

**Universidad de las Ciencias Informáticas**

**Empresa para la Tecnología de la Información para la Defensa (XETID)**

**SCADA**



**APLICACIÓN DE GESTIÓN Y VISUALIZACIÓN DE CÁMARAS IP  
PARA LA PLATAFORMA ANDROID**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

**Autor:** Alejandro Avelino Infante de la Torre.

**Tutores:** Ing. Yadiel Nápoles Rosales

Ing. Yunieski Orlando Vasconcelo Mir

La Habana, 2020

“Año 62 del triunfo de la Revolución”



## DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Alejandro Avelino Infante de la Torre, soy el autor principal del trabajo final de tesis de pregrado que se titula: "APLICACIÓN DE GESTIÓN Y VISUALIZACIÓN DE CÁMARAS IP PARA LA PLATAFORMA ANDROID". El cuál ha sido desarrollado como parte del trabajo del Centro de Tecnologías para la Formación. Autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

Y para que así conste, se firma la presente declaración jurada de autoría en La Habana, a los \_\_14\_\_ días del mes de \_\_septiembre\_\_ del año 2020.

---

Alejandro Avelino Infante de la Torre.

---

Ing. Yadiel Nápoles Rosales

---

Ing. Yunieski Orlando Vasconcelo Mir



## AGRADECIMIENTOS

- *A mi familia que siempre me dio su apoyo incondicional*
- *A mis compañeros de apartamento y edificio que siempre estuvieron ahí para apoyarme y salvarme la vida.*
- *A Meylan Sánchez por todo, ya que no se puede resumir en un documento lo q ha hecho por mí.*
- *A mis profesores.*
- *A mi tribunal y oponente.*
- *A mis tutores Yadriel Nápoles Rosales y Yunieski Orlando Vasconcelo Mir que de una forma u otra me ayudaron.*
- *A la profesora principal del año Disnayle Jorge Chacón que incluso en tiempos muy difíciles me presto su ayuda.*
- *Y a la UCI en general por haberme dado la oportunidad de estudiar en tan maravilloso centro educacional.*

## RESUMEN

Los sistemas de videovigilancia tienen cada vez mayor demanda en la actualidad. Especialmente para garantizar la seguridad de interiores y alrededores de las empresas e instituciones. La evolución tecnológica ha facilitado la obtención e instalación de videocámaras sin requerir enormes inversiones económicas. Actualmente para la visualización y gestión de las numerosas cámaras de seguridad se utilizan sistemas instalados en plataformas de escritorio tanto en software libre como privativo: Sistemas basados en Linux y todos sus derivados, Windows o Mac; esto trae como consecuencia que se requiera la monitorización por parte del personal de forma constante de dicho sistema. Esto sería un problema en instalaciones muy complejas o extensas ya que requeriría mucho más personal tanto para la monitorización de las pantallas como para los recorridos. Por tanto, es necesario crear una aplicación que permita la movilidad para la solución de esta deficiencia.

El presente trabajo tiene como objetivo implementar una aplicación de visualización y gestión de cámaras IP. Para su desarrollo se propone la utilización de librerías gratuitas compatibles con los estándares elegidos, todo para la tecnología móvil específicamente la plataforma Android, la cual es software libre.

Como resultado se pretende crear una herramienta que permita mejorar la seguridad en las instituciones del país a las que se les preste el servicio. Esto facilitaría en gran medida el control y vigilancia de las entidades cliente.

**Palabras clave:** Android, cámara de seguridad, móvil, seguridad, vigilancia.

# ÍNDICE

INTRODUCCIÓN.....	1
1. CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA ASOCIADA AL DOMINIO DEL PROBLEMA DE LA GESTIÓN Y VISUALIZACIÓN DE CÁMARAS IP PARA LA PLATAFORMA ANDROID.....	5
1.1 Introducción .....	5
1.3 Soluciones informáticas relacionadas con el objeto de estudio de la investigación.....	7
1.4 Propuesta de solución .....	11
1.5 Ambiente de desarrollo.....	11
1.6 Conclusiones del capítulo .....	17
2 CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA APLICACIÓN PARA LA VISUALIZACIÓN Y GESTIÓN DE CÁMARAS IP PARA LA PLATAFORMA ANDROID (AVIGESCIP):.....	18
2.1 Introducción .....	18
2.2 Modelo de propuesta de solución.....	18
2.3 Planeación.....	20
2.4 Diseño de la propuesta de solución .....	6
2.5 Conclusiones del capítulo .....	11

CONCLUSIONES GENERALES .....	12
RECOMENDACIONES.....	13
REFERENCIAS BIBLIOGRÁFICAS .....	14
ANEXOS.....	16

## Introducción

La Videovigilancia IP es una tecnología visual que combina los beneficios analógicos de los tradicionales CCTV (*Circuito Cerrado de Televisión*) con las ventajas digitales de las redes de comunicación IP (*Internet Protocol*) (Allmer, Thomas, 2012), permitiendo la supervisión local y/o remota de audio e imágenes, así como el tratamiento digital de las mismas, para aplicaciones como el reconocimiento de matrículas, reconocimiento facial, entre otras.

El despliegue resulta más sencillo y económico que un CCTV, puesto que aprovecha la red informática empresarial, es decir, el mismo cableado que se emplea para la comunicación de datos, acceso a Internet o correo electrónico, sin necesidad de desplegar una infraestructura de cableado coaxial específica para nuestra red de videovigilancia. La mayoría de las instalaciones más modernas están abandonando la tecnología analógica en favor de la videovigilancia IP, dada su versatilidad, funcionalidad, sencillez y optimización de las infraestructuras existentes en las compañías.

En 1996 la compañía *Axis Communications* publica la primera cámara que utiliza el protocolo de internet (IP), posibilitando el nacimiento de lo que vendría a ser la videovigilancia hoy en día. Desde el inicio la videovigilancia no ha tenido como meta la sustitución de las personas sino la de asistirles y mejorar la efectividad del servicio. Esta se ha convertido en una necesidad para todas las entidades; ha sido influenciada por el crecimiento de las tecnologías tanto ópticas como de infraestructura de red, de almacenamiento, y mayormente la móvil ya que es la que más aceptación tiene mundialmente debido a lo intuitiva y amigable que es la interfaz de las aplicaciones para esa plataforma. Uniendo las ventajas de estas se garantiza así la calidad de imagen necesaria, el medio para el envío de esta y la velocidad para posibilitar dicho envío, la capacidad del almacenamiento necesario para salvar esa información (Petersen, J.K., 2012).

En Cuba se han realizado grandes inversiones en nuevas tecnologías para instituciones tales como hospitales, bancos, empresas, universidades, entre otras. Un ejemplo de lo anteriormente mencionado es la Universidad de las Ciencias Informáticas, la cual es una institución en la que se encuentran diversos equipos como computadoras, aires acondicionados, refrigeradores, televisores, ventiladores en grandes

cantidades para su uso diario, además de los almacenes en los que se puede encontrar más de estos equipos y piezas de repuesto para estos medios, lo cual potencia los intentos de actos delictivos.

En dicha universidad radica el Centro SCADA perteneciente a la Empresa de Tecnologías para la Defensa (XETID), que tiene implementado un Sistema de videovigilancia para escritorio; este sistema funciona correctamente, con la desventaja de ser estático, ya que es necesario la visualización de los monitores constantemente para así poder cumplir con su función correctamente; en cuanto a la economización de los recursos, la tecnología de escritorio es más cara que la móvil. Otra deficiencia que tiene este tipo de videovigilancia es que el operador humano tiene que estar todo el tiempo en el puesto sin moverse del mismo ya que tiene que estar analizando constantemente la secuencia de video. Este tipo de vigilancia restringe que la persona se mueva personalmente por las diferentes áreas de la instalación. Estos sistemas también presentan una interfaz agresiva ya que los usuarios estándar requieren de una capacitación especializada para la utilización del software. Otra desventaja de este sistema de escritorio es que si falla algún hardware implicado en la visualización o conexión a la red local ocasionaría una brecha en la seguridad de las áreas protegidas por este servicio.

Por los elementos anteriormente expuestos se plantea como **problema de investigación**: ¿Cómo desarrollar una aplicación que permita la visualización y gestión de las cámaras IP desde la plataforma Android?

