



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

**CONTROL REMOTO DE MOVIMIENTO
DE CÁMARA WEB CON
TRACKERPOD EN C++.**

Alumno: Inés Jiménez Martínez

Tutor: Pérez Lorenzo, José Manuel
Viciano Abad, Raquel

Depto.: Ingeniería de Telecomunicación

Septiembre, 2017

Universidad de Jaén
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

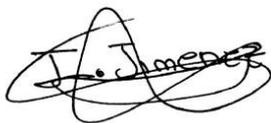
CONTROL REMOTO DE MOVIMIENTO DE CÁMARA WEB CON TRACKERPOD EN C++.

Alumno: Inés Jiménez Martínez.

Tutor: José Manuel Pérez Lorenzo.

Raquel Viciano Abad

Depto.: Ingeniería de Telecomunicación.



Firma del autor /a

Firma del tutor /a

Índice General

Escuela Politécnica Superior de Linares	2
1. Resumen	5
2. Introducción	6
2.1 Antecedentes.	7
2.1.1 Primera cámara web.	7
2.1.2 Utilidades de una cámara web.	8
2.2 Descripción del trabajo desarrollado.	9
3. Objetivos	11
4. Estado del arte	12
4.1 Protocolos de tiempo real y/o streaming HTTP	13
4.2 Protocolos de video Streaming	15
4.2.1 Protocolos de streaming no basados en HTTP.	16
4.2.2 Protocolos de streaming basados en HTTP	18
4.2.3 Formatos y protocolos de video Streaming	21
4.3 Aplicaciones existentes para el control remoto de una cámara de vídeo	24
4.3.1 Programas para ordenador	24
4.3.2 Aplicaciones para Smartphone	25
4.4 Selección de la cámara web y servomecanismo	26
4.5 Sistema de monitoreo remoto	28
4.6 Software actual de control para TrackerPod	29
4.6.1 PTZ Driver vs Trackercam	29
4.6.2 Métodos de comunicación con TrackerPod	30
5. Implementación de plataformas de prueba.	32
5.1 Herramientas de desarrollo	32
5.1.1 Lenguaje de programación C++	32
5.1.2 PHP	32
5.1.3 HTML	33

5.1.4	JavaScript	34
5.1.5	XAMPP	34
5.2	Diseño e implementación de la aplicación.	34
5.2.1	Comunicaciones en red.	34
5.2.2	Funcionamiento	35
5.2.3	Librerías de video.	36
5.2.4	Desarrollo de un modelo cliente/servidor	37
5.3	Control del dispositivo de movimiento de la cámara de forma remota.	39
5.4	Streaming de video o transmisión de video en tiempo real.	42
5.4.1	Descripción de la alternativa de Streaming.	44
6.	Pruebas y resultados	47
6.1	Metodología.	48
6.2	Experimentos	49
7.	Conclusiones y líneas de futuro.	60
7.1	Líneas de futuro.	61
8.	Anexos	62
8.1	Manual de descarga de programas y librerías.	62
8.1.1	Microsoft Visual C++ 2010	62
8.1.2	TrackedPod y OpenCV.	62
8.1.3	VLC	66
8.2	Instalación del servicio TrackerPod.	66
8.3	Streaming VLC vía HTTP	69
8.4	Tabla comparativa de las distintas aplicaciones	71
9.	Índice de Figuras	72
10.	Índice de Tablas	74
11.	Referencias bibliográficas	75

1. Resumen

El término cámara web (webcam) se ha convertido en los últimos años en uno de los conceptos más conocidos en el ámbito de la comunicación para la informática de usuario. Actualmente, es impensable la idea de tener un ordenador y no disponer de este dispositivo, al que se le pueden dar infinidad de usos, como puede ser la comunicación en tiempo real a través de audio y de video con otros usuarios, la video vigilancia, videojuegos más interactivos, aplicaciones comerciales, etc.

En la actualidad la presencia o telepresencia es un concepto que se está integrando en aplicaciones que pretenden hacer más cercana la interacción remota entre personas comunicadas a través de un ordenador y una webcam o un sistema de videoconferencia más avanzado. Así los desarrollos basados en este enfoque tratan de dotar a la aplicación de la sensación de no mediación de la tecnología, introduciendo, por ejemplo, la capacidad de mover el enfoque de la cámara remota en base a los movimientos de la cabeza del usuario local.

En el presente trabajo fin de grado, tiene como principal objetivo la evaluación de distintas alternativas de desarrollo de una aplicación que permita el control remoto de una cámara web. Se persigue evaluar alternativas de bajo coste que permitan desarrollar un sistema para el uso de una webcam que permita su incorporación en aplicaciones de usuario más avanzadas. Esto se ha conseguido implementando el movimiento de una cámara web a través de un dispositivo TrackerPod.

Por lo tanto, este trabajo consta de dos partes, por un lado, se ha desarrollado una aplicación cliente/servidor a través de la cual se puede controlar el movimiento del Trackerpod de manera remota y por otro lado el estudiar las distintas alternativas para llevar a cabo el *streaming* de video de la webcam.

La aplicación desarrollada se ha programado mediante C++, PHP, JavaScript y mediante el uso de librerías propias del dispositivo TrackerPod y librerías específicas para el procesado de imagen y programación de sockets.

2. Introducción

Las siguientes páginas sirven para introducir al lector del Trabajo Fin de Grado (TFG) en el tema abordado de dicho trabajo. Es por esto por lo que se hablará de los comienzos y la historia de las webcams.

Hoy en día podemos encontrar una gran variedad de cámaras web que cuentan con software propio que permite ponerlas en línea para que un número de usuarios (que dispongan de permisos previamente autorizados por el administrador de la aplicación de la cámara) puedan tener acceso a los distintos recursos, ya sea movimiento, video, configuración, etc.

Para poder realizar retransmisiones vía streaming es necesario el uso de protocolos de streaming multimedia. Estos tipos de protocolos de transferencia de datos pueden determinar variables como el método de compresión de datos, el método de comprobación de errores y la indicación de fin de archivo. Si todas las redes estuvieran organizadas de la misma forma y todo el software y equipos de las redes se comportarán de forma similar, sólo sería necesario un protocolo para todas las transmisiones de datos. Sin embargo, Internet está formada por millones de redes distintas con una amplia gama de combinaciones de hardware y software. [15] Es por esto que el uso de protocolos para el intercambio de información multimedia cobra mucha relevancia, ya que la capacidad de transmitir de forma fiable contenido multimedia digital a los clientes depende de varios protocolos de gran complejidad. Es por ello que a la hora de transmitir contenido multimedia existen varias alternativas como son el uso de protocolos en tiempo real o bien el uso de HTTP Streaming, en la sección 4.1 y 4.2 se describen en profundidad cada uno de ellos.

Internet ha experimentado un gran incremento en el número de datos transferidos y en el número de usuarios. Hoy en día, la mayoría de los servicios que se ofrecen en nuestra sociedad consiguen valor añadido mediante el uso de Internet. Este hecho justifica el continuo desarrollo de aplicaciones software sobre redes de comunicaciones entre ordenadores que permitan la adaptación a este nuevo entorno. [1]

Las nuevas formas de utilizar Internet han dado lugar a que el tipo de información que se transmite también haya cambiado, haciendo que hoy día los servicios multimedia generen una gran cantidad de tráfico que viaja a través de la red. Además, las expectativas de crecimiento para aplicaciones como voz sobre IP (Voice over Internet Protocol), videoconferencia o telepresencia indican que irá en aumento.

Como consecuencia de esta evolución, y para soportar el creciente número de usuarios y sus necesidades, las tecnologías de acceso a Internet se han diversificado. Las características heterogéneas de los diferentes accesos a Internet, junto con las exigencias de los usuarios, hacen necesaria la definición de la calidad de servicio (QoS) que ofrecen, particularmente cuando se trata de dar servicio a aplicaciones en tiempo real. Actualmente existen accesos digitales de banda ancha, tanto cableados como inalámbricos. Estos accesos presentan características muy heterogéneas: diferente ancho de banda y retardo, asimetría, tamaño de trama variable, etc., lo que se traduce en diferentes niveles de QoS. [1] A lo largo del tiempo se han desarrollado diversas herramientas de estimación de parámetros relacionados con la QoS, como son el ancho de banda, el retardo o la tasa de pérdida de paquetes. No obstante, el ancho de banda ha sido el parámetro tradicionalmente usado por los usuarios finales para cuantificar las prestaciones de su acceso a Internet. [1]

Las aplicaciones multimedia en tiempo real suelen usar el protocolo RTP (Real-time Transport Protocol), el cual, a su vez, se transporta sobre el protocolo UDP (User Datagram Protocol). Como consecuencia del distinto comportamiento de los protocolos TCP y UDP, los test de velocidad de la conexión a Internet existentes en la actualidad no son tan útiles para las aplicaciones multimedia en tiempo real. Por tanto, se hace necesaria una herramienta que sirva de estimador fiable de las prestaciones de los accesos a Internet en cuanto al uso de aplicaciones multimedia en tiempo real se refiere. [1]

2.1 Antecedentes.

2.1.1 Primera cámara web.

La primera cámara web surgió en 1991 de la mano de Quentin Stafford-Fraser y Paul Jardetzky, que trabajaban en el Departamento de Informática de la Universidad de Cambridge, en el que la cafetera estaba situada en el sótano, estos, hartos de bajar tres plantas y encontrarse la cafetera vacía decidieron diseñar un protocolo cliente-servidor que conectándolo a una cámara, transmitía una imagen de la cafetera a una resolución de 128 x 128, de esta manera desde la pantalla de su ordenador, sabían cuándo había café y quien era quien la dejaba vacía y no volvía a llenarla. El protocolo se llamó XCoffee y después de un tiempo lo comercializaron, saliendo así en 1992 la primera cámara web, la XCam. [3]



Figura 1 Imagen tomada primera cámara

2.1.2 Utilidades de una cámara web.

Una cámara web es básicamente una cámara conectada a un ordenador o a la red, a la que se puede acceder desde el ordenador a la que se encuentra conectada o de manera remota desde otro ordenador o dispositivo con acceso a la red.

A partir de la comercialización de la primera cámara sus usos se han multiplicado, encontrándole utilidad para infinidad de usos, hoy día por ejemplo es impensable el tener un ordenador o portátil que no venga con cámara web incorporada.

Uno de sus usos más extendidos es la videoconferencia, que permite a los usuarios a través de aplicaciones como Skype o Messenger, entre otras, el establecer videoconferencias en tiempo real y de buena calidad con otros usuarios, de una manera mucho más barata y personal que la tradicional conferencia telefónica. Otro de los usos más importantes, útil y más extendido es la video vigilancia, el cual ha avanzado mucho desde ese primer uso que se le dio para vigilar la cafetera del Departamento de Informática de Cambridge.

Hoy día prácticamente todos los establecimientos públicos o privados disponen de un sistema de video vigilancia interno, y con la llegada de la cámara web estos tienen la capacidad de conectarse en tiempo real mediante una aplicación o programa a la cámara o cámaras web que se encuentren situadas en su establecimiento, accediendo así a las imágenes en tiempo real o las grabaciones que dichas cámaras hayan realizado. Pudiendo acceder a ellas desde cualquier sitio o lugar siempre que dispongan de una conexión a internet.

Aparte de estos dos usos que podríamos decir que son los principales hay muchas otras aplicaciones comerciales que hacen uso de las cámaras web, como por ejemplo los videojuegos que las usan para hacerlos mucho más interactivos con el usuario, desde la cámara dreameye que lanzó SEGA en el 2000 hasta el dispositivo Kinect de Microsoft, por poner unos ejemplos.

2.2 Descripción del trabajo desarrollado.

Como ya se ha visto, una de las aplicaciones más importantes y usadas de la cámara web es la video vigilancia, a la hora de vigilar una habitación, establecimiento, o espacio abierto es muy importante el colocar la cámara de manera que el ángulo de visión sea el máximo posible, es por esto que el poder dotar de movimiento la cámara es una gran ventaja, ya que en la mayoría de los casos estas cámaras cuentan con un software dedicado apoyado en un servidor web para facilitar la configuración y el acceso a la funcionalidad de la cámara.

Hoy día uno de los modelos de cámara para video vigilancia comercial más conocidos y que disponen de la capacidad para ponerlas en línea son las AXIS, que también disponen de cámaras dotadas de movimiento, sin embargo, el coste de estas es bastante elevado.

Es por ello que se pensó en utilizar el TrackerPod, fabricado por la compañía americana Eagletron, gracias al cual tenemos un sistema que nos permite la integración de la cámara y un servomecanismo gracias al cual podemos dotar a la misma de movimiento de manera mucho más barata.

TrackerPod es un pequeño mecanismo robótico donde se monta la cámara web. Este cuenta con dos motores a pasos acoplados a un juego de engranajes que le permite generar movimientos horizontales de hasta 160° y verticales de hasta 110° a la cámara. Además, cuenta con un circuito que controla la posición y el grado del paso de los motores, estos dependen de los comandos que el circuito reciba desde el ordenador.

Los principales objetivos de este trabajo son dos, por un lado, el crear una aplicación cliente/servidor a través de la cual poder controlar de manera remota el movimiento del TrackerPod, y por otro lado el análisis de las distintas alternativas para llevar a cabo el *streaming* de video obtenido de la webcam y la integración con el control remoto del movimiento de la cámara.

El proyecto por tanto consta de dos partes diferenciadas:

1. Implementación de una aplicación cliente/servidor para el control de movimientos de la cámara.

La implementación de la aplicación se realiza mediante las librerías propias del TrackerPod y utilizando el lenguaje de programación en C++ y en PHP. Para el intercambio de información, es necesario establecer un protocolo de aplicación con ciertos comandos básicos, de modo que cuando un usuario envíe algún parámetro, como, por ejemplo, las coordenadas de movimiento, se envíe un comando a la aplicación servidor. El servidor recibirá este comando y envía una respuesta, de manera que este lo interpreta y se producen los cambios necesarios a través del uso del controlador de la cámara.

En la comunicación se puede optar entre dos protocolos de nivel de transporte, como son TCP y UDP. El protocolo escogido para este trabajo ha sido el TCP, ya que este tipo de servicio nos garantiza que no se pierda la información involucrada en el proceso de recibir o emitir información.

2. Integración del streaming de video o envío de información de video.

La implementación del *streaming* de video se realizará con las librerías propias del OpenCV y empleando un lenguaje de programación en C++. Al utilizar UDP es una ventaja para las aplicaciones de streaming de video o audio, ya que es muy importante la recepción rápida de los datos, algo bastante importante en el tipo de aplicación que se aborda en el proyecto. Como otra alternativa para el desarrollo de streaming de video en tiempo real se ha implementado mediante el uso de http streaming. En particular, como se describe en la sección 5.4.1., se ha desarrollado mediante el uso del programa VLC como servidor y a través del uso de funciones en JavaScript poder mostrar en una página lo que la cámara está transmitiendo en tiempo real, mediante el uso de un servidor y cliente HTTP.

3. Objetivos

El objetivo principal de este Trabajo Fin de Grado, es estudiar la posibilidad de controlar el movimiento de una webcam de forma remota usando un soporte motorizado, en concreto un dispositivo TrackerPod del fabricante Eagletron, así como de evaluar las distintas alternativas para el streaming de video.

Para ello se partirá de la API facilitada por el fabricante para el lenguaje C++. Además, se estudiarán las distintas alternativas para llevar a cabo el streaming de video de la webcam.

Para cumplir este objetivo necesitaremos abordar las siguientes tareas:

- Estudio de la documentación técnica del soporte motorizado TrackerPod.
- Evaluación de alternativas de implementación del control remoto.
- Implementación de los distintos módulos del cliente y servidor, así como del interfaz gráfico de usuario.
- Evaluación de alternativas para realizar el streaming de video.
- Integración del streaming de video.
- Evaluación de las prestaciones de las alternativas desarrolladas en términos de calidad del servicio.

4. Estado del arte

La telepresencia es un término creado por Marvin Minsky, significa la presencia remota. En otras palabras, es un medio que proporciona a la persona la sensación de estar físicamente en otro lugar por medio de una escena creada por ordenador. Los sistemas de telepresencia sitúan al observador en un mundo real que es capturado por medio de videocámaras ubicadas en distintos lugares y que permiten la manipulación remota de objetos reales mediante brazos robotizados. Es decir, el usuario puede interactuar en un mundo real pero que se encuentra ubicado en un sitio distinto a donde se encuentra en ese momento. [12]

Huawei es una de las empresas que inicia una nueva etapa para las reuniones gracias a productos de conferencia de video y telepresencia, diseñados para la toma de decisiones y la interacción a alta velocidad de hoy en día. Ofrece una alta definición, seguridad, estabilidad, interoperabilidad y gran capacidad, lo que permite que los participantes establezcan comunicaciones de gran realismo y sin ningún tipo de inconvenientes. [11]

El hecho de que se viva cada vez más separados de nuestros familiares y amigos, está provocando que las empresas se vuelquen más en la realización de estos servicios con la mayor sensación de proximidad cuando se interacciona con las personas de forma remota.

Además, podemos encontrar proyectos de investigación en el que se captura el punto en el que una persona dirige su mirada con el sistema *eyetracking*, usando esta información para cambiar el punto de vista en una habitación de manera remota en la que se está empleando una cámara que pueda teleoperarse para su visualización. También, se utiliza para juegos en la red en los que se puede mantener la información del jugador externo mediante el uso de las webcams o incluso por medio de animar un avatar remoto con las capturas de la información de un usuario.

Una cámara web o webcam suelen ser los dos términos utilizados para definir a cualquier cámara que genera imágenes a las que se puede acceder a través de un servidor de Internet o bien conectarse directamente al ordenador, generalmente a través de un puerto USB. [3]

Una cámara web es una cámara que se encuentra conectada a un ordenador, ya sea directamente o de manera inalámbrica, y nos permite captar imágenes que hacen posible la visualización remota.

