

**SISTEMA DE VIDEO-MONITOREO REMOTO PARA HOGARES VÍA CELULAR**

**JOSE ANTONIO FALLA LUNA  
OSCAR DANIEL PINZÓN HEREDIA**

**TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE INGENIERO ELECTRÓNICO**

**DIRECTOR**

**ING. GERMAN ALEXIS CORTES HERNANDEZ**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE ELECTRÓNICA  
BOGOTÁ D.C.  
NOVIEMBRE 2010**

**RECTOR MAGNÍFICO:**

**DECANO ACADÉMICO:**

**DECANO DEL MEDIO UNIVERSITARIO:**

**DIRECTOR DE CARRERA:**

**DIRECTOR DEL TRABAJO DE GRADO:**

**R.P. JOAQUÍN SÁNCHEZ GARCÍA S. J.**

**Ing. FRANCISCO J. REBOLLEDO M.**

**R.P. SERGIO BERNAL RESTREPO S. J.**

**Ing. JUAN MANUEL CRUZ M.Edu**

**Ing. GERMAN ALEXIS CORTES H.**

**ARTÍCULO 23 DE LA RESOLUCIÓN N° 13 DE JUNIO DE 1946**

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus trabajos de grado. Solo velará porque no se publique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien que se vea en ellos el anhelo de buscar la verdad y la justicia”

## TABLA DE CONTENIDO

1. INTRODUCCIÓN .....	9
2. MARCO TEÓRICO.....	11
2.1 Captura de imagen.....	12
2.2 Análisis de imágenes .....	13
2.3 Almacenamiento de video. ....	16
2.5 Envío de mensajes .....	20
2.6 Visualización de video en tiempo real .....	23
3. ESPECIFICACIONES .....	24
3.1 Dispositivos necesarios: .....	24
3.2 Programas y librerías utilizadas para crear la herramienta:.....	24
3.3 Especificaciones básicas del PC donde se ejecutara el programa.....	25
3.4 Requisitos ambientales .....	25
3.5 Formatos de video permitidos.....	25
3.6 Otros requisitos. ....	25
3.7 Parámetros variables Dentro del Código para detección de movimiento.....	25
3.8 Especificaciones de funcionamiento. ....	26
3.9 Modos de funcionamiento. ....	26
4. DESARROLLOS .....	26
4.1 Captura de imagen.....	27
4.2 Grabación de video.....	28
4.3 Análisis de video. ....	29
4.4 Carga de archivos. ....	31
4.5 Envío de mensajes .....	32
4.6 Visualización de Video en Tiempo Real .....	33
4.7 Página de Internet (servidor).....	34
5. ANÁLISIS DE RESULTADOS .....	38
5.1 Pruebas Unitarias. ....	38
5.1.1 Captura y grabación de video.....	38
5.1.2 Análisis de video. ....	41
5.1.3 Modo Visualizacion.....	49
5.1.4 Carga de archivos .....	50
5.1.5 Envió de mensajes .....	54
5.1.6 Visualización de los videos desde el browser. (Desde celular y PC).....	56
5.2 Pruebas Integrales .....	57
5.2.1 Rendimiento. ....	57
5.2.2 Tiempos.....	60

5.2.3 Veracidad y calidad de la información .....	60
6. CONCLUSIONES .....	62
7. BIBLIOGRAFÍA .....	63
8. ANEXOS.....	65
Anexo 1. Captura y análisis de video.....	65
Anexo 2. FTP (Protocolo para carga de archivos).....	65
Anexo 3. PHP (Protocolo para envío de mensajes) .....	65
Anexo 4. HTML Modo visualización. ....	65

## LISTA DE FIGURAS

Figura 1. Esquema General Herramienta de Monitoreo.....	10
Figura 2. Imagen de Cálculo Piramidal.....	15
Figura 3. Diagrama general de FTP [15].....	17
Figura 4. Esquema FTP en <i>modo</i> Activo [17].....	18
Figura 5. Esquema FTP en <i>modo</i> Pasivo [17].....	18
Figura 6. Camino evolutivo de la plataforma GSM [18].....	20
Figura 7. Envío E-mail a SMS directo.....	22
Figura 8. Envió E-mail to SMS con servidor corporativo.....	23
Figura 9. Diagrama de Bloques Captura de Imagen.....	27
Figura 10. Diagrama de bloques Grabación de video.....	28
Figura 11. Diagrama de Bloques Análisis de Imagen.....	29
Figura 12. Diagrama de flujo programa análisis de video.....	30
Figura 13. Diagrama de Bloques Carga de Archivos.....	31
Figura 14. Diagrama de Bloques Envío de Mensajes.....	32
Figura 15. Diagrama de Bloques Streaming y Modo Visualización.....	33
Figura 16. Site <a href="http://www.videonlinemonitoring.com">www.videonlinemonitoring.com</a> Página de inicio.....	35
Figura 17. <a href="http://www.videonlinemonitoring.com">www.videonlinemonitoring.com</a> Página de videos y Modo Visualización.....	36
Figura 18. cPanel <a href="http://www.videonlinemonitoring.com">www.videonlinemonitoring.com</a> .....	36
Figura 19. Matriz de píxeles desplazada.....	39
Figura 20. Pantallas desplegadas por OpenCv sin Imagen.....	40
Figura 21. Pantallas desplegadas por OpenCv con Imagen.....	40
Figura 22. Capturas de imagen en dos cámaras Web diferentes (a) Cámara 1(b) Cámara 2.....	41
Figura 23. Archivos almacenados en formato .avi en disco duro.....	41
Figura 24. Captura de imagen con cambios en distancia (a) 6m (b) 3m (c) 2m (d) 1m. Con variables fijas en Cal_g=10, Minp=12, Cps=20 y tamaño de la persona aproximado 180x50 cm.....	43
Figura 25. Prueba de distancia con movimmiento a 60 m.....	44
Porcentaje de imagen 5%.....	44
Figura 26. Cambios de velocidad con variables fijas en minp = 7, maxd = 11, mind = 5. Puntos encontrados 0.....	44
Porcentaje de imagen 20%.....	44
Figura 27. Pruebas para fijar filtro en el valor mas recomendado, con variación en velocidad, variables fijas en minp = 7, maxd = 80, mind = 1.....	45
Porcentaje de imagen. 25%.....	45
Figura 28. Con variables definidas en minp = 7, maxd = 11, mind = 5. Mintiem_vid=13.....	46
Porcentaje de imagen 23%.....	46
Figura 29. Pruebas de barrera de tiempo con variables fijas en minp = 7, maxd = 11, mind = 5. Mintiem_vid=1.....	47
Figura 30. Pruebas con barrera de tiempo mayor, con variables fijas en minp = 7, maxd = 11, mind = 5. Mintiem_vid=20.....	47
Figura 31. Pruebas de cambio de luz.....	48
Intensidad de luz 15%.....	48
Luego de realizar las pruebas mencionadas anteriormente, se obtuvieron los siguientes resultados en cuanto a la sensibilidad. En la siguiente tabla se puede observar cómo se comporta la sensibilidad de la herramienta ante la variación de las variables.....	48
Figura 32. Carga de archivos con FTP ejecutado desde C.....	49
Figura 33. Prueba de modo visualizacion.....	49
Figura 34. Error carga de archivo cuando está en uso.....	50
Figura 35. Archivos cargados en el servidor.....	51
Figura 36. Pantalla de Filezilla (Carga y descarga de archivos).....	51

