aumentar la seguridad de las empresas y hogares.

En el trabajo de diploma se realiza un análisis de los conceptos, las tecnologías y herramientas que se deben utilizar para la implementación y diseño del módulo, entre las que 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. Además, se utiliza la biblioteca LibVLC para el acceso y retransmisión de flujos de video y audio en tiempo real, se selecciona los protocolos de comunicación HTTP y RTSP. Se obtuvo un módulo a partir de la arquitectura cliente servidor, definiendo los requisitos funcionales y no funcionales que posee el sistema. Por último, se realizarán las pruebas unitarias y de integración, para garantizar la calidad, eficiencia y el correcto funcionamiento del módulo.

**Palabras claves:** módulo, streaming, cámara, tecnologías, herramientas, protocolos de comunicación.

## Índice

Dedicatoria.....	IV
Agradecimientos .....	V
Resumen .....	VI
Índice .....	VII
Introducción .....	1
Capítulo 1: Fundamentación teórica .....	4
1.1 Sistema de Medición y Adquisición de datos AREX. ....	4
1.2 Conceptos.....	5
1.2.1 Streaming .....	5
1.2.2 Cámaras IP .....	6
1.2.3 Protocolos de comunicación: .....	9
1.3 Análisis de soluciones existentes .....	12
1.3.1 Comparación de las soluciones existentes.....	13
1.4 Análisis y selección de las herramientas y tecnologías .....	14
1.4.1 Metodología de desarrollo del software.....	14
AUP-UCI .....	14
1.4.2 Lenguaje de programación: C++ .....	15
1.4.3 Herramientas para transmitir Streaming: Biblioteca LibVLC .....	16
1.4.4 Entorno de desarrollo: Qt Creator .....	16
1.4.3 Lenguaje de Modelado: UML .....	16
1.4.6 Herramienta CASE.....	16
Conclusiones Parciales.....	16
Capítulo 2: Diseño de la Propuesta de Solución .....	18
2.1 Propuesta de Solución .....	18
2.1.1 Propuesta de petición de comandos entre el cliente y el servidor streaming. ....	18
2.1.2. Propuesta de transmisión de streaming entre la cámara y el servidor:.....	19
2.1.3. Propuesta de retransmisión de streaming entre el cliente y el servidor: .....	19
2.2. Arquitectura:.....	19
2.3 Requisitos .....	19
2.3.1 Requisitos Funcionales .....	20

2.3.2 Requisitos no funcionales .....	20
2.4 Historias de usuarios.....	20
2.5 Diagramas de Clases.....	21
2.6 Patrones de Diseño.....	23
2.6.1 Patrones GRASP .....	23
2.6.2 Patrones GoF.....	23
2.7 Diagrama de Despliegue.....	24
Conclusiones parciales .....	25
Capítulo 3: Implementación y Prueba .....	26
3.1 Implementación.....	26
3.1.1 Estándares de codificación .....	26
3.2 Pruebas de Software.....	27
3.2.1 Niveles de Pruebas .....	27
3.2.2 Métodos de prueba .....	28
3.2.3 Diseño de casos de prueba.....	28
3.2.4 Pruebas Unitarias.....	29
3.2.5 Pruebas de Integración. ....	29
3.2.6 Resultados de las pruebas:.....	30
3.3 Conclusiones Parciales.....	31
Conclusiones Generales .....	32
Recomendaciones .....	33
Referencia .....	34
Anexos.....	37



## Índice de tablas y figuras

Tabla 1.1 : Comparación de las soluciones existentes .....	14
.....	18
Figura 2.1: Propuesta de solución.....	18
Figura 2.5: Arquitectura del sistema: Cliente Servidor. ....	19
Tabla 2.1: HU 1 Cargar archivos de configuración del sistema AREX .....	21
Figura 2.6: Diagrama de clases .....	22
Figura 2.7: Diagrama de despliegue del Sistema .....	24
Tabla 3.1: Estándares de codificación .....	27
Tabla 3.2: Prueba de caja negra: cargar el archivo de configuración de AREX .....	29
Tabla 3.8: No Conformidades.....	30
Figura 3.1: Gráfica de No Conformidades.....	30
Tabla 2.1: HU 2: Conectarse a las cámaras y recibir streaming de video. ....	38
Tabla 2.2: HU 3 Eliminar conexión a las cámaras .....	38
Tabla 2.3: HU 4 Atender las solicitudes del cliente para acceder al streaming. ....	39
Tabla 2.4: HU 5 Cargar el plugins que permita la conexión y ejecución de comandos a las cámaras.....	39
Tabla 3.3: Prueba de caja negra: Realizar la conexión a las cámaras y recibir el streaming .....	39
Tabla 3.4: Prueba de caja negra: Atender las solicitudes del cliente para acceder al streaming. ....	40
Tabla 3.5: Prueba de caja negra: Eliminar conexión a las cámaras.....	40
Tabla 3.6: Prueba de caja negra: Cargar el plugins que permita la conexión y ejecución de los comandos a las cámaras. ....	41

## Introducción

La seguridad es un componente esencial en todos los procesos empresariales, haciéndose notar en las soluciones informáticas, que tributan al control automatizado de las variables. La creación e integración de software para el manejo de dispositivos de seguridad tales como los sensores y las cámaras de vigilancia, han venido a dar respuesta de forma inteligente para el manejo de la seguridad en las empresas y los hogares (Gouvernement, 2018).

Las cámaras de vigilancias o cámaras IP son dispositivos encargadas de captar, grabar videos de todo lo que ocurra en un espacio; casas, departamentos y negocios (Imsel, 2019). Se puede usar para monitorear un área de forma remota y también se puede controlar a través de una computadora o teléfono inteligente (Seguí, 2020). Estos dispositivos están integrados a los sistemas domóticos, que son capaces de reconocer información proveniente de sensores o entradas, procesarlas y emitir órdenes a través de actuadores o salidas. Contribuyen a una mejor calidad de vida de los usuarios, aportando seguridad mediante la vigilancia automática de personas, animales y bienes, así como de incidencias y averías (Simon, 2021).

La cámara IP (*Internet Protocol*) revolucionó el mercado de la videovigilancia ya que es el dispositivo de video seguridad más completo en la actualidad. Gracias a la tecnología con que cuenta otorga una versatilidad notable, puede ser integrada con otros sistemas de seguridad y funcionar de forma autónoma con una tarea preconcebida. Puede desarrollar tareas de analítica de video desde software externos a ella (Seguí, 2020).

En Cuba son de vital importancia los sistemas de video vigilancia ya que permiten la seguridad de las viviendas y las empresas, para la detección delictiva, proporcionando evidencias para incriminar a las personas en los casos penales. En los últimos años se han desarrollado algunos de estos sistemas para proporcionar un mayor nivel de seguridad a la población.

Actualmente se tiene la necesidad de aumentar la seguridad y el control interno en nuestras instituciones mediante la implementación de un sistema de video vigilancia, con la confiabilidad e integridad de la información que se trasmite y se almacena. En este momento esto constituye un área poco explotada en nuestro país, pero muy utilizada en todas las empresas e instituciones a nivel internacional, llegando a ser en la actualidad una normativa.

La Universidad de las Ciencias Informáticas (UCI) es una universidad científica e innovadora, fundada en el 2002, por el comandante en jefe Fidel Castro Ruz. En esta institución, se cuenta con un Centro de Tecnologías Interactivas (VERTEX) en la línea de trabajo dedicada a la Informática Industrial, donde se han realizado componentes y aplicaciones para la captura y visualización de Streaming de video mediante cámaras. Además, se encuentra el "Sistema de Medición AREX", el cual, está diseñado para el monitoreo de procesos de pequeña y mediana complejidad (Leyva, 2016).

El sistema mencionado anteriormente no cuenta con mecanismos que permitan soportar el trabajo multimedia, por lo que provoca que no se pueda realizar un manejo adecuado de la información

proveniente de fuentes de adquisición como cámaras IP, que requieren un tratamiento en tiempo real.

El centro se ha propuesto incorporarle al Sistema AREX un módulo de streaming, el cual bajo la tecnología en que está desarrollado el sistema, permita cargar los archivos de configuración del sistema AREX, conectarse a las cámaras y recibir streaming de video.

Basado en lo expuesto anteriormente se define como **Problema de investigación:** ¿Cómo contribuir a la seguridad de locales monitorizados mediante cámaras IP?

A partir de la situación problemática que se presenta se define como **objeto de estudio:** Los servidores de Streaming de video desde cámaras IP, específicamente en el **campo de acción:** Los servidores de Streaming de video desde cámaras IP para Sistema de Medición y Adquisición de datos AREX.

Para dar solución al problema de investigación anteriormente se propone como **Objetivo General:** Desarrollar un módulo servidor de Streaming de video desde cámaras IP para el Sistema de Medición y Adquisición de datos AREX.

#### **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.
- Caracterizar las tecnologías actuales para streaming de video.
- Investigar protocolos que utilizan las cámaras IP.
- Describir la propuesta de solución e identificar los requisitos funcionales y no funcionales para el desarrollo de la solución propuesta.
- Aplicar las pruebas a la aplicación implementada para verificar su correcto funcionamiento.
- Desarrollar un módulo de servidor de streaming de video desde cámaras IP para el sistema AREX.

#### **Métodos de investigación**

Para desarrollar esta investigación y lograr los objetivos planteados se utilizan métodos teóricos y empíricos, mediante los cuales se obtiene una idea más detallada de lo que se desea lograr. A continuación, se enuncian los métodos utilizados para la investigación:

#### **Métodos Teóricos:**

- **Histórico–Lógico:** Se utilizó para realizar el estudio del arte, es decir, para realizar el estudio acerca de los sistemas o soluciones similares, además de los lenguajes, herramientas y metodologías para el desarrollo del sistema que se propone.
- **Modelación:** Se empleó para realizar las descripciones de las historias de usuarios necesarios para darle cumplimiento a los requisitos funcionales y no funcionales asociados al sistema. Además, se utilizó en la construcción de diagramas que brinden un mejor entendimiento de la solución.

- **Análisis y Síntesis:** Se utilizó para realizar el análisis y estudio de las bibliografías existentes sobre el tema en cuestión y lograr obtener de manera sintetizada el contenido necesario y suficiente para la realización del presente trabajo.