Las posibilidades que brinda este dispositivo hacen que la tecnología detrás de la webcam sea ampliamente utilizada por todo tipo de usuarios de todo el mundo para diferentes propósitos.

Desde la aparición de la primera cámara web en 1991, las webcams se han introducido en los hogares, empresas, edificios públicos, etc. Es por ello, que estos dispositivos se han convertido en unos de los más utilizados en todo el planeta para diversos usos. [3]

En el caso de su utilización en el hogar, generalmente la cámara web es usada para establecer comunicaciones entre parientes y/o amigos que se encuentran distanciados geográficamente mediante el uso de un programa de video llamada. Es así que las webcams también están siendo utilizadas para seguridad, es decir, para que diversos usuarios puedan controlar de forma remota qué es lo que sucede en su casa. Los diversos usos que se le pueden dar a una cámara web son ilimitados. [3]

En lo que respecta a las empresas, estas suelen utilizarse para llevar a cabo videoconferencias y así poder realizar reuniones con todos los miembros que componen la compañía, sin importar en el lugar en el que estos se encuentren.

Por último y no menos importante, hay que mencionar que la tecnología que acerca la cámara web es también utilizada por una gran cantidad de entidades públicas, que a través de las webcams brindan a la gente acceso a una gran variedad de información. [3]

En este apartado se va a realizar un estudio previo sobre el estado actual de las herramientas disponibles en la actualidad, para el desarrollo de este trabajo.

4.1 Protocolos de tiempo real y/o streaming HTTP

Desde que se produjo la creación de Internet, y con el aumento del número de equipos y de la velocidad de la red, han ido apareciendo tecnologías tales como:

- Radio por Internet
- Videoconferencia
- VoIP
- Televisión por Internet o emisiones en directo. [19]

Actualmente Internet es la herramienta de comunicación más importante del mundo en la que los usuarios cada vez se animan más a transmitir todo tipo de contenidos y compartirlos con el resto de usuarios. Estas necesidades han generado la posibilidad de que los usuarios hayan decidido a retransmitir su contenido vía streaming.

Todos estos servicios requieren que el envío y la recepción de los datos se produzcan en tiempo real, para así poder ser reproducidos en el momento sin necesidad de que se produzcan interrupciones.

El Streaming, es la distribución multimedia a través de una red de ordenadores de manera que el usuario consume el producto al mismo tiempo que se descarga. La palabra streaming se refiere a una corriente continua de tráfico sin interrupción. Esta tecnología funciona mediante un búfer de datos que va almacenando los paquetes que se van descargando para poder mostrársela al usuario. Este término se aplica habitualmente para la difusión de audio o video.

Características y tecnología:

- Se envía la información a través de la red y el cliente la reproduce en tiempo real al recibirla.
- No se produce descarga de la información a un disco local.
- Existen tecnologías de transmisión a través de redes de medios continuos, principalmente audio y video.

El funcionamiento que tiene el streaming es de la siguiente forma:

- Conexión con el servidor: El reproductor cliente conecta con el servidor remoto y este comienza a enviarle el archivo.
- Buffer: el cliente comienza a recibir el fichero y construye un buffer donde empieza a guardarlo.
- Inicio de la reproducción: cuando el buffer se ha llenado, el reproductor cliente comienza a mostrarlo mientras continua en segundo plano con la descarga.
- Caídas de la velocidad de conexión: Si la conexión experimenta ligeros descensos de velocidad durante la reproducción, el cliente podría seguir mostrando el contenido consumiendo la información almacenada en el buffer. Si consume todo el buffer se detendría hasta que se volviera a llenar. [17]

Los servicios de Streaming multimedia tienen como finalidad la difusión de contenido de audio y/o video de manera que el usuario pueda reproducir el contenido al mismo tiempo que se descarga. Las aplicaciones que demandan servicio de tráfico en tiempo real pueden ser divididas en dos categorías:

- **Aplicaciones interactivas en tiempo real:** voz sobre internet (VoIP), video conferencia.
- **Aplicaciones no interactivas:** estas se subdividen en dos categorías:
 - Streaming de audio/video almacenado: música, video clips, películas, etc. Almacenados en servidores, el usuario solicita bajo demanda contenido que previamente está codificado y almacenado por el proveedor de contenidos, algunas de las aplicaciones podrían ser Real Player, Apple Quick Time, Microsoft Windows Media, entre otros.
 - Streaming de audio/video en vivo: es empleado para aplicaciones de audio y video similares a la señal de televisión tipo broadcast de la televisión o radio convencional, es decir, el usuario solicita contenido que está siendo generado en tiempo real.

4.2 Protocolos de video Streaming

En la actualidad hay disponibles una gran variedad de protocolos y técnicas que permiten realizar la distribución del video streaming a través de Internet. Podemos destacar el protocolo HTTP que emplea el protocolo de transporte TCP, para poder asegurar una transferencia de datos fiable. Por otro lado, podemos encontrar protocolos como son: el protocolo de transporte en tiempo real (RTP), el protocolo de streaming en tiempo real (RTSP), que utiliza como protocolo de transporte RTP y UDP y finalmente el protocolo de control de transporte en tiempo real (RTCP).

UDP, es un protocolo no orientado a conexión por lo que no se garantiza la entrega de datos, por lo que no es fiable. Como punto a favor la latencia de este protocolo es mucho menos que si se utiliza HTTP sobre TCP, ya que no solicita una retransmisión de los paquetes que se han perdido. Como punto en contra, el protocolo RTSP puede ser bloqueado por los firewalls, es por ello que su uso no ha crecido tanto comparado con HTTP.

Los protocolos empleados para realizar el video streaming se pueden clasificar en dos grupos: Protocolos basados en HTTP, y por lo tanto en TCP, es decir, su transporte es fiable y por otro lado los protocolos que emplean un protocolo de transporte no fiable, como es UDP. A continuación, se explicarán de una forma más detallada:

4.2.1 Protocolos de streaming no basados en HTTP.

4.2.1.1 Protocolo de Transporte en Tiempo-Real (RTP).

El Protocolo de Transporte en tiempo real fue diseñado para realizar emisiones multicast de tráfico en tiempo real, aunque también se puede emplear en emisiones unicast, además puede ser empleado para el video bajo demanda y en servicios tales como telefonía en Internet. El objetivo de este protocolo es el de proporcionar el transporte de datos extremo a extremo de servicios de datos con características de tiempo real, tales como audio y video interactivos.

RTP multiplexa los paquetes de varios flujos de datos y los manda a través de la red en un único flujo de paquetes UDP, cuando el usuario recibe estos paquetes, el reproductor ira re-ensamblando todos los paquetes que le van llegando y los irá almacenando en el buffer hasta que pueda reproducirlo. Los paquetes que se pierdan, serán ignorados, manteniendo así una sincronización entre el video y audio durante su reproducción.

RTP, se usa conjuntamente con el protocolo de control de transporte en tiempo real (RTCP), ya que mientras que RTP encapsula los streams de video y audio que son transmitidos, RTCP realiza las tareas de monitorización de la comunicación, manteniendo de esta manera un flujo de datos multimedia adecuado entre el cliente y el servidor. RTCP, no transporta contenido multimedia, solamente transmite información sobre el estado de la sesión entre los clientes y el servidor, permitiendo así conocer el estado de la conexión en todo momento.

RTP puede ser empleado para diferentes tipos de aplicaciones multimedia en tiempo real con las que se da cobertura a un amplio espectro de posibilidades en Internet. Las diferentes aplicaciones que puede tener son: [4]

- Conferencia de Audio Simple Multicast.
- Conferencia de Audio y Video.
- Mezcladores y Traductores.
- Codificación Escalable.

Para que se pueda reproducir el contenido en los navegadores Web, es necesario un Plug-in que permite este tipo de streaming, actualmente contamos también con algunos reproductores como Flash, Windows Media Player, QuickTime Player, entre otros, soportan RTP.

4.2.1.2 Protocolo de Streaming en Tiempo-Real (RTSP)

Es protocolo que trabaja a nivel de aplicación, su función es poder establecer y controlar las sesiones multimedia entre el servidor y el cliente con el objetivo de que los datos se entreguen correctamente.

Es un protocolo no orientado a conexión, por lo que el servidor mantiene una sesión asociada a un identificador, en la mayoría de los casos RTSP usa TCP para datos de control del reproductor y UDP para los datos de audio y video. Durante una sesión RTSP, un cliente puede abrir y cerrar varias conexiones de transporte hacia el servidor con tal de satisfacer las necesidades del protocolo. [10]

RTSP dispone de múltiples ventajas que los demás protocolos no pueden ofrecer como son:

- Transmitir en vivo
- Comprensión del video digital.
- Transmisión de los datos en paquetes que son leídos por el cliente mientras llegan.
- Utilización de buffer de reserva para aminorar los retardos inherentes a la red.
- No requiere almacenamiento en el cliente.
- No se produce un desperdicio de ancho de banda.
- Las difusiones son multicast.
- Pueden comprimirse pistas individuales en un film a partir de un servidor, es decir, no importa donde se encuentre el servidor.
- Es escalable. [10]

Por otro lado, las desventajas que ofrece RTSP son:

- Un video se detiene si la tasa de datos excede la velocidad de la conexión.
- Puede ser detenido por firewalls o el NAT.
- Requiere de un servidor de Streaming o un broadcaster.
- No puede usar interactividad. [10]

4.2.1.3 Protocolo de Mensajes en Tiempo-Real (RTMP)

Protocolo propietario, desarrollado por Adobe y publicado en el año 2009. Fue diseñado principalmente para la transmisión de flujos de datos, video y audio, empleando la plataforma Adobe Flash. La latencia de este protocolo es baja, debido al uso de Flash Player para poder reproducir el contenido bajo demanda y en directo. Al disminuir la latencia, favorece que la entrega del contenido multimedia sea fluida. Este protocolo trabaja sobre TCP, por lo que el control de congestión es más efectivo que RTP, que trabaja con UDP.

RTMP, emplea la técnica de flujo dinámico, donde la calidad del video se ajusta automáticamente ante los cambios que se produzcan en el ancho de banda. El ajuste de la calidad del video, es posible debido a que este protocolo divide el contenido a transmitir en pequeños bloques de información de igual tamaño, por lo que la calidad de video se va ajustando a las condiciones de la red, por ejemplo, a las variaciones del ancho de banda.

4.2.2 Protocolos de streaming basados en HTTP

HTTP es un protocolo basado en el método de petición/respuesta sobre el protocolo de transporte TCP; es un protocolo sin estado, esto quiere decir que el servidor no mantiene información sobre las peticiones que un cliente ha realizado.

HTTP posee la ventaja de que no presenta problemas a la hora de atravesar firewalls, ya que al ser un protocolo usado por los navegadores Web es poco probable que un administrador de red bloquee el puerto que utiliza este protocolo. Otra de las ventajas que presenta frente a otros protocolos es a la hora de distribuir el contenido multimedia ya que es utilizado por una técnica llamada Adaptive Bitrate Streaming (ABR), la cual es utilizada para optimizar la entrega de contenido multimedia a los usuarios finales. Todo ello ha hecho que HTTP sea entre los protocolos preferidos a la hora de transmitir video streaming sobre Internet. A continuación, se va a hablar de algunos protocolos que emplean HTTP.

4.2.2.1 HTTP Live Streaming (HLS)

Es un protocolo de comunicación desarrollado por Apple para realizar el streaming de contenidos multimedia, basado en HTTP. Proporciona un medio fiable y rentable, con la capacidad para realizar un video continuo y con un formato largo a través de Internet.

Su funcionamiento se basa en la segmentación del contenido en pequeños ficheros a transmitir, estos segmentos se descargan empleando el protocolo HTTP. Según se vaya reproduciendo el contenido y dependiendo de la conexión, el cliente podrá modificar el bitrate de los archivos descargados, esto lo consigue simplemente seleccionando el siguiente segmento con una compresión diferente.

Dado que HLS funciona utilizando el protocolo estándar HTTP, no tiene ningún tipo de problemas para atravesar los firewalls o servidores proxy. Esto no ocurre por ejemplo en otros protocolos basados en UDP, como por ejemplo el protocolo RTP. Este protocolo es adecuado para su utilización en emisiones que sean en directo o para video bajo demanda.

Para poder crear un servidor de este tipo, se puede usar un servidor web ordinario, lo único que necesitaremos es un proceso previo del video o audio para poder realizar la codificación y la segmentación. La figura 2 nos muestra un esquema típico de un servidor HLS. [18]

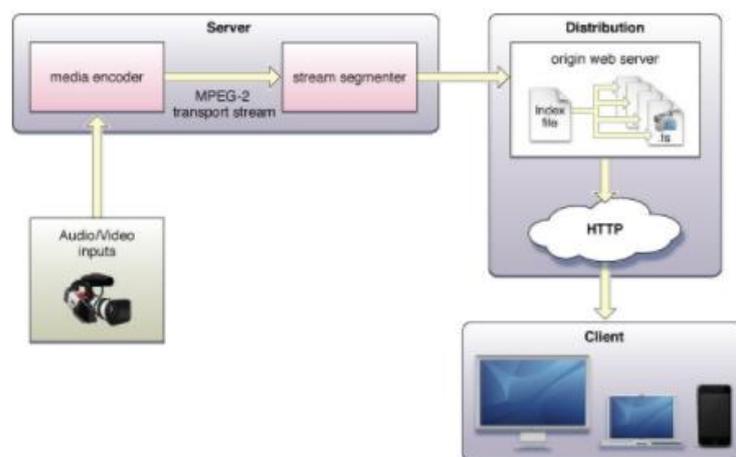


Figura 2 Esquema servidor HLS

HLS, es ampliamente soportado por servidores de streaming de varios fabricantes como es Adobe, Microsoft, etc.

4.2.2.2 HTTP Dynamic Streaming (HDS)

Es una tecnología desarrollada conjuntamente por Adobe y Akamai en el año 2010, el objetivo que persigue es la aplicación en los servidores de Adobe Flash Media. Está basado en la técnica de streaming adaptativo, permitiendo de esta manera poder manejar el streaming de video usando protocolos HTTP y RTMP.

Este protocolo combina las características de los métodos de descarga progresiva y las de streaming, permitiendo a los usuarios poder tener una reproducción instantánea y la capacidad de cambiar la reproducción en cualquier punto en el video. [20]

4.2.2.3 Dynamic Adaptive Streaming over HTTP (Dash)

Tecnología desarrollada por el comité MPEG, conocida como MPEG-DASH, publicada en la ISO en el año 2012. El comité buscaba con DASH, poder contar con un estándar que resuelva la complejidad a la hora del envío de contenido multimedia a través de diferentes clientes que operaban bajo estándares propietarios como HLS y HDS entre otros.

Este nuevo estándar se puede utilizar para la transmisión bajo demanda o contenido de eventos en directo. DASH define un archivo Manifest llamado MPD, el cual describe las características multimedia almacenadas en el servidor. MPD no es más que un documento en XML constituido por periodos, cada periodo dispone de un identificador con un tiempo de inicio y la duración de este.

Para la reproducción del contenido, un cliente primero tiene que obtener el archivo MPD, el cual puede ser entregado por múltiples vías como es HTTP, email, etc. Una vez obtenido el MPD, el cliente puede conocer las características del contenido multimedia. Usando esta información, el cliente escogerá las características que más se ajusten a su perfil, para posteriormente comenzar con la descarga y reproducción del contenido, además el reproductor del cliente realiza la monitorización de la red y las características del equipo del cliente.

Los beneficios de utilizar esta tecnología, es el hecho de que la mayoría de las compañías multimedia como son Netflix, Google, Microsoft, entre otras, formen parte en su desarrollo de manera directa o indirectamente, hace que este protocolo elimine los problemas técnicos en la entrega del contenido y la codificación del mismo, es decir, combina tecnologías en una sola.

4.2.2.4 Microsoft Smooth Streaming (MSS)

Protocolo desarrollado por Microsoft en el año 2009, basado en el método de descarga progresiva sobre HTTP. El contenido multimedia se codifica con distintos bitrates y niveles de calidad de video, posteriormente el contenido se divide en pequeños fragmentos llamados chunks. Las descargas HTTP mediante los chunks, permiten que el contenido multimedia sea más fácil y económico de almacenar en las caches Web de los servidores que proveen de este contenido a los usuarios finales.

Microsoft cuenta con reproductor llamado Silverlight, que permite la reproducción del contenido que le cliente solicita. Este se encarga de mantener y manejar la información indexada en el archivo Manifest XML, que contiene información acerca de los fragmentos codificados.

4.2.3 Formatos y protocolos de video Streaming

Los formatos del video streaming es muy importante ya que de ello dependerá la calidad que tenga el video cuando llegue a los usuarios finales. Un generador de contenidos multimedia necesita que el contenido vaya en el formato adecuado para poder ser transmitido. El contenido tendrá que pasar por un proceso de codificación que le permita reducir el tamaño en comparación con el video original. Para ello se hace uso de los códecs de audio y video, que nos permiten mejorar la calidad de la imagen y audio reduciendo la tasa de bits necesarios para la reproducción.

A continuación, se va a describir las distintas alternativas de codecs de video que se utilizan hoy día.

4.2.3.1 Formatos de video streaming

Hoy día se dispone de muchos formatos de video que permiten almacenar el contenido multimedia en un único archivo, dependiendo del formato que el video posea tendremos una calidad determinada y distintos tamaños de video. Todo esto depende de los tipos de codecs empleados para la comprensión del video. Los formatos de video están compuestos por los contenedores y codecs de video y audio.

- Formatos contenedores:

Un archivo audiovisual puede tener varios audios, videos y una cantidad variable de metadatos, todo este conjunto se compila en formatos contenedores.