Figura 37. Prueba de caga de archivos con FTP ejecutado en consola Java. ....	52
Figura 38. Prueba de carga de archivos con FTP ejecutado en consola Java y con inclusión de archivo dentro de proyecto. ....	52
Figura 39. Prueba de carga de archivos con FTP con argumento. ....	53
Figura 40. Prueba de carga de archivos con FTP con argumento con carga exitosa. ....	53
Figura 41. Prueba de carga de archivos ejecutando .jar desde DOS. ....	54
Figura 42. Pantalla de Mozilla con mensaje de envío de mensaje exitoso, mediante el uso de PHP sin Query. ....	55
Figura 43. Pantalla Mozilla FireFox con mensaje “Message Delivery Failed” ....	55
Figura 44. Pantalla de Mozilla con mensaje de envío de mensaje exitoso, mediante el uso de PHP con Query. ....	55
Figura 45. Prueba de reproducción de video desde el Site. ....	56
Figura 46. Pruebas de rendimiento de maquina, en equipo Celeron (a)(b) y equipo Pentium 4 (c)(d) con variables fijas en 15cps. Minp=7 cal_g=10. ....	58
Figura 47. Pruebas de rendimiento de máquina, en equipo Core i5 con variables fijas en 15cps. Minp=7 cal_g=10 (a).....	59
(b).....	59

## LISTA DE TABLAS

Tabla 1. Ejemplos de extensiones y tipos de transferencia.....	19
Tabla 2. Características servidor www.videonlinemonitoring.com.....	37
Tabla 3. Prueba de variación de movimiento.....	43
Tabla 4. Prueba de variación de velocidad en movimiento.....	45
Tabla 5. Análisis de sensibilidad.....	48
Tabla 6. Características teléfonos celulares.....	57
Tabla 7. Equipos en los que se realizo prueba de rendimiento.....	59
Tabla 8. Prueba tiempos de compilación.....	60
Tabla 9. Prueba tiempos de ejecución.....	60
Tabla 10. Pruebas tiempos de respuesta.....	60
Tabla 11. Resultados de pruebas de calidad de la información.....	61



## 1. INTRODUCCIÓN

En la actualidad constituye una necesidad imperiosa implementar mecanismos que mejoren la sensación de seguridad dentro de un ambiente cada vez más hostil. Este efecto se refuerza, de manera positiva, al tenerse conciencia que tanto las personas como los bienes materiales objeto de especial interés de protección, son objeto de observación y/o control, en momentos en que puedan estar expuestos a algún riesgo o peligro por parte de quienes pretendan atentar contra ellos.

Esta sensación de seguridad debe ser una constante en el entorno que rodea la actividad rutinaria de cada día tanto en lo familiar como en los negocios. Hoy en día, se precisa realizar desplazamientos que al dejar expuestas personas y/o bienes que al requerir supervisión de una manera constante, afectan la tranquilidad personal del responsable de éstos, lo que a su vez repercute en un progresivo deterioro de su salud.

Por otro lado, si bien es cierto que el desarrollo actual de tecnología ha traído consigo herramientas como alarmas, seguros, puertas antirrobo, sistemas de monitoreo, los cuales representan altos costos para su implementación, excluyen ciertos sectores sociales que no poseen los medios económicos requeridos para acceder a éstas herramientas u opciones.

Dada esta problemática, se advierte la necesidad de generar mecanismos que contribuyan a generar el sentimiento de seguridad de las personas que se movilizan de manera constante por diversos motivos laborales, comerciales o familiares, etc. y brindarle a estas personas, una herramienta asequible para la mayoría y que les permita saber de sus activos mientras se encuentran lejos de ellos, alertando, por ejemplo, cuando éstos están siendo sustraídos.