#### **Método Empírico:**

- **Entrevista:** Se empleo para la recolección de información, en la que se realizaron preguntas objetivas al Jefe de Proyecto para conocer el funcionamiento del Sistema de Medición y Adquisición AREX, y también conocer los problemas más frecuentes de dicho sistema (ver anexo 1).

El presente documento está estructurado en tres capítulos:

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

## **Capítulo 1: Fundamentación teórica**

En el siguiente capítulo se introducen los elementos teóricos fundamentales necesarios para la realización del módulo de Streaming de video de cámaras IP para el sistema de Medición y Adquisición de datos AREX. Además, se describen las principales herramientas y tecnologías a utilizar para la realización y ejecución de la propuesta de solución.

### **1.1 Sistema de Medición y Adquisición de datos AREX.**

Un Sistema de Medición de Datos (SMD) es la colección de instrumentos, indicadores, normas, operaciones, métodos, herramientas, software, personal, ambiente y suposiciones usadas para cuantificar las unidades de medición o evaluación de la característica que se mide. Un proceso de medición puede ser visto como un proceso de manufactura que produce números (datos) para sus producciones. El objetivo de un SMD es indicar el valor de una o varias variables. Para ello toma del medio medido la información y la transforma en una cantidad obtenida a través de un sensor (Leyva, 2016).

Por otra parte, un Sistema de Adquisición de Datos está diseñado fundamentalmente para la medida y el análisis de una o varias magnitudes físicas, características de un sistema real, tomando varias muestras de las mismas (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otro dispositivo de cómputo (sistema digital) (Leyva, 2016). Estos sistemas ofrecen flexibilidad de procesamiento, posibilidad de realizar las tareas en tiempo real o en análisis posteriores (a fin de analizar los posibles errores), gran capacidad de almacenamiento, facilidad de automatización, rápido acceso a la información y toma de decisión, con su utilización se adquieren gran cantidad de datos para poder analizar, posibilidad de emular una gran cantidad de dispositivos de medición y es posible activar varios instrumentos al mismo tiempo (Lapido, 2015).

AREX es un sistema de medición de datos para medir procesos de baja y mediana complejidad, a partir de la adquisición y procesamiento de datos en tiempo real, que puede ser ejecutado en hardware con bajas prestaciones. Para lograr que el sistema aumente la competitividad, es necesario que los despliegues gráficos puedan ser visualizados sobre teléfonos inteligentes o similares, debido al auge que han alcanzado estos dispositivos en el mundo de la tecnología moderna. Para medir el proceso de forma continua AREX debe (Lapido, 2015):

- Recolectar y procesar la información recibida, en forma continua y confiable.
- Representar gráficamente y de forma animada los datos y las variables de proceso.
- Monitorizar las variables de proceso por medio de alarmas.
- Configurar y modificar la evolución del proceso (variables, alarmas, despliegues).
- Transmitir información a dispositivos de campo y otros.
- Alertar al operador de cambios detectados durante un proceso, tanto aquellos que no se consideren normales (alarmas) como cambios que se produzcan en la operación diaria (eventos).
- Almacenar los datos de las variables, eventos y alarmas ocurridas en el sistema.

AREX está formado por cuatro elementos fundamentales: HMI-Edición, HMI-Ejecución, Recolector y Tarjetas de adquisición de datos. Se tienen en cuenta aspectos esenciales en este tipo de sistemas, por ejemplo, el manejo de puntos y alarmas. La comunicación con las tarjetas de adquisición se realiza mediante las interfaces de red Ethernet y Serie sobre el protocolo Modbus en sus variantes TCP y RTU (Leyva, 2016).

A continuación, se describen brevemente los principales módulos de AREX:

- **Recolector:** Este módulo es el encargado de gestionar la comunicación con las Tarjetas de adquisición de datos mediante el protocolo Modbus en sus variantes TCP y RTU. El recolector recibe la configuración de los dispositivos además de sus variables y procede mediante un manejador a la creación de bloques de encuesta, actividad que se realiza atendiendo a ciertos parámetros como son: el tipo de dato, el periodo de muestreo de las variables, el tamaño máximo permitido, la densidad del bloque, entre otros. Este componente de adquisición posee un planificador que permite adicionar y ejecutar tareas de lectura y de escritura sobre las tarjetas de adquisición de datos. La información de los valores de las variables se publica en un servidor de datos (Lapido, 2015).
- **HMI-Edición:** Este módulo puede ser ejecutado en una computadora convencional, permite al usuario crear y configurar despliegues que representen lógicamente el ambiente donde va a ser desplegado el sistema. Permite además la configuración de los dispositivos, sus variables asociadas, así como las alarmas. Las variables son vinculadas a elementos gráficos que forman parte de los despliegues. Finalmente, las configuraciones son exportadas en formato XML (Lapido, 2015).
- **HMI-Ejecución:** Ofrece opciones para el manejo de las alarmas generadas en el sistema. Otra de las funcionalidades soportadas es el envío de comandos de escritura, que posibilita al operador ejercer control sobre los procesos (Lapido, 2015).

El sistema provee las funcionalidades necesarias para contribuir a la seguridad de un local, tributar al ahorro energético y al confort de los habitantes de un recinto mediante la medición de variables como: estado de las luces, temperatura, humedad, apertura o cierre de puertas y ventanas, detección de intrusos (presencia), detección de humo, detección de fuga de gas, detección de inundaciones, detección de polvo (Lapido, 2015).

## **1.2 Conceptos**

En el siguiente epígrafe se hará referencia a los conceptos fundamentales relacionados con la investigación.

### **1.2.1 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 (Hackathon, 2022). 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 (Verizon, 2022).

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 (Hackathon, 2022). 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 (*Hypertext Transfer Protocol*), es poco el trabajo que se ha realizado en cuanto a la generación de herramientas, para realizar pruebas de estrés sobre servidores de Streaming (Verizon, 2022). La transmisión de Streaming de video se realiza preferiblemente con los protocolos de comunicación RTSP y HTTP (Hackathon, 2022). 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 (Verizon, 2022).

### **Ventajas del streaming de video**

- La monetización es uno de los principales beneficios. Hacer una transmisión en streaming es económico y puede servirte para ganar dinero con las emisiones (Pedro, 2019).
- La comunicación que se puede conseguir con una buena transmisión es otra de las principales ventajas del streaming (Pedro, 2019).
- Se puede llegar mejor a los clientes o a integrantes de la plantilla de trabajo si son capaces de crear el mensaje correcto y conseguir que llegue a destino de la mejor manera.
- La aceptación del contenido que ofrece la empresa y el interés por el producto o servicio que se ofrece debe mantenerse potencialmente activo en el destinatario (Pedro, 2019).
- Existen servicios de streaming como la comunicación multidireccional o el vídeo 360°, que permiten que el usuario se sienta más cerca, más comprometido con el evento (Pedro, 2019).
- Otra de las ventajas del streaming es poder configurar las emisiones para que se hagan en directo o en diferido. Se satisface de esta forma a la audiencia que prefiere ver los contenidos en cualquier momento del día, en cualquier horario (Pedro, 2019).

### **Desventajas del streaming de video**

- El principal inconveniente que puede llegar a tener una transmisión en streaming es que no tenga un ancho de banda lo suficientemente amplio como para una emisión de calidad (Pedro, 2019).
- Un servicio de streaming permite la visualización de contenido sin necesidad de descarga y eso requiere de un ancho de banda considerable, que deberá aumentarse a medida que crezca el número de usuarios que se conecten (Pedro, 2019).
- Otra desventaja de la transmisión en streaming es que se deben conocer con *precisión* las prestaciones del servicio y, generalmente, esto es motivo de consulta a empresas especializadas como la nuestra (Pedro, 2019).

### **1.2.2 Cámaras IP**

Las 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 (*Unshielded Twisted Pair*). Una cámara IP es una cámara analógica que en su salida tiene un conversor que hace digital la imagen (López, 2019). 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). Esta cámara 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 (López, 2019). 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 (López, 2019).

Una cámara IP (también conocida como cámara de red) es un dispositivo que contiene:

- 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 (López, 2019).

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 (*Dynamic Domain Name System*) (López, 2019).

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 sitio web 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 video-grabadoras(dvr), combinado con las ventajas de la conexión en red, la captura y manipulación de imágenes digitales y la inteligencia de las cámaras (López, 2019).

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 las interfaces 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 (López, 2019).

Las principales ventajas de las cámaras IP son (Movistar, 2022):

- 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 (Movistar, 2022):

- **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 (*Pan Tilt Zoom*) y señales de control a través de un solo cable conocido como POE (*Power over Ethernet*), 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. Además, cifran y comprimen los datos antes de transportarlo a través de internet a su servidor y son compatibles con VPN (*Virtual Private Network*).
- **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. 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 integrada:** 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.

### 1.2.3 Protocolos de comunicación:

Un protocolo de comunicación es un conjunto de reglas establecidas que determinan cómo se transmiten los datos entre diferentes dispositivos electrónicos. Podríamos decir que es similar a un lenguaje hablado, donde las reglas determinan una conversación, las reglas gramaticales, así como el vocabulario son las que determina el idioma. Los protocolos se pueden apilar en forma de capas, de tal forma que cada protocolo tiene una función determinada. La capa más baja es la base y sobre ella se levantan los diferentes elementos de una transmisión de datos. Desde la capa física, hasta el gobierno de paquetes, pasando por el transporte y terminando en la capa de aplicación (Valluerca, 2021).

Streaming es una transmisión mediante un flujo de datos que normalmente se refiere a video y/o audio, la característica que tiene conceptualmente es que permita comenzar su procesamiento antes de ser terminado. El streaming se realiza a través de una red de datos, bien sea una red privada, pública o internet (Valluerca, 2021).