El **objeto de estudio** está enmarcado en el servicio de videovigilancia, delimitándose como **campo de acción** las aplicaciones de videovigilancia, móviles.

Para dar respuesta al problema antes mencionado se establece como **objetivo general**: Desarrollar una aplicación para gestionar y visualizar cámaras IP desde la plataforma Android.

Por el objetivo previamente planteado se definen como **objetivos específicos** los siguientes:

1. Realizar una investigación enfocada a los sistemas de videovigilancia que utilicen la plataforma Android.
2. Describir la propuesta de la aplicación para gestionar y visualizar cámaras IP desde la plataforma Android.



3. Implementar la aplicación para gestionar y visualizar cámaras IP desde la plataforma Android.

4. Probar la aplicación para gestionar y visualizar cámaras IP desde la plataforma Android.

La investigación se sustenta en la siguiente **hipótesis**: El desarrollo de una aplicación que permita la visualización y gestión de cámaras IP desde la plataforma Android en las instituciones que lo requieran, contribuirá a mejorar la seguridad de las áreas de dichas instituciones.

A continuación, se explican los **métodos científicos** que se utilizan en la investigación.

### **Métodos teóricos**

**Histórico – Lógico**: se utilizó para satisfacer la necesidad de realizar un estudio acerca de las soluciones informáticas relacionadas con el objeto de estudio, así como las tecnologías y estándares utilizados para su desarrollo.

**Modelación**: se utilizó con el fin de modelar los procesos de negocios, así como crear los diagramas ingenieriles de la propuesta de solución.

**Analítico – Sintético**: empleado para estudiar y analizar documentos y bibliografías de diferentes autores para poder realizar una amplia investigación sobre los elementos que se relacionan con el objeto de estudio.

### **Métodos empíricos**

**Entrevista**: se utilizó para obtener información de cómo se lleva a cabo el proceso de videovigilancia en la XETID. Además de conocer el estado de opinión del operador humano a cargo de dicho servicio, pudiendo determinar cuáles son las principales deficiencias del proceso.

**Observación**: el método permite un mayor acercamiento y familiarización con la organización a la cual se le desea informatizar los procesos que realizan. Se utilizó para observar cómo se desarrolla el proceso de videovigilancia en la XETID.

## Estructura del documento

**Capítulo 1. Fundamentación teórica asociada de la aplicación para la visualización y gestión de cámaras IP para la plataforma Android (AvigesCIP):** En este capítulo se dan a conocer los elementos que son importantes para el proceso de vigilancia y gestión de cámaras desde una plataforma Android. También se realiza un estudio referente a otras soluciones infáticas que responden a este proceso, así como el marco de trabajo, tecnologías y herramientas a utilizar para el desarrollo de la aplicación.

**Capítulo 2. Análisis y diseño de la aplicación para la visualización y gestión de cámaras IP para la plataforma Android:** En este capítulo se explican los principales aspectos de diseño de la aplicación de gestión y visualización de cámaras IP.

# 1. Capítulo 1. Fundamentación teórica asociada al dominio del problema de la gestión y visualización de cámaras IP para la plataforma Android.

## 1.1 Introducción

En este capítulo se dan a conocer los elementos que son objeto de estudio en el proceso de gestión y visualización de cámaras IP para la plataforma Android. También se evidencia un estudio del estado referente a las aplicaciones existentes que responden a este proceso, así como el marco de trabajo, tecnologías y herramientas a utilizar para el desarrollo de la aplicación.

## 1.2 Definiciones asociadas.

Con el objetivo de lograr un mejor entendimiento del tema de investigación, se mencionan varios de los principales conceptos relacionados con el tema.

**Videovigilancia:** es una tecnología de vigilancia visual que combina los beneficios analógicos de los tradicionales CCTV (Circuito Cerrado de Televisión) con las ventajas digitales de las redes de comunicación IP (Internet Protocol), permitiendo la supervisión local y/o remota de imágenes y audio, así como el tratamiento digital de las imágenes, para aplicaciones como el reconocimiento de matrículas o reconocimiento facial, entre otras. (Francisco Javier García Mata, 2010).

**IP:** El Protocolo de Internet (en inglés Internet Protocol o IP) es un protocolo de comunicación de datos digitales clasificado funcionalmente en la capa de red según el modelo internacional OSI. (Charles M. Kozierok, 2005)

**Android:** Android es un sistema operativo móvil basado en una versión modificada del kernel de Linux y otro software de código abierto, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes y tabletas. Android es desarrollado por un consorcio de desarrolladores conocido

como Open Handset Alliance, con el principal contribuyente y comercializador comercial Google. (Bretel King, 2016).

**Autenticación:** es el acto de probar una afirmación, como la identidad de un usuario de un sistema informático. En contraste con la identificación, el acto de indicar la identidad de una persona o cosa, la autenticación es el proceso de verificar esa identidad. Puede implicar validar documentos de identidad personal, verificar la autenticidad de un sitio web con un certificado digital, determinar la edad de un artefacto por datación por carbono, o asegurarse de que un producto o documento no sea falsificado (Turner, Dawn M, 2016).

**Play Store:** Google Play Store es un servicio de tienda virtual que permite distribuir aplicaciones (apps) para funcionar en el sistema Android y Chrome OS. Este servicio es desarrollado y mantenido por la empresa Google, y es usado principalmente en dispositivos móviles como teléfonos inteligentes. (Leandro Alegsa, 2016).

**Cámara PTZ:** El termino PTZ se refiere a un acrónimo de pan-tilt-zoom. Las cámaras PTZ pueden rotar alrededor de dos ejes, uno horizontal y otro vertical, así como acercarse o alejarse (zoom) para enfocar un área u objeto de forma manual o automática. Dicho de otra forma, este tipo de cámaras es capaz de rotar en un plano vertical (tilt en inglés) y en un plano horizontal (panning), además de acercarse o alejarse de forma manual o automática (Del Bimbo y otros, 2009).

**ACID:** En bases de datos se denomina ACID a las características de los parámetros que permiten clasificar las transacciones de los sistemas de gestión de bases de datos. Cuando se dice que es ACID compliant se indica -en diversos grados- que éste permite realizar transacciones. En concreto ACID es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español (ISO/IEC 10026-1: 1992 sección 4).

**API:** Una API representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no

necesariamente) entre los niveles o capas inferiores y los superiores del software. Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas del API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio (Fisher, Sharon, 1989).

### **1.3 Soluciones informáticas relacionadas con el objeto de estudio de la investigación.**

A continuación, se realiza un estudio de soluciones informáticas existentes que visualizan y gestionan cámaras IP desde la plataforma Android.

#### **1.3.1 Soluciones existentes en el ámbito internacional.**

##### **RTSP Camera Server**

MIV Dev es un grupo de desarrolladores independientes que se enfocan en aplicaciones de videovigilancia para la plataforma Android.

Esta aplicación es capaz de servir vista de la cámara en vivo o grabado a los clientes RTSP. Además, permite:

- La grabación de vídeo -loop
- El método de captura -Quick
- El Trabajo en segundo plano como un servicio
- Soporte De las diferentes resoluciones de la cámara y FPS
- Se Puede elegir entre la cámara hacia atrás y la cámara frente a frente
- Encoders H264 y el CAC (Video Surveillance.com, 2019)

No es recomendable esta aplicación debido a que tiene malas reseñas en Play Store, lo que demuestra cierto patrón de inconformidad de los usuarios hacia este software.

### **OpticaMobile**

Optica es reconocida como un fabricante líder de video vigilancia IP de alta definición y cámaras PTZ. Comparables con otras marcas líderes como Axis y Vivotek, todas las cámaras IP Optica vienen con un conjunto básico de características integradas como resolución Full HD, amplio rango dinámico, almacenamiento integrado MicroSD, detección de manipulaciones, modo de vista vertical y gestión de video OpticaCMS software. Optica ofrece una amplia selección de estilos de cámaras IP, incluidos domos, balas y modelos PTZ para satisfacer las diversas necesidades de organizaciones más grandes y profesionales de la seguridad.