El monitoreo por medio de imágenes digitales se ha convertido en una necesidad, principalmente a nivel industrial, empresarial y, en algunos casos, en unidades residenciales. Desde hace varios años, el análisis de imágenes digitales ha estado aportando al desarrollo de muchas tecnologías, a tal punto que, varias de ellas se han masificado y están al alcance de la gran mayoría de personas. Lo mismo ha ocurrido con las redes globales para la transmisión de datos (La Internet) y con las redes de celular para transmisión de datos móviles (por ejemplo la red 3G).

Conforme a lo expuesto con anterioridad, surge la idea de hacer una herramienta doméstica de bajo costo, de fácil manejo y operación que sea accesible para cualquier persona. Estos recursos de uso masivo, están en el mercado a un precio económico, por lo que con un desarrollo de ingeniería como el propuesto, pueden ser utilizados para la problemática planteada.

Para resolver esta situación, con el presente Trabajo de Grado se diseña y se desarrolla una herramienta tecnológica para el monitoreo remoto del hogar, utilizando recursos y tecnologías ya existentes, integrándolas y desarrollando aplicaciones de software para la funcionalidad y efectividad de la herramienta.

Esta herramienta es un sistema de monitoreo de imágenes remoto que consta, principalmente, de un celular, un computador, una cámara Web y redes de comunicaciones. Ver FIGURA 1.

Para el desarrollo de esta herramienta se partió de los siguientes supuestos:

- El usuario debe tener en su hogar un computador encendido con cámara Web y acceso a Internet.
- En el sitio remoto, el usuario debe tener un celular con acceso a internet que cumpla con las características requeridas para el funcionamiento de la herramienta.

La herramienta es utilizada para prestar un servicio que funciona de la siguiente manera:

1. El usuario recibirá un SMS en su celular, cuando en el sistema se presente un evento (El software detecte el cambio en una imagen captada en el lugar específico).
2. El usuario por medio del celular, podrá ver una imagen en tiempo real del lugar a monitorear en el momento que lo desee, o consultar los archivos de video o imagen que se encuentren almacenados en un servidor.

Este Proyecto pretende disminuir el costo de la herramienta, hacerlo asequible para cualquier persona y generar una propuesta acertada y efectiva a las diferentes necesidades que se tengan. Esta herramienta es versátil, en la medida que puede ser utilizada para resolver diferentes necesidades del usuario. Algunos usos pueden ser: vigilar un activo, niños, supervisión de empleados, de mascotas, integridad física del lugar, etc.

En la siguiente imagen, se puede ver un esquema general del funcionamiento de la herramienta.

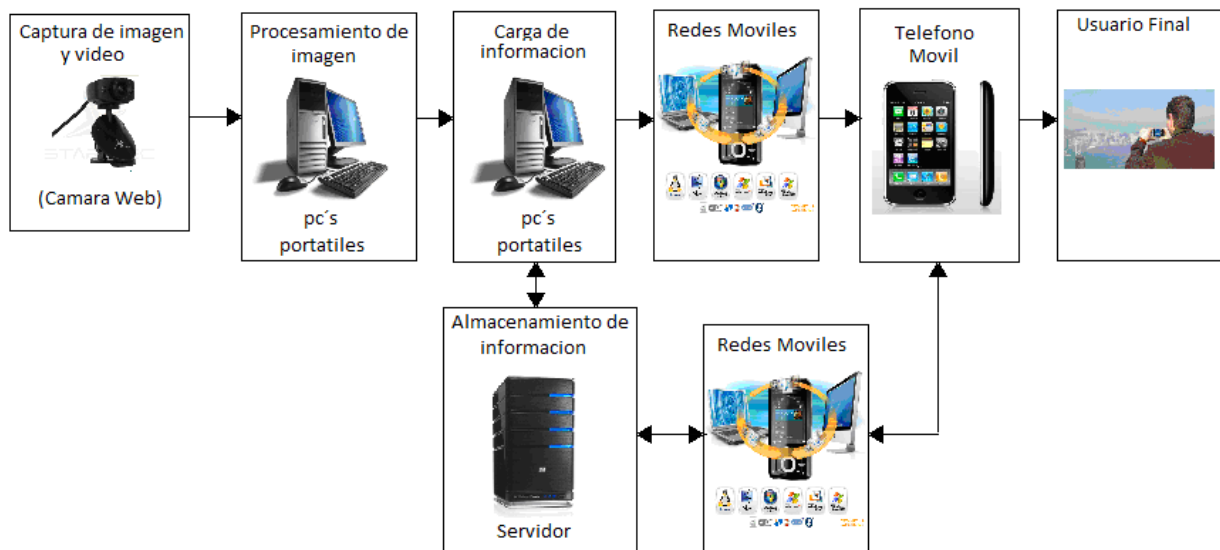


Figura 1. Esquema General Herramienta de Monitoreo

Entre los aspectos que justifican el Proyecto se tiene los siguientes:

La posibilidad de acreditar, como evidencia probatoria ante las autoridades correspondientes, los registros de video que se obtengan, cuando de ellos se advierta la comisión de un delito, tales como hurto, daños en bien ajeno, maltratos, etc.

Empresas como Telefónica, ETB y Telmex, (actualmente y una vez realizada una investigación de mercado), están validando la posibilidad de prestar este servicio en forma masiva, que ven como una necesidad para el usuario en el hogar y como una gran oportunidad de negocio.

El impacto real del Proyecto es lograr que la herramienta sea una solución viable, económica y con un enfoque doméstico y personalizado. Hacer que el usuario se sienta cómodo y confiado, generando sensación de seguridad y tranquilidad en la vida común del usuario.