El protocolo de streaming es el conjunto de reglas y capas que permiten hacer una transmisión de video y audio desde un punto de difusión a un punto de recepción, siempre a través de una cada

de red de datos, publica como internet o privada. Existen varios protocolos para poder hacer llegar el vídeo y audio a los clientes de un servidor, ejemplo (Valluerca, 2021):

- **HTTP (*Hypertext Transfer Protocol*)**: El Protocolo de transferencia de hipertexto es un protocolo de comunicación basado en TCP (*Transmission Control Protocol*) / IP que permite transferencias de información, se realiza una petición de datos entre cliente que inicia la conversación y un servidor, no existe control sobre la transmisión dado que lo guarda ningún dato entre dos peticiones en la misma sesión. Cuando TCP y HTTP sufren un error de transmisión, se vuelve a intentar la entrega de los paquetes perdidos hasta conseguir una confirmación de que la información llegó en su totalidad. En transmisión de video para directo, se utiliza el HLS.
- **HLS (*HTTP Live Streaming*)**: Se trata de un protocolo robusto y eficaz, la transmisión en vivo HTTP (HLS) es un protocolo de transmisión de velocidad de bits adaptable basado en HTTP desarrollado por Apple para permitirles eliminar Flash en los dispositivos iPhone. Es un protocolo muy extendido y compatible con casi todos los navegadores web, dispositivos y teléfonos móviles. Su principal característica es que soporta una transmisión con tasa de bits adaptativa, por lo que la calidad se adapta al ancho de banda del cliente Player.
- **RTMP (*Real-Time Messaging Protocol*)**: El protocolo de mensajería en tiempo real fue desarrollado por Macromedia (propiedad de Adobe), para la transmisión de muy baja latencia en tiempo real de video, audio y datos entre un servidor y un reproductor Flash. Fiable y eficiente, funcionó muy bien para la transmisión en vivo. Si bien Flash está en fase de desaparición, RTMP sigue siendo un protocolo de uso común para la transmisión en vivo dentro de los flujos de trabajo.
- **HDS (*HTTP Dynamic Streaming*)**: HTTP Dynamic Streaming es la evolución de RTMP por parte de Adobe. Es un protocolo de transmisión basado en flash que está en proceso de desuso. Permite la transmisión adaptativa y posee una alta calidad. HDS posee baja latencia, si bien es mayor que el RTMP debido al proceso de fragmentación y encriptación.
- **RTP (*Real Time Protocol*)**: Es un protocolo de Internet para la transmisión en tiempo real de datos multimedia en modo unicast o multicast. Trabaja sobre protocolo UDP (*User Datagram Protocol*), tiene una serie de características especiales de tiempo real, como código de tiempo sobre el vídeo y número de secuencias. Posee una serie de limitaciones, no incluye recuperación de pérdida de paquetes, tiene mecanismos para compensar cualquier pérdida menor de datos, pero no garantizar el envío, ni garantizar que los paquetes lleguen ordenados, tampoco que los paquetes lleguen a tiempo ni la calidad de servicio. Requiere el uso de un buffer que se carga antes de comenzar la reproducción para asegurar la calidad del servicio, este buffer es el amortiguador frente a las limitaciones de calidad en el envío y pérdida de paquetes de datos.
- **RTP-FEC (*Real-Time Transport Protocol with Forward Error Correction*)**: RTP con corrección de errores de reenvío, posee una amplia aplicabilidad para reparar

automáticamente cualquier corrupción y pérdida de paquetes. El uso de este protocolo requiere más ancho de banda que RTP.

- **RTCP (Real Time Control Protocol):** RTCP se usa para monitorear las estadísticas de transmisión y proporcionar retroalimentación de calidad de servicio (QoS). RTCP ayuda a la sincronización de múltiples flujos de media del RTP.
- **RSP (Resilient Streaming Protocol):** El protocolo de transmisión resiliente, es la primera tecnología de transmisión en vivo que protege completamente la pérdida de calidad de audio y video durante una transmisión, independientemente de las interrupciones de la red. RSP garantiza la entrega de video y audio completos sin errores, además permite atravesar el firewall de una conexión HTTP. Tiene latencia baja y predefinida para garantizar no almacenar nada en el búfer o tener pérdida de paquetes por inestabilidad de la red en el sitio de transmisión.
- **RTSP (Real Time Streaming Protocol):** Protocolo primitivo de transmisión en tiempo real con baja latencia, utiliza TCP como protocolo de transporte y permite la recepción desde servidores, el cliente establece y controla uno o varios flujos sincronizados de audio y vídeo, está específicamente diseñado para ser usado en transmisiones de vídeo. Permite controlar los servidores para streaming, establece una sesión de control entre el cliente receptor y el emisor y actúa como un control remoto de los servidores multimedia. También puede usar UDP como su protocolo de transporte, pero no es tan fiable y seguro como TCP RTSP es similar en algunos aspectos al protocolo HTTP Live Streaming (HLS). Los servidores RTSP a menudo funcionan junto con el Protocolo de transporte en tiempo real (RTP) y el Protocolo de control en tiempo real (RTCP) para entregar flujos de medios. Es una excelente opción para sistemas de vigilancia, dispositivos IoT (*Internet Of Things*) y SDK móviles (*Software Development Kit*).
- **RIST (Reliable Internet Stream Protocol):** RIST, que significa protocolo de flujo de Internet confiable, es un protocolo de baja latencia y alta disponibilidad adecuado para aplicaciones de larga distancia. Es un protocolo de transporte de código abierto y especificación abierta diseñado para la transmisión confiable de video a través de redes con pérdidas (incluido Internet) con baja latencia y alta calidad. Actualmente se encuentra en desarrollo en el marco del "Grupo de actividades RIST" del Foro de servicios de vídeo.
- **SRT (Secure Reliable Transport):** Se trata de un protocolo de código abierto impulsado por Haivision, está orientado al transporte de señal de vídeo y audio con ultra baja latencia bajo UDP. Es un protocolo de transporte puro, independiente del códec, garantiza que lo enviado y recibido en el decodificador es idéntico y sin pérdida. Permite atravesar el firewall en todo el flujo de trabajo al admitir los modos de envío y recepción. Posee la capacidad de detectar el rendimiento de la red con respecto a la latencia, la pérdida de paquetes, la fluctuación y el ancho de banda disponible. Las integraciones avanzadas de SRT pueden utilizar esta

información para guiar el inicio de la transmisión o incluso para adaptar los puntos finales a las condiciones cambiantes de la red.

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 a través de un navegador web cumpliendo con los requisitos del Sistema AREX. Además, el protocolo que utilizan las cámaras IP es el RTSP ya que permite la conexión, transferencia y transmisión de datos en tiempo real provenientes de ellas, conectados a la red, y el protocolo HTTP permite realizar una petición de datos entre cliente que inicia la conversación y un servidor.

### **1.3 Análisis de soluciones existentes**

Actualmente en el mercado mundial existen varios sistemas que poseen implementada la funcionalidad de streaming de video desde las cámaras IP, los cuales fueron identificados a través de un estudio bibliográfico realizado para llevar a cabo la investigación. Realizar un análisis sobre cada uno de ellos permitió además de caracterizarlos, identificar los principales elementos a incluir en la propuesta a desarrollar. A continuación, se describen las principales características identificadas de cada uno de los sistemas analizados.

#### **Suria:**

Suria es un sistema de video vigilancia soportado sobre tecnología IP, desarrollado en la Universidad de las Ciencias Informáticas (UCI) en el Departamento de Componentes del Centro de Geoinformática y Señales Digitales (GEYSED). Esta solución de software permite la grabación, visualización, recuperación, análisis y gestión de los flujos de videos provenientes de las cámaras IP. Este sistema está formado por 8 módulos: Interfaz general, Grabación, Análisis, Servidor de Administración, Administración2, Recuperador y Visor. Este proyecto se desarrolló bajo la metodología AUP-UCI, utilizaron el protocolo RTSP, lenguaje de modelado UML, como herramienta Case Visual Paradigm (Rosabal Peña, 2017).

#### **Xyma Safe Vision**

En Cuba, la empresa Desarrollo de Aplicaciones, Tecnologías y Sistemas (DATYS), dedicada al desarrollo de aplicaciones informáticas, cuenta entre sus productos con Xyma Safe Vision, software de video vigilancia profesional basado en tecnología IP que tiene como objetivo el monitoreo en tiempo real y la vigilancia de instalaciones y exteriores. Este software permite realizar análisis sobre las grabaciones, permitiendo la navegación sobre las mismas para facilitar el proceso. Sin embargo, esta acción solo se puede realizar sobre videos grabados, no sobre las transmisiones en vivo (Mena, 2015).

#### **iSpy**

iSpy es un software de código abierto válido tanto para usuarios domésticos que únicamente necesiten una o pocas cámaras de vigilancia como para negocios que necesiten un sistema de vigilancia a pequeña escala para ser controlado localmente. Se trata de un programa informático totalmente gratuito que se puede descargar sin coste alguno desde su página web. Una de sus

principales características es que permiten la navegación sobre las grabaciones, permitiendo desplazarse en el tiempo, pero solo sobre las grabaciones, además de permitir la motorización de tantas cámaras como se desee. También, permite grabar video y audio a demanda, incluso programando horarios. Este software no permite la navegación en tiempo real, pero si reafirma el uso de la herramienta LibVLC para la transmisión y procesamiento de los videos (Velasco, 2014).

**TinyCam Monitor (Free)**

Es una aplicación Android para la vigilancia o control remoto de sus cámaras privadas o públicas de red/IP, codificadores de video, DVR (*Digital video recorder*) y webcams. Tiene dos versiones: FREE y PRO, y soporta cámaras IP que presentan características como (Cárdenas, 2018) :

- Formatos multimedia donde cada fotograma o campo entrelazado de una secuencia de video digital es comprimida por separado como una imagen JPEG (MJPEG),
- Métodos para la compresión digital de audio y vídeo (MPEG4),
- Codificación de video de alta frecuencia (H265)
- Protocolo de transmisión en tiempo real (RTSP),
- Pueden comunicarse con otros dispositivos y software en red mediante ONVIF (*Open Network Video Interface Forum*)
- Permite a individuos comunicarse y compartir información con otros individuos sin necesidad de un servidor central que facilite la comunicación mediante las tecnologías P2P (*Peer to Peer*) que son un tipo de arquitectura para la comunicación entre aplicaciones.

TinyCam Monitores la única aplicación en el mercado que soporta tanto detección de movimiento en la aplicación como en la cámara. Posee detección de rostro, escáner de LAN para la detección automática de cámaras, procesamiento en tiempo real de audio (silencio y alarmas) para utilizarlo como monitor de bebé con gráfica de audio, monitorización audio de múltiples cámaras a la vez y múltiples audios de fondo. Brinda soporte de sensores para algunas cámaras; accesos directos para widgets, ventanas flotantes y pantalla de inicio; plugin de automatización Tasker/Locale; soporte para Android Weare interfaz TV (Cárdenas, 2018) .

**Software de Video Vigilancia de la XETID**

Es una empresa de las Fuerzas Armadas Revolucionarias, 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.