OpticaMobile es una aplicación de monitoreo móvil diseñada para dispositivos iPhone y Android. Esta aplicación de monitoreo le permiten controlar su sistema de cámara IP directamente desde su teléfono o tableta con funciones como el modo de visualización cuádruple, transmisiones de video en vivo, marcadores, audio, instantáneas, soporte de transmisión dual y control de panorámica / inclinación / zoom de Optica PTZ cámaras domo de alta velocidad (Video Surveillance.com, 2019). Sin embargo, es una aplicación de pago y privativa, razón que no permite modificaciones en el código para satisfacer las necesidades de otras organizaciones.

### **Milestone XProtect Mobile**

Milestone es reconocido en todo el mundo por su software de gestión de video de última generación y sus herramientas de inteligencia de video que maximizan el rendimiento del video IP. XProtect, la principal plataforma VMS de Milestone, se puede combinar con numerosas marcas de cámaras IP y NVR para una implementación universal. Hay una gran cantidad de complementos VMS que aseguran que tenga la mejor vista de todas sus cámaras y, por lo tanto, puede proteger su propiedad mejor que nunca. Completamente escalable, XProtect VMS es perfecto para empresas en crecimiento que es probable que compren cámaras

de videovigilancia IP adicionales en el futuro. El análisis de video es bastante impresionante y se encuentra entre las tecnologías más pioneras que ofrecen los proveedores de VMS en la actualidad.

Milestone XProtect Mobile tiene tres aplicaciones para teléfonos inteligentes disponibles que son compatibles con iPhone, Android y iPad. A través de estas aplicaciones de monitoreo móvil, descubrirá lo fácil que es vigilar de cerca sus cámaras, reproducir videos grabados, enviar instantáneas por correo electrónico y MMS, y buscar videos archivados. Las posibilidades son infinitas y atienden a los usuarios que viven vidas ocupadas y no siempre tienen el tiempo para estar en la ubicación de sus cámaras IP (Video Surveillance.com, 2019). Esta aplicación presenta una puntuación de 3.5 puntos en Play Store, entre los comentarios se puede apreciar que presenta inestabilidad para varios dispositivos, lo cual elimina la posibilidad de ser utilizada por un sistema de Videovigilancia de forma empresarial.

### **Smartvue**

Smartvue es el líder de la industria en videovigilancia remota y móvil, así como en la configuración de la nube plug-and-play. Cuando compra uno de los NVR plug-and-play de Smartvue, obtiene acceso gratuito a su servicio patentado Cloudvue y a la aplicación Smartvue diseñada para usar con iPad, iPhone y Android. Smartvue es conocido por admitir una larga lista de marcas y modelos líderes de cámaras IP. Cuando se conecta a Smartvue desde una ubicación remota, puede administrar, grabar y ver su sistema completo de cámara IP.

La aplicación Smartvue para Android permite acceder a videovigilancia remota. Apreciará lo fácil que es deslizarse por las distintas pantallas de visualización de sus cámaras IP y ver el video grabado a través de una función de búsqueda básica. Smartvue se compromete a garantizar que pueda controlar su propiedad en cualquier momento, de día o de noche, ya sea que esté en un teléfono inteligente, una PC o una Mac (Video Surveillance.com, 2019). Esta aplicación es de código cerrado o privativa impidiendo que se modifique su código para así adaptarlo a alguna condición específica, por tanto, no debe ser elegible para la resolución del objetivo general.

### **Vivotek iViewer**

Vivotek se compromete a proporcionar servicios de monitoreo remoto a sus usuarios. Cuando compra un sistema de cámara IP Vivotek, puede acceder a video en tiempo real desde cualquier parte del mundo con acceso a Internet. También tenga en cuenta que puede autorizar a varios usuarios a acceder a la misma cámara IP simultáneamente para que más de una persona pueda vigilar su negocio o propiedad. Su plataforma de monitoreo remoto le permite ver, controlar y administrar todas sus cámaras IP a través de varios navegadores web. Es conveniente, simple e ideal para usuarios de vigilancia activa.

Vivotek ofrece la aplicación de cámara iViewer para iPhone y Android diseñada para brindarle acceso completo a su sistema de cámara IP mientras viaja. Es imposible estar en el sitio todo el tiempo; ahora puede controlar sus cámaras según sea necesario mientras está físicamente alejado de sus cámaras de seguridad. Con Vivotek iViewer puede ver la alimentación de su cámara en vivo, elegir entre numerosas opciones de visualización, desplazarse por las imágenes grabadas, controlar las funciones PTZ, reproducir videos grabados y tomar instantáneas de imágenes. Se puede ver en vivo en la palma de tu mano (Surveillance, 2019). Esta aplicación es de código cerrado o privativa impidiendo que se modifique su código para así adaptarlo a alguna condición específica, por tanto, no debe ser elegible para la resolución del objetivo general.

### **MobileFocus**

El "MobileFocus de EverFocus" le permite conectarse a sus DVR y cámaras IP EverFocus desde su teléfono inteligente. Puede ver transmisiones de video y controlar las cámaras PTZ fácilmente. Sus características serían: gestión de dispositivos, vista en vivo, control de canales, control PTZ, control PTZ por gestos, zoom, instantánea, pantalla completa, transmisión móvil y notificación de inserción (Video Surveillance.com, 2019). Esta aplicación tiene una puntuación de 3.1 en Play Store, principalmente presenta problemas con la inestabilidad en la conexión a las cámaras IP; por tanto, no podría ser seleccionada como solución al objetivo general planteado.



## 1.4 Propuesta de solución

Teniendo en cuenta estudio realizado sobre algunas de las soluciones informáticas a nivel internacional, se determinó que utilizar una de estas no es lo más recomendable, lo que evidencia la necesidad de desarrollar una aplicación de videovigilancia que permita la visualización y gestión de cámaras IP desde la plataforma Android. Sin embargo, fue posible identificar un conjunto de funcionalidades a tener en cuenta en el desarrollo de la nueva propuesta de solución.

El objetivo de esta aplicación no es sustituir el sistema de videovigilancia que se encuentra en funcionamiento, sino, servir como un sistema de respaldo y se eliminaría la desventaja de ser un sistema estático, ya que la aplicación propuesta permitiría las mismas funciones de forma móvil. La ventaja que presenta es que los guardias de seguridad podrían monitorizar las cámaras IP desde cualquier lugar dentro de la instalación. En caso de afectarse la conexión ethernet del ordenador, en el que este instalado el sistema de videovigilancia de escritorio, no afectaría ninguna de las funcionalidades de la aplicación a desarrollar.

## 1.5 Ambiente de desarrollo

A continuación, se describen las herramientas y tecnologías a utilizar en el desarrollo de la propuesta de solución, siguiendo la base tecnología definida por la XETID para el desarrollo de las aplicaciones de videovigilancia.

### 1.5.1 Lenguaje de programación

#### Java

Potente lenguaje orientado a objetos y multiplataforma. La Plataforma Java se compone de un amplio abanico de tecnologías, cada una de las cuales ofrece una parte del complejo de desarrollo o del entorno de ejecución en tiempo real. Por ejemplo, los usuarios finales suelen interactuar con la máquina virtual de Java y el conjunto estándar de bibliotecas. Además, las aplicaciones Java pueden usarse de forma variada,

como por ejemplo ser incrustadas en una página web. Para el desarrollo de aplicaciones, se utiliza un conjunto de herramientas conocidas como JDK (Java Development Kit, o herramientas de desarrollo para Java). Este presenta una amplia documentación facilitando el trabajo con él. Java tiene una facilidad de cumplir varias funciones al mismo tiempo, gracias a su función de multihilos ya que por cada hilo que el programa tenga se ejecutaran en tiempo real muchas funciones al mismo tiempo. Java es considerado de alto rendimiento por ser tan veloz en el momento de correr los programas y por ahorrarse muchas líneas de código (Binstock, Andrew, 2015).

### **1.5.2 Marco de trabajo para Java**

Puede decirse que un marco de trabajo (framework) es el esqueleto de una aplicación que debe ser adaptado por el programador para desarrollar una aplicación específica.

El framework oficial de trabajo con Android es Android Studio. Esta publicado de forma gratuita y es fácil de acceder a él. Este es un marco de trabajo que está basado en el patrón Cliente-Servidor, ya que la aplicación Android como cliente pediría toda información necesaria del servidor.