Crear una herramienta que permita hacer un monitoreo de imágenes o video remoto a un determinado lugar, a partir de la integración de tecnologías ya existentes y el desarrollo de diferentes aplicaciones en software.

-Investigar y analizar las redes de comunicaciones celulares existentes en Colombia para la transmisión de datos (3G, GSM, GPRS, EDGE, HSPA).

-Lograr la transferencia de datos entre PC-Celular-PC y la red pública (Internet) utilizando alguna red de datos ya existente (3G, GSM, GPRS, EDGE, HSPA).

-Integrar las tecnologías que ya existen y que necesitamos con los desarrollos realizados para la creación de la herramienta.

-Investigar la arquitectura de las diferentes configuraciones, pensando en la red de datos y en los equipos que capturan, almacenan y procesan la imagen.

-Analizar los desarrollos para detección de movimiento en una imagen utilizando flujo óptico

## **2. MARCO TEÓRICO**

Antecedentes: Durante la realización del Proyecto se tuvieron en cuenta trabajos desarrollados por estudiantes y profesores de la Pontificia Universidad Javeriana, trabajos que desarrollan formas y métodos para la estimación de flujo óptico en videos. Entre estos trabajos, tomamos principales referencias los siguientes:

-Trabajo de Grado Calderón Francisco Urrego Germán “Conteo Automático de Vehículos”.

-Trabajo de Grado Néstor Romero y Carolina García “Detección y Seguimiento de Personas en Cruce peatonal”.

-Trabajo de Grado David Mora, Andrés Páez y Julián Quiroga Detección de Objetos Móviles en una Escena Utilizando Flujo Óptico.

A continuación se expone de qué consta la herramienta propuesta y cómo fue constituida. Esta es, principalmente, por la integración de sistemas, por lo tanto, se procederá a documentar una por una, indicando posteriormente resultados de pruebas unitarias y pruebas integrales.

## 2.1 Captura de imagen

En esta herramienta la captura de imagen se realiza mediante la ejecución de una función de OpenCv (CvCapture), función que se encuentra dentro de la librería cv.h. Esta función permite obtener un cuadro de video de la cámara Web instalada en el computador del lugar a monitorear.

OpenCv es un conjunto de bibliotecas de código abierto u Open Source desarrolladas en un principio por Intel disponibles desde <http://SourceForge.net/projects/opencvlibrary>. Están desarrolladas en lenguaje C y C++ y se pueden ejecutar desde Windows, Linux y Mac OS X.[1].

Estas Bibliotecas ofrecen código diseñado eficientemente, orientado a aplicaciones capaces de ejecutarse en tiempo real en procesadores modernos. Estas bibliotecas tienen como objetivo proveer las funciones más usadas en Visión por computadora y tiene más de 500 funciones implementadas. Se usan ampliamente en vigilancia, imágenes médicas, inspección industrial, interfaces de usuarios, calibración de cámaras y, recientemente, se han usado estas bibliotecas en imágenes aéreas y mapas de calles, por ejemplo, en la herramienta *Google's Street View*. Se hace uso de calibración de cámara y de funciones de OpenCV para hacer el proceso de *Stitching*, que consiste en combinar múltiples imágenes para producir una imagen panorámica o una imagen de alta resolución.[1].

Ya que esta herramienta genera almacenamiento de video y de análisis, se hace necesario tener en cuenta los recursos del computador que serán necesarios durante la ejecución del programa. Recursos de almacenamiento y velocidad de procesamiento. Con el fin de optimizar el uso de estos recursos, se hace necesaria la utilización de un Códec de video, previamente instalado en el equipo, donde se ejecutará el programa.

El Códec ( Xvid 1.2.2) optimiza la compresión de video mediante el uso de algoritmos matemáticos, que resuelven la compresión y descompresión de los archivos que se trabajan durante la ejecución del programa.

Xvid es un Códec de distribución libre que instala en el equipo los paquetes de codificación necesarios para desarrolladores de software, como principal ventaja se advierte que este software es compatible con la mayoría de sistemas operativos y con las diferentes arquitecturas de hardware que actualmente existen. Xvid también permite ser ejecutado desde diferentes plataformas de compilación, trabajando la compresión sobre la imagen, lo que genera mayor utilidad del Códec en resoluciones bajas.

## 2.2 Análisis de imágenes

Para el análisis de imágenes en la herramienta se escogió el programa Dev C++ como entorno de programación, en razón a que tiene las siguientes ventajas sobre los demás: es un programa fácil de usar, con la misma interfaz de Windows y de libre distribución. Permite compilar código de C o C++ y, por último, posee librerías y funciones dedicadas a la captura y análisis de video.

Dentro de las librerías usadas para el tratamiento de imágenes se ha venido usando Open CV, una librería libre con bastantes aplicaciones, con versiones para Linux, Mac OS y Windows, dentro de la infinidad de aplicaciones se tienen, entre otras, el tratamiento de imágenes, control de procesos, detección de rostros, detección de objetos y detección de movimiento. La librería maneja funciones dedicadas a los programas relacionados con procesos de visión.

Las siglas OpenCv provienen de los términos anglosajones “Open Source Computer Vision Library”. Por lo tanto, OpenCv es una librería de tratamiento de imágenes, destinada principalmente a aplicaciones de visión por computador en tiempo real.[19].

Dentro de las funciones de la librería OpenCv se cuenta con dos en particular (Lucas-Kanade y Good features to track), funciones que ayudarán de forma acertada en el desarrollo de este Proyecto.