**1.3.1 Comparación de las soluciones existentes**

A continuación, se muestra una tabla resumen donde se realiza una comparación entre las soluciones identificadas teniendo en cuenta diferentes criterios:

Criterios de comparación	Suria	Xyma Safe Vision	iSpy	TinyCam Monitor (Free)	Software de Video Vigilancia de la XETID

Código abierto	no	no	si	no	no
Transmisión en vivo	si	no	no	si	
Lenguaje de programación	C++	C++	C++	Android	C++
Herramienta de desarrollo	LibVLC 3.0	LibVLC 3.0	LibVLC 3.0	LibVLC 3.0	OpenCV
Protocolos de comunicación	RTSP	HTTP/ RSTP	HTTP/ RSTP	HTTP/ RSTP	HTTP/ RSTP

Tabla 1.1 : Comparación de las soluciones existentes (Rosabal Peña, 2017) (Velasco, 2014) (Mena, 2015) (Cárdenas, 2018)

Luego de analizar las diferentes soluciones existentes a nivel nacional e internacional se puede concluir que, en Cuba, a pesar de existir empresas dedicadas al desarrollo de sistemas de video vigilancia, no todas las soluciones permiten la transmisión de streaming de video en tiempo real. La mayoría de los productos investigados a nivel internacional son softwares privativos por lo que no se puede obtener su código fuente, lo que imposibilita su integración con el módulo a desarrollar. A pesar de que los productos investigados no brindan solución a la problemática planteada el estudio de sus funcionalidades permitió identificar el uso de los protocolos de comunicación HTTP/RTSP y la herramienta LibVLC 3.0 para la gestión y edición de flujos de video y audio. Por lo anteriormente planteado, se concluye que las soluciones analizadas no satisfacen las necesidades, por tanto, se requiere crear un módulo de streaming de video desde cámaras IP, en el Sistema AREX.

#### **1.4 Análisis y selección de las herramientas y tecnologías**

El proyecto Sistema de Medición AREX cuenta con un marco tecnológico establecido para su desarrollo. A continuación, se presenta detalladamente las características de estas tecnologías y herramientas.

##### **1.4.1 Metodología de desarrollo del software**

Las metodologías son sistemas complejos de técnicas que influyen procedimientos paso a paso, productos resultantes, funciones, herramientas y normas de calidad para la terminación del ciclo de vida completo del desarrollo de sistemas. Es un conjunto de actividades llevadas a cabo para desarrollar y poner en marcha un sistema de información (Universidades, 2020).

##### **AUP-UCI**

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable, se decidió hacer una variación de la metodología AUP (Proceso Unificado Ágil), de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI (Benítez, 2019).

##### **Descripción de las Fases:**

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre (Benítez, 2019).

Esta versión de AUP define **cuatro escenarios** en los que se puede ubicar el desarrollo de una aplicación de acuerdo a sus características, los cuales son (Benítez, 2019):

- Escenario 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta.
- Escenario 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio.
- Escenario 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.
- Escenario 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos.

Siguiendo la política de desarrollo de software de la institución, se define como metodología a emplear la AUP-UCI en el escenario número 4, debido a la necesidad de una metodología que responda con facilidad a los cambios continuos, por estar el cliente siempre acompañando el desarrollo y por estar bien definido el negocio.

#### **1.4.2 Lenguaje de programación: C++**

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina (Stroustrup, 2017).

C++ posee las siguientes características (Stroustrup, 2017):

- Es un lenguaje orientado a objetos.
- Es muy potente y práctico, debido a su robustez, para la creación de sistemas complejos.
- Es independiente de la plataforma.
- Es muy utilizado en el mundo por lo que existe abundante información sobre su uso.
- Existen muchos algoritmos ya desarrollados en C++, de manera que pueden tomarse y amoldarse a la solución deseada.
- Sirve para programar sistemas empotrados.



Además, es el lenguaje definido por el proyecto para el desarrollo e integración con los demás componentes.

#### **1.4.3 Herramientas para transmitir Streaming: Biblioteca LibVLC**

VLC es un reproductor multimedia libre y de código abierto multiplataforma y un framework que reproduce la mayoría de archivos multimedia, así como DVD, Audio CD, VCD y diversos protocolos de transmisión (libbluray, 2022).

VLC es una herramienta imprescindible, ya que no sólo reproduce una lista interminable de archivos multimedia en varios sistemas operativos, sino que también transmite vídeos de manera eficiente en varios dispositivos, sin necesidad de codificación. (Gonzalez, 2022).

#### **1.4.4 Entorno de desarrollo: Qt Creator**

El IDE Qt Creator es una plataforma o soporte, que los desarrolladores de software usan para generar aplicaciones complejas, de forma más sencilla y segura. La plataforma puede constar de varias utilidades, como herramientas, bibliotecas, enlaces multilenguaje, comunicaciones estándar con bases de datos de diferente índole, etc. Incluye (Frameworks, 2019):

- Un avanzado editor de código C++.
- GUI Integrado de diseño y las formas de diseño.
- Herramientas de gestión de proyectos y construcción.
- Integrado, sistema de ayuda sensible al contexto.
- Depurador de Visual.
- Herramientas de navegación de código rápido.
- Soporta múltiples plataformas.

#### **1.4.3 Lenguaje de Modelado: UML**

Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Se utilizó este lenguaje gráfico para visualizar, especificar, construir y documentar el sistema de software. Además, ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (lucidchart).

#### **1.4.6 Herramienta CASE**

Una herramienta CASE es un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en algunas de sus fases (Parscal, 2020).

**Visual Paradigm:** La misma propicia un conjunto de funcionalidades que apoyan al desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Cornelio, 2016).

#### **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 el streaming de video desde cámaras IP. Se escogió el lenguaje de programación C++ 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 seleccionaron 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 permite el flujo de video en tiempo real.

## Capítulo 2: Diseño de la Propuesta de Solución

En el presente capítulo se describe la propuesta de solución, que incluyen los requisitos funcionales y no funcionales que posee el módulo y la descripción de dichos requisitos. Además, se detallan las principales clases que va a poseer el módulo, y aspectos de diseño (el estilo y patrón arquitectónico, los patrones de diseño, el modelo de clases y diagramas de secuencias).

### 2.1 Propuesta de Solución

Para darle respuesta a la problemática planteada se propone desarrollar un módulo que transmita streaming de video desde las cámaras IP para el Sistema de Medición y Adquisición de Datos ARES. La solución estará diseñada para permitir que el módulo a desarrollar se autentique en el módulo seguridad para obtener la configuración del servidor y detectar la configuración de las cámaras definidas en el módulo Editor de ARES. Además, debe permitir, utilizando la configuración obtenida del servidor, realizar la configuración de cada una de las cámaras. El módulo debe poder escuchar las peticiones de los clientes a través de la ejecución de comandos mediante una URL y a su vez el módulo debe pedir la ejecución de comandos a la cámara. Después realizada la ejecución de comandos la cámara debe envía el streaming de video a el servidor streaming mediante una URL y este se la debe retransmitir al cliente por medio de otra URL.

A continuación, se muestra una imagen donde se podrá visualizar la propuesta descrita anteriormente:



Figura 2.1: Propuesta de solución.

#### 2.1.1 Propuesta de petición de comandos entre el cliente y el servidor streaming.

Para la petición y ejecución de comandos se propone la siguiente URL: `http//IP: Puerto/NúmeroCámara? Comando`, donde los posibles valores que pueden tomar los comandos son los siguientes:

- rotate: left, right, up, down, home
- brightness: more, less, [-10 a 10]
- contrast: more, less, [0 a 20]

- saturation: more, less, [0 a 20]
- hue: more, less, [-10 a 10]
- zoom: in, out, more, less, [0 a 100]

Por ejemplo:

`http://10.30.20.1:5000/1?brightness=10&saturation=10`

### 2.1.2. Propuesta de transmisión de streaming entre la cámara y el servidor:

Para la transmisión de streaming se propone la siguiente URL (`http://Ip:Puerto/urlVideo` o `rtsp://Ip:Puerto/urlVideo`)

Por ejemplo:

`http://10.30.20.1:5000/image`

### 2.1.3. Propuesta de retransmisión de streaming entre el cliente y el servidor:

Para la retransmisión de streaming se propone la siguiente URL (`http://Ip:PuertoCámara/stream`) donde el puerto de la cámara se debe calcular por la fórmula  $\text{PuertoCámara} = \text{PuertoServidor} + \text{NúmeroCámara}$ .

Por ejemplo:

`http://10.30.20.1:5001/stream`

## 2.2. Arquitectura:

La arquitectura cliente servidor tiene dos partes claramente diferenciadas, por un lado la parte del servidor y por otro la parte de cliente o grupo de clientes donde lo habitual es que un servidor sea una máquina bastante potente con un hardware y software específico que actúa de depósito de datos y funcione como un sistema gestor de base de datos o aplicaciones. En esta arquitectura el cliente suele ser estaciones de trabajo que solicitan varios servicios al servidor, mientras que un servidor es una máquina que actúa como depósito de datos y funciona como un sistema gestor de base de datos, este se encarga de dar la respuesta demandada por el cliente. Esta arquitectura se aplica en diferentes modelos informáticos alrededor del mundo donde su propósito es mantener una comunicación de información entre diferentes entidades de una red mediante el uso de protocolos establecidos y el apropiado almacenaje de la misma (Andrés Schiaffario, 20119).

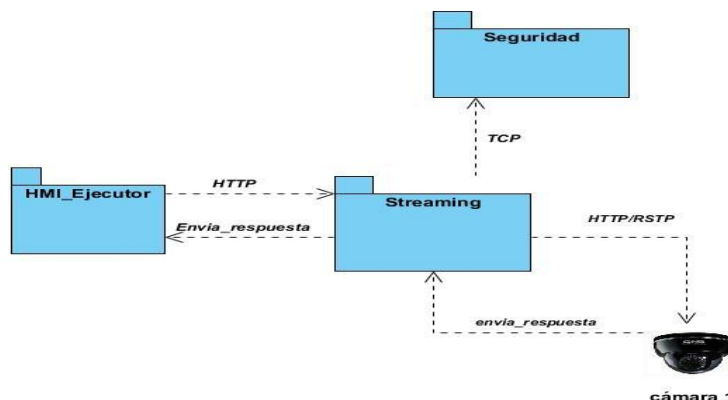


Figura 2.5: Arquitectura del sistema: Cliente Servidor.