En este tipo de formato contenedor están las componentes de video y audio. Cada una de ellas esta codificada mediante un códec, es decir, un archivo de video tiene un formato contenedor y un códec, y dentro de un formato contenedor puede haber distintos tipos de códec. A continuación, en la figura 3 se muestra el formato del contenedor. [22]



Figura 3 Formato Contenedor

Dentro de los formatos contenedores podemos encontrar los siguientes tipos:

- AVI: contenedor estándar de los equipos de Microsoft. Este tipo de archivo contiene audio y video en un contenedor estándar que permite la reproducción en simultáneo.
- MP4 que es el contenedor genérico empleado para los archivos codificados en MPEG-4. Se emplea para almacenar medios con las definiciones de ISO/IEC Moving Picture Experts Group, aunque también puede almacenar otros tipos de medios. Permite streaming de contenido bajo demanda vía Internet, así como el multiplexado de múltiples pistas de audio y video en un archivo único.
- MOV: es el contenedor estándar de QuickTime, del fabricante Apple. Este formato permite contener pistas comprimidas con otros codecs tales como Cinepak, MP-, JPEG, DivX o PNG o una referencia a un medio almacenado en otro archivo o ubicación de red.

Por otro lado, los formatos contenedores más avanzados son MKV u Ogg permiten almacenar audio, video, subtítulos y meta-datos, junto con la información de sincronización requerida para la reproducción. [21]

- Codecs de video y audio.

El video sin compresión requiere una enorme cantidad de recursos, es por ello que se requieren técnicas de compresión digital del video original para reducir así la redundancia del mismo. Muchos estándares de codecs han sido propuestos desde los años 80, los cuales especifican las técnicas de codificación que emplea las diferentes aplicaciones que se muestra en la siguiente imagen.

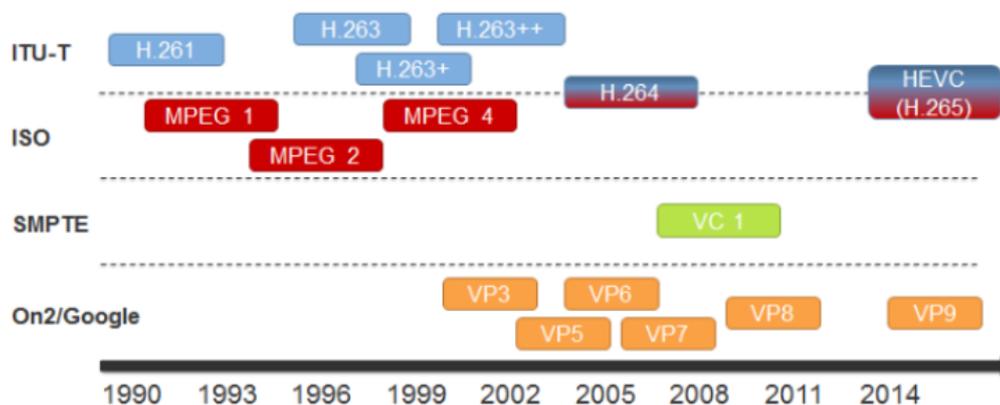


Figura 4 Codecs de Video

Por otro lado, los codecs de audio también tienen una gran importancia en el servicio de video streaming. Actualmente, los códecs de audio más utilizados son el formato MP3, el Ogg Vorbis, el cual se caracteriza por ser un códec perteneciente al software libre, y el AC3 que suele utilizarse en compresiones de DVDs ya que se trata de un códec multicanal. Debido a su gran popularidad obtenida en los últimos años gracias a su masiva difusión en Internet, el formato líder de compresión de audio es sin lugar a dudas el MP3.

En cuanto a los códecs con pérdida, tanto el formato MP3 como el Ogg Vorbis y el AC3, comprimen con cierta pérdida de información, al igual que los códecs MP1, MP2, AAC, WMA, ADPCM, ADX, ATRAC y muchos otros. Por el contrario, existen gran cantidad de códecs que permiten una compresión de la fuente original de audio, sin la necesidad de pérdida de datos, tales como FLAC, LPAC, APE, WavPack, entre otros. [2]

4.3 Aplicaciones existentes para el control remoto de una cámara de vídeo

En el mercado de las tecnologías podemos encontrar múltiples aplicaciones para poder controlar desde nuestro ordenador o Smartphone una cámara web, esta cámara puede ser una cámara IP, la web cam de nuestro portátil, o incluso un Smartphone que tengamos en desuso. El objetivo que esto persigue es poder controlar a los niños, mascotas o hasta poder ver la televisión mientras no nos encontramos en casa, directamente desde nuestro dispositivo móvil Android, BlackBerry, iPhone y Windows Phone, o desde nuestro ordenador.

Además, es una alternativa para tener un sistema de seguridad en casa mucho más simple y más económico. A continuación, se van a mostrar una serie de aplicaciones que se encuentran disponibles como las mejores valoradas del mercado actual, lo que implica que, aunque hay versiones gratuitas para los dispositivos mencionados anteriormente, hay más alternativas. A continuación, se va a proceder a explicar este tipo de aplicaciones:

4.3.1 Programas para ordenador

Si nos adentramos en el mundo de la videovigilancia encontramos gran cantidad de programas que funcionan con cámaras IP en el mercado, el precio de estos programas puede ir desde unos 1500€ hasta versiones gratuitas, dependiendo de las funciones que incorporen y el número de cámaras capaces de soportar, esto último también dependerá de la capacidad de nuestro ordenador, si nuestro ordenador no es lo suficientemente potente no será capaz de gestionar tanta información. Algunos de estos programas son:

- **Blue Iris:** Es un software desarrollado para Windows que permite al usuario controlar la gran mayoría de marcas de cámaras IP existentes en el mercado, como AXIS, Foscam, AirLink, BOSCH, y un largo etc. Es compatible también con webcams USB. Es un programa multicámara capaz de soportar hasta 64 cámaras, y dispone de funciones como PTZ, grabación de imágenes y video con/sin audio, alertas de detección de sonido y movimiento, y muchas más funciones.

- Webcam XP: Software para Windows que permite al usuario conectarse remotamente usando otras computadoras o su teléfono móvil, soporta una gran selección de cámaras (más de 1500 modelos), según la versión del programa que se haya adquirido, free, private, pro puede soportar un máximo de 1, 5, o un número ilimitado de cámaras. Soporta el control PTZ de la cámara tanto local como remoto, sensores de movimiento ópticos o acústicos, etc.
- IP Camera Viewer: Es un software desarrollado para macOS 10.52 o posterior, este software permite conectarse a un número ilimitado de cámaras, así como grabar vídeo siempre, según un horario, o según los movimientos que detecte. Entre algunas de las funciones está el envío de imágenes por email, subir vídeos a un servidor FTP, etc.
- ZoneMinder: Es un programa de código abierto para Linux que permite conectar desde una cámara hasta todas las que se quieran, así como conectar cámaras IP o incluso cámaras analógicas, soporta visión nocturna, resolución 4k. Este, además cuenta con una aplicación para móvil zmNinja, disponible para móviles Android e IOS, pudiendo controlarlo así todo tanto desde el pc como desde nuestro dispositivo móvil.

4.3.2 Aplicaciones para Smartphone

Al igual que para PC hay multitud de programas en el mercado que nos permiten convertir nuestro PC en un sistema de vigilancia también hay infinidad de aplicaciones que permiten hacer lo mismo con nuestro Smartphone o Tablet, muchas de estas aplicaciones como zmNinja, o Blue Iris Client son clientes para móvil del correspondiente programa para PC. Algunas de estas aplicaciones son:

- IP Cam Viewer: Esta aplicación está disponible para los dispositivos Android, con ella podremos controlar diversos dispositivos de cámara, incluidas las de nuestro ordenador, con lo que podemos convertir nuestro teléfono en un sistema de videovigilancia. Cuenta con dos versiones gratuitas, pero la más completa cuesta poco más de tres euros. [6]

- Mobiscope: Pasamos de Android a BlackBerry, esta opción es algo más compleja, Mobiscope ofrece un servicio de monitorización online con un sistema de prueba gratuito tras pasado ese periodo requiere de un coste adicional, aun así, sigue siendo más barato que tener un sistema de seguridad y cuenta con una función de envío de un correo electrónico si la cámara detecta movimiento. [6]
- Air Cam Live Video: Es quizás una de las mejores alternativas para nuestro dispositivo iPhone, dispone de dos modalidades, una de ellas gratuitas y otra de pago. La versión Lite únicamente funciona sobre la misma red, por lo que es ideal para poder controlar el resto de estancias, si lo que queremos es controlar nuestra vivienda fuera de casa debemos de pasar a la aplicación de pago.

4.3.2.1

- WEB CAM Remote: Esta última aplicación nos lleva hasta el sistema operativo Windows Phone, nos permitirá monitorear la webcam del PC desde el teléfono gracias a este tipo de aplicación. Su uso es muy sencillo, únicamente necesitamos descargar el servidor remoto en nuestro ordenador desde el link que nos facilita la aplicación y tras ello podemos abrir la aplicación y escoger nuestro pc como puerta de enlace a las imágenes que capte la webcam.

En el anexo 8.4 se incluye a modo informativo una tabla comparativa de algunas de las aplicaciones existentes. [6]

4.4 Selección de la cámara web y servomecanismo

Uno de los modelos de cámara comercial más conocidos con la capacidad de ponerlas en línea son las AXIS, cuentan con un servomecanismo que les permita tener movimiento, pero el precio de estas es elevado. Es por esto que se ha decidido buscar otras alternativas que permitan tener la fusión de la cámara y el servomecanismo o un servo al que se le pudiera acoplar una cámara web y que al mismo tiempo fuera económica.

Después de buscar y comparar varios modelos de cámaras se encontró un sistema que conjunta la cámara y el servomecanismo, conocido como Trackercam. Este sistema cuenta con muchas características que lo hacen muy completo.

Está conformado por una cámara web de la marca Logitech C920 HD Pro. Este modelo de cámara es recomendado por ser muy fácil de acoplar al servomecanismo, y sus especificaciones son las siguientes:

- Videoconferencias Full HD 1080p (hasta 1920 x 1080 píxeles) con las versiones más recientes de Skype para Windows.
- Videoconferencias HD 720p (1280 x 720 píxeles) con clientes compatibles.
- Grabaciones de video Full HD (hasta 1920 x 1080 píxeles)
- Compresión de video H.264
- Micrófonos estéreo integrados con reducción de ruido automática.
- Corrección automática de iluminación escasa.
- Clip universal compatible con trípodes para monitores LCD, CRT o portátiles.
-

Sin embargo, no es una condición el usar este modelo de cámara, ya que se puede utilizar cualquier otro que se adapte al servomecanismo y que cuente con un controlador para que el software del sistema pueda reconocerla.



Figura 5. Cámara Logitech C920 HD Pro

La otra parte que conforma el sistema del servomecanismo que se conoce como **Trackerpod**. Este es un pequeño mecanismo robótico donde se monta la cámara web. Este sistema cuenta con un software inteligente que permite controlar los movimientos de la cámara desde el ordenador, además de que permite conectar la Trackercam a través de la web, es decir, la pone en línea para que se pueda acceder remotamente desde una página web a los recursos de video, movimiento y zoom digital y así monitorear.

Este software permite que utilizando cualquier cámara web se puedan escalar las imágenes desde un 50% hasta un 300% para así tener una vista más detallada o mejores acercamientos y más aún si se cuenta con una cámara con buena resolución.



Figura 6. Servomecanismo Trackerpod

4.5 Sistema de monitoreo remoto

TrackerCam es una aplicación de software que permite poner una cámara web en Internet. Suele ser utilizado para la vigilancia y para hacer cosas como acceder a servicios de video de un teléfono o cargar imágenes a un servidor FTP. TrackerCam es un software totalmente gratuito.

Trackercam tiene numerosas ventajas, y una de las más importantes es que se puede conectar directamente a una computadora conectada a Internet. Tanto la cámara como la base giratoria se conectan a través de un puerto USB. Con la configuración adecuada y creando los permisos necesarios, la cámara se encuentra lista para ser usada remotamente desde una página web con la dirección IP del ordenador al que se encuentre conectado la cámara. En la figura 7 se muestra el esquema que se emplea para el monitoreo remoto.

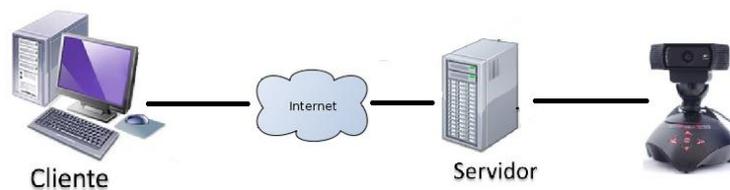


Figura 7. Sistema de Monitoreo Remoto

4.6 Software actual de control para TrackerPod

4.6.1 PTZ Driver vs Trackercam

PTZdriver: Es un software de control básico para la base móvil TrackerPod.

TrackerCam: Es un software de control avanzado que trabaja conjuntamente con la base móvil TrackerPod y con la cámara web utilizada. [7]

FUNCIONES	TRACKERCAM	PTZDRIVER
Funciona con cámaras web/ video	Sí / Sí	No / Sí
Control Pan/Tilt/Zoom	Sí / Sí / Sí	Sí / Sí / Sí
Seguimiento de movimiento/personas	Sí / Sí	No / No
Servidor de video en página Web	Sí	No
Captura de video/imágenes	Sí / Sí	No / Sí
Subida de imágenes a servidores FTP	Sí	No

Tabla 1 TrackercamVS PTZDriver

4.6.2 Métodos de comunicación con TrackerPod

4.6.2.1 Peticiones/respuestas HTTP POST

Post es un método HTTP con el cual enviamos información desde el cliente para que esta sea procesada y actualice o agregue información en el servidor, como podría ser la carga o actualización en sí de una noticia.

Cuando enviamos (request) datos a través de un formulario, estos son procesados y luego a través de una redirección devolvemos alguna página con información (response). Este método sólo se puede usar con formularios, y la información no se envía por la URL, sino que es invisible al usuario.

Este es el método de control más simple para manejar la base del TrackerPod. Se puede utilizar desde cualquier lenguaje de programación. Las peticiones POST desencadenan la ejecución de funciones PHP por parte del servidor integrado en TrackerPod. [7]

4.6.2.2 Usando la librería C++

El uso de esta librería es el siguiente:

- Ejecutar el programa "trackerpod_server.exe"
- Incluir el fichero de cabeceras "TrackerPod.h"
- Enlazar el proyecto con el fichero "TrackerPod_ml.lib" / "TrackerPod_mt.lib"

Internamente la librería funciona a través de la ejecución del programa llamado "Administrador de TrackerPod".

4.6.2.3 Usando el componente Component Object Model (COM)

COM es una plataforma de Microsoft para componentes de software, introducida en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología. El término COM es a menudo usado en el mundo del desarrollo de software como un término que abarca las tecnologías OLE, OLE Automation, ActiveX, COM+ y DCOM. Si bien COM fue introducido en 1993, Microsoft no hizo énfasis en el nombre COM hasta 1997.

Esencialmente COM es una manera de implementar objetos neutrales con respecto al lenguaje, de manera que pueden ser usados en entornos distintos de aquel en que fueron creados, a través de fronteras entre máquinas. Para componentes bien creados, COM permite la reutilización de objetos sin conocimiento de su implementación interna, porque fuerza a los implementadores de componentes a proveer interfaces bien definidas que están separados de la implementación.

Las diferentes semánticas de reserva de memoria están acomodadas haciendo a los objetos responsables de su propia creación y destrucción por medio del contador de referencias. El método preferido de herencia en COM es la creación de sub-objetos a los que se delegan las llamadas a métodos (llamado agregación). [5]

El uso de esta librería es el siguiente:

- Ejecutar el programa "compodsrv_setup.exe" para instalar el componente "trackerpod_com.dll".
- Crear una instancia del componente.
- Usar la interfaz ITrackerPodC del componente.

Internamente el componente funciona a través de la ejecución del programa llamado "Administrador de TrackerPod". [7]

5. Implementación de plataformas de prueba.

5.1 Herramientas de desarrollo

Para el desarrollo de este Trabajo Fin de Grado emplearemos el lenguaje de programación C++, PHP, así como el entorno de desarrollo integrado Visual Studio 2010.

A continuación, se describirán todas estas herramientas de forma más detallada:

5.1.1 Lenguaje de programación C++

C++ es un lenguaje de programación orientado a objetos que toma la base del lenguaje C.

C++ es un lenguaje de programación diseñado a mediados de los años 80 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido. Posteriormente se añadieron facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos (programación estructurada y la programación orientada a objetos).

En la actualidad, el C++ es un lenguaje versátil, potente y general. El C++ mantiene las ventajas del C en cuanto a la riqueza de operadores y expresiones, flexibilidad, concisión y original. [13]

5.1.2 PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de código abierto de propósito general, multiplataforma y que está especialmente adecuado para el desarrollo web de contenido dinámico, y que puede ser embebido en páginas HTML.

Fue uno de los primeros lenguajes de programación del lado del servidor al que podían incorporar directamente en el documento HTML, en lugar de llamar a un archivo externo que procese los datos. [14]

PHP tal y como se conoce hoy en día es en realidad el sucesor de un producto llamado PHP/FI. Fue creado en 1994 por Rasmus Lerdorf, la primera encarnación de PHP era un conjunto simple de ficheros binarios Common Gateway Interface (CGI) escritos en el lenguaje de programación C. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Publica General de GNU, debido a las restricciones del uso del término PHP.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero además ofrece muchas características avanzadas para los programadores profesionales.