*Good features to track* es uno de los tres detectores de puntos característicos que maneja la librería OpenCv, este detector se basa en el detector de Harris, el cual mide las diferencias entre las áreas de la imagen según tres casos: **(i)** región plana: intensidad constante, cambios pequeños; **(ii)** borde: cambios pequeños en dirección del eje, pero altos en dirección perpendicular al borde y **(iii)** esquina: en donde todos los cambios son altos.[20].

Teniendo en cuenta que la detección de movimiento de esta herramienta se basa en el flujo óptico, se hace necesario saber que éste, es un patrón de movimiento dado por el cambio de lugar de un borde o superficie (puntos de referencia Good features to track) dentro de un clip de video o escena visual.

De forma más precisa, el flujo óptico puede definirse como el aparente movimiento de los pixeles midiendo su intensidad en blanco y negro. Se dice aparente, porque el flujo óptico puede coincidir o no con el movimiento real. En ocasiones, cuando se cumple con ciertos parámetros, se puede relacionar directamente el movimiento con el flujo óptico. Existen algunas formas para encontrar el flujo óptico: **(i)** por medio de comparación de regiones; **(ii)** por comparación de fase, técnicas basadas en energía y **(iii)** técnicas basadas en radiantes. A partir de éstos, se desarrollaron métodos entre los cuales sobresale el Método de Lucas-Kanade.

Para cumplir la ecuación de flujo óptico, se debe cumplir la hipótesis de que los parámetros de intensidad en blanco y negro se encuentran constantes entre cuadros, es decir, los puntos de las intensidades de los objetos, permanecen constantes en el tiempo. Esto da lugar a la ecuación de flujo óptico.

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Ecuación de flujo óptico

En donde  $I(x,y,t)$  es igual a la intensidad en niveles de grises de la imagen.

Al expandir en Series de Taylor se obtienen las correspondientes ecuaciones con sus vectores horizontal y vertical para cada pixel en el tiempo. Para su solución, no existe una única solución, para lo cual, se utilizan algunas restricciones en los videos.

$$I(x, y, t) = I(x, y, t) + dx \frac{\partial I}{\partial x} + dy \frac{\partial I}{\partial y} + dt \frac{\partial I}{\partial t} + \varepsilon$$

Ecuación de flujo óptico expandida

Lukas-Kanade es una función que se utiliza para la estimación de flujo óptico mediante la utilización de barridos de pixeles. Esta función es una de las más populares y utilizadas para este tipo de análisis. Este método trata de calcular el movimiento entre dos puntos particulares de la imagen cuando se presenta movimiento continuo en la imagen capturada (Webcam). Razón por la cual se seleccionó este método.

Este método asume que el flujo es constante sobre la región a calcular.

$$I_x(p_i)u + I_y(p_i)v = -I_t(p_i) \quad \forall p_i \in R$$

Organizando en matrices:

$$\underbrace{\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_N) & I_y(p_N) \end{bmatrix}}_A \underbrace{\begin{bmatrix} u \\ v \\ d \end{bmatrix}}_d = - \underbrace{\begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_N) \end{bmatrix}}_b$$

Multiplicando por la transpuesta de la matriz A se despeja d y se tiene.

$$d = (A^T A)^{-1} A^T b$$

Donde

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix}$$

Su inversa debe existir y debe ser bien condicionada “acercarse a la matriz identidad”

## Lukas y Kanade Piramidal

Si se tienen dos imágenes en escala de grises, la idea es encontrar la ubicación de un pixel en la siguiente imagen, para esto se puede definir una ventana llamada “ventana de integración” que es donde se va a encontrar el siguiente pixel, esto genera un problema ya que si se escoge una ventana de integración muy pequeña no se pueden encontrar movimientos muy grandes, en cambio si se escoge una ventana de integración muy grande, se puede presentar inconsistencia en los movimientos debido a que muy posiblemente se encontrarán varios puntos que representen dicho movimiento.

Para darle solución a este problema, Lucas y Kanade Piramidal propone encontrar primero en las imágenes más pequeñas el punto del flujo óptico de la imagen en el tiempo  $i$  a la imagen en el tiempo  $i + 1$  ó siguiente cuadro; luego se aumenta el tamaño en un factor de dos y se busca el flujo óptico de la imagen en el tiempo  $i + 1$  sin aumentar el factor a la imagen  $i$  con el factor aumentado; Esto se itera hasta llegar al tamaño real. Esta solución trata de disminuir la probabilidad de no encontrar el punto real de movimiento del objeto. El resultado del desplazamiento es el que se obtiene de la suma de todos los pequeños desplazamientos anteriormente mencionados.

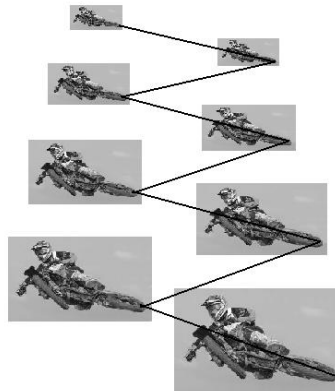


Figura 2. Imagen de Cálculo Piramidal

## Algoritmo de Shi y Tommasi

Shi y Tommasi identifica un conjunto de puntos con propiedades que se basan específicamente en características de texturas, los puntos extremos o bordes por medio de la intensidad espacial. Para determinar un conjunto de “los mejores puntos para seguir”, se trabaja con el desplazamiento de los puntos en el interior de una ventana fija definiendo un error y tratando de disminuirlo.