### **1.5.3 Metodología de desarrollo**

Para lograr una mayor organización en el proceso de desarrollo de software se seleccionan las metodologías de desarrollo, las cuales constituyen una filosofía de trabajo que proporciona una base de procesos para llevar a cabo con éxito cualquier proyecto informático. Las metodologías de desarrollo brindan soporte a la toma de decisiones en un equipo de trabajo.

eXtreme Programing (XP) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Para el

desarrollo de la propuesta de solución se escoge XP como metodología que guiará dicho proceso, teniendo en cuenta los aspectos que se listan a continuación:

- El proyecto es de corta duración y todo el trabajo de codificación es llevado a cabo por un programador
- No es necesaria la generación excesiva de artefactos y roles ya que el proyecto es pequeño y está diseñado para ser realizado en el menor tiempo posible.
- Los requisitos suelen cambiar con frecuencia en la medida en que avanza el desarrollo del proyecto, así el cliente puede ir añadiendo Historias de Usuario, dividir las para agilizar el trabajo o eliminarlas simplemente. XP permite al equipo de trabajo modificar los planes de desarrollo conforme a lo anterior.
- Tanto cliente como desarrolladores forman parte del equipo de trabajo de forma tal que se logra una interacción constante entre ellos. Esto posibilita la retroalimentación, corrección de errores y finalmente la realización de un producto que cumpla las expectativas de las partes involucradas en el proceso.

#### **1.5.4 Android SDK Tools 23.0.5**

El SDK16 de Android provee las bibliotecas API y las herramientas de desarrollo necesarias para compilar, probar y depurar aplicaciones Android. Se utiliza esta versión del SDK debido a que es la más estable disponible en los repositorios de herramientas de software de la UCI en el momento en que se inicia la presente investigación.

#### **Android API 21**

Cada aplicación Android se desarrolla en base a una versión específica del framework<sup>17</sup> API que ofrece el SDK, en pos de mantener la compatibilidad de la misma con versiones anteriores del sistema operativo. El nivel API es un valor entero que identifica de forma exclusiva la revisión del framework API que ofrece una versión de la plataforma Android. ([www.google.com/apis](http://www.google.com/apis)) Las aplicaciones pueden utilizar el framework API para interactuar con el sistema Android subyacente. Cada revisión del API consiste en:

1. Un conjunto básico de paquetes y clases.
2. Un conjunto de elementos y atributos XML<sup>18</sup> para declarar un archivo de manifiesto.
3. Un conjunto de elementos y atributos XML para declarar el acceso a los recursos.
4. Un conjunto de Intents<sup>19</sup> para gestionar las acciones.
5. Un conjunto de permisos que las aplicaciones pueden solicitar, así como refuerzos de permisos incluidos en el sistema.

Cada versión sucesiva de la plataforma Android puede incluir actualizaciones del framework de aplicaciones que ofrece. La aplicación Android propuesta en el presente trabajo se implementará utilizando el framework API 28, lo cual implica que la compatibilidad de la misma se garantiza hasta la versión 9 de Android, Pie. Dicha versión es la más estable soportada en la revisión del SDK que se utilizará en el desarrollo de la aplicación.

### **1.5.5 Visual Paradigm para UML 8.0**

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computador) que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

Permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta, controla que el modelado con UML sea correcto, al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades permite integrarse con otras aplicaciones, como herramientas ofimáticas, lo cual aumenta la productividad, permite el trabajo en grupo, proporcionando herramientas de compartición de trabajo, permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software y posibilita generar diversos informes a partir de la información introducida en la herramienta (Ruiz, 2013). Se propone la utilización de esta herramienta en su versión 8.0.

### **1.5.6 Entorno de desarrollo integrado (IDE).**

El uso de los entornos de desarrollo integrados se ratifica y afianza en los 90 y hoy en día se cuenta con infinidad de Entornos de Desarrollo Integrados (IDE), tanto de licencia libre como privada (Comesaña, 2012).

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android (Android Developers, 2019).

Debido a estas razones se decide usar el Android Studio 3.5 como IDE de programación para el desarrollo de la aplicación.

### **1.5.7 Real Time Streaming Protocol (RTSP)**

El protocolo de transmisión en tiempo real (RTSP) es un protocolo de control de red diseñado para su uso en sistemas de entretenimiento y comunicaciones para controlar servidores de transmisión de medios. El protocolo se utiliza para establecer y controlar sesiones de medios entre puntos finales. Los clientes de los servidores de medios emiten comandos de estilo VHS, como reproducir, grabar y pausar, para facilitar el control en tiempo real de la transmisión de medios desde el servidor a un cliente (Video On Demand) o desde un cliente al servidor (Grabación de voz) (H. Schulzrinne, 1998). En el proyecto toda comunicación con las cámaras IP se realizará mediante este protocolo.

### **1.5.8 Native Development Kit (NDK)**

Un kit de desarrollo de software (SDK) es una colección de herramientas de desarrollo de software en un paquete instalable. Facilitan la creación de aplicaciones al tener compilador, depurador y quizás un marco

de software. Normalmente son específicos de una combinación de plataforma de hardware y sistema operativo. Para crear aplicaciones con funcionalidades avanzadas como anuncios, notificaciones push, etc. La mayoría de los desarrolladores de software de aplicaciones utilizan kits de desarrollo de software específicos (Shamsee y otros, 2015). Se utilizará para cargar las librerías programadas en C++, como los códec de video necesarios para reproducir la transmisión enviada por el protocolo RTSP.

### 1.5.9 Gestor de base de datos

Un Sistema Gestor de Base de Datos (SGBD) es un software de propósito general que facilita los procesos de definición, construcción y manipulación de la base de datos para distintas aplicaciones (Garzón, 2010). Estos sistemas facilitan el control sobre los datos almacenados y posibilitan garantizar la integridad y confiabilidad de la información a través de las políticas de seguridad que implementan.

Actualmente existen muchos SGBD, algunos más robustos y confiables que otros, pero todos inclinados a la manipulación sencilla de la información residente en una base de datos.

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kiB)<sup>2</sup> biblioteca escrita en C. SQLite es un proyecto de dominio público<sup>1</sup> creado por D. Richard Hipp.

A diferencia del sistema de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un solo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción (sqlite, 2019).

## 1.6 Conclusiones del capítulo

En este capítulo se trataron los elementos teóricos que dan sustento a la propuesta de solución del problema planteado, arribando a las siguientes conclusiones:

- ✓ El estudio de los conceptos asociados a la problemática planteada permitió lograr un mejor entendimiento de la investigación que se realiza.
- ✓ El análisis de las funcionalidades que brindan algunos de los sistemas que permiten la videovigilancia desde la plataforma Android, permitió identificar requerimientos a tener en cuenta en la implementación.
- ✓ La selección de las tecnologías, metodología y herramientas con soporte multiplataforma y basadas en software libre, permitió obtener una base tecnológica enfocada en las políticas de desarrollo definidas en la XETID.

## **2 Capítulo 2. Análisis y diseño de la aplicación para la visualización y gestión de cámaras IP para la plataforma Android (AvigesCIP):**

### **2.1 Introducción**

En este capítulo se explican los principales aspectos de diseño de la aplicación de gestión y visualización de cámaras IP. Su contenido incluye la descripción del modelo de propuesta de solución, así como las fases de planeación y diseño correspondientes a la metodología de desarrollo de software seleccionada.

### **2.2 Modelo de propuesta de solución.**

La solución propuesta está diseñada para aquellas entidades que utilicen un CCTV de forma estática, dispositivos móviles que operan con sistema operativo Android, las cámaras IP para la transmisión de datos, y que posea a su vez una infraestructura adecuada. Su objetivo es permitir la gestión y visualización de cámaras IP junto con el audio, así como el control de movimiento de las cámaras PTZ, gestión de las vistas, la filtración las cámaras listadas, la autenticación usuario contraseña para el servidor, y la posibilidad de iniciar y detener una grabación manual del flujo de video. La solución debe permitir la visualización del flujo de video de las cámaras IP conectadas en el CCTV, así como del cumplimiento de las políticas de seguridad informática establecidas.

AvigesCIP primeramente solicita la autenticación antes de continuar mediante el protocolo HTTPS, luego será capaz de mostrar el flujo de video y el audio de cada cámara IP configurada y conectada en el CCTV de la entidad por medio del protocolo RTSP, además de la gestión de las características de cada cámara IP especificando si es PTZ o no, por ejemplo, de estas últimas va a permitir el control de las mismas y el establecimiento de presets para el rápido posicionamiento de las cámaras.