PHP puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin coste alguno. El lenguaje PHP se encuentra instalado en más de 200 millones de sitios web y en un millón de servidores. Además, presenta la capacidad de por ser ejecutado en la mayoría de los sistemas operativos, tales como Microsoft Windows, Mac OS X o Linux, así como interactuar con los servidores web más populares ya que existe en versión CGI, modulo para Apache e ISAPI.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como es C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje corta. [14]

5.1.3 HTML

Es el acrónimo de “HyperText Markup Language”, se trata de un lenguaje de marcado empleado para la creación de páginas web, además es un lenguaje muy sencillo que nos permite definir documentos de hipertexto mediante el uso de etiquetas que marcan partes del documento dándoles así una estructura, para que visualmente el texto se presente de una forma ordenada y agradable. Permite la inserción de multimedia, ya sea fotos, video, audio, etc. [9]

5.1.4 JavaScript

Lenguaje de programación que surgió con el objetivo de extender las capacidades del lenguaje HTML. Los scripts de JavaScript pueden ser introducidos en las páginas web. Permite dar una respuesta a los eventos iniciados por el usuario, tales como la entrada de un formulario o pinchar sobre un enlace determinado, por lo tanto, permite la verificación y validación de datos del usuario, que se desarrollan en el lado del cliente, sin ser necesaria la transmisión de dichos datos al servidor para su validación. [9]

5.1.5 XAMPP

Es un servidor independiente de software libre, que consiste principalmente en la base de datos de MySQL, el servidor Web Apache y los interpretes para lenguajes de script como son, PHP y Perl.

El programa se encuentra liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X. Es un software que se puede utilizar en cualquier ordenador ya que no necesita muchos recursos. [16]

5.2 Diseño e implementación de la aplicación.

Para el siguiente apartado se va a proceder explicar detalladamente los pasos que se han llevado a cabo para el desarrollo correcto de la aplicación cliente/servidor, tanto para el control remoto, como para el streaming de video.

5.2.1 Comunicaciones en red.

La manera más sencilla y eficaz de comunicar programas dio origen a la aparición de los sockets, que permitían la entrega de paquetes de datos provenientes de la tarjeta de red.

Los sockets, emplean una serie de primitivas para establecer el punto de comunicación, es decir, para conectarse a una maquina remota que este asociado a un

determinado puerto que se encuentre disponible, para poder escuchar en él, leer o escribir y publicar información, y finalmente poder desconectarse.

Para que los programas puedan comunicarse, deben de cumplir los siguientes requisitos:

- Ambos programas deben de ser capaces de localizarse.
- Que puedan intercambiarse datos entre sí.

Para establecer un socket es necesario, como mínimo los siguientes recursos:

- Un protocolo de comunicación, necesario para el intercambio de datos.
- Dos direcciones de red que identifiquen el origen y el destino de la comunicación.
- Un par de números de puertos que identificaran los procesos que se ejecutaran a nivel de aplicación.

De esta forma, se permite a implementar una arquitectura cliente/servidor.

5.2.2 Funcionamiento

Un servidor se ejecuta sobre una máquina, y dispone de un socket que responde en un puerto específico. El servidor únicamente permanece escuchando a través del socket, esperando a que un cliente realice una petición.

El cliente conoce el nombre del host en el cual el servidor se encuentra ejecutándose y el número del puerto en el que está conectado. Para que el cliente pueda realizar una petición de conexión, este debe intentar conectar con el servidor, que se encuentra en una máquina con una dirección y puerto especificado.

Si todo se ha realizado correctamente, el servidor aceptará la conexión y crea un nuevo socket sobre un puerto diferente, ya que, en el que se ha realizado el establecimiento de la conexión permanecerá a la escucha de otras posibles peticiones por parte de otros clientes.

Por parte del cliente, si la conexión se ha realizado satisfactoriamente, el socket se creará correctamente y puede ser usado para comunicarse con el servidor.

Ahora el cliente y el servidor se podrán comunicar desde sus respectivos sockets.

La librería de sockets más utilizada es la siguiente:

- Winsock: esta es una biblioteca dinámica de funciones DLL para Windows que se hizo con la finalidad de poder implementar aplicaciones basadas en el protocolo TCP/IP. Incluyen soporte para el envío y recepción de paquetes de datos a través de sockets.

5.2.3 Librerías de video.

OpenCV (Open Computer Vision) es una librería multiplataforma muy extendida en el ámbito del tratamiento de imágenes. Se distribuye bajo la licencia BSD y es de código abierto. Se encuentra disponible para Windows, Linux y Mac OSX, escrita en los lenguajes de programación en C/C++, Python y Java. Inicialmente fue desarrollada por el departamento Intel Research, perteneciente a la compañía Intel, en la actualidad el desarrollo y soporte de la librería lo continua la fundación OpenCV.org

Actualmente, contiene más de 2500 algoritmos de procesamiento de imágenes, como son las siguientes: funciones que abarcan una gama de áreas en el proceso de visión, como es el reconocimiento de objetos, calibración de cámaras, visión estera y visión robótica. [4]

Fue diseñada para ser eficiente en aplicaciones en tiempo real y proveer de un framework de visión que fuera sencillo de utilizar, permitiendo así la construcción rápida de aplicaciones de visión que fueran potentes y robustas. [8]

Las librerías de OpenCV están compuestas por cinco módulos:

- CV: Contiene las funciones principales, tales como procesamiento de imágenes, análisis de la estructura de la imagen, detección del movimiento y rastreo de objetos, entre otros.
- CXCORE: Define las estructuras de datos y funciones de soporte para algebra lineal, persistencia de objetos, transformación de datos y manejo de errores entre otros.
- HighGUI: Empleada para construcción de interfaces de usuario sencillas y muy ligeras.
- CVAUX: Formada por un conjunto de funciones auxiliares de OpenCV.
- Machine Learning: Implementa funciones para agrupación, clasificación y análisis de datos. [8]

5.2.4 Desarrollo de un modelo cliente/servidor

Para la implementación de un modelo cliente servidor se ha usado las librerías de programación con sockets, los cuales ya han sido mencionados anteriormente, a través de la cual una aplicación pueda enviar y recibir información.

Un socket está definido por:

- La dirección IP (32 bits) que consiste en el direccionamiento a nivel de red, la cual se debe conocer del equipo donde se ejecuta el proceso.
- Los puertos, que son los identificadores usados para asociar los datos entrantes a un proceso específico de la máquina.

Existen dos modelos, según el protocolo de transporte que se vaya a utilizar, por un lado, tenemos los sockets TCP, que están orientados a conexión (stream sockets), y por otro lado los UDP, que no están orientados a conexión (datagram sockets), ambos sobre IP.

Ambas conexiones utilizan una serie de primitivas, las cuales se utilizan para establecer un punto de comunicación, para conectarse a una maquina remota en un puerto determinado que se encuentre disponible, para escuchar, leer o escribir, y finalmente desconectarse.

Para la transmisión y recepción de las funciones control remoto se ha usado un tipo de conexión TCP, ya que este tipo de servicio garantiza que no se pierda la información involucrada en el proceso de recibir o emitir información. Los datos se transfieren sin encuadrarlos en registros asegurando de esta manera que los datos lleguen al destino en el orden de transmisión.

La estructura que se lleva a cabo es:

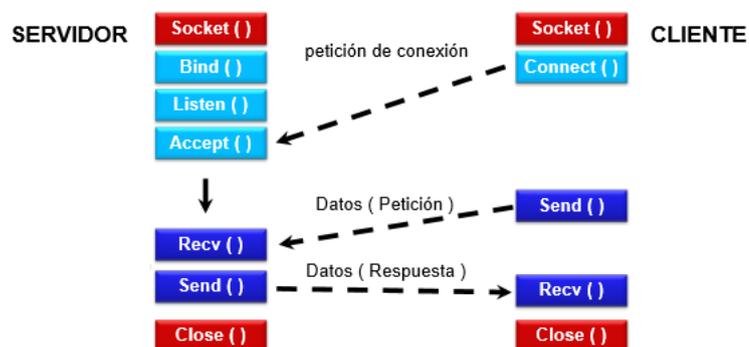


Figura 8 Diagrama estructura TCP.

Donde las principales primitivas en este tipo de servicio son las siguientes:

- Listen: Pone al socket en modo pasivo y crea la cola de espera para almacenar solicitudes de conexión.
- Accept: Espera una solicitud de conexión.
- Connet: Inicia conexión con conector remoto.
- Send: Envía el mensaje.
- Recv: Recibir el mensaje.

Por otro lado, para el streaming de video, se ha usado una conexión UDP, en este tipo de servicio no se establece conexión. Es más eficiente que el servicio TCP, pero no garantiza la fiabilidad ya que los datos se envían y reciben en paquetes independientes, cuya entrega no está garantizada. Los paquetes se pueden perder, duplicar o llegar en un orden distinto al que se envió.

Para este caso en el que se requiere una transmisión de datos en tiempo real, es más eficaz el uso de datagramas que, aunque no garantizan un servicio óptimo, es bastante útil para no perder tiempo en la verificación de datos. Además de ser un servicio menos complejo y de menor sobrecarga.

La estructura que se realiza es la siguiente:

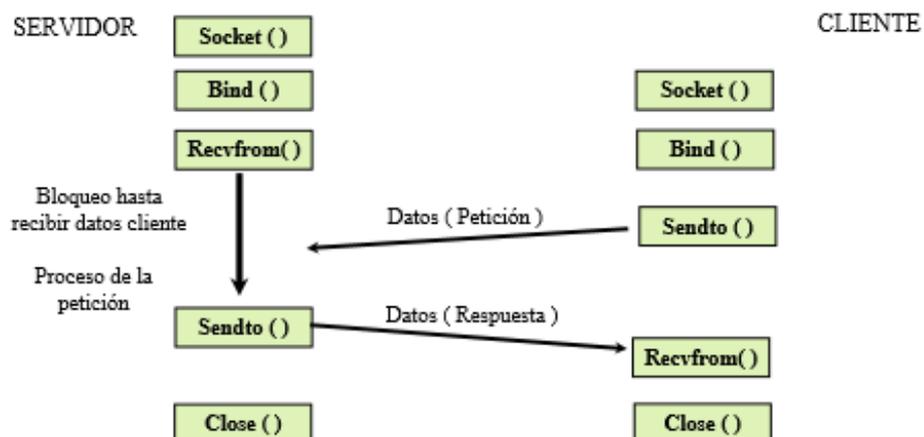


Figura 9 Diagrama del servicio UDP

UDP emplea las siguientes primitivas:

- Socket: este crea el descriptor de socket.
- Close: cierra el socket.
- Bind: asocia una dirección local con un socket.
- Sendto: envía un mensaje.
- Recvfrom Recibe el mensaje.

5.3 Control del dispositivo de movimiento de la cámara de forma remota.

Para la realización del control remoto, se ha implementado un protocolo de aplicación sencillo sobre TCP compuesto por las primitivas para la creación, vinculación y destrucción de sockets en este tipo de servicio que son:

- Socket: Crea un descriptor de socket. Toma como parámetro la familia de protocolos y el tipo de servicio, en este caso "STREAM".
- Listen: Pone al socket en modo pasivo y crea la cola de espera para almacenar solicitudes de conexión.
- Close: cierra la conexión y libera el socket.
- Bind: Asocia una dirección local con un socket.
- Sendto: Envía el mensaje.
- Recvfrom: Recibe el mensaje.
- Accept: Espera una solicitud de conexión.
- Connet: Inicia conexión con conector remoto.

Este protocolo básicamente una vez establecida la conexión, el servidor se mantiene a la espera de que el cliente le mande el tipo de movimiento que desee realizar.

Se ha empleado como dispositivo el servomecanismo TrackerPod, utilizando para ello el uso de las librerías propias que ofrece este tipo de mecanismo, en concreto dos:

- trackerpod_md.lib.
- trackerpod_mt.lib.

La implementación y el diseño del movimiento del servomecanismo se realizada con las librerías anteriormente citadas, las cuales engloban las funciones y estructuras que

son necesarias para el desarrollo del movimiento. Además, tanto en la aplicación cliente como servidor se ha de incluir “TrackerPod.h”.

Para la parte del servidor, inicialmente se ha realizado una llamada y una creación del objeto CTrackerpod, para posteriormente realizar una inicialización del dispositivo y poder ver las características de este, como son el nombre, versión, ID y descripción. Una vez inicializado el dispositivo, se ha realizado un testeo inicial del Trackerpod, para comprobar su correcto funcionamiento.

El servidor tiene asociados dos tipos de comandos para realizar el movimiento, “MOVE_TO” y “MOVE_BY”, estos están asociados a unas coordenadas que serán mandados desde la aplicación cliente. El servidor únicamente lo que realiza es permanecer a la escucha de que la aplicación cliente le mande por una petición el tipo de movimiento que quiere realizar, el servidor lo recibe comprueba que es correcto y procede a realizar el movimiento que el cliente le ha solicitado. La diferencia que hay entre los dos comandos citados anteriormente es el tipo de movimiento que realiza cada uno. MOVE_TO tiene asociado el comando “reset” mediante el cual TrackerPod vuelve al punto de inicio (0,0) en el eje de coordenadas, independientemente de la posición actual en la que este se encuentre. Por otro lado, MOVE_BY el cual explicaremos mediante un plano cartesiano, para una mejor comprensión.

El plano cartesiano está compuesto por dos rectas numéricas, a la izquierda del origen, en el eje de abscisas, se encuentran los valores negativos, y a la derecha los valores positivos. En el eje de ordenadas, del origen hacia arriba se encuentran los valores positivos, y hacia abajo, los valores negativos.

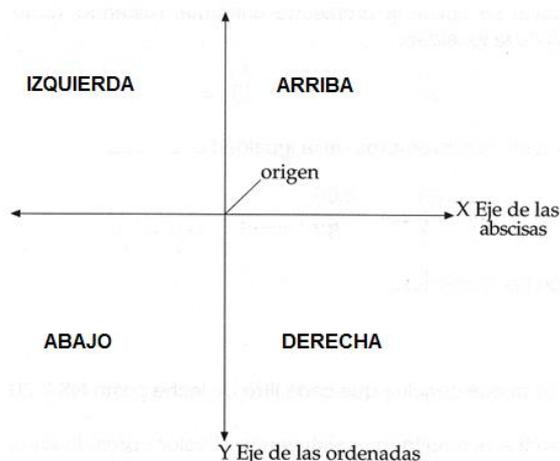


Figura 10 Eje de Coordenadas

Arriba: Abscisa positiva y ordenada positiva.

Izquierda: Abscisa negativa y ordenada positiva.

Abajo: Abscisa negativa y ordenada negativa.

Derecha: Abscisa positiva y ordenada negativa.

Un punto en el plano se localiza con una pareja ordenada de valores (x, y), llamadas coordenadas. La primera componente (x) se localiza en el eje de las abscisas, y la segunda componente (y), se localiza en el eje de ordenadas.

Si utilizamos el comando “salto 50 60”, la primera componente (50) se localiza en el eje de abscisas y la segunda (60) en el eje de ordenadas.

Para que el usuario pueda realizar el control remoto existen dos alternativas claramente diferenciadas:

- En la primera alternativa el usuario debe introducir los comandos anteriormente citados (izquierda, derecha, arriba, abajo, reset, salto), para así poder desplazar el TrackerPod sobre el eje cartesiano de manera ininterrumpida hasta un máximo de 40 unidades, que han sido previamente fijadas.
- En la segunda alternativa se da la posibilidad de que el usuario pueda elegir el número de unidades que desee mover el TrackerPod. Si por ejemplo el usuario desea mover Trackerpod hacia la izquierda con un ángulo reducido, basta con que este introduzca por teclado izquierda (N), siendo “N” el número de unidades de desplazamiento que en este caso sería un valor pequeño, por ejemplo 5.

Finalmente, para la implementación del control remoto utilizando JavaScript se ha establecido mediante PHP una conexión hacia el servidor de C++ de Trackerpod donde se encuentra definido cada tipo de movimiento a realizar, esto se encuentra definido en el script “Enviar_formulario.php”. En el script “Camara.php”, dependiendo del tipo de movimiento que se quiera realizar se escoge el tipo de flecha, esta se encuentra asociada a una función en JavaScript que manda el comando que se quiere enviar haciendo una llamada al script “Enviar_formulario.php”.

5.4 Streaming de video o transmisión de video en tiempo real.

Para la realización del streaming de video se ha utilizado como dispositivo una webcam y se han hecho uso de las librerías de video de OpenCV.

OpenCv tiene una estructura modular, por lo que se deben de incluir distintas bibliotecas compartidas o estáticas en nuestra aplicación, dependiendo de la función que queramos desempeñar.

La estructura que se realiza para llevar a cabo la captura de imagen es la siguiente:

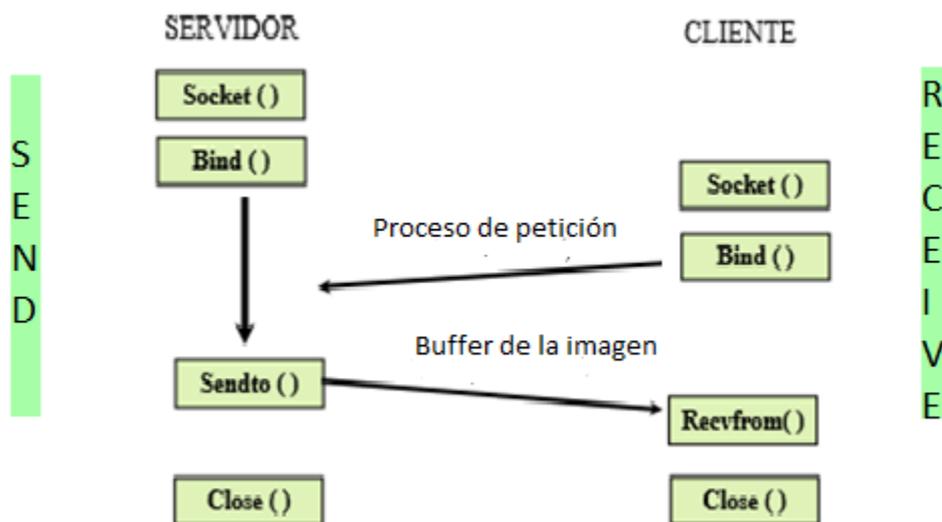


Figura 11 Diagrama Transmisión de video UDP

La implementación y diseño de la captura de video se hace uso de las bibliotecas HighGUI, CV y CXCORE las cuales engloban las funciones y las estructuras que son necesarias para realizar la captura de video la comprensión y descompresión de imágenes en JPG.