Estos puntos son los apropiados para estimar el flujo óptico de una manera más confiable, debido a que los puntos encontrados por el algoritmo de Shi y Tommasi dan solución a la matriz  $A^T A$ . De la misma manera, se obtienen las características que deben cumplir dichos puntos. Sean  $\lambda_1$  y  $\lambda_2$  los valores propios de la matriz  $A^T A$  para cierta región  $R$  de la imagen, entonces para ser un buen punto para seguir, se debe cumplir que:

Textura:

- $\min(\lambda_1, \lambda_2) > \lambda_{\min} \in \mathbb{R}^+$ , donde  $\lambda_{\min}$  es un valor fijo establecido, lo cual garantiza que  $A^T A$  es invertible y la región no es ruidosa.

Bordes:

- $\frac{\lambda_1}{\lambda_2} < \tau$ , lo que garantiza que  $A^T A$  está bien condicionada y no se presenta bordes en una sola dirección [13].

### 2.3 Almacenamiento de video.

Para el almacenamiento de video de la herramienta se utilizó un servidor (www.hosting24.com). Este funciona como servidor, proporcionándole a la herramienta un lugar con alta capacidad en almacenamiento, funciones como mensajería, transferencia de archivos, acceso remoto y, además, un espacio programado bajo lenguaje HTML para la visualización desde un Web browser. Al mismo tiempo, se cuenta con un dominio propio, un panel de control “cPanel”.

Este servidor utiliza los más avanzados sistemas de hardware y software para garantizar la mejor calidad en la prestación del servicio. La información siempre va a estar segura, debido a que utiliza sistemas de copia de seguridad sobre arreglos RAID. Esta copia de seguridad se genera cada 7 días y se almacena en servidores remotos con acceso ilimitado a la cuenta de administrador, para que pueda acceder en cualquier momento y desde cualquier lugar.

Este servidor consta de una aplicación llamada Web Site builder, que permite que el usuario pueda configurar y diseñar, de manera personalizada y fácil, el aspecto de la página.

Un aspecto importante a tener en cuenta para escoger el servidor, es el acceso a la información en todo momento. En este caso se cuenta con un servidor robusto, que consta de un SW que soluciona el 99% de los errores en forma automática, sin necesidad de que exista interacción de personas con la máquina. Esto garantiza un funcionamiento confiable y constante. En términos de seguridad y monitoreo, éste es un aspecto primordial y definitivo para evaluar la eficiencia del sistema.

### 2.4 Carga de Archivos

Para la carga de archivos se utilizó un protocolo programado en Java (FTP File Transfer Protocol), modificado en Eclipse (Editor de lenguaje Java). El protocolo FTP permite enviar o recibir archivos utilizando la tipología de cliente servidor. Este software permite copiar, modificar o borrar los archivos en el servidor. Además, es un modelo de aplicación que utiliza las capas de red TCP/IP, y los puertos 20 y 21.

La tipología cliente servidor es independiente del sistema de archivos que se esté utilizando en cada máquina, por lo tanto, el hecho de que éstas tengan sistemas operativos diferentes, no impacta en el desempeño de la herramienta, las entidades que se comunican entre sí para establecer la transferencia de



datos son los PI y los DTP. También, es importante destacar que la conexión de datos puede ser bilateral, en razón a se puede usar simultáneamente para enviar y recibir información.

### Servidor FTP

El servidor FTP se ejecuta en un servidor normalmente, conectado a internet, como un programa normal. Este programa es utilizado para permitir el intercambio de información entre diferentes ordenadores. Entre las aplicaciones más comunes de los servidores FTP, encontramos la de alojamiento.

### Cliente FTP

El cliente FTP es un programa que se instala y ejecuta en el ordenador del usuario, sin embargo, su función es la misma, la de transferencia de archivos. No obstante, para poder usar el cliente FTP, es necesario conocer la información (contraseñas, nombres de usuario, etc.) del servidor al que se le desean transferir los archivos.

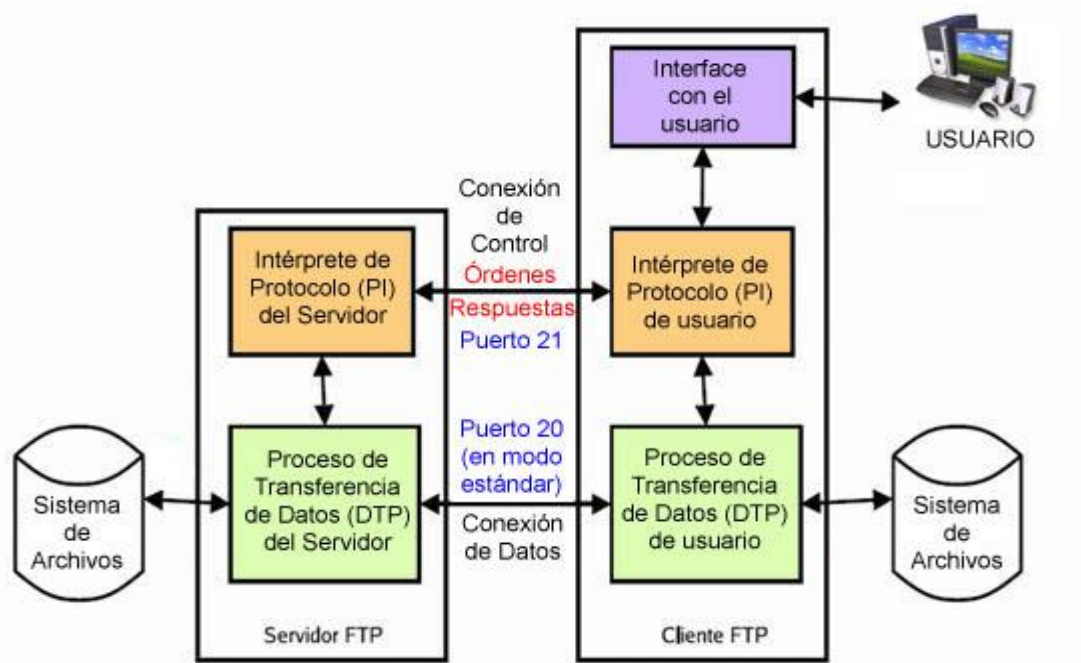


Figura 3. Diagrama general de FTP [15]