#### **2.2.1 Autenticación**

Android incluye un componente llamado Authenticator, para manejar las cuentas asociadas cada aplicación, el cual es vital para el esquema de sincronización. El framework de sincronización asume que la



transferencia ocurre entre datos almacenados en el dispositivo y datos almacenados en un servidor remoto que requiere autenticación, por esa razón requiere que se incluya el componente Authenticator como parte de la estructura del Sync Adapter. Entre los beneficios asociados su implementación se encuentra:

- Simplificación de la autenticación para el desarrollador
- Forma estándar de autenticación de usuarios
- Seguridad en el almacenamiento y uso de credenciales
- Manejo de casos de acceso denegado

En la solución propuesta se implementará este componente para gestionar la creación de la cuenta local asociada a la aplicación y el manejo seguro de las credenciales de autenticación de un usuario del sistema GRHS. Estas credenciales se utilizarán para la obtención del token de seguridad necesario para la realización de solicitudes al servidor.

### 2.2.2 Consulta y manipulación de datos

La consulta y manipulación de los datos almacenados en la base de datos local de AvigesCIP se realizará mediante un componente llamado Content Provider, el cual, gestiona el acceso a un repositorio central de datos y ofrece una interfaz estándar y consistente para los mismos. Este componente maneja la comunicación entre procesos y el acceso seguro a los datos.

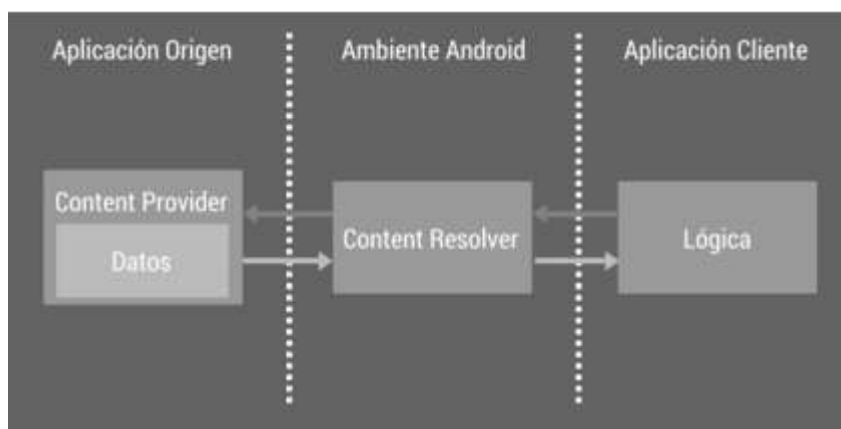


Figura 2.1: Estructura lógica de un Content Provider

La aplicación accede a los datos de un Content Provider mediante un objeto cliente Content Resolver. Dicho objeto provee los métodos CRUD básicos para el almacenamiento persistente y accede a los datos del Content Provider mediante la revisión de las URI de contenido que el segundo utiliza para encapsular su implementación SQLite, protegiendo así los datos del acceso no autorizado por parte de aplicaciones externas.

### 2.2.3 Gestión de almacenamiento de datos en el dispositivo

Tomando en consideración el gran tamaño que la base de datos de GRHS puede llegar a alcanzar (por defecto se necesitan 4GB o más) y que usualmente los dispositivos móviles cuentan con un espacio de almacenamiento interno más limitado que el de los ordenadores tradicionales, se hace vigente la necesidad de implementar un mecanismo que permita controlar cuánta información puede descargar y mantener almacenada la aplicación AvigesCIP.

Ante cada solicitud de sincronización realizada por el usuario, el sistema primero verificará un parámetro en las preferencias de la aplicación. Dicho parámetro corresponde al porcentaje de espacio libre mínimo que debe existir en el dispositivo para autorizar una descarga de datos del servidor. Si el porcentaje de espacio libre, en el momento de la solicitud, es menor que este parámetro, la aplicación abortará el proceso de sincronización y mostrará al usuario un mensaje advirtiéndole que cambie sus preferencias o libere espacio en su dispositivo. De esta forma se evita la ocurrencia de errores de Lectura/Escritura y se mantiene la integridad de los datos.

## 2.3 Planeación

En la etapa de Planeación comienza la interacción con el cliente para identificar los requerimientos del sistema y se identifican las iteraciones y ajustes necesarios a la metodología según las características de la solución. En la misma se describen tres elementos fundamentales, los cuales son: Historia de Usuario (HU),

Tiempo de Ejecución del Proyecto y Plan de Entrega, en ellos se recogen las características principales del sistema a construir y se concreta el tiempo que se necesitará para implementarlas.

### 2.3.1 Características del sistema.

Luego de la interacción con el cliente se definieron 7 requisitos no funcionales y 14 requisitos funcionales, de los cuales se establecieron 6 con una prioridad alta, 5 con una prioridad media y 3 con una prioridad baja. Los requisitos funcionales establecidos son:

No.	Requisito Funcional	Prioridad en el negocio	Nivel de complejidad
RF 1	Autenticar usuario	alta	media
RF 2	Listar cámaras	media	baja
RF 3	Crear cámara	baja	baja
RF 4	Mostrar información de la cámara	baja	baja
RF 5	Actualizar cámara	baja	baja
RF 6	Eliminar cámara	baja	baja
RF 7	Visualizar flujo de video de las cámaras	alta	alta
RF 8	Controlar Angulo de las cámaras PTZ	alta	alta
RF 9	Seleccionar vista	baja	baja
RF 10	Enmudecer el flujo de video	baja	baja
RF 11	Seleccionar preset para las cámaras PTZ	alta	media
RF 12	Grabar el flujo de video entrante	alta	alta
RF 13	Mostrar grabaciones guardadas	alta	media
RF 14	Filtrar grabaciones guardadas por fecha	media	baja

Y por ultimo se establecen como requisitos no funcionales los siguientes:

No.	Requisito No Funcional	Clasificación	Descripción
-----	------------------------	---------------	-------------

RNF 1	Sistema operativo Android	Software	Es necesario debido a que la aplicación se está realizando para esta plataforma.
RNF 2	Resolución de 1080x1920 (WxH) o superior	Hardware	Es necesario para la correcta disposición de los elementos en la pantalla, de esta forma manteniendo la claridad de la imagen se necesita una resolución igual o superior.
RNF 3	Fuente de alimentación superior a 4000 mA	Hardware	El dispositivo debe contar con una batería como fuente de alimentación de 4000 mA o superior para permitir su movilidad por largos periodos de tiempo sin conectar a la alimentación eléctrica.
RNF 4	Versión de Sistema Operativo superior a 6.0 Marshmallow	Software	La versión de sistema operativo tiene que ser mayor que 6.0 Marshmallow debido a que se requieren características del sistema que comienzan a partir de esta versión
RNF 5	Conexión segura	Servidor	La conexión entre AvigesCIP y los servicios del servidor de streaming debe realizarse mediante el protocolo HTTPS, en pos de proteger las credenciales de autenticación.
RNF 6	RNF6. Consistencia de los datos	Servidor	Los datos mostrados en AvigesCIP mantendrán la consistencia con los datos almacenados en la base de datos de las cámaras en el servidor de streaming, esto quiere decir que al descargar un objeto desde el servidor (por ejemplo, una nueva lista de cámaras), la aplicación verificará si el elemento a insertar ya existe en la base de datos local y lo actualizará si ha sufrido algún cambio. Esto evitará la existencia de datos duplicados y la realización de inserciones innecesarias.
RNF 7	Cuenta única	Servidor	Solo existirá una cuenta de GRHS a la vez en un mismo dispositivo. Solo va a poder iniciar sesión una persona designada para ello en ese dispositivo, de

			forma tal que si ocurre alguna incidencia con el mismo exista un responsable.
--	--	--	---

### 2.3.2 Historias de usuario

Kent Beck define a las HU como una parte importante de XP. Constituyen el cimiento de los requisitos del sistema; son el artefacto primario utilizado para definir lo que el equipo de desarrollo estará implementando en una iteración dada. El cliente es el encargado de asignar una prioridad a cada HU y es el equipo de desarrollo el encargado de asignar un costo, que se traduce en las semanas que llevará el desarrollo de las mismas (Mike Cohn, 2004). En la tabla siguiente se muestra un ejemplo de las HU especificadas para el desarrollo de AvigesCIP. Las HU restantes se encuentran en el **Anexo I**.