Tanto la captura de video, como su posterior visualización se ha desarrollado de manera independiente, es decir, se procede a la captura de la imagen en la aplicación servidor y se envía el video de imágenes comprimidas, en el formato citado anteriormente a la aplicación cliente. Además, se tiene en cuenta que al estar usando el protocolo de transporte UDP hay un límite en el tamaño de los datagramas, por lo que se debe de controlar el buffer de las imágenes y no sobrepasar dicho límite.

Para el servidor streaming llamado “sercam”, se ha creado un socket en la función send donde se indica la dirección del cliente para poder enviar la información del video. Una vez implementado el socket se procede a la captura de video y así poder enviar las imágenes por medio de la primitiva send to, asegurándonos del que buffer no sobrepase un determinado tamaño que hemos fijado.

El cliente en la función receive implementa la descripción del socket, y haciendo uso de la primitiva recvfrom recibe dicha información, rellenando así el buffer de la imagen que se van a ir almacenando por medio de la función imdecode haciendo uso en una estructura del tipo Mat, y posteriormente con la función cv_load_image_color, mostramos la imagen final descomprimida.

Utilizando el tipo de clase videocapture, podemos comenzar a capturar desde nuestra webcam. Le pasamos el parámetro con el valor dos, si solo hay una única cámara, por defecto se le asigna el valor 0, pero puede ocurrir que haya varias cámaras conectadas, por lo que se le asignara su valor correspondiente, o se le puede asignar su nombre específico. La información que proviene de la webcam se almacena en VideoCapture, que se emplea para la lectura de archivos y secuencias de imágenes.

Para que la información recogida en la webcam quede almacenada y después puedan ser procesadas las imágenes recogidas por ella se hace uso de la estructura CV::Mat. Es muy importante ya que las funciones de OpenCv requieren que los tamaños de las imágenes de la fuente y el destino coincidan.

- CV::Mat: representa una matriz n-dimensional, puede usarse para almacenar vectores y matrices reales o complejos, imágenes de escala de grises o de color, volúmenes, etc.

Con la función resize, lo que hacemos es preparar la imagen, para posteriormente realizar el recorte de la imagen, recalándola al valor que nosotros queramos, en este caso (300*300).

Para poder ejecutar tanto el servicio de movimiento, como de streaming se ha creado una aplicación cliente multihebra, de esta manera cada tarea se ejecuta de manera independiente dentro del sistema, dejando por separado los servidores de movimiento y cámara ya que el streaming puede llegar a más de un usuario, pero el control de la cámara sea para un solo usuario.

Para crear una hebra la API de Windows proporciona la función CreateThread. Cada hebra cuenta con una pila propia, a la cual se le puede especificar el tamaño y llamar a la función que queremos que la hebra ejecute, por lo que para ejecutar el video se llama a la función correspondiente para la recepción del video.

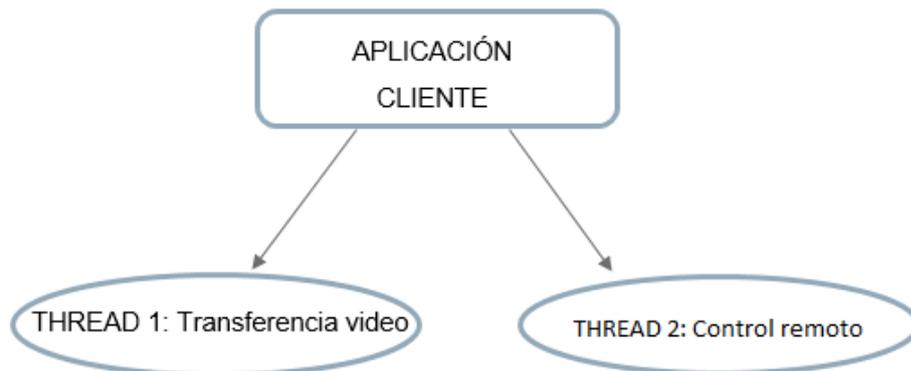


Figura 12 Esquema hebras Cliente

A continuación, en la figura 13 se muestra una imagen con el resultado final de dicha aplicación.

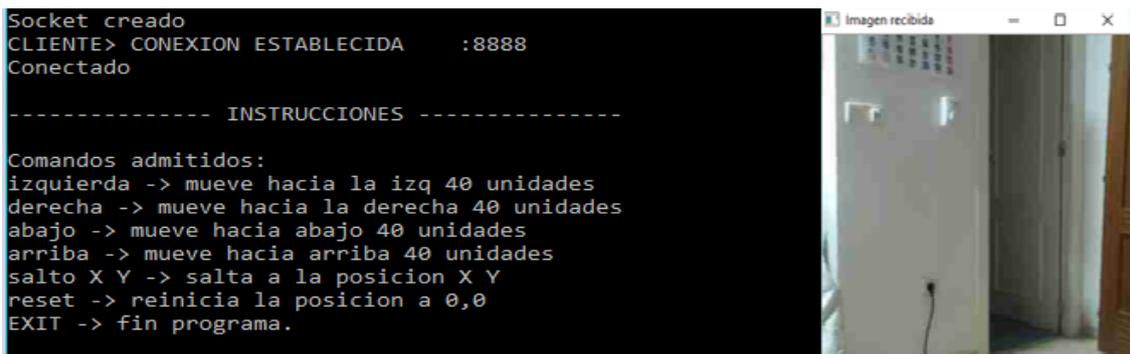


Figura 13 Aplicación Cámara

5.4.1 Descripción de la alternativa de Streaming.

Como alternativa para poder realizar el streaming de video, se ha escogido la utilización de VLC Media Player, es un reproductor gratuito multiplataforma que permite reproducir audio y video sin necesidad de instalar códec adicionales en el equipo Este software nos da la posibilidad de realizar el streaming de video y audio a los equipos disponibles. Además, VLC permite realizar el envío de streaming con o sin transcodificación, variar el tamaño del video y escoger el tipo de protocolo entre otros.

VLC Media Player nos da la posibilidad de escoger el tipo de protocolo del destino al cual queremos mostrarle la emisión, como puede ser HTTP, MMSH, RTSP, RTP, UDP e IceCast. En nuestro caso el destino será un cliente HTTP, que mostrará el contenido utilizando el servicio Web “Xampp”, el cual leerá la emisión empleando el puerto 8095.

Para poder leer el contenido multimedia de la emisión “8095” se ha utilizado un script de JavaScript el cual nos permite leer diferentes fuentes de video compatibles con HTML5 y mostrarlo en un servidor Web que sea compatible con PHP. Esto ha quedado integrado en la web en la que podemos realizar el control remoto del TrackerPod quedando todo ello integrado para poder ver la emisión y mover la cámara cuando se desee.

VLC, nos muestra una amplia gama de codificadores que podemos emplear para realizar nuestro streaming de video y ver cual se adapta mejor a las capacidades de nuestro equipo para que el streaming sea de la mayor calidad posible. Los códec que nos ofrece VLC Media Player son: MPEG-1, MPEG-2, MPEG-4, DivX, MP3, OGG, MOV, RAM, AVI, FLV, entre otros. Entre las distintas pruebas que se han realizado y el codificador que muestra un menor retardo y mejor calidad de la imagen emitida se ha escogido el OGG.

OGG es un contenedor que soporta tanto códec de audio como de video. Disponemos de OGG Vorbis que codifica audio, mientras que OGG Theora codifica video.

OGG Vorbis, es un formato de compresión de datos de audio desarrollado por Xiph.org. Este formato reduce alguno de los datos de audio. Esto quiere decir que se elimina alguno de los datos de audio, como por ejemplo frecuencias inaudibles, con el fin de obtener el mayor grado de compresión posible y así generar un archivo de salida que suene lo más parecido al original.

OGG Theora, es un codec de video de propósito general con bajo consumo de CPU. Está basado en el codec de video VP3, pero amplía las características con respecto a las disponibles en este códec. Theora, es un método de compresión de video con pérdidas. El video comprimido puede ser almacenado en cualquier contenedor multimedia, pero generalmente se encuentra en el formato contenedor OGG que es frecuentemente usado con el formato de audio Vorbis.

A continuación, se muestra el escenario que se ha creado para la realización de HTTP Streaming, y el cliente HTTP con la respuesta en la figura 14. La configuración para realizar la emisión con VLC se muestra en el anexo 8.3.

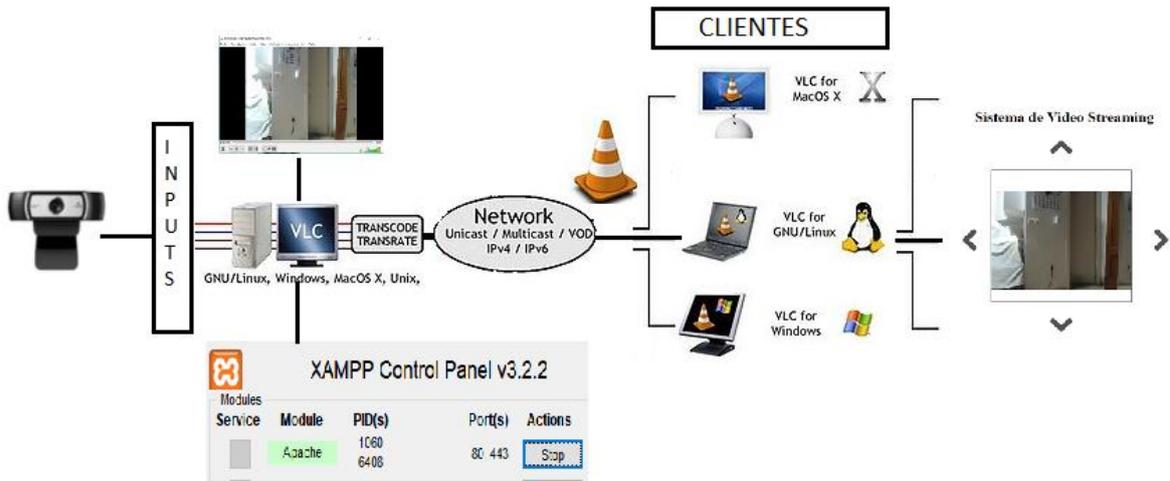


Figura 14 Esquema Streaming HTTP

6. Pruebas y resultados

En las siguientes páginas se recogen las pruebas realizadas para el servicio de streaming de video en diferentes condiciones y valorando así los resultados obtenidos.

Los parámetros de QoS con los que vamos a trabajar son:

- Tasa de error
- Delay
- Ancho de banda
- Jitter

A continuación, haremos una breve descripción de cada uno de ellos:

DELAY

El delay o retardo es un problema general de las redes de telecomunicaciones consistente en el tiempo que tarda un paquete en llegar desde la fuente al destino.

Las comunicaciones full-duplex y en tiempo real son sensibles a este efecto, que se encuentra en enlaces lentos o congestionados y el retardo entre el punto inicial y final de la comunicación debe de ser inferior a 150 ms.

No hay una solución que podamos implementar de una manera sencilla. Mucha de las veces depende de los equipos por los que pasan los paquetes, es decir, de la propia red. Se puede intentar reservar un ancho de banda de origen a destino o señalar los paquetes con valores de TOS (Type Of Service) para intentar que los equipos sepan que se trata de tráfico en tiempo real y lo traten así con mayor prioridad.

JITTER

El jitter es un efecto de las redes de datos no orientadas a conexión y basadas en conmutación de paquetes. La información se discretiza en paquetes y cada uno de ellos puede seguir una ruta distinta hasta el destino. El jitter es la variación en el tiempo en la llegada de los paquetes, causada por la pérdida de sincronización, congestión de red o por las diferentes rutas seguidas por los paquetes para llegar al destino. El jitter entre el punto inicial y final de la comunicación debería ser inferior a 100ms. Si es menor a este valor puede ser compensado, en caso contrario, no.

ANCHO DE BANDA

En las redes de ordenadores, el ancho de banda a menudo se utiliza como sinónimo para la tasa de transferencia de datos que se puedan llevar de un punto a otro en un período dado (generalmente un segundo). Se expresa generalmente en bits (de datos) por segundo (bps). En ocasiones, se expresa como bytes por segundo (Bps).

En general, una conexión con ancho de banda alto es aquella que puede llevar la suficiente información como para sostener la sucesión de imágenes en una presentación de video.

Debe recordarse que una comunicación consiste principalmente en una sucesión de conexiones, cada una con su propio ancho de banda. Si una de estas conexiones es mucho más lenta que el resto actuará como cuello de botella enlenteciendo la comunicación.

6.1 Metodología.

En este apartado se van analizar el efecto del uso de distintos protocolos en las cuatro variables analizadas anteriormente y en base al cual se pueden extraer conclusiones. Además, se va a analizar el efecto que estos resultados pueden tener en el uso de códec distintos y de resoluciones o tamaños de imagen diferentes, para lo cual se cambiaran los parámetros de configuración de VLC por un lado y del código implementado para transmitir el streaming directamente sobre UDP.

Para la realización de las siguientes pruebas y su posterior valoración hay que tener en cuenta los equipos empleados, ya que de ellos va a depender los resultados obtenidos.

El primer factor que se ha de tener en cuenta para un rendimiento y funcionamiento correcto del streaming y control remoto, son los equipos empleados, ya que para realizar las pruebas se han usado dos equipos con características diferentes.

TIPO	EQUIPO 1	EQUIPO 2
HARDWARE	Intel (R). Core i5-4200 CPU 2.3GHz RAM 4.00GB Disco duro 500GB	Intel (R). Core i7-3630QM CPU 3.6 RAM 16.00GB Disco duro SSD 250GB
SOFTWARE	Windows 10 VLC OpenCV 3.1.0	Ubuntu 16.4 VLC OpenCV 3.1.0

Tabla 2 Comparativa de Equipos empleados

Otro de los factores que se ha de tener en cuenta es el tipo de conexión en red. Para la alternativa de OpenCV se ha empleado un cable cruzado para realizar la conexión entre ambos equipos por el puerto Ethernet, proporcionando como velocidad máxima de 1Gbps. Mientras que para el uso de VLC se ha empleado una conexión inalámbrica.

6.2 Experimentos

Para la comparativa de las distintas alternativas de streaming se ha empleado distintos tipos de codificación para poder observar como son las variaciones de tiempo, tasa de error, ancho de banda y jitter.

Para ello se ha hecho uso del software Wireshark, este es un sniffer de red el cual nos da la posibilidad de ver todo el tráfico que está pasando por la red, pudiendo filtrar a todos los niveles de la torre de protocolos TCP/IP.

Para la realización de los experimentos lo clasificaremos en dos apartados:

1. Medidas del efecto de usos en distintos protocolos:

En este apartado las pruebas que se han realizado son similares en el caso de la transmisión con HTTP y UDP. Con VLC se puede trabajar no solamente con transmisión tipo HTTP, sino que también se puede realizar con RTP sobre UDP entre otros, por lo que analizaremos el efecto.

A nosotros nos interesa especialmente ver el streaming de video por ello filtraremos el trafico RTP, pudiendo analizar directamente diferentes parámetros en la transmisión, como son jitter, delay, ancho de banda o la tasa de error.

En la primera prueba se ha analizado en Wireshark utilizando una red inalámbrica el jitter de los paquetes RTP, como se muestra en la figura 15.

	Packet	Sequence	Delta (ms)	Jitter (ms)	Skew
Forward					
	256	4894	9.47	13.57	-22.93
SSRC 0x9a89c316	258	4895	6.96	13.15	-29.89
Max Delta 44.63 ms @ 8771	259	4896	7.99	12.83	-37.88
Max Jitter 18.30 ms	260	4897	8.38	12.55	-46.26
Mean Jitter 14.93 ms	264	4898	8.59	12.30	-54.85
Max Skew 117.21 ms	266	4899	6.56	11.95	-61.41
RTP Packets 4861	267	4900	7.99	11.70	-69.41
Expected 4861	269	4901	7.28	17.40	33.56
Lost 0 (0.00 %)	271	4902	8.88	16.87	24.68
Seq Errs 0	272	4903	7.58	16.29	17.10
Duration 37.80 s	275	4904	7.49	15.74	9.60
Clock Drift 86 ms	276	4905	8.03	15.26	1.57
Freq Drift 44201 Hz (0.23 %)	277	4906	7.36	14.76	-5.79
Reverse	280	4907	7.77	14.33	-13.56
	283	4908	7.99	13.93	-21.55
SSRC 0x9a89c316	284	4909	8.09	13.57	-29.64
Max Delta 0.00 ms @ 0	287	4910	7.94	13.22	-37.59
Max Jitter 0.00 ms	298	4911	7.98	12.89	-45.57
Mean Jitter 0.00 ms	306	4912	8.00	12.58	-53.58
Max Skew 0.00 ms	308	4913	7.21	12.25	-60.79
RTP Packets 0	309	4914	7.58	11.96	-68.37
Expected 1	310	4915	7.24	17.61	33.98
Lost 1 (100.00 %)	312	4916	7.68	16.98	26.30
Seq Errs 0	315	4917	9.66	16.53	16.64
Duration 0.00 s					

Figura 15 Jitter

En la figura 16, se muestra el ancho de banda consumido al realizar el streaming de video por medio del protocolo RTP, siendo este un ancho de banda de 1.5Mbps.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	4898	100.0	6940343	1468 k	0	0	0
Ethernet	100.0	4898	1.0	68572	14 k	0	0	0
Internet Protocol Version 4	100.0	4898	1.4	97960	20 k	0	0	0
User Datagram Protocol	100.0	4898	0.6	39184	8392	0	0	0
Real-Time Transport Protocol	100.0	4935	97.2	6744358	1427 k	4861	6724896	1423 k
MicroBlast Neighbor Discovery Protocol	0.0	37	0.1	9731	2039	0	0	0

Figura 16 Ancho de Banda Wireshark red inalámbrica.