El FTP permite 2 modos de conectividad. Se conectan en modo activo (que funciona cuando se envían comandos tipo PORT por el canal de conexión) y en modo pasivo, (en donde los comandos que se envían son de tipo PASV).

### Modo Activo.

El modo activo el servidor se conecta utilizando el puerto 20, mientras que en el cliente, el canal de datos se asocia a un puerto superior al puerto 1024. Para esto, el cliente indica el número de puerto enviando un comando de tipo PORT al servidor utilizando el canal de control, para que de esta manera el servidor pueda habilitar la conexión cliente-servidor e iniciar la transferencia de datos.

Para ello, el cliente manda un comando PORT al servidor por el canal de control indicándole el número de puerto, de manera que el servidor pueda abrirle una conexión de datos, por donde se transferirán los archivos y los listados, en el puerto especificado.

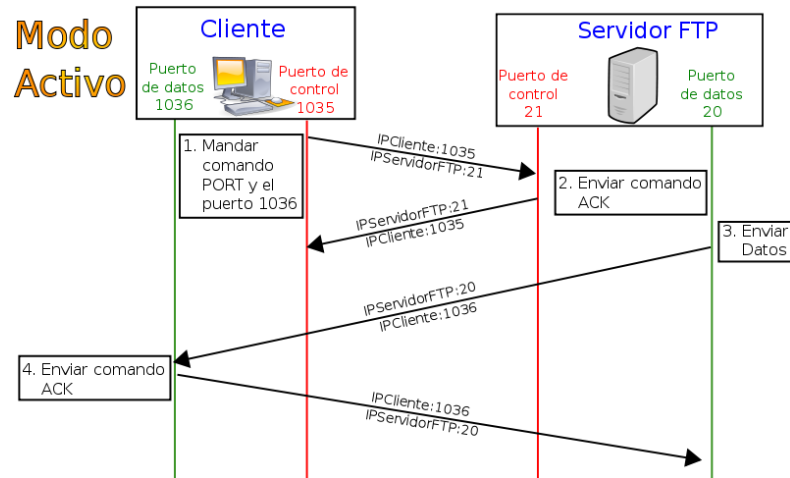


Figura 4. Esquema FTP en modo Activo [17]

Lo anterior tiene un gran inconveniente, ya que la máquina cliente estaría dispuesta a aceptar cualquier conexión de entrada en un puerto superior al 1024. Para solucionar este problema, se desarrolló el modo pasivo.

### Modo Pasivo.

En el modo pasivo el cliente envía un comando tipo PASV utilizando el canal de control, para que de esta forma el servidor FTP le indique por el mismo canal por cual puerto debe conectarse. El puerto debe ser mayor a 1023. Ej.: 2030. Posteriormente, el cliente inicia una conexión desde el siguiente puerto, al puerto control, hacia el puerto que el servidor tiene definido. En este caso sería el 2030.

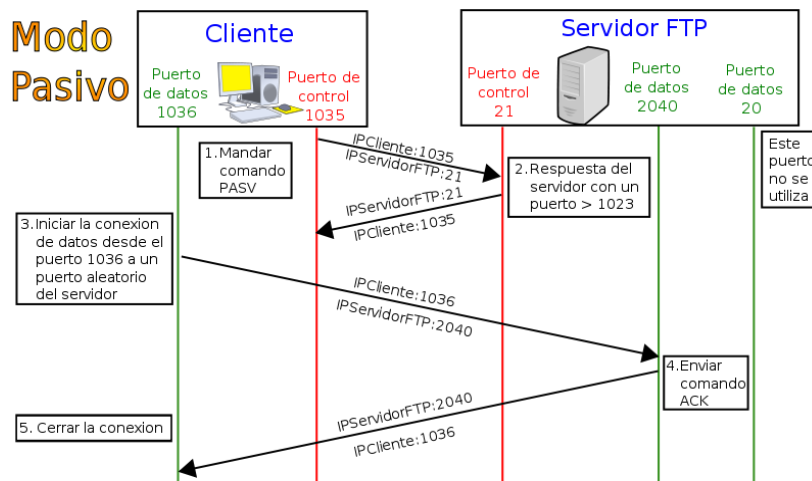


Figura 5. Esquema FTP en modo Pasivo [17]

Cada vez que se va a iniciar una nueva transferencia, ya sea en modo activo o en modo pasivo, el cliente debe enviar el comando de control (PASV o PORT según el modo), y el servidor detectará esa conexión para transferencia de datos usando un nuevo puerto aleatorio si está en modo pasivo, o por el puerto 20 si está en modo activo. En el protocolo FTP existen 5 tipos de transferencia para ASCII y para binarios.

## Tipos de transferencia de archivos en FTP

Para no dañar la información del archivo, es necesario conocer como se debe transportar la información por la red. Si no se utilizan las opciones adecuadas, se pueden perder datos o inclusive se puede llegar a dañar completamente el archivo. Para que esto no ocurra, se debe ejecutar el FTP utilizando los comandos ASCII o Binario.