**Tabla 2.1:** Ejemplo de Historia de Usuario

<b>Historia de usuario</b>	
<b>Numero:</b> HU-1	<b>Nombre de la Historia de Usuario:</b> Autenticar usuario
<b>Modificación de la historia de usuario:</b> 3	
<b>Usuarios:</b> Alejandro Infante de la Torre, Yunieski Orlando Vasconcelo Mir	<b>Iteración asignada:</b> 1
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 2
<b>Descripción:</b> El servidor de streaming contiene un servicio de autenticación, el cual va a ser usado como base para la identificación del usuario dentro de la aplicación requiriendo un nombre de usuario y una contraseña. Asegurando así la confidencialidad y la verificación de roles para la utilización de las distintas características de la aplicación.	
<b>Observaciones:</b> Ninguna	
<b>Prototipo de interfaz:</b>	

**La prioridad en el negocio:**

**Alta:** Son aquellas HU que constituyen funcionalidades fundamentales en el desarrollo el sistema, a las que el cliente define como principales para el control integral del sistema.

**Media:** Son las funcionalidades a tener en cuenta por el cliente, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

**Baja:** Es otorgada a las HU que son funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo.

### **Leyenda de las Historias de Usuario:**

**Número:** número de identificación para las HU, sería incremental en el tiempo.

**Nombre historia de usuario:** es el nombre de la HU, sirve para identificarla fácilmente tanto para los desarrolladores como para los clientes.

**Modificación de historia de usuario:** cantidad de modificaciones que se le ha realizado a la HU (de no tener modificaciones se pone ninguna, si no la cantidad de veces que ha sido modificada).

**Usuario:** nombre del programador encargado de implementar la HU.

**Prioridad del negocio:** qué tan importante es para el cliente, se clasifica en Alta, Media y Baja.

Iteración asignada: iteración en la que se desarrollará la HU.

**Puntos estimados:** tiempo en semanas que se le asignará, donde un punto de estimación corresponde a una semana de 5 días con 8 horas de trabajo diarias.

**Descripción:** es la descripción de la historia, detallando las operaciones del usuario y las respuestas del sistema.

**Observaciones:** informaciones de interés, como glosarios, detalles del usuario, entre otros.

**Prototipo de Interfaz:** contiene la imagen de una de las interfaces de usuario relacionadas con la historia de usuario.

De acuerdo a lo planteado por Cohn y Paul, en XP no es necesario desarrollar una especificación formal de requisitos del sistema. En su lugar, se espera que un conjunto de HU informales realice esa función. Las

HU no son un conjunto único, normalizado, vinculado, completo y consistente. Sin embargo, satisfacen las propiedades de ser abstractas, limitadas, modificables, trazables, comprobables, configurables y granulares. El concepto de HU de XP también satisface la mayor parte de los requisitos de categorización (Identificación, Prioridad, Criticidad, Viabilidad, Riesgo y Fuente) de la guía IEEE.

### 2.3.3 Tiempo de ejecución del proyecto

El tiempo de ejecución del proyecto es un elemento que le permite al equipo de desarrollo completar las historias de usuario en una determinada iteración. Este elemento se calcula sumando los puntos estimados de las historias de usuario realizadas en una iteración. En este sentido un punto estimado equivale a una semana ideal de programación donde se trabaje sin interrupciones, ocho horas diarias durante cinco días. Dado que la planificación se realiza basándose en el tiempo de implementación de una historia de usuario, el tiempo de ejecución se utiliza para declarar cuántas historias de usuario se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de las mismas. La tabla siguiente muestra la estimación del costo en tiempo realizada a partir de la especificación de las HU, la cual provee una idea de lo que tardará el desarrollo de la solución propuesta en el presente trabajo.

**Tabla 2.2:** Tiempo de ejecución del proyecto

Numero	Nombre de la historia de usuario	Puntos Estimados
HU-1	Autenticar usuario	1
HU-2	Listar cámaras	1
HU-3	Crear cámara	1
HU-4	Mostrar información de la cámara	1



HU-5	Actualizar cámara	1
HU-6	Eliminar cámara	1
HU-7	Visualizar flujo de video de las cámaras	2
HU-8	Controlar Angulo de las cámaras PTZ	2
HU-9	Seleccionar vista	2
HU-10	Enmudecer el flujo de video	1
HU-11	Seleccionar preset para las cámaras PTZ	2
HU-12	Grabar el flujo de video entrante	1
HU-13	Mostrar grabaciones guardadas	1
HU-14	Filtrar grabaciones guardadas por fecha	1

La estimación realizada da como resultado una duración aproximada de cuatro meses para el desarrollo de la solución propuesta. Este resultado permite dividir el proceso de desarrollo en etapas para facilitar su realización. Estas etapas en la tercera fase de XP se denominan iteraciones. (Beck, K, 2000)

#### 2.3.4 Iteraciones

Para cada iteración, se define un número determinado de historias de usuario a implementar y al final de cada iteración se obtiene como resultado la entrega de las funcionalidades correspondientes. La duración de cada iteración no debe ser superior a tres semanas. Esto ocurre conjuntamente con la aceptación, por parte del cliente, de los requerimientos solucionados. Idealmente, al final de la última iteración el sistema estará listo para entrar en producción. La tabla siguiente muestra las iteraciones en que estará dividido el

proceso de desarrollo de la solución propuesta en el presente trabajo, así como la distribución de las HU a desarrollar por cada iteración definida.

**Tabla 2.3:** Iteraciones

<b>Iteración</b>	<b>Título historia de usuario</b>	<b>Duración total</b>
1	1. Autenticar usuario 2. Listar cámara	2
2	3. Crear cámara 4. Mostrar información de la cámara. 5. Actualizar cámara	3
3	6. Eliminar cámara 7. Visualización del flujo de video de las cámaras.	3
4	8. Controlar Angulo de las cámaras PTZ 9. Seleccionar vista	3
5	10. Enmudecer el flujo de video 11. Seleccionar preset para las cámaras PTZ 12. Grabar el flujo de video entrante	3
6	13. Mostrar grabaciones guardadas 14. Filtrar grabaciones guardadas por fecha	2

Los resultados anteriores permiten estimar la duración de cada iteración, utilizando para ello los puntos estimados de las HU correspondientes. Esto permite elaborar un plan de entrega de funcionalidades implementadas al final de cada iteración.

### 2.3.5 Plan de entrega

El plan de entrega consiste en establecer, de manera conjunta entre el equipo de desarrollo y el cliente, la duración y fecha de entrega de cada iteración hasta lograr el producto final. En este plan se detalla la fecha fin de cada iteración, mostrando una versión desarrollada del producto en ese momento, hasta lograr el producto final en la fecha establecida. En la siguiente tabla se desglosa el plan de entrega para las funcionalidades implementadas en cada iteración del proceso de desarrollo.

**Tabla 2.4:** Plan de entrega

Iteraciones	Fecha de entrega
1	26-feb
2	18-mar
3	8-abr
4	29-abr
5	20-may
6	3-jun

## 2.4 Diseño de la propuesta de solución

En esta sección se realiza la descripción de la arquitectura de software, los patrones de diseño aplicados a la propuesta de solución y la descripción de las principales clases de la aplicación. Estos elementos muestran una vista técnica y lógica del software a desarrollar, describen la arquitectura utilizada, basada en las necesidades de la solución, así como las clases y sus relaciones; además, son el punto de partida para la posterior codificación.