## 2.3 Requisitos

Los requisitos son una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio (pmoinformatica, 2017).

### 2.3.1 Requisitos Funcionales

Los requerimientos funcionales son las descripciones explícitas del comportamiento que debe tener una solución de software y que información debe manejar (pmoinformatica, 2017).

Los requisitos funcionales que presenta el sistema son:

- **RF1:** Cargar el archivo de configuración del sistema AREX.
- **RF2:** Realizar la conexión a las cámaras y recibir streaming de video.
- **RF3:** Eliminar conexión a las cámaras.
- **RF4:** Atender las solicitudes del cliente para acceder al streaming.
- **RF5:** Cargar el plugins que permita la conexión y ejecución de comandos a las cámaras.

### 2.3.2 Requisitos no funcionales

Los requisitos no funcionales especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que estos corresponden a los funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento. Son las restricciones o condiciones que impone el cliente al programa que necesita, por ejemplo, el tiempo de entrega del programa, el lenguaje o la cantidad de usuarios (pmoinformatica, 2017).

Los requisitos no funcionales que presenta el sistema son:

- Requisitos no funcionales del software
  - Sistema operativo Windows 10
  - Framework Qt
- Requisitos no funcionales del hardware
  - El sistema requiere de 1GB de memoria RAM
  - El sistema requiere de 30 GB de disco duro

## 2.4 Historias de usuarios

Las historias de usuario forman parte del enfoque Ágil, los cuales, describen las características y necesidades de un software desde la perspectiva de un usuario, ayudando a alinear expectativas y evitar errores críticos en el futuro. Puede considerarse como una preparación para establecer los requisitos del software (Vergara, 2021).

La descripción de historias de usuario permite planificar, priorizar y estructurar el trabajo, mediante un enfoque visual donde se podrá ver, priorizar y evaluar los requisitos (Vergara, 2021).

<b>Descripción del requisito Cargar archivos de configuración del sistema AREX</b>	
Número: 1	<b>Nombre del requisito:</b> Cargar archivos de configuración del sistema AREX
<b>Programador:</b> Rosalena Hasty	<b>Iteración asignada:</b> V1.0
<b>Prioridad:</b> Alta	<b>Tiempo estimado:</b> 3 semana

<b>Riesgo en desarrollo:</b> El programador puede presentar problemas de salud.	<b>Tiempo real:</b> 3 semana
<b>Descripción:</b> El sistema explora el archivo de configuración del sistema AREX, reconoce el identificador correspondiente a cada una de las cámaras de video, carga la configuración de cada cámara.	

Tabla 2.1: HU 1 Cargar archivos de configuración del sistema AREX

## 2.5 Diagramas de Clases

Los diagramas de clase describen los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir entre ellos. Los diagramas de clase se convierten así en la técnica más potente para el modelado conceptual de un sistema software, la cual suele recoger los conceptos clave del modelo de objetos subyacente al método orientado a objetos que la incorpora, en este caso UML 1.1 (lucidchart, 2018).

El propósito de un diagrama de clase es describir las clases que conforman el modelo de un determinado sistema. Dado el carácter de refinamiento iterativo que caracteriza un desarrollo orientado a objetos, el diagrama de clase va a ser creado y refinado durante las fases de análisis y diseño, estando presente como guía en la implementación del sistema (lucidchart, 2018).

Se puede decir que existen tres perspectivas diferentes desde las cuales se pueden utilizar los diagramas de clase (lucidchart, 2018):

- **Conceptual:** El diagrama de clase representa los conceptos en el dominio del problema que se está estudiando. Este modelo debe crearse con la mayor independencia posible de la implementación final del sistema (lucidchart, 2018).
- **Especificación:** El diagrama de clase refleja las interfaces de las clases, pero no su implementación. Aquí las clases aparecen más cercanas a los tipos de datos, ya que un tipo representa una interfaz que puede tener muchas implementaciones diferentes (lucidchart, 2018).
- **Implementación:** Esta vista representa las clases tal cual aparecen en el entorno de implementación. Los diagramas de clase de UML pueden utilizarse en cualquiera de las tres perspectivas (lucidchart, 2018).

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 (lucidchart, 2018).

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 instancias 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 (Lucidchart, 2018).

A continuación, se muestra el diagrama de clases correspondiente a la solución a desarrollar:

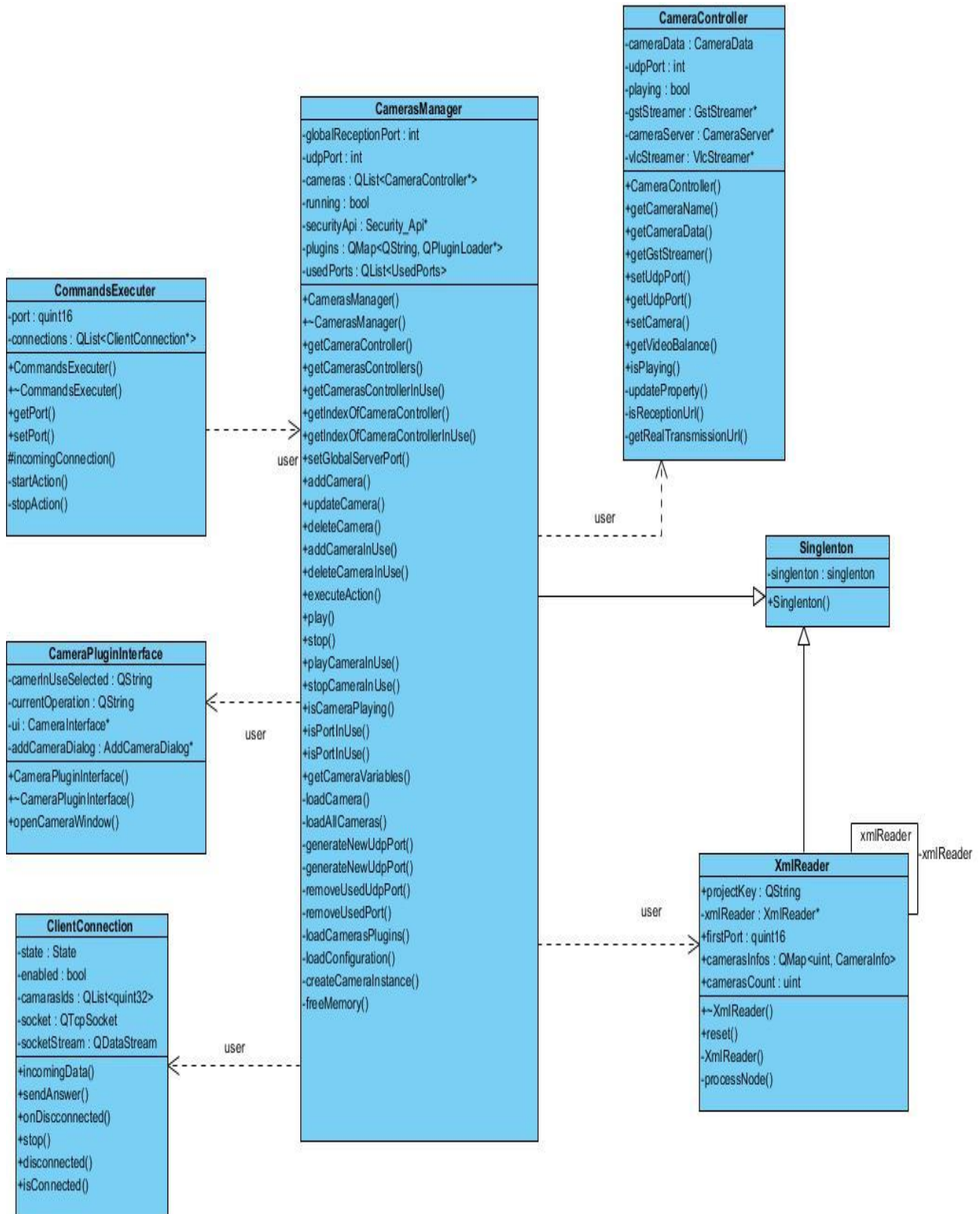


Figura 2.6: Diagrama de clases

## 2.6 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”. 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 (Canelo, 2017).

### 2.6.1 Patrones GRASP

Los patrones GRASP 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 (Canelo, 2017).

Los patrones GRASP tienen nombres concisos como Experto en Información, Creador, Variaciones Protegidas (Canelo, 2017).

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 (Canelo, 2017).

#### 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 (Canelo, 2017).

Este patrón se pone de manifiesto en las clases **CameraManager** 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.

#### Creador

El patrón Creador es 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 (Canelo, 2017).

El Creador sugiere que la clase contenedor o registro es una buena candidata para asignarle la responsabilidad de crear lo que contiene o registra (Canelo, 2017).

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

### 2.6.2 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 (Canelo, 2017):

- **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:** Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella (Gamma, 1994).

Este patrón se empleó en la clase **XmlReader** para garantizar que exista una única instancia de dicha clase para todo el proyecto.

## 2.7 Diagrama de Despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos (Lucidchart).

El diagrama de despliegue que se muestra a continuación representa la distribución física del sistema a través de nodos. Está compuesto por un nodo llamado Módulo Streaming, que es el módulo que se desea desarrollar, un nodo llamado Módulo Seguridad, un nodo Cliente (Módulo Ejecutor) y el nodo Cámara IP, que se comunican con el Módulo Streaming, donde la comunicación entre ellos se lleva a cabo a través de Protocolos de Comunicación.

### Descripción de los conectores:

- **RTSP:** esta conexión se realizará entre el Módulo Streaming y la Cámara IP, mediante una URL.
- **HTTP:** esta conexión se realizará entre el Cliente y el Módulo Streaming, y también entre el Módulo Streaming y la Cámara IP, mediante una URL.
- **TCP:** esta conexión se realizará entre el Módulo Streaming y el Módulo Seguridad.