A continuación, se muestra en la figura 17 y 18 la tasa de error y el delay, siendo el valor de la tasa de error igual a 0, ya que nos situamos en una red local por lo que la pérdida es nula, por otro lado, la media de reenvío de paquetes está en torno a 0.4369 segundos. Esta media se ha calculado con aproximadamente 10 paquetes en los cuales se han encontrado la marca de tiempo.

Expected	4861
Lost	0 (0.00 %)
Seq Errs	0
Duration	37.80 s
Clock Drift	86 ms
Freq Drift	44201 Hz (0.23 %)

Figura 17 Tasa de error

Time	delta
6 0.079921	0.516884000

Figura 18 Delay Red Inalámbrica

A continuación, se va a mostrar los parámetros anteriores, pero empleando para ello una red cableada:

En la figura 19, podemos observar que el valor del jitter es igual a 0.00. Esto es debido a que se está empleando una red cableada, por lo que la desviación de la media del retardo es casi nula.

Packet	Sequence	Delta (ms)	Jitter (ms)	Slew	Bandwidth	Marker	Status
2	42107	0.00	0.00	0.00	11.42		✓
3	42108	0.00	0.00	0.00	22.85		✓
4	42109	0.00	0.00	0.00	34.27		✓
5	42110	0.00	0.00	0.00	45.70		✓
6	42111	0.00	0.00	0.00	57.12		✓
7	42112	0.00	0.00	0.00	68.54		✓
8	42113	0.00	0.00	0.00	79.97		✓
9	42114	0.00	0.00	0.00	91.39		✓
10	42115	0.00	0.00	0.00	102.82		✓
11	42116	0.00	0.00	0.00	114.24		✓
12	42117	0.00	0.00	0.00	125.66		✓
13	42118	0.00	0.00	0.00	137.09		✓

Figura 19 Jitter (red cableada)

A continuación, en la figura 20 se muestra una captura donde se ve el ancho de banda consumido a través de la red cableada, siendo este un ancho de banda de 0.6 Mbps.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	4153	100.0	5772580	689 k	0	0	0
Ethernet	100.0	4153	1.0	58142	6948	0	0	0
Internet Protocol Version 4	100.0	4153	1.4	83060	9925	0	0	0
User Datagram Protocol	100.0	4153	0.6	33224	3970	0	0	0
Real-Time Transport Protocol	103.2	4287	97.6	5633396	673 k	4019	5562912	664 k
Mikrotik Neighbor Uncovery Protocol	3.2	134	0.0	33242	4211	0	0	0
Malformed Packet	3.2	134	0.0	0	0	134	0	0

Figura 20. Ancho de banda (red cableada)

Por ultimo en la figura 21 y 22 se muestran los valores de la tasa de error y delay, siendo la perdida de paquetes de un 3.53% y 5.708 ms respectivamente, este retardo se ha calculado igual que para la red inalambrica.

Expected	4166
Lost	147 (3.53 %)
Seq Errs	10
Duration	60.88 s
Clock Drift	0 ms
Freq Drift	1 Hz (0.00 %)

Figura 21 Tasa de error

o.	Time	delta
186...	0.004403	0.010111000

Figura 22 Delay Red Cableada

A continuación, mostramos la pérdida de un paquete ya que el paquete con numero de secuencia 42154 no se ha recibido, por lo que el 42155 (marcado en rojo) nos da un aviso informando que se ha roto la secuencia de números (Ver Figura 23).

		Forward	Reverse	Graph					
		Packet	Sequence	Delta (ms)	Jitter (ms)	Skew	Bandwidth	Marker	Status
Forward		35	42140	0.00	0.00	0.00	388.42		✓
SSRC	0x2446acc1	36	42141	0.00	0.00	0.00	399.84		✓
Max Delta	0.00 ms @ 0	37	42142	0.00	0.00	0.00	411.26		✓
Max Jitter	0.00 ms	38	42143	0.00	0.00	0.00	422.69		✓
Mean Jitter	0.00 ms	39	42144	0.00	0.00	0.00	434.11		✓
Max Skew	0.00 ms	40	42145	0.00	0.00	0.00	445.54		✓
RTP Packets	4019	41	42146	0.00	0.00	0.00	456.96		✓
Expected	4166	42	42147	0.00	0.00	0.00	468.38		✓
Lost	147 (3.53 %)	43	42148	0.00	0.00	0.00	479.81		✓
Seq Errs	10	44	42149	0.00	0.00	0.00	491.23		✓
Duration	60.88 s	45	42150	0.00	0.00	0.00	502.66		✓
Clock Drift	0 ms	46	42151	0.00	0.00	0.00	514.08		✓
Freq Drift	1 Hz (0.00 %)	47	42152	0.00	0.00	0.00	525.50		✓
Reverse		48	42153	0.00	0.00	0.00	536.93		✓
SSRC	0x2446acc1	70	42155	0.00	0.00	0.00	548.35	Wrong sequence number	✗
Max Delta	0.00 ms @ 0	71	42156	0.00	0.00	0.00	559.78		✓
Max Jitter	0.00 ms	72	42157	0.00	0.00	0.00	571.20		✓
Mean Jitter	0.00 ms	73	42158	0.00	0.00	0.00	582.62		✓
Max Skew	0.00 ms	74	42159	0.00	0.00	0.00	594.05		✓
RTP Packets	0	75	42160	0.00	0.00	0.00	605.47		✓
Expected	1	76	42161	0.00	0.00	0.00	616.90		✓
Lost	1 (100.00 %)	77	42162	0.00	0.00	0.00	628.32		✓
Seq Errs	0	78	42163	0.00	0.00	0.00	639.74		✓
Duration	0.00 s	79	42164	0.00	0.00	0.00	651.17		✓
Clock Drift	0 ms	80	42165	0.00	0.00	0.00	662.59		✓
Freq Drift	1 Hz (0.00 %)	81	42166	0.00	0.00	0.00	674.02		✓

Figura 23 Perdida de paquetes

La segunda alternativa a analizar es mediante la realización de streaming multimedia por medio de HTTP streaming, por lo que el análisis será muy similar al que se ha realizado anteriormente, para ver si los resultados se parecen a los obtenidos con RTP sobre UDP.

Con el uso de HTTP streaming el ancho de banda es de 0.7Mbps, como se muestra en la siguiente figura:

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	8090	100.0	8188206	745 k	0	0	0
Ethernet	100.0	8090	1.4	113260	10 k	0	0	0
Internet Protocol Version 4	100.0	8090	2.0	161800	14 k	0	0	0
Transmission Control Protocol	100.0	8090	96.6	7913146	720 k	7863	7580030	690 k
SPDY	0.0	2	71.8	5879105	535 k	0	0	0
Data	2.8	227	4.0	327028	29 k	227	327028	29 k

Figura 24 Ancho de banda red cableada

La segunda prueba se ha realizado empleando una red inalámbrica para comprobar el ancho de banda que se está consumiendo, en este caso 0.62Mbps.

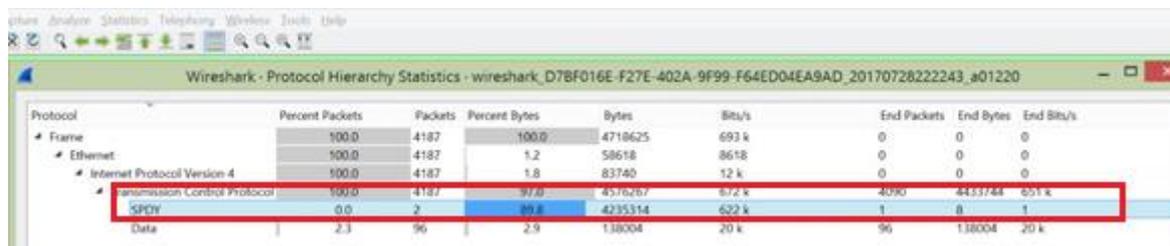


Figura 25 Ancho de banda red inalámbrica.

Para la tasa de error tanto en la red cableada como la inalámbrica, al ser una emisión HTTP sobre TCP, sabemos que los errores serán nulos, debido a que TCP reenviará de manera transparente los posibles paquetes perdidos.

Por último, la forma de comprobar el retardo o delay, se ha realizado a través de un parámetro de TCP llamado "Time Since Request", el cual nos indica el retardo desde que se ha realizado la petición hasta que se ha recibido. Este tiempo solo estará presente en paquetes que necesitan ser reenviados debido a alguna pérdida. En este caso sabemos cuándo hemos pedido el reenvío de paquetes a TCP y cuando hemos recibido de nuevo el paquete.

La media de reenvío de paquetes está en torno a 0.54 segundos. Esta media se ha calculado con aproximadamente 10 paquetes en los cuales se han encontrado la marca de tiempo.

Time	TCP delta
266 0.538850	0.939054000

Figura 26 Time Since Request Red Inalambrica.

Como anteriormente hemos realizado las pruebas empleando una red inalámbrica y una cableada, en el caso de HTTP hemos realizado la misma prueba usando para ello el mismo valor de “Time Since Request”, obteniendo en este caso unos valores significativamente inferiores. Para el cálculo de este hemos tomado 8 paquetes, los cuales han tenido que ser reenviados. La media del tiempo de retardo es de aproximadamente 0.20 segundos.

Time	TCP delta
177 0.073175	0.272276000

Figura 27 Time Since Request Red cableada.

Finalmente, se ha realizado un análisis similar a los anteriores pero esta vez empleando el protocolo UDP, donde se ha hecho uso de OpenCV, enviando directamente frames en formato JPEG, podemos apreciar una notable calidad en la imagen, además de tener un menor retardo, siendo casi instantánea la transmisión. Para comprobar el ancho de banda consumido hemos utilizado la aplicación Wireshark capturando una gran cantidad de tráfico, en la figura 28 podemos ver como el ancho de banda consumido por OpenCV sobre UDP es notablemente superior (~13Mbps), en UDP se hace una compresión rudimentaria tipo MPEG trama a trama por lo que el ancho de banda es superior en comparación con las pruebas realizadas anteriormente con RTP y HTTP, también podemos apreciar una mayor calidad de imagen y como acabamos de mencionar apenas tenemos retardo.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	2043	100.0	775251	96 k	0	0	0
Ethernet	100.0	2043	3.7	28602	3551	0	0	0
Internet Protocol Version 6	0.7	14	0.1	972	120	0	0	0
Internet Protocol Version 4	99.5	2033	5.2	40660	5048	0	0	0
User Datagram Protocol	99.4	2031	2.1	16248	2017	0	0	0
Teredo IPv6 over UDP tunneling	0.2	4	0.0	340	42	0	0	0
Simple Service Discovery Protocol	0.3	7	0.2	1215	150	7	1215	150
Real-Time Transport Protocol	0.2	5	0.0	80	9	5	80	9
QUIC (Quick UDP Internet Connections)	0.5	11	0.5	3579	444	11	3579	444
NetBios Name Service	0.7	15	0.1	750	93	15	750	93
Mikrotik Neighbor Discovery Protocol	6.4	130	3.2	24747	3072	0	0	0
Malformed Packet	6.4	130	0.0	0	0	130	0	0
Link-local Multicast Name Resolution	0.4	9	0.0	198	24	9	198	24
Dropbox LAN sync Discovery Protocol	0.1	3	0.2	1314	163	3	1314	163
Domain Name System	0.2	4	0.0	164	20	4	164	20
data	90.2	1843	14806.3	111685225	13 M	1843	111685225	13 M
Internet Control Message Protocol	0.1	2	0.1	641	79	2	641	79

Figura 28 Ancho de banda Red cableada OpenCV.

La prueba anterior se ha repetido en un escenario diferente, utilizando para ello una red inalámbrica, subjetivamente no hemos notado cambios en la calidad de la imagen o en el retardo, pero como podemos ver en la figura 29 el ancho de banda consumido es de ~ 8.9 Mbps.

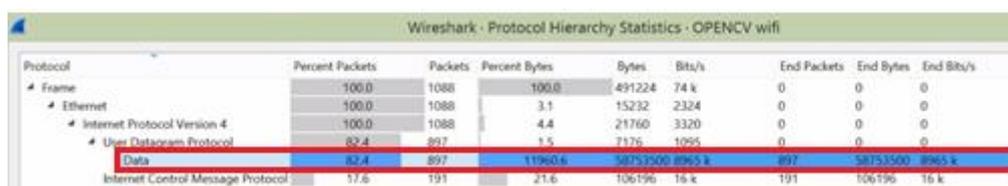


Figura 29 Ancho de banda Red Inalámbrica Opencv.

2. Realización del streaming de video modificando los diferentes parámetros:

En este apartado se van a realizar diferentes pruebas variando la calidad de la imagen y tamaño tanto con OpenCV, como en HTTP streaming.

Con OpenCV vamos a variar la calidad de los parámetros para ver la diferencia subjetiva además del ancho de banda, el parámetro en el que nos centraremos es el que especifica la compresión de imagen.

El parámetro con el que se han realizado las pruebas, es el parámetro de la calidad de compresión de imagen que nos proporciona la librería OpenCV a la hora de transmitir los frames capturados en formato JPEG. Este parámetro se especifica haciendo uso del método `imencode` de OpenCV. Esta función guarda la imagen en el buffer pasándole un valor que asigna para JPEG un nivel de calidad que puede ir de 0 a 100.

En primer lugar, se va a disminuir la calidad de imagen en un 20% con respecto a las pruebas que acabamos de realizar, obteniendo una drástica disminución del ancho de banda pasando de 8.9 Mbps a 0.62 Mbps como podemos ver en la figura 30. Sin embargo, pese a la drástica reducción del ancho de banda no hemos apreciado una notable caída en la calidad de imagen.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	108	100.0	39568	6282	0	0	0
Ethernet	100.0	108	3.8	1512	240	0	0	0
Internet Protocol Version 4	100.0	108	5.5	2160	342	0	0	0
User Datagram Protocol	100.0	108	2.2	864	137	0	0	0
Simple Service Discovery Protocol	0.9	1	0.3	137	21	1	137	21
Mikrotik Neighbor Discovery Protocol	34.3	37	24.6	9731	1544	0	0	0
Malformed Packet	34.3	37	0.0	0	0	37	0	0
Link-local Multicast Name Resolution	3.7	4	0.2	84	13	4	84	13
Data	61.1	66	10925.5	4323000	686 k	66	4323000	686 k

Figura 30 Reducción calidad un 20%

Se ha aumentado de forma drástica la compresión de imagen para observar el ancho de banda consumido y si se apreciaba una notable caída en la calidad de imagen, se ha reducido en un 50% la calidad, siendo mayor la compresión. Como podemos ver en la figura 31 el ancho de banda ha caído hasta los 0.151 Mbps notando en este caso un empeoramiento de la calidad de imagen.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	302	100.0	432722	106 k	0	0	0
Ethernet	100.0	302	1.0	4228	1043	0	0	0
Internet Protocol Version 4	100.0	302	1.4	6052	1494	0	0	0
User Datagram Protocol	4.3	13	0.0	104	25	0	0	0
Simple Service Discovery Protocol	0.3	1	0.0	137	33	1	137	33
Mikrotik Neighbor Discovery Protocol	2.6	8	0.5	2104	519	0	0	0
Malformed Packet	2.6	8	0.0	0	0	8	0	0
Link-local Multicast Name Resolution	0.3	1	0.0	21	5	1	21	5
Internet Group Management Protocol	1.0	3	0.0	36	9	1	36	9
Data	95.7	289	142.2	815412	151 k	289	815412	151 k

Figura 31 Reducción calidad de 50%

Finalmente nos hubiese gustado saber el porcentaje de pérdidas de paquetes, pero debido a que en OpenCV se transmite directamente sobre UDP sin ningún protocolo de control no tenemos forma de saber cuántos paquetes se han podido perder por el camino.