## 2.4.1 Arquitectura de software

La arquitectura de software de un sistema o programa de computación es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades externamente visibles de estos elementos, y las relaciones entre ellos. (ISO/IEC/IEEE 42010, 2013)

### Arquitectura Cliente/Servidor

El estilo arquitectónico Cliente/Servidor describe sistemas distribuidos que implican un sistema independiente de clientes y servidores, y una red que los conecta. La forma más simple de sistema Cliente/Servidor consiste en una aplicación servidor a la que se accede directamente por varios clientes, a la cual se refiere como un estilo arquitectónico de 2 niveles. (Sun Microsystems, 2009)

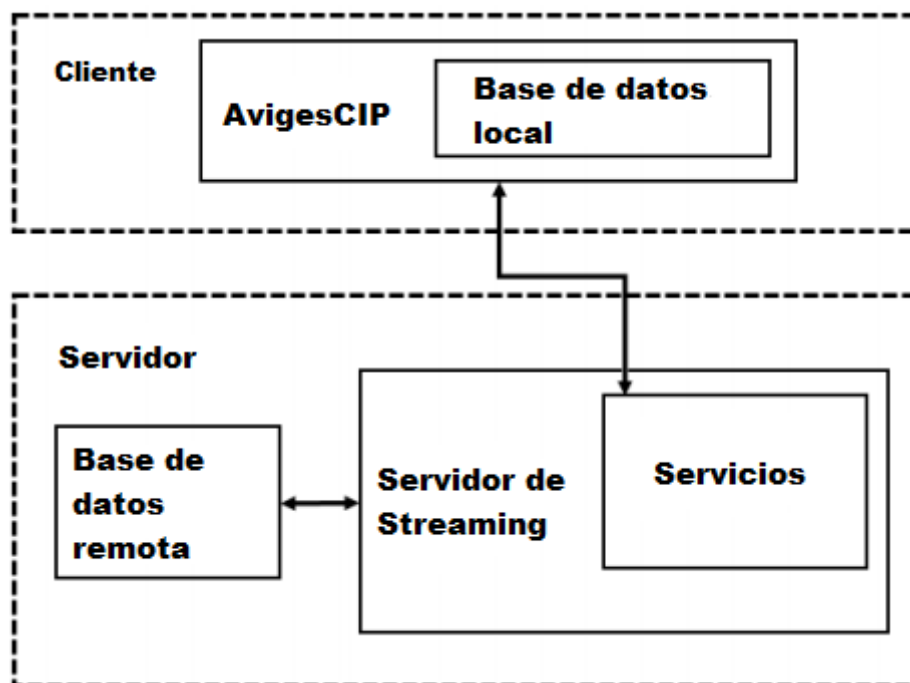


Figura 2.2: Arquitectura Cliente/servidor desglosada en sus dos niveles

En el contexto de la solución propuesta, la aplicación AvigesCIP ocupa el nivel Cliente, ya que la misma será el demandante de información del servidor de streaming y es el software que se ejecuta del lado del usuario. El modelo de solución plantea un escenario en el que múltiples instancias de AvigesCIP solicitan información al servidor, primeramente, la autenticación, y con esta se encarga de decidir si el usuario puede acceder a esta o no dependiendo de sus privilegios o rol.

### **2.4.2 Patrones de diseño**

Un patrón de diseño provee un esquema para refinar componentes de un sistema de software y la forma en que se relacionan entre sí. De manera más simple, un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos (Graham Paul, 2002).

Puede asumirse que un patrón describe una estructura de comunicación de componentes, generalmente recurrente, que resuelve un problema de diseño general dentro de un contexto particular. Por tanto, un patrón de diseño es una solución demostrada, eficiente y reutilizable en diferentes contextos y escenarios de desarrollo de software.

#### **Patrones GRASP**

Los patrones GRASP (General Responsibility Assignment Software Patterns) son un conjunto de patrones de diseño de software que:

*Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (Graham Paul, 2002).*

Los patrones pertenecientes a este conjunto son: Experto, Creador, Alta Cohesión y Bajo Acoplamiento. En la siguiente sección se explica el uso de algunos de estos patrones en el diseño de AvigesCIP.

#### **Alta Cohesión**

La alta cohesión indica que los datos y responsabilidades de una entidad están fuertemente ligados a la misma en un sentido lógico. La información que maneja una entidad de software tiene que estar conectada lógicamente con esta, no deben existir entidades con atributos que describan comportamientos que en realidad no le corresponden. Este patrón permite la obtención de clases fácilmente actualizables y reutilizables durante el proceso de desarrollo de software. Según se plantea, se considera que un diseño de clases provee una alta cohesión cuando: Una clase tiene una responsabilidad moderada en un área funcional y colabora con otras clases para llevar a cabo las tareas (Graham Paul, 2002).

Un diseño con alta cohesión puede observarse en la relación de las clases del paquete **cu.xetid.avigescip.dbproviders**, donde se maneja la capa de acceso a datos locales en AvigesCIP. La clase `DataBaseContract` contiene la definición estructural de la base de datos local de la aplicación, así como los tipos de dato de cada elemento que se maneja en la misma. Colabora con las clases `DataBaseHelper`, responsable de la creación y actualización del esquema de la base de datos, y `DataBaseContentProvider`, clase que implementa los métodos necesarios para realizar las operaciones sobre los datos locales.

### **Bajo Acoplamiento**

El acoplamiento en términos de software se refiere a la relación que se guardan entre los módulos de un sistema y la dependencia entre ellos. El bajo acoplamiento dentro de un sistema indica que los módulos no conocen o conocen muy poco del funcionamiento interno de otros módulos, evitando la fuerte dependencia entre ellos (Graham Paul, 2002).

Ejemplo de bajo acoplamiento sería como el resto de clases no conocen el funcionamiento interno de las clases dentro del paquete **cu.xetid.avigescip.dbproviders**, ya que fuera de ellos solamente se trabaja con `Array List` e internamente se realiza la conversión necesaria para el trabajo con la base de datos usada.

### **Patrón Maestro/Esclavo**

El patrón de diseño Maestro/Esclavo es usado cuando se tienen dos o más procesos que necesitan ejecutarse simultánea y continuamente, pero a diferentes velocidades. National Instruments lo describe

como un patrón de diseño que consiste en múltiples ciclos paralelos. Cada ciclo puede ejecutar tareas a velocidades distintas. De estos, uno actúa como el maestro y los otros como esclavos. El ciclo maestro controla todos los ciclos esclavos y se comunica con ellos utilizando arquitecturas de mensajería (Graham Paul, 2002).

En el diseño de AvigesCIP, existen dos tipos de hilos de ejecución que se ejecutan simultáneamente en el proceso principal de la aplicación. Uno de estos es la UI que maneja la presentación lógica de datos al usuario y su interacción con ellos. El segundo es el de la sincronización con el flujo de video, además de la conexión con el servidor para la obtención de estos flujos de video o la autenticación. En este caso el hilo maestro es el de la UI, ya que controla la ejecución del hilo de sincronización mediante las acciones del usuario. La comunicación entre ambos hilos se realiza mediante Intents.

### 2.4.3 Tarjetas Clase-Responsabilidad-Colaborador

Scott Ambler define las tarjetas Clase-Responsabilidad-Colaborador (CRC) como tarjetas de índice estándar que han sido divididas en tres secciones, una que indica el nombre de la clase que la tarjeta representa, un inventario de las responsabilidades de la clase, y la tercera con los nombres de las otras clases con las que ésta colabora para cumplir sus responsabilidades. En el Manifiesto para el Desarrollo Ágil de Software se recomienda el uso de este artefacto para inventariar las clases que son relevantes para el incremento actual del software. La siguiente tabla muestra una de las tarjetas CRC elaboradas para el modelo de la solución propuesta en el presente trabajo. Las tarjetas CRC restantes se encuentran en el Anexo II Tarjetas CRC (Kent y otros, 2012).

**Tabla 2.5:** Ejemplo de tarjeta CRC

<b>Nombre de la clase:</b> DataBaseCamera
<b>Responsabilidades de la clase:</b> Se encarga de la creación de la Base de datos a utilizar por la aplicación además de la creación de un sistema de versiones necesario por si en el transcurso del proyecto la misma sufre cambios en su estructura.

**Colaboradores:**

- class SQLiteOpenHelper
- class DataBaseHelper
- class SQLiteDatabase
- Interface Cursor

## 2.5 Conclusiones del capítulo

En este capítulo se elaboraron los artefactos correspondientes a la etapa de planeación de la metodología XP, determinando como resultado que el proceso de desarrollo durará un total de 16 semanas, de forma tal que el equipo de desarrollo y el cliente obtendrían un producto funcional al concluir cada iteración, posibilitando a este último plantear su conformidad o no con la aplicación obtenida.

Se especificó la arquitectura de software y los patrones de diseño, lo que permitió el diseño de una aplicación que podrá ser mantenida, auditada y actualizada con relativa facilidad, por parte de los desarrolladores en futuras versiones. Se confeccionaron además las tarjetas CRC, las cuales permitieron establecer el marco de relaciones para las clases del diseño. Estos artefactos ayudarán a una mejor comprensión de la manera en que se encuentra estructurada la aplicación, lo cual facilita su análisis y, de ser necesario, su modificación.