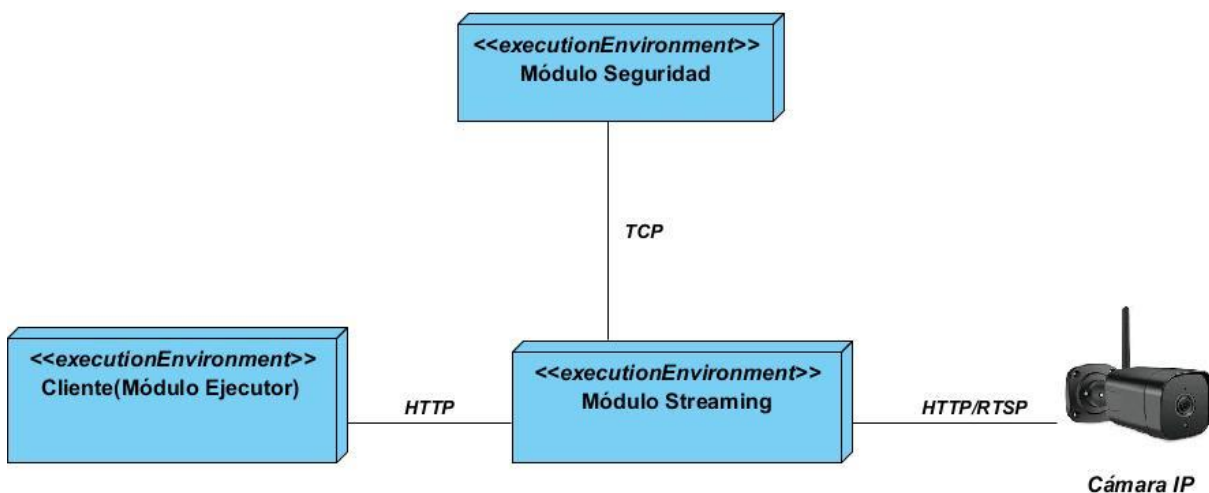


Figura 2.7: Diagrama de despliegue del Sistema

## **Conclusiones parciales**

En el presente capítulo se describió la propuesta de solución para el módulo a desarrollar, así como los elementos que lo componen y sus principales funciones, escogiéndose 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. Además, se definieron los principales requisitos funcionales y no funcionales con que cuenta el sistema a partir de las historias de usuario, se crea el diagrama de clases que permite recoger las clases de objetos y sus asociaciones entre ellas. También se describieron los patrones de diseño utilizados para lograr una correcta estructuración del sistema y se desarrolló el diagrama de despliegue del sistema que consiente estructurar la arquitectura del sistema desde el punto de vista del despliegue de los artefactos del software.

### Capítulo 3: Implementación y Prueba

En el presente capítulo se abordan los aspectos más importantes referentes a la implementación y a las pruebas del módulo de Streaming de video desde cámaras IP. Además, se enuncian los estándares de codificación que se siguieron para generar el código fuente y por último se realizan pruebas al módulo implementado a través de los casos de pruebas.

#### 3.1 Implementación

La fase de implementación del software posee como entradas los artefactos de diseño, como: diagramas de clases, especificación de arquitectura, patrones a emplear en el sistema, entre otros. En la implementación se define el estándar de codificación a emplear y se realizan las implementaciones a las historias de usuarios.

##### 3.1.1 Estándares de codificación

Descripción	Ejemplo
<b>Definición de objetos, clases, funciones y atributos</b>	
Todos los nombres de las clases implementadas comenzarán con letra mayúscula. En caso de poseer un nombre compuesto se escribirán de acuerdo a la normativa CamelCase-UpperCamelCase.	<pre>class Foo {     cuerpo de la clase }  class FooFirst {     cuerpo de la clase }</pre>
La declaración de funciones o métodos siempre comenzarán con letra inicial minúsculas. En caso de ser un nombre compuesto se regirá por la normativa CamelCase-lowerCamelCase.	<pre>&lt;Tipo dato retorno&gt; funcionDobleCompuesta ()</pre>
Los atributos siempre estarán escritos con letra minúscula. En caso de ser un nombre compuesto se regirá por la normativa CamelCase-lowerCamelCase.	<pre>&lt;Tipo dato&gt; atributo;  &lt;Tipo dato&gt; atributoNombreCompuesto;</pre>
<b>Definición de parámetros dentro de las funciones y constructores de clases</b>	
Los nombres de los identificadores de los parámetros en las funciones deben estar escritos con minúsculas separados a un	<pre>&lt;Tipo dato retorno&gt; función(&lt;tipo&gt;&lt;id1&gt;, &lt;tipo&gt;&lt;id2&gt;, &lt;tipo&gt;&lt;id3&gt;)</pre>

espacio cada uno y en caso de ser compuesto utilizar normativa CamelCaseLowerCamelCase.	
Los identificadores de los parámetros dentro de los constructores de las clases deben estar escritos con minúsculas separados a un espacio cada uno y en caso de ser compuesto utilizar normativa CamelCaseLowerCamelCase.	Clase(<tipo><id1>, <tipo><id2>, <tipo><idN>)
<b>Comentario en el código según el estándar de c++</b>	
Comentarios pequeños.	<code>/* comentario sencillo */</code>
Otros comentarios.	<code>/* *Comentario */</code>
Comentario de versión, descripción de clase y otras características de la clase o paquete.	<code>/* *****  *Comentario amplio *  ***** */</code>

Tabla 3.1: Estándares de codificación

### 3.2 Pruebas de Software

El proceso de pruebas de software tiene los siguientes objetivos (Sommerville, 2006):

- Demostrar al desarrollador y al cliente que el software satisface sus requisitos. Esto significa que debería haber al menos una prueba para cada requerimiento o característica que se incorporará a la entrega del producto.
- Descubrir defectos en el software en el que el comportamiento de este es incorrecto, no deseable o no cumple su especificación. La prueba de defectos está relacionada con la eliminación de todos los tipos de comportamientos del sistema no deseables, tales como caídas del sistema, interacciones no permitidas con otros sistemas, cálculos incorrectos y corrupción de datos.

#### 3.2.1 Niveles de Pruebas

Para aplicarle pruebas a un sistema se deben tener en cuenta una serie de objetivos en diferentes escenarios y niveles de trabajo, debido a que las pruebas son agrupadas por niveles que se

encuentran en distintas etapas del proceso de desarrollo. Los niveles de prueba son (Sommerville, 2006):

- **Prueba Unitaria o de Unidad:** se centra en el proceso de verificación de la menor unidad del diseño del software: el componente de software o módulo. Se emplea para detectar errores debidos a cálculos incorrectos, comparaciones incorrectas o flujos de control inapropiados. Las pruebas del camino básico y de bucles son técnicas muy efectivas para descubrir una gran cantidad de errores en los caminos.
- **Prueba de Integración:** es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados mediante la prueba unitaria y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.
- **Pruebas de Sistema:** está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas. Algunas de estas son: pruebas de recuperación, seguridad, resistencia, entre otras.
- **Pruebas de Aceptación:** es la realización de una serie de pruebas de caja negra que demuestran la conformidad con los requisitos. Un plan de prueba traza la clase de pruebas que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos.

### 3.2.2 Métodos de prueba

El principal objetivo del diseño de casos de prueba es obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Para llevar a cabo este objetivo, se emplearán los dos métodos de prueba (Sommerville, 2006):

- **Prueba de Caja Blanca:** se centran en la estructura de control del programa. Se obtienen casos de prueba que aseguren que durante la prueba se han ejecutado, por lo menos una vez, todas las sentencias del programa y que se ejercitan todas las condiciones lógicas.
- **Prueba de Caja Negra:** son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de un programa, solo se fijan en las funciones que realiza el software. Las técnicas de prueba de caja negra se centran en el ámbito de información de un programa, de forma que se proporcione una cobertura completa de prueba.

### 3.2.3 Diseño de casos de prueba

Los Casos de Pruebas han sido realizados sobre la base de las Historias de Usuarios y tienen como objetivo fundamental encontrar la mayor cantidad posible de deficiencias existentes en las funcionalidades implementadas.

<b>Escenario 1.1 Existe el archivo de configuración</b>	
Descripción	Cargar el archivo de configuración de AREX, donde vengan especificados los datos para conectarse a cámaras IP.
Respuesta del sistema	Se carga el archivo de configuración del sistema.
Flujo central	El módulo 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
<b>Escenario 1.2 No existe el archivo de configuración</b>	
Descripción	Cargar el archivo de configuración de AREX, donde vengan especificados los datos para conectarse a cámaras IP.
Respuesta del sistema	No se puede establecer la comunicación con las cámaras, ya que no existe el archivo de configuración del sistema.
Flujo central	El módulo no se inicia automáticamente con el sistema operativo como proceso.

Tabla 3.2: Prueba de caja negra: cargar el archivo de configuración de AREX

### **3.2.4 Pruebas Unitarias.**

Las pruebas unitarias son pruebas automatizadas que verifican la funcionalidad en el componente, clase, método o nivel de propiedad (Quijano, 2020). 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, 2020).

Para la realización de esta prueba se utilizó la herramienta Qt Test para conservar el estado del código, garantizar la cobertura del código, detectar errores y fallos antes de que lo hagan los clientes. Al realizar esta prueba se detectaron algunas no conformidades las cuales fueron resueltas.

### **3.2.5 Pruebas de Integración.**

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupo. Se centra principalmente en probar la comunicación entre los componentes y sus comunicaciones ya sea hardware o software (Wik, 2022).

Se realizaron las pruebas de integración para identificar los posibles errores que puede presentar el módulo al integrarlo con el sistema AREX, donde se realizaron pruebas de transmisión de streaming usando el reproductor VLC, se comprobaron la ejecución de comandos por mensaje de consola y se simularon las peticiones de clientes usando el navegador web Firefox, llegando a la conclusión que el módulo se integra satisfactoriamente al Sistema AREX.