Por último, se ha realizado de nuevo una prueba emitiendo con VLC variando la resolución del video que estamos transmitiendo. Como primera opción hemos puesto una resolución de video baja y el ancho de banda que este consume es de 0.81 Mbps tal como se puede apreciar en la figura 32.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	1230	100.0	1396122	838 k	0	0	0
Ethernet	100.0	1230	1.2	17220	10 k	0	0	0
Internet Protocol Version 4	99.8	1228	1.8	24576	14 k	0	0	0
User Datagram Protocol	1.2	15	0.0	120	72	0	0	0
Transmission Control Protocol	98.3	1209	96.7	13503611	811 k	1204	1347075	809 k

Figura 32 Ancho de banda baja resolución

La segunda variación que se ha realizado con VLC ha sido poniendo una resolución de video de alta calidad, en la figura 33 se puede ver que el ancho de banda consumido es levemente mayor al anterior, en este caso es de 0.84 Mbps, se aprecia un retardo algo mayor pero la calidad de la imagen que se está transmitiendo es bastante buena.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	898	100.0	943210	902 k	0	0	0
Ethernet	100.0	898	1.3	12572	12 k	0	0	0
Internet Protocol Version 4	99.8	896	1.9	17932	17 k	0	0	0
User Datagram Protocol	10.1	91	0.1	728	696	0	0	0
Transmission Control Protocol	89.3	802	94.0	886728	848 k	737	819362	783 k

Figura 33 Ancho de banda alta resolución

A continuación, en la tabla 4 se va a exponer una comparativa mostrando los resultados obtenidos tras analizar los diferentes protocolos de streaming que se han empleado tanto en red cableada como inalámbrica fijándonos en el ancho de banda consumido y el retardo. En el caso de UDP para el retardo, se ha realizado una estimación subjetiva, además los valores distintos de ancho de banda se deben a que en cada sistema se ha usado codecs distintos para la transmisión de video:

RED CABLEADA	ANCHO DE BANDA	RETARDO	RED INALAMBRICA	ANCHO DE BANDA	RETARDO
RTP	0.6 Mbps	5.708 ms	RTP	1.5 Mbps	4369 ms
HTTP	0.7 Mbps	0.20 ms	HTTP	0.62 Mbps	0.54 ms
UDP	13 Mbps	X	UDP	8.9 Mbps	X

Tabla 3 Comparativa de resultados

A modo de conclusión y tras las pruebas realizadas analizando los diferentes protocolos con variaciones de codecs/calidad de imagen, podemos concluir que si disponemos de una buena red la mejor opción sería sin duda utilizar el protocolo UDP empleando OpenCV debido a que nos proporciona la mayor calidad de imagen y el retardo subjetivo es prácticamente nulo, así mismo este no nos permitiría tener parámetros de conexión solo se podría saber el ancho de banda que consumimos. Aun habiendo utilizado en RTP, el protocolo UDP para la transmisión de datos con el servidor de streaming VLC hemos notado una mayor calidad con OpenCV. Con respecto al protocolo HTTP este no es apropiado para el streaming de video ya que al ir sobre TCP y necesitar ACK, si tenemos una pérdida de paquetes el protocolo TCP pedirá el reenvío de este aun no teniendo sentido debido a que el tiempo de visionado de ese momento ya ha pasado, por lo que tendremos aún más ancho de banda consumido y un mayor retardo en la imagen.

7. Conclusiones y líneas de futuro.

En este trabajo se han alcanzado los objetivos propuestos, los cuales engloban el desarrollo del control remoto del servomecanismo TrackerPod, en el que un usuario a partir de la arquitectura cliente/servidor creada mediante la programación con socket, así como la inclusión de la transferencia de video por una webcam, para su posterior visualización con la utilización de las herramientas que proporciona la librería OpenCV.

Además, por otro lado, se ha desarrollado una alternativa para el control remoto haciendo uso de un cliente en PHP que se conecta al servidor en C++ para poder manejarlo correctamente y que posteriormente se puede manejar a través de nuestra web, también se ha incluido el streaming de video empleando para ello un cliente en HTTP que recibe la imagen del servidor VLC Media Player.

A continuación, se detalla el grado en el que se han cumplido los objetivos de partida de este trabajo:

- Posibilidad de controlar el movimiento de una webcam de forma remota usando el soporte motorizado TrackerPod.
- Integración de OpenCV para la transferencia de video en tiempo real, pudiendo escoger la calidad de video que se desee.
- Desarrollo de una alternativa de streaming de video a través de un cliente HTTP Streaming.
- Se ha conseguido mediante la arquitectura cliente/servidor proporcionada por la librería Winsock el control remoto y el streaming de video en tiempo real.
- La posibilidad de que todo el software necesario para la implementación de dicha aplicación sea open source, comprobando además que se adaptan a las necesidades requeridas.

Ante los resultados obtenidos tras el desarrollo del control remoto y el streaming de video en tiempo real, hemos comprobado la infinidad de herramientas que existen actualmente y la gran variedad de recursos que pueden ofrecer.

Finalmente, también se ha podido comprobar que hoy en día existen diferentes comunidades en Internet donde se desarrollan librerías de código abierto para la implementación de nuevas funciones en aplicaciones futuras.

7.1 Líneas de futuro.

Tras la realización de este proyecto aparecen varias líneas de trabajo diferentes a seguir en el futuro. Se enumeran a continuación esas posibles mejoras:

- Una línea de las líneas de futuro que se podría ajustar bastante a este tipo de trabajo, consistiría en la posibilidad de analizar el uso de nuevas tecnologías de distribución de video como node.js y HTML5 que son bastantes utilizados en la actualidad.
- Integración de sensores para la detección de movimiento.
- Analizar cómo hacer el streaming del audio para distribuir también la información capturada por los micrófonos que tienen las webcams.

8. Anexos

8.1 Manual de descarga de programas y librerías.

8.1.1 Microsoft Visual C++ 2010

Se encuentra disponible en la página oficial de Microsoft de manera gratuita (<https://www.microsoft.com/es-es/download/details.aspx>). La instalación de este compilador es muy sencilla, solo basta con hacer “doble clic” en el archivo obtenido en la página oficial una vez se haya descargado.

Como cualquier otro compilador, se pide que se acepte la licencia de uso, y se nos creará una carpeta en la unidad C de nuestro equipo, que es donde se instalará nuestro programa, finalmente, al finalizar el proceso de instalación se nos creará un acceso directo en el menú de inicio de Windows.

8.1.2 TrackedPod y OpenCV.

Para obtener las librerías que se han de emplear con TrackerPod debemos de acceder a la página oficial del dispositivo y descargarnos TrackerPod COM Client: <http://www.trackercam.com/TCamWeb/API.HTM>

Para proceder a la instalación del servicio TrackerPod debemos de seguir los pasos a seguir que se nos muestra en fichero.txt que se encuentra en la carpeta que anteriormente se ha descargado.

Para poder utilizar las librerías de TrackerPod es necesario que en Visual Studio 2010 las vinculemos, esto se hará dirigiéndonos a las propiedades de nuestro proyecto y allí vincular las librerías, para que nuestra aplicación cliente/servidor pueda hacer uso de ellas (Ver figura 34).

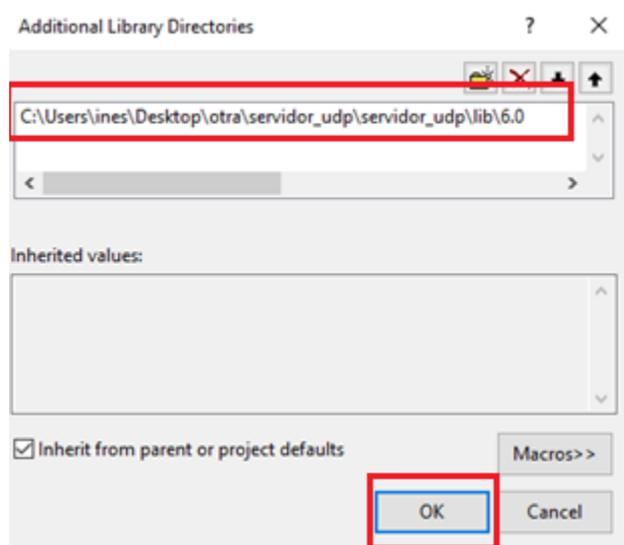


Figura 34 Librerías incluidas en modo Debug

Las librerías de OpenCV también han sido descargadas de su página oficial, una vez descargadas se procede a extraerlas en el disco duro C (Ver Figura 35).

<http://opencv.org/>

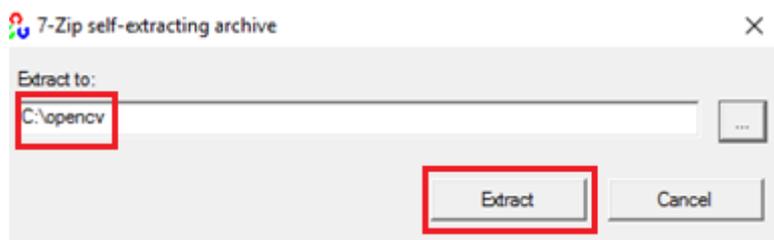


Figura 35 Ubicación de la librería OpenCV

Una vez hallamos extraído las librerías debemos de asegurarnos de que se han agregado las variables de entorno. Para ello debemos de irnos al Panel de Control del sistema y una vez en sistema nos vamos a configuración avanzada del sistema (Ver Figura 36,37 y 38).

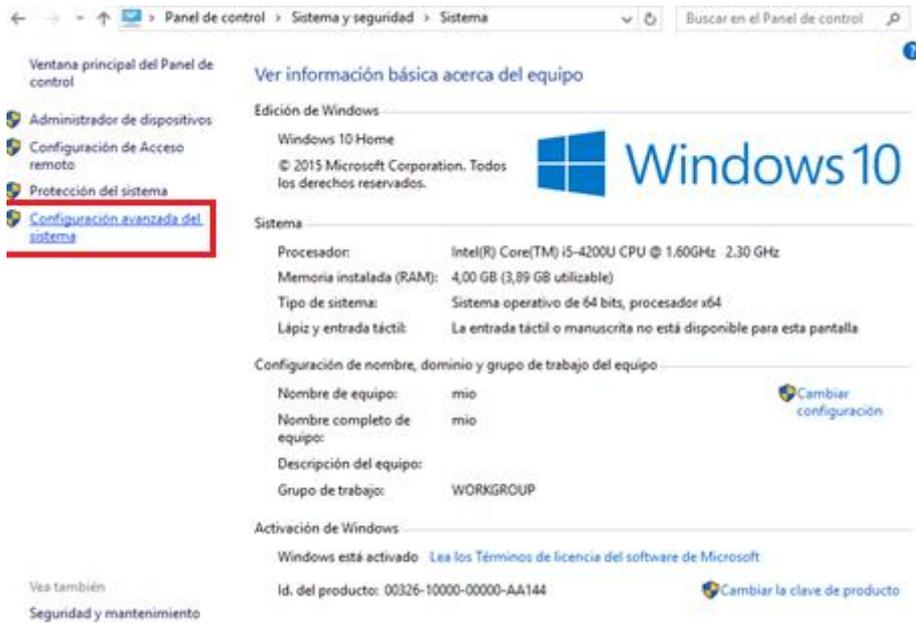


Figura 36 Panel de control del sistema

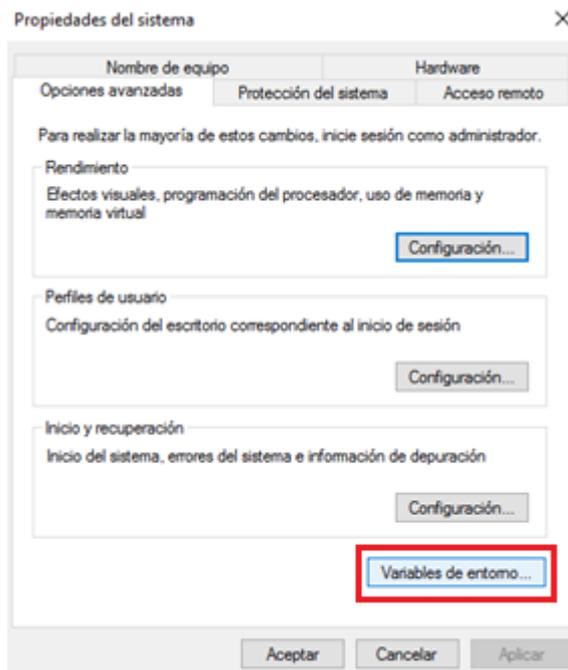


Figura 37 Variables de entorno

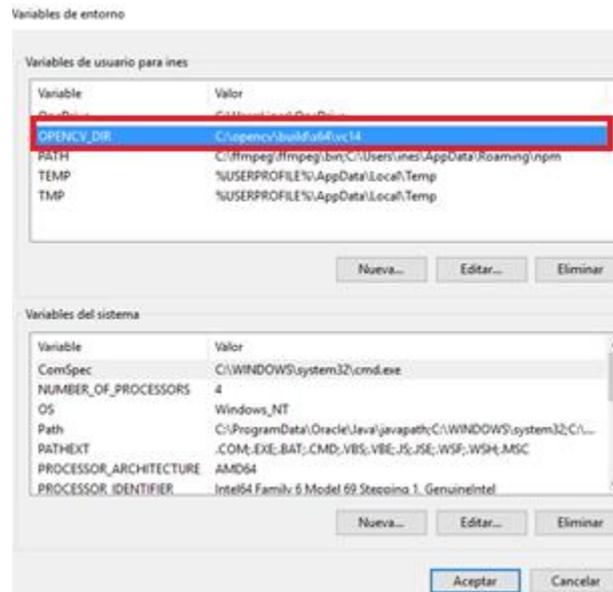


Figura 38 Variables de entorno OpenCV

Una vez que se ha comprobado que las variables de entorno están disponibles procedemos a reiniciar el equipo.

Al igual que hemos realizado con las librerías de TrackerPod, debemos vincular en Visual Studio las librerías propias de OpenCV que anteriormente han sido extraídas para poder hacer uso de ellas (Ver Figura 39).

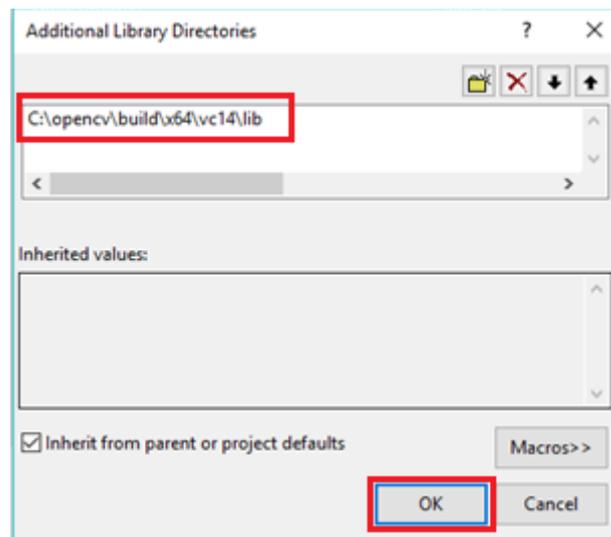


Figura 39 Librerías incluidas modo Release

8.1.3 VLC

Para poder realizar la segunda alternativa de streaming de video necesitaremos la aplicación VLC, esta aplicación podemos descargarla de la página oficial:

<https://www.videolan.org/vlc/index.es.html>

Para su instalación una vez descargado lo único que debemos es ejecutarlo y se instalará automáticamente (Ver Figura 40).

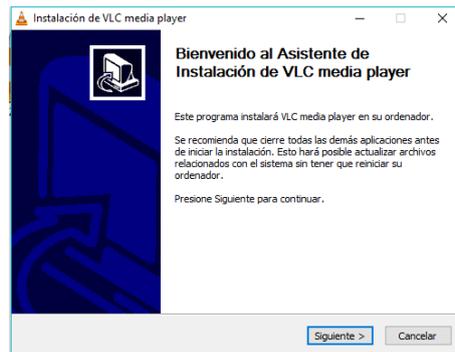


Figura 40 Instalación VLC

8.2 Instalación del servicio TrackerPod.

Para poder inicializar el servicio TrackerPod, debemos de seguir los siguientes pasos para que el funcionamiento sea el correcto:

1. Copiamos el ejecutable "TrackerPodSvcSvr.exe", en nuestro caso en el directorio siguiente C:\Program Files (x86)\Eagletron, si no existe debemos de crearlo (Ver Figura 41).

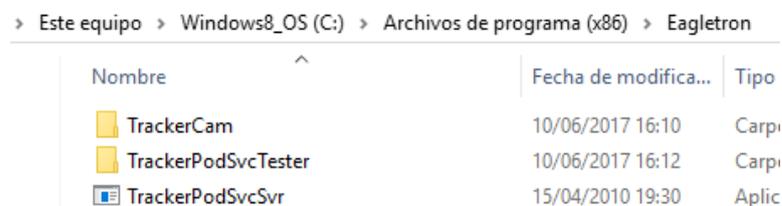


Figura 41 Directorio Servicio TrackerPod

2. Debemos de instalar Visual C++ 2008 SP1 Redistributable, que lo podremos descargar sin ningún problema de la página de Microsoft. El enlace para su descarga sería el siguiente:

<https://www.microsoft.com/es-es/download/details.aspx?id=2092>

3. Nos vamos a símbolo del sistema, lo ejecutamos como administrador y ponemos: `sc create "Eagletron TrackerPod Service"`, con esto lo que hacemos es crear el servicio (Ver Figura 42).

```
C:\WINDOWS\system32>sc create "Eagletron TrackerPod Service"
DESCRIPCIÓN:
    Crea una entrada de servicio en el Registro y en la base de datos
    de servicio.
USO:
    sc <servidor> create [nombre del servicio] [binPath= ] <opción1> <opción2>...
OPCIONES:
Nota: el nombre de la opción incluye el signo de igual.
    Se requiere un espacio entre el signo de igual y el valor.
type= <own|share|interact|kernel|filesys|rec|adapt|userown|usershare>
      (default = own)
start= <boot|system|auto|demand|disabled|delayed-auto>
      (default = demand)
error= <normal|severe|critical|ignore>
      (default = normal)
binPath= <Nombre_de_ruta_binaria_al_archivo_.exe>
group= <Grupo_de_orden_de_carga>
tag= <yes|no>
depend= <Dependencias(separado por / (barra diagonal))>
obj= <Nombre_de_cuenta|Nombre_de_objeto>
      (default = LocalSystem)
DisplayName= <nombre para mostrar>
password= <contraseña>
```

Figura 42 Creación del servicio

4. A continuación, abrimos el registro y buscamos en las carpetas que nos van saliendo los siguientes registros:
 - HKEY_CLASSES_ROOT\AppID\{97E2D3ED-86CB-42AE-845B-8209624B1EC3}.
 - "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDLLs" (para 32 bits).
 - "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\SharedDLLs" (para 64 bits).

A la derecha de las carpetas nos saldrá una entrada en la que tenemos que cambiar "LocalService" a "tpodservice".

Ahora buscaríamos el siguiente directorio:

"C:\Program Files (x86)\Common Files\Eagletron\TrackerPodSvcSvr.exe

Y modificamos el valor del registro sumándole 1 al valor que halla.



Figura 43 Apertura del registro.