## Conclusiones Generales

Al término de la presente investigación se concluye lo siguiente:

- ✓ El estudio de los sistemas similares permitió identificar características que sirvieron como base para elaborar la propuesta de solución.
- ✓ El estudio del proceso de videovigilancia propicio un mejor dominio del funcionamiento en la ejecución de dicho proceso.
- ✓ La obtención, descripción y validación de los requisitos definieron las características técnicas y funcionales de la propuesta de solución.

## Recomendaciones

Se recomienda la finalización de la aplicación AvigesCIP debido a la ocurrencia a nivel global de la pandemia Covid-19.

## Referencias Bibliográficas

Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J., 2002. "Agile software Development methods Review and analysis". VTT Publications.

Beck, K, 2000 "Extreme Programming Explained. Embrace Change", Pearson Education, 1999. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley,

Newkirk, J., Martin R.C., 2001. "Extreme Programming in Practice". Addison-Wesley.

Fisher, Sharon, 1989. "OS/2 EE to Get 3270 Interface Early". Citado marzo de 2020

Odersky, Martin; Spoon, Lex; Venner, Bill, 2008. "Combining Scala and Java". Citado marzo de 2020

Thompson, Jr., R.J., 2011. Crystal Clear: The Struggle for Reliable Communications Technology in World War II, Hoboken, NJ: Wiley.[www.google.com/apis](http://www.google.com/apis)

Baker, R. Jacob, 2011. CMOS: Circuit Design, Layout, and Simulation Citado abril de 2020

Lazar, Irwin, 2011. "The WAN Road Ahead: Ethernet or Bust?" Citado abril de 2020

Turner, Dawn. 2013 "Understanding eIDAS" Citado marzo de 2020

Turner, Dawn M., 2015 "eIDAS from Directive to Regulation - Legal Aspects"

Turner, Dawn M., 2016 "The Difference Between an Electronic Signature and a Digital Signature" "Technical committees". International Organization for Standardization. Cited from the original on 19 September 2012.

Binstock, Andrew, 2014. "Continuous Delivery: The Agile Successor"

Andrew Binstock, 2010. "InfoWorld review: Top Java programming tools"

Owens, Michael, 2006. The Definitive Guide to SQLite. Apress. doi:10.1007/978-1-4302-0172-4\_1.

Huurdeman, Anton A, 2003. The Worldwide History of Telecommunications.

Emerson, D. T., 1997. "The work of Jagadis Chandra Bose: 100 years of MM-wave research"

Asif, Saad, 2018. 5G Mobile Communications: Concepts and Technologies.

Ambardar, Ashok, 1999. Analog and Digital Signal Processing. Citado marzo de 2020

Bohn, Dieter, 2019. "Google deserts desserts: Android 10 is the official name for Android Q".

Van Camp, 2011. "Google activates 500,000 Android devices a day, may reach 1 million in October"

Wiggers, Kyle, 2017. "Google's virus-scanning Verify Apps feature for Android now reveals its secrets".

Ganapati, Priya, 2010. "Study Shows Some Android Apps Leak User Data Without Clear Notifications | Gadget Lab".

Block, Ryan, 2007. "Google is working on a mobile OS, and it's due out shortly".

Srinivas, Davanum, 2007. "Android — Invoke JNI based methods (Bridging C/C++ and Java).

Chu, Eric 2009. "ADC 2 Round 1 Scoring Complete". Citado mayo de 2020

Fruhlinger, Josh 2010. "Myriad CTO: J2Android moves MIDlets to "beautiful" Android framework".

Westfall, 2009. "Backup & Restore Android Apps Using ADB". Citado mayo de 2020

Morrill, Dan, 2009. "A Note on Google Apps for Android". Citado mayo de 2020

Segall, Bob, 2009. "Tapping your cell phone" Citado mayo de 2020

Greenberg, Andy, 2017. "How the CIA Can Hack Your Phone, PC, and TV".

John Adams 2015, "Martin Gren: IP CCTV's Founding Father". Citado agosto de 2020

A. Del Bimbo, F. Dini, F., 2009 "Pernici Multi-Camera Networks, Principles and Applications".

## Anexos

### Anexo 1 Historias de Usuario

Historia de usuario	
<b>Numero:</b> HU-1	<b>Nombre de la Historia de Usuario:</b> Autenticar usuario.
<b>Modificación de la historia de usuario:</b> 3	
<b>Usuarios:</b> Alejandro Infante de la Torre, Yunieski Orlando Vasconcelo Mir	<b>Iteración asignada:</b> 1
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Descripción:</b> El servidor de streaming contiene un servicio de autenticación, el cual va a ser usado como base para la autenticación dentro de la aplicación requiriendo un nombre de usuario y una contraseña. Asegurando así la seguridad y la verificación de roles para la utilización de las distintas características de la aplicación.	
<b>Observaciones:</b> Ninguna	
<b>Prototipo de interfaz:</b> si	

Historia de usuario	
<b>Numero:</b> HU-7	<b>Nombre de la Historia de Usuario:</b> Visualizar el flujo de las cámaras.
<b>Modificación de la historia de usuario:</b> 3	
<b>Usuarios:</b> Alejandro Infante de la Torre, Yunieski Orlando Vasconcelo Mir	<b>Iteración asignada:</b> 2
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 3

<b>Descripción:</b> Se implementa los controladores necesarios para la visualización del flujo de video utilizando los códecs necesarios para la codificación H264, además de la interfaz de usuario necesaria para verlo.
<b>Observaciones:</b> Ninguna
<b>Prototipo de interfaz:</b> si

Historia de usuario	
<b>Numero:</b> HU-8	<b>Nombre de la Historia de Usuario:</b> Controlar el Angulo de las cámaras PTZ.
<b>Modificación de la historia de usuario:</b> 3	
<b>Usuarios:</b> Alejandro Infante de la Torre, Yunieski Orlando Vasconcelo Mir	<b>Iteración asignada:</b> 4
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 2
<b>Descripción:</b> En caso de que este seleccionada una cámara IP tipo PTZ se volverá visible el control de movimiento 3D para el control de la posición de visualización de la misma.	
<b>Observaciones:</b> Ninguna	
<b>Prototipo de interfaz:</b> si	

Historia de usuario	
<b>Numero:</b> HU-11	<b>Nombre de la Historia de Usuario:</b> Seleccionar preset para las cámaras PTZ
<b>Modificación de la historia de usuario:</b> 3	
<b>Usuarios:</b> Alejandro Infante de la Torre, Yunieski Orlando Vasconcelo Mir	<b>Iteración asignada:</b> 5
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 2

<b>Descripción:</b> Se implementan las posiciones predefinidas así como la velocidad de giro de las cámaras PTZ.
<b>Observaciones:</b> Ninguna
<b>Prototipo de interfaz:</b> si

Historia de usuario	
<b>Numero:</b> HU-12	<b>Nombre de la Historia de Usuario:</b> Grabar el flujo de video entrante.
<b>Modificación de la historia de usuario:</b> 3	
<b>Usuarios:</b> Alejandro Infante de la Torre, Yunieski Orlando Vasconcelo Mir	<b>Iteración asignada:</b> 5
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Descripción:</b> Se implementan los métodos necesarios para la grabación del flujo de video de forma manual en el directorio establecido en las opciones.	
<b>Observaciones:</b> Por defecto el directorio sería: /scard/Grabación/	
<b>Prototipo de interfaz:</b> si	

Historia de usuario	
<b>Numero:</b> HU-13	<b>Nombre de la Historia de Usuario:</b> Mostrar grabaciones guardadas.
<b>Modificación de la historia de usuario:</b> 3	
<b>Usuarios:</b> Alejandro Infante de la Torre, Yunieski Orlando Vasconcelo Mir	<b>Iteración asignada:</b> 6
<b>Prioridad en el negocio:</b> Alta	<b>Puntos estimados:</b> 1

**Descripción:** Se implementan los métodos necesarios para el listado de las grabaciones realizadas hasta el momento dentro de la carpeta predefinida.

**Observaciones:** Por defecto el directorio seria: /scard/Grabación/

**Prototipo de interfaz:** si