### 3.2.6 Resultados de las pruebas:

La realización de pruebas permitió identificar una serie de no conformidades descritas a continuación:

No.NC	Requisito Funcional	Descripción	Complejidad	Estado
1	RF:1	No se encontró el archivo de configuración de AREX	Alta	Resuelta
2	RF:2	Las cámaras no se conectaban al módulo	Alta	Resuelta
3	RF:2	Las cámaras no reconocían los comandos	Media	Resuelta
4	RF:3	La petición de la web está mal escrita	Baja	Resuelta
5	RF:3	La cámara no enviaba el streaming de video al servidor streaming	Media	Resuelta
6	RF6	El módulo no permitió cargar el plugins que permite la conexión y ejecución de los comandos a las cámaras.	Alta	Resuelta

Tabla 3.8: No Conformidades

En la siguiente gráfica se muestran los datos correspondientes a cada iteración de prueba por las que transitó el módulo:

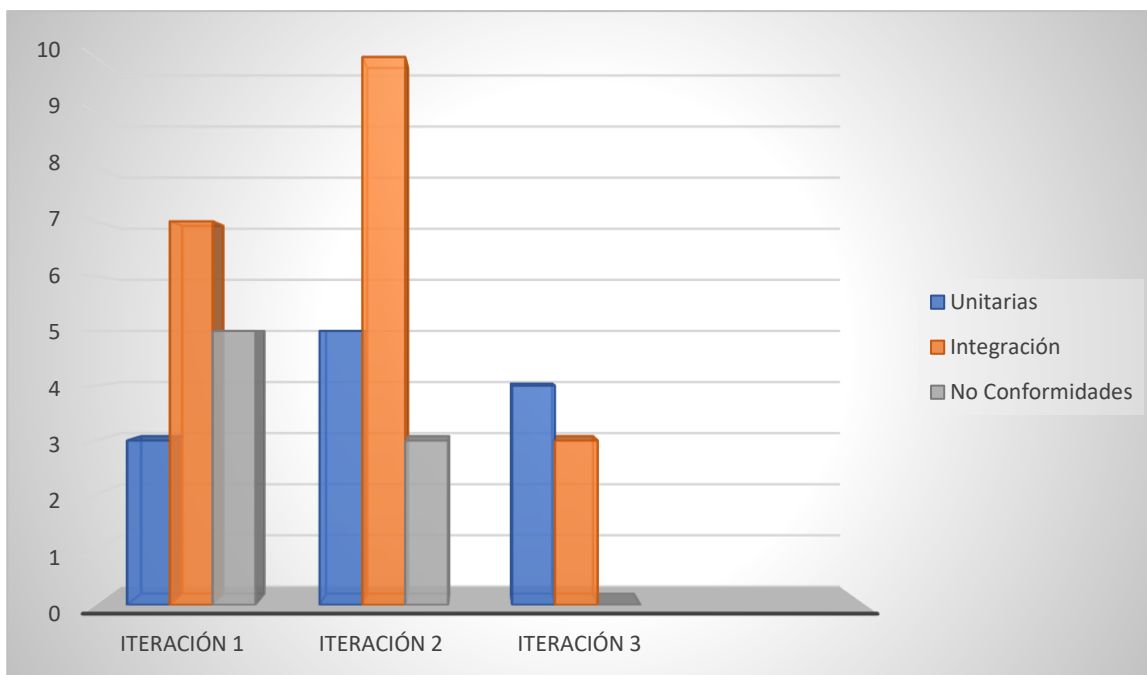


Figura 3.1: Gráfica de No Conformidades

La eficacia de las funcionalidades implementadas para el módulo servidor de streaming de video desde cámaras IP, está avalada por los resultados de las pruebas realizadas, que demostraron el correcto funcionamiento de cada una de ellas.

### **3.3 Conclusiones Parciales**

En el presente capítulo se caracterizaron los estándares de codificación establecidos por el grupo de investigación para la implementación de las clases y los métodos definidos en la fase de diseño. Se ejecutaron las pruebas adecuadas para garantizar la calidad, eficiencia y buen funcionamiento del módulo de las cuales se seleccionaron las pruebas unitarias para verificar el funcionamiento de cada componente individualmente. Además, se realizaron pruebas de integración, para comprobar que el módulo se integraba correctamente al sistema. El resultado de estas pruebas permitió detectar una serie de no conformidades que fueron corregidas, garantizando una mejor calidad de la solución.



## **Conclusiones Generales**

Como resultado del proceso de Investigación-Desarrollo realizado se arribó a las siguientes conclusiones generales:

- El marco teórico elaborado permitió fundamentar teóricamente los conceptos, las tecnologías y herramientas a utilizar para la implementación y diseño del módulo.
- Se utilizó la biblioteca LibVLC para el acceso y retransmisión de flujos de video y audio en tiempo real y se seleccionaron los protocolos de comunicación HTTP y RTSP.
- Se realizó las pruebas unitarias y de integración, para garantizar la calidad, eficiencia y el correcto funcionamiento del módulo.
- Se obtuvo un módulo para el servidor de Streaming de video desde cámaras IP para el Sistema de Medición y Adquisición de datos AREX.

## **Recomendaciones**

Al concluir la presente investigación se propone, a manera de recomendación:

- Utilizar la inteligencia artificial en el reconocimiento de patrones para detectar figuras y reconocimientos de formas.

## Referencia

**Andrés Schiaffario. 20119.** *Arquitectura cliente servidor.* 20119.

**August. 2021.**

<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwil0uzc0fb3AhUwRTABHTzYCHkQFnoECBIQAw&url=https%3A%2F%2Fdocs.citrix.com%2Fes-es%2Fcitrix-adc%2Fcurrent-release%2Fcitrix-adc-support-for-telecom-service-providers%2FIsn>.  
[En línea] 2021.

**Benítez, Drymon Alfonso. 2019.** *Herramienta para generar productos de trabajos de la metodología variación AUP-UCI.* 2019.

**Canelo, Miriam Martínez. 2017.** <https://profile.es/blog/patrones-de-diseno-de-software/>. [En línea] 2017.

**Cárdenas, Grettel Hevia. 2018.** *Aplicación de monitoreo de cámaras en dispositivos móviles para el sistema de Video Vigilancia Digilante.* 2018.

**Cornelio, Lionel R. Baquero Hernández Dayana Mendoza Peña Osviel Rodríguez Valdés**

**Omar Mar. 2016.** <https://publicaciones.uci.cu/index.php/serie/article/view/822>. [En línea] 2016.

**Estapé, Juan Antonio Pascual. 2021.** <https://computerhoy.com/reportajes/tecnologia/mejores-cameras-vigilancia-ip-595981>. [En línea] 2021.

**Frameworks, Qt Development. 2019.** [https://es.wikipedia.org/wiki/Qt\\_Creator](https://es.wikipedia.org/wiki/Qt_Creator). [En línea] 2019.

**Gonzalez, Manuel. 2022.** [uniconverter.wondershare.es](https://uniconverter.wondershare.es). [En línea] 2022.

**Gouvernement. 2018.** <https://www.inspq.qc.ca/es/centro-collaborador-oms-de-quebec-para-la-promocion-de-la-seguridad-y-prevencion-de-traumatismos/definicion-del-concepto-de-seguridad>.  
[En línea] 17 de agosto de 2018.

**Gamma. 1994.** *Patrones de diseño.* 1994.

**Hackathon, Wikimedia. 2022.** <https://es.wikipedia.org/wiki/Streaming>. [En línea] 2022.

**Hintelholher, Rina Marissa Aguilera. 2013.** *Identidad y diferenciación entre Método y Metodología.* 2013.

**HTTPS, Protocolo. 2020.** <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>. [En línea] 2020.

**2019.** <https://www.lucidchart.com/pages/es/diagrama-de-secuencia>. [En línea] 2019.

**Imsel. 2019.** <https://www.imsel.com/que-es-cctv-y-cual-es-su-funcion/>. [En línea] 2019.

**ionos. 2020.** <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/>. [En línea] 2020.

**Lapido, Yoenia Ramírez. 2015.** *HMI para el sistema AREX.* 2015.

**Leyva, Enrique Santos. 2016.** *Módulo de seguridad del sistema de medición y Adquisición Arex.* 2016.

**libbluray. 2022.** [www.videolan.org](https://www.videolan.org). [En línea] 2022.

**Londoño. 2019.** *Pruebas de Aceptacion .* 2019.

**López, Arlet Galguera. 2019.** *Módulo de recolección de variables mediante cámaras para el Sistema de Medición y Adquisición Arex.* 2019.

**lucidchart.** <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>. [En línea]

—. **2018.** <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>. [En línea] 2018.

**Lucidchart.** <https://www.lucidchart.com/pages/es/tutorial-de-diagramas-de-despliegue>. [En línea]

**Maydalis Hernández Pérez, Raimel Sobrino Duque. 2017.** *Módulo de comunicación síncrona para la Plataforma Educativa ZERA 2.0* . 2017.

**Mena, Araujo. 2015.** <http://www.datys.cu/spa/site/product/8>. [En línea] 2015.

**Movistar. 2022.** <https://blog.prosegur.es/camara-ip-o-una-camara-cctv/>. [En línea] 2022.

**Nivian. 2019.** <https://www.nivianhome.com/es/que-es-camara-ip-como-funciona/>. [En línea] 2019.

**Overview, Qt Test.** <https://doc.qt.io/qt-6/qttest-overview.html>. [En línea]

**Parscal, Trevor. 2020.** [https://es.wikipedia.org/wiki/Herramienta\\_CASE](https://es.wikipedia.org/wiki/Herramienta_CASE). [En línea] 2020.

**Pedro. 2019.** <https://www.bambooaudiovisual.com/desventajas-y-ventajas-del-streaming/>. [En línea] 2019.

**pmoinformatica. 2017.** <http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>. [En línea] 2017.

**Protocolos. 2022.**  
[https://es.wikipedia.org/wiki/Protocolo\\_de\\_transmisi%C3%B3n\\_en\\_tiempo\\_real](https://es.wikipedia.org/wiki/Protocolo_de_transmisi%C3%B3n_en_tiempo_real). [En línea] 2022.

**Quijano. 2020.** *Pruebas unitarias.* 2020.

**Rosabal Peña, Yunior Luis. 2017.** *Componente de navegación sobre los flujos de video transmitidos en vivo por las cámaras IP.* 2017.

**Seguí, Pablo Seguí. 2020.** <https://ovacen.com/cameras-de-seguridad/>. [En línea] 2020.

**Simon, Inside. 2021.** <https://www.simonelectric.com/blog/que-tipos-de-sistemas-domoticos-hay-en-la-actualidad>. [En línea] 2021.

**Sommerville. 2006.** *Ingeniería de software.* 2006.

**Stroustrup, Bjarne. 2017.** <https://es.wikipedia.org/wiki/C%2B%2B>. [En línea] 2017.

**Universidades, Santander. 2020.** <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [En línea] 2020.

**Valluerca, Asier Anitua. 2021.** Telefónica Servicios Audiovisuales.  
<https://www.telefonicaserviciosaudiovisuales.com/articulos-de-divulgacion/protocolos-de-streaming/>. [En línea] 2021.