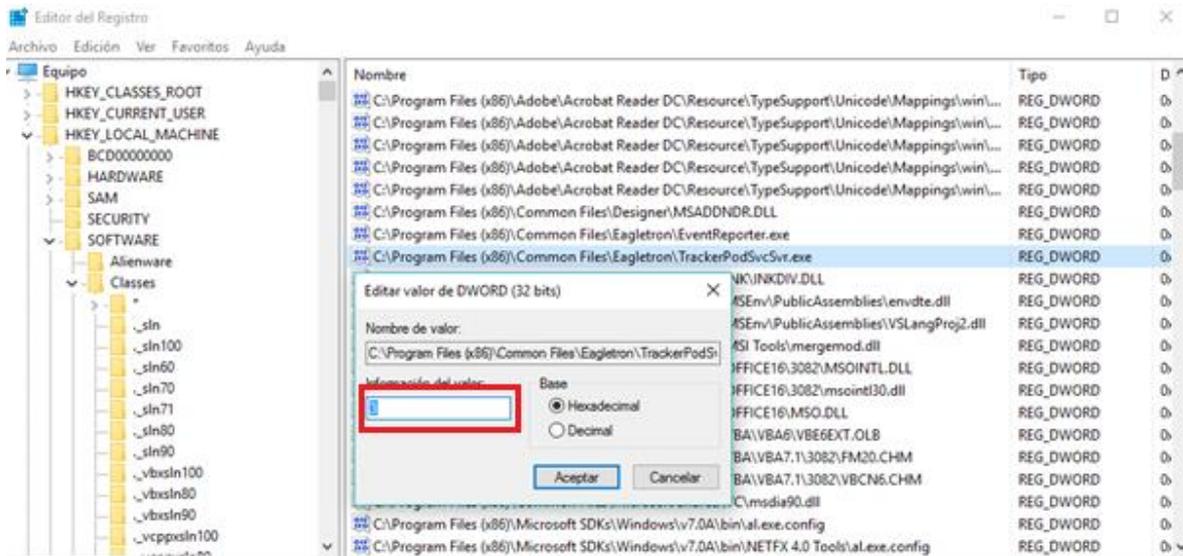


Figura 44 Cambio en los registros

- Finalmente nos vamos a servicios desde el inicio de Windows buscamos Eagletron TrackerPod y lo ponemos con Inicio-Automático (Ver Figura 45).

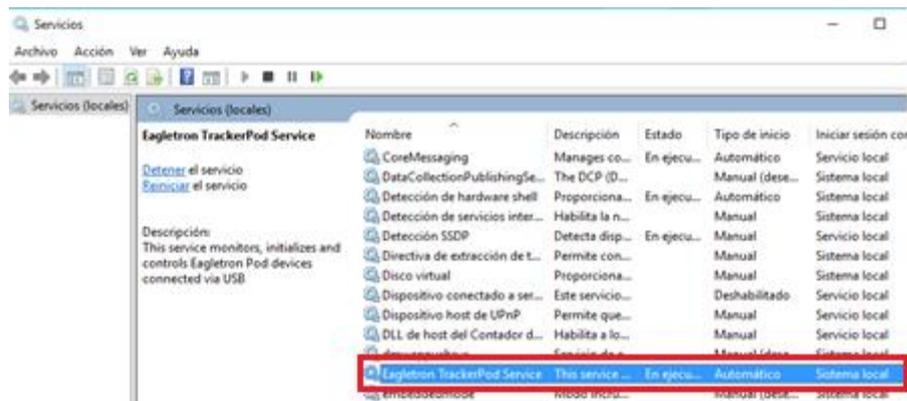


Figura 45 Servicio Automático

Una vez seguidos estos pasos el servicio TrackerPod estará inicializado.

8.3 Streaming VLC vía HTTP

Para realizar streaming en directo mediante VLC y poder mostrarlo a través de un servidor web, los pasos a seguir serán los siguientes:

1. Al finalizar la instalación de VLC, se podrá iniciar el streaming de video accediendo a la pestaña de “Medio” y la opción de Emitir (Ver Figura 46).

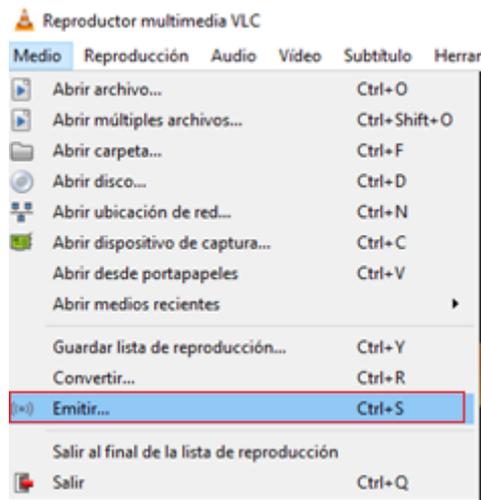


Figura 46 Emitir VLC

2. Una vez seleccionada esta opción, aparece una ventana en la cual se seleccionará el dispositivo de captura, en nuestro caso será Logitech y haremos clic en el botón emitir. (Ver Figura 47)

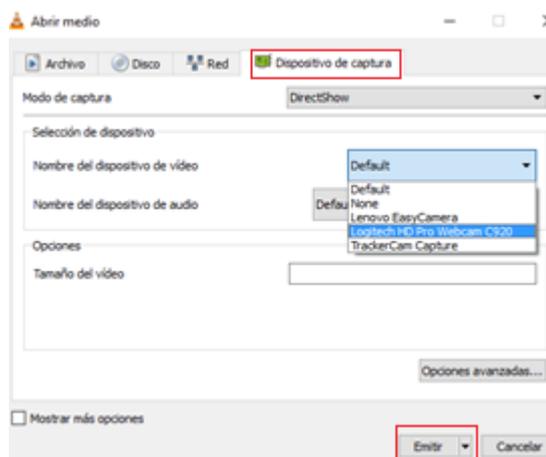


Figura 47 Dispositivo de Captura

3. A continuación, le damos a mostrar en local, si queremos ver en nuestro equipo lo que se está emitiendo en ese momento, añadimos el protocolo HTTP, en nuestro caso configurado con el puerto 8095, y con nombre de la ruta “cam3”. (Ver Figura 48)

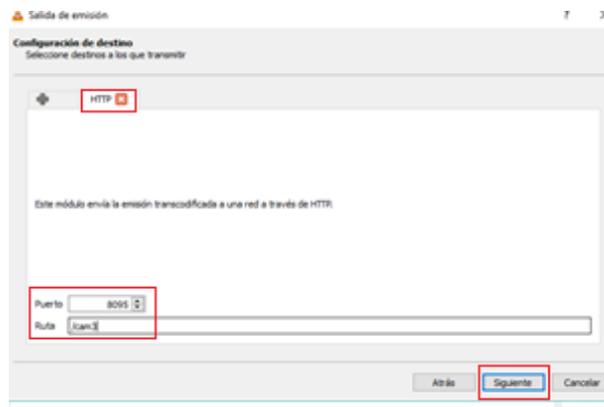


Figura 48 Selección del Protocolo y Puerto

4. Para finalizar, le damos a siguiente y se abre la pantalla de configuración para la transcodificación del contenido a emitir. Dependiendo del material a transmitir se elegirá una opción diferente de la lista desplegable (Ver Figura 49). Para nuestro caso se escoge tipo archivo de video “Theora+Vorbis(OGG)”.

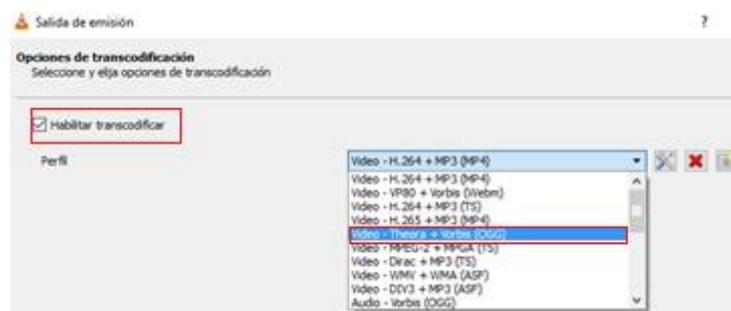


Figura 49 Codificaciones disponibles

8.4 Tabla comparativa de las distintas aplicaciones

Nombre	Plataforma	Edición	Precio	Compatibilidad con compresión		Nº máx. De cámaras	Detecta movimiento	Grabación vídeo	FTP	Sonido
				MJPEG	H.264					
Blue Iris	Windows	Full	46 €	si	si	64	si	si	si	si
Webcam XP	Windows	Free	Gratis	si	no	1	no	no	si	
		Private	45 €			5	no	no	si	
		Pro	89,95 €			Ilimitadas	si	si	si	
IP Camera Viewer	IOS					Ilimitadas			si	
ZoneMinder	Linux		Gratis			Ilimitadas	si	si	si	
IP Cam Viewer	Android	Basic/Lite	Gratis	si	si	?		no	no	
		Pro	3 €	si	si	Ilimitadas		si	si	
Mobiscope	BlackBerry OS		Gratis				si	si		

Tabla 4 Comparativa de aplicaciones

9. Índice de Figuras

Figura 1 Imagen tomada primera cámara	8
Figura 2 Esquema servidor HLS	19
Figura 3 Formato Contenedor	22
Figura 4 Codecs de Video	23
Figura 5. Cámara Logitech C920 HD Pro.....	27
Figura 6. Servomecanismo Trackerpod	28
Figura 7. Sistema de Monitoreo Remoto	29
Figura 8 Diagrama estructura TCP.	37
Figura 9 Diagrama del servicio UDP.....	38
Figura 10 Eje de Coordenadas.....	40
Figura 11 Diagrama Transmisión de video UDP	42
Figura 12 Esquema hebras Cliente.....	44
Figura 13 Aplicación Cámara	44
Figura 14 Esquema Streaming HTTP.....	46
Figura 15 Jitter	50
Figura 16 Ancho de Banda Wireshark red inalámbrica.....	51
Figura 17 Tasa de error	51
Figura 18 Delay Red Inalámbrica	51
Figura 19 Jitter (red cableada)	52
Figura 20. Ancho de banda (red cableada)	52
Figura 21 Tasa de error	52
Figura 22 Delay Red Cableada	53
Figura 23 Perdida de paquetes	53
Figura 24 Ancho de banda red cableada.....	53
Figura 25 Ancho de banda red inalámbrica.	54
Figura 26 Time Since Request Red Inalambrica.....	54
Figura 27 Time Since Request Red cableada.	55
Figura 28 Ancho de banda Red cableada OpenCV.....	55
Figura 29 Ancho de banda Red Inalámbrica Opencv.	56
Figura 30 Reducción calidad un 20%	57
Figura 31 Reducción calidad de 50%	57
Figura 32 Ancho de banda baja resolución	57
Figura 33 Ancho de banda alta resolución	58

Figura 34 Librerías incluidas en modo Debug	63
Figura 35 Ubicación de la librería OpenCV	63
Figura 36 Panel de control del sistema	64
Figura 37 Variables de entorno	64
Figura 38 Variables de entorno OpenCV	65
Figura 39 Librerías incluidas modo Release	65
Figura 40 Instalación VLC	66
Figura 41 Directorio Servicio TrackerPod	66
Figura 42 Creación del servicio	67
Figura 43 Apertura del registro	67
Figura 44 Cambio en los registros	68
Figura 45 Servicio Automático	68
Figura 46 Emitir VLC	69
Figura 47 Dispositivo de Captura	69
Figura 48 Selección del Protocolo y Puerto	70
Figura 49 Codificaciones disponibles	70

10. Índice de Tablas

Tabla 1 TrackercamVS PTZDriver	29
Tabla 2 Comparativa de Equipos empleados	49
Tabla 3 Comparativa de resultados	58
Tabla 4 Comparativa de aplicaciones.....	71

11. Referencias bibliográficas

[1] NAVAJAS, J. Fernández, et al. Evaluación de QoS en accesos a Internet para aplicaciones Multimedia (EQoSIM). Proc. Mundo Internet, 2005, vol. 2, p. 179-185.

[Accessed 1 July 2017].

[2] Códecs de audio y video más utilizados. Informática, tecnología y mucho más [online].

[Accessed 19 July 2017]. <http://www.informatica-hoy.com.ar/aprender-informatica/Codecs-de-audio-y-video-mas-utilizados.php>

[3] LIZETH CORREA, ADMIN AT ADMINISTRADORA DE PAGINAS SEGUIR. Dispositivos de entrada y salida. LinkedIn SlideShare [online]. [Accessed 1 July 2017].

Available from: <https://es.slideshare.net/angielizethsolartecorrea/dispositivos-de-entrada-y-salida-35892809>

[4] Tema 4: Comunicaciones Multimedia. PDF [online]. [Accessed 1 July 2017].

Available from: <http://docplayer.es/1933875-Tema-4-comunicaciones-multimedia.html>

[5] Arquitecturas Basadas en Componentes. Scribd [online]. [Accessed 1 July 2017].

Available from: <https://es.scribd.com/document/51349062/Arquitecturas-Basadas-en-Componentes>

[6] @RMRQUZ, Mgn. Controla la webcam de tu casa desde el móvil con estas aplicaciones. Nobbot [online]. 4 April 2013. [Accessed 18 July 2017].

Available from: <http://www.nobbot.com/redes/controla-la-webcam-de-tu-casa-desde-el-movil-con-estas-aplicaciones/>

[7] Manuela Ruiz Montiel, Javier Ríos Pérez, Jesús Manuel Rodríguez Sánchez, Daniel Héctor Stolfi Rosso [online]. 28 November 2008. [Accessed 18 July 2017].

Available from: <http://www.danielstolfi.com/vigia/archivos/formacion.pdf>

- [8] Curso Desarrollo Videojuegos 3Ed. Scribd [online]. [Accessed 18 July 2017].
Available from: <https://es.scribd.com/doc/282300551/Curso-Desarrollo-Videojuegos-3Ed>
- [9] Tfg b.158 - Documents. Docslide.us [online]. 22 July 2015. [Accessed 19 July 2017].
Available from: <https://docslide.us/documents/tfg-b158.html>
- [10] Gary McGath. Basics of Streaming Protocols [online]. [Accessed 18 July 2017].
Available from: <http://www.garymcgath.com/streamingprotocols.html>
- [11] Telepresencia y conferencias de vídeo. Telepresencia y conferencias de vídeo - Productos de Huawei [online]. [Accessed 1 July 2017]. Available from: <http://e.huawei.com/es/products/enterprise-networking/telepresence-video-conferencing>
- [12] Infotechnology. Como aprovechar la telepresencia. Infotechnology.com [online]. 5 October 2010. [Accessed 19 July 2017].
Available from: <http://www.infotechnology.com/historico/Como-aprovechar-la-telepresencia-20101005-0003.html>
- [13] Multimedia. LENGUAJE DE PROGRAMACIÓN C. LENGUAJE DE PROGRAMACIÓN C [online]. 1 January 1970. [Accessed 1 July 2017].
Available from: <http://lenguajedeprogramacion21.blogspot.com.es/>
- [14] IMECAF México, S.C. El Blog del IMECAF [online]. [Accessed 19 July 2017].
Available from: <http://www.imecaf.com/blog/2012/10/15/que-es-php/>
- [15] Download Transmisión de Datos (3era CLASE) - Documents. documents.mx [online]. [Accessed 19 July 2017]. Available from: <http://documents.mx/download/link/transmision-de-datos-3era-clase>
- [16] Xampp, Desarrollo de Aplicaciones Web. Rafael Menéndez-Barzanallana Asensio. Universidad de Murcia (DIS). Xampp, Desarrollo de Aplicaciones Web. Rafael Menéndez-Barzanallana Asensio. Universidad de Murcia (DIS) [online]. [Accessed 19 July 2017].

Available from: <http://www.um.es/docencia/barzana/DAWEB/Desarrollo-de-aplicaciones-web-Xampp.html>

[17] ¿Qué es el streaming? - ite.educacion.es. [online]. [Accessed 20 July 2017]. Available from:

<http://www.bing.com/cr?IG=4DC6A4059FB2442B903A76028A7C2C68&CID=1A9CBC834B29666D216BB6484A2F6726&rd=1&h=CyfZrd78WxC8wLBV1smKbjQB6SfQWdLBTUWUfRvWDO4&v=1&r=http%3a%2f%2fwww.ite.educacion.es%2fformacion%2fmateriales%2f107%2fcd%2fvideo%2fvideo0103.html&p=DevEx,5064.1>

[18] Multimedia. Multimedia [online]. [Accessed 6 August 2017]. Available from:

<http://www.itech.ua.es/dadm/restringido/web/sesion07-apuntes.html#%28HLS%29>

[19] MOROCHO, Jefferson. PROTOCOLOS DE TIEMPO REAL RTP, RTCP Y RTSP. prezi.com [online]. 2 August 2016. [Accessed 6 August 2017]. Available from:

https://prezi.com/9wk_j_iejkn/protocolos-de-tiempo-real-rtp-rtcp-y-rtsp/

[20] Adobe HTTP Dynamic Streaming (HDS) Technology Center. Adobe HTTP Dynamic Streaming (HDS) Technology Center | Adobe Developer Connection [online]. [Accessed 6 August 2017]. Available from: <http://www.adobe.com/devnet/hds.html>

[21] Formatos Contenedores Multimedia - DivXLand.org. Formatos Contenedores Multimedia - DivXLand.org [online]. [Accessed 6 August 2017]. Available from:

http://www.divxland.org/es/article/12/formatos_contenedores_multimedia#.WYdsUFFJbIV

[22] Formatos contenedores. Guía 4K 709 V.2 [online]. [Accessed 6 August 2017]. Available from: <http://guia4k.709mediaroom.com/formatos-y-codecs/formatos-contenedores/>