**Velasco, Rubén. 2014.** <https://www.redeszone.net/2014/09/13/ispy-el-software-de-videovigilancia-opensource-mas-completo/>. [En línea] 2014.

**Vergara, Seigio. 2021.** <https://www.atlassian.com/es/agile/project-management/user-stories>. [En línea] 2021.

**Verizon. 2022.** <https://espanol.verizon.com/info/definitions/streaming/>. [En línea] 2022.

**Wik. 2022.** Wikipedia. *Pruebas de integración*. [En línea] 18 de octubre de 2022.

**yeeply.** <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>. [En línea]

## **Anexos**

### **Anexo 1: Entrevistas para la realización del proyecto**

#### **Estimado profesor:**

La tesista Rosalena Hasty Rojas de la Facultad 4 de la carrera de Ingeniería en Ciencias Informáticas realiza la siguiente encuesta con el objetivo de identificar los elementos que estructurarán la información que van a contribuir al cumplimiento del objetivo formativo de nuestro proyecto que se trata sobre una la realización de un Módulo de Streaming de video desde cámaras IP para el Sistema de Medición y Adquisición de Datos AREX. Para ello pedimos su colaboración. Muchas gracias por su dedicación y ayuda.

Escriba según corresponda sus datos.

Nombre y apellidos: Jordanis Viltres Chávez

Centro de trabajo actual: Universidad de las Ciencias Informáticas

Email: jviltres@uci.cu

Teléfono: 53609167

Labor que desempeña: Especialista del centro VERTEX en la línea de trabajo dedicada a la informática industrial.

1. Para conocer sobre el Sistema de Medición y Adquisición de datos AREX responda a las siguientes interrogantes:
  - a) Diga que es el Sistema de Medición y Adquisición de datos AREX.

El Sistema de Medición y Adquisición de datos AREX es un sistema de medición de datos para medir procesos de baja y mediana complejidad, a partir de la adquisición y procesamiento de datos en tiempo real, que puede ser ejecutado en hardware con bajas prestaciones. Para lograr que el sistema aumente la competitividad, es necesario que los despliegues gráficos puedan ser visualizados sobre teléfonos inteligentes o similares, debido al auge que han alcanzado estos dispositivos en el mundo de la tecnología moderna.

- b) Mencione el objetivo fundamental de la realización del módulo para el Sistema AREX.

El objetivo fundamental es incorporarle al sistema un módulo streaming de video desde las cámaras IP para la transmitir y visualizar videos en tiempo real.

- c) Marquen con una X que otros módulos del Sistema AREX se van a relacionar con el módulo a desarrollar:

Recolector

HMI\_Ejecutor

HMI\_Editor

2. Para conocer más sobre el sistema, descríballo:

(Ver descripción en el capítulo 1 epígrafe 1.1)

**Muchas gracias por su atención.**

**Anexo 2: Descripción del requisito Conectarse a las cámaras y recibir streaming de video.**

<b>Descripción del requisito Conectarse a las cámaras y recibir streaming de video.</b>	
Número: 2	<b>Nombre del requisito:</b> Conectarse a las cámaras y recibir streaming de video.
<b>Programador:</b> Rosalena Hasty	<b>Iteración asignada:</b> V1.0
<b>Prioridad:</b> Alta	<b>Tiempo estimado:</b> 1 semana
<b>Riesgo en desarrollo:</b> El programador puede presentar problemas de salud.	<b>Tiempo real:</b> 1 semana
<b>Descripción:</b> Realizar la conexión a las cámaras con el objetivo de obtener el streaming de video.	

Tabla 2.1: HU 2: Conectarse a las cámaras y recibir streaming de video.

**Anexo 3: Descripción del requisito Eliminar conexión a las cámaras.**

<b>Descripción del requisito Eliminar conexión a las cámaras.</b>	
Número: 3	<b>Nombre del requisito:</b> Eliminar conexión a las cámaras
<b>Programador:</b> Rosalena Hasty	<b>Iteración asignada:</b> V1.0
<b>Prioridad:</b> Alta	<b>Tiempo estimado:</b> 5 semana
<b>Riesgo en desarrollo:</b> El programador puede presentar problemas de salud.	<b>Tiempo real:</b> 5 semana
<b>Descripción:</b> El sistema debe eliminar las conexiones a las cámaras por falta de cliente que se conecten a estas.	

Tabla 2.2: HU 3 Eliminar conexión a las cámaras

**Anexo 4: Descripción del requisito Atender las solicitudes del cliente para acceder al streaming.**

<b>Descripción del requisito Atender las solicitudes del cliente para acceder al streaming.</b>	
Número: 4	<b>Nombre del requisito:</b> Atender las solicitudes del cliente para acceder al streaming.
<b>Programador:</b> Rosalena Hasty	<b>Iteración asignada:</b> V1.0
<b>Prioridad:</b> Alta	<b>Tiempo estimado:</b> 2 semana
<b>Riesgo en desarrollo:</b> El programador puede presentar problemas de salud.	<b>Tiempo real:</b> 2 semana
<b>Descripción:</b> Permite al cliente que realice peticiones para obtener el streaming de video y audio.	

Tabla 2.3: HU 4 Atender las solicitudes del cliente para acceder al streaming.

**Anexo 5: Descripción del requisito Cargar el plugins que permita la conexión y ejecución de comandos a las cámaras.**

<b>Descripción del requisito Cargar el plugins que permita la conexión y ejecución de comandos a las cámaras.</b>	
Número: 5	<b>Nombre del requisito:</b> Cargar el plugins que permita la conexión y ejecución de comandos a las cámaras.
<b>Programador:</b> Rosalena Hasty	<b>Iteración asignada:</b> V1.0
<b>Prioridad:</b> Alta	<b>Tiempo estimado:</b> 4 semana
<b>Riesgo en desarrollo:</b> El programador puede presentar problemas de salud.	<b>Tiempo real:</b> 5 semana
<b>Descripción:</b> El sistema debe permitir cargar los archivos de conexión y los comandos para ejecutar las cámaras.	

Tabla 2.4: HU 5 Cargar el plugins que permita la conexión y ejecución de comandos a las cámaras.

**Anexo 6: Escenario 2.1 Realizar la conexión a las cámaras y recibir el streaming**

<b>Escenario 2.1 Realizar la conexión a las cámaras y recibir el streaming</b>	
Descripción	Conectarse a las cámaras y recibir el streaming de video
Respuesta del sistema	Se realizó la conexión a las cámaras y se recibió el streaming de video
Flujo central	El módulo se conecta a la cámara y recibe el flujo de video mediante un streaming.
<b>Escenario 2.2 No se realiza la conexión a las cámaras y no se recibe el streaming</b>	
Descripción	Conectarse a las cámaras y recibir el streaming de video
Respuesta del sistema	No se puede conectar con la cámara, por tanto, no recibe el streaming de video.
Flujo central	El módulo no se conecta con la cámara, por tanto, no recibe el streaming de video

Tabla 3.3: Prueba de caja negra: Realizar la conexión a las cámaras y recibir el streaming

**Anexo 7: Escenario 3.1 Atender las solicitudes del cliente para acceder al streaming**

<b>Escenario 3.1 Atender las solicitudes del cliente para acceder al streaming</b>	
Descripción	Atender solicitudes de clientes para acceder al streaming.



Respuesta del sistema	Se atiende la solicitud del cliente para acceder al streaming de video
Flujo central	El cliente accede al módulo escribiendo la siguiente dirección en un navegador web: <code>http://applp:commandsPort/1?stream</code> , donde <code>applp</code> sería el ip donde se ejecuta el módulo, <code>commandsPort</code> el puerto de la cámara, el <code>1</code> el índice de la cámara y <code>stream</code> , el comando para acceder al streaming. Mediante esa petición el cliente accede al streaming.
<b>Escenario 3.2 La petición web está mal escrita</b>	
Descripción	La petición web está mal escrita
Respuesta del sistema	El sistema no devuelve las solicitudes de los clientes para la cámara 1.
Flujo central	El cliente accede al módulo escribiendo la siguiente dirección en un navegador web: <code>http://applp:commandsPort/1?</code> , no se escribe el comando <code>stream</code> por tanto no se pueden acceder a las variables. Esto es solo un ejemplo de una url mal escrita.

Tabla 3.4: Prueba de caja negra: Atender las solicitudes del cliente para acceder al streaming.

#### **Anexo 8: Escenario 4.1 Eliminar conexión a las cámaras**

<b>Escenario 4.1 Eliminar conexión a las cámaras</b>	
Descripción	Eliminar la conexión de las cámaras cuando no haya ningún cliente conectado al sistema
Respuesta del sistema	Se elimina la conexión de las cámaras
Flujo central	El módulo elimina la conexión a de las cámaras cuando el cliente se desconecta del módulo.
<b>Escenario 4.2 No eliminar la conexión a las cámaras</b>	
Descripción	Eliminar la conexión de las cámaras cuando no haya ningún cliente conectado al sistema
Respuesta del sistema	Se elimina la conexión de las cámaras
Flujo central	El módulo no elimina la conexión de las cámaras, por lo tanto, las cámaras siguen conectadas y retransmitiendo streaming de video sin que el cliente esté conectado al módulo.

Tabla 3.5: Prueba de caja negra: Eliminar conexión a las cámaras

#### **Anexo 9: Escenario 7.1 Cargar el plugins que permita la conexión y ejecución de los comandos a las cámaras.**

<b>Escenario 7.1 Cargar el plugins que permita la conexión y ejecución de los comandos a las cámaras.</b>	
Descripción	Cargar el plugins que permita la conexión y ejecución de los comandos a las cámaras
Respuesta del sistema	Se cargo el plugins que permite la conexión y ejecución de los comandos a las cámaras.
Flujo central	El módulo permitió cargar el plugins que permite la conexión y ejecución de los comandos a las cámaras.
<b>Escenario 7.2 no carga el plugins de configuración y ejecución de los comandos a las cámaras</b>	
Descripción	Cargar el plugins que permita la conexión y ejecución de los comandos a las cámaras
Respuesta del sistema	Se cargo el plugins que permite la conexión y ejecución de los comandos a las cámaras.
Flujo central	El módulo no permitió cargar el plugins que permite la conexión y ejecución de los comandos a las cámaras.

Tabla 3.6: Prueba de caja negra: Cargar el plugins que permita la conexión y ejecución de los comandos a las cámaras.