- TIPO ASCII

Este se utiliza para transferir archivos que contengan únicamente caracteres, pero teniendo en cuenta que no pueden ser archivos en formatos de editores de texto. Es usado para transferir como por ejemplo, páginas HTML.

- TIPO Binario

Este tipo se utiliza cuando se van a transferir archivos con imágenes, sonidos, videos, ejecutables, comprimidos y documentos con formato de editor de texto.

<b>Extensión de Archivo</b>	<b>Tipo de Transferencia</b>
txt (texto)	ASCII
HTML (página WEB)	ASCII
doc. (documento)	Binario
ps (postscript)	ASCII
hqx (comprimido)	ASCII
Z (comprimido)	Binario
ZIP (comprimido)	Binario
ZOO (comprimido)	Binario
Sit (comprimido)	Binario
pit (comprimido)	Binario
shar (comprimido)	Binario
uu (comprimido)	Binario
ARC (comprimido)	Binario
tar (empaquetado)	Binario

Tabla 1. Ejemplos de extensiones y tipos de transferencia [17]

Tanto Java como Eclipse son versiones de software libres abiertas a desarrolladores de código.

## 2.5 Envío de mensajes

Ya que en este trabajo de grado se utiliza la red celular 3G de Comcel, se hace necesario tener en cuenta o mencionar parte de la teoría existente acerca de las redes celulares que se utilizan hoy en día.

Como primera medida, es necesario entender que la red GSM (Global System For Mobile Communications), es una red que surge a partir de la necesidad de unificar los estándares de comunicación celular y con el fin de crear sistemas de comunicación compatibles entre sí. En la figura que se muestra a continuación se muestra el desarrollo que ha venido teniendo la plataforma GSM.

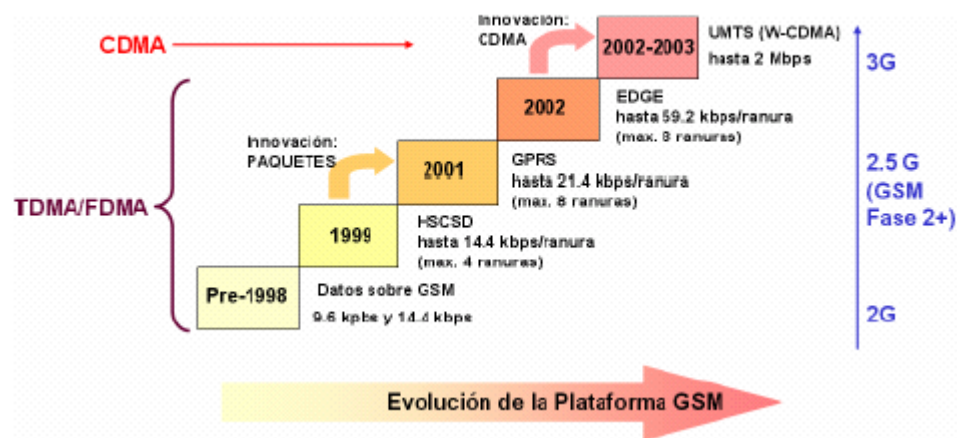


Figura 6. Camino evolutivo de la plataforma GSM [18]

Como podemos ver en el gráfico se parte de la red GSM, esta red se caracteriza por ofrecer transferencia de datos con un buen rango de velocidades de transmisión, en el que encontramos velocidades desde 300 bps, hasta 9.6 Kbps. Esta red trabaja sobre las bandas 850/900/1800/1900 MHz

Es importante resaltar que el protocolo de aplicaciones (WAP) apareció como un estándar internacional sobre esta red y permitió la conexión de dispositivos móviles a través de redes inalámbricas.

Siguiendo el gráfico, se puede observar la red de transmisión de datos HSCSD (High Speed Circuit Switched Data), la cual maneja una velocidad constante, recomendable para transmisión de datos, en especial, archivos electrónicos (e-mail) con datos o archivos adjuntos de tamaño considerable. Red bastante recomendada, a su vez, para la navegación en páginas Web.

En la figura se encuentra la red GPRS (General Packet Radio Service), la cual busca el mejoramiento de la comunicación para la transmisión de datos. La red GPRS mejora los servicios existentes en la red HSCSD, servicios como WAP, Palm.net e I-node. La diferencia de esta red con las dos anteriores, se basa en establecer una velocidad de transferencia de datos variable (siempre acorde al requerimiento del usuario), lo que permite generar una mayor eficiencia en el manejo del ancho de banda. Por otro lado, esta red permite la conexión permanente al servicio, sin la necesidad de previa marcación. Debido a que esta red se basa en la conmutación de paquetes de información, los usuarios ven afectada su velocidad de transferencia de datos, acorde con la cantidad de usuarios conectados a la red.

Luego de la red GPRS, se cuenta con la red EDGE (Enhanced Data Rates for Global Evolution). Esta red es una evolución de las redes GSM y GPRS, que busca el mejoramiento de las tasas de transmisión de datos, en especial, para el manejo de datos multimedia, esta red provee servicios de telefonía que alcanzan velocidades de transmisión de datos de hasta 348 Kbps teóricos y entre 50 Kbps y 100 Kbps prácticos.

Por último, se cuenta con la red de transmisión de datos UMTS o W-CDMA (Wideband Code Division Multiple Access), esta red se basa en el acceso múltiple por división de código de banda ancha, lo que genera una velocidad de transferencia en datos mucho mayor (348 Kbps a 2 Mbps).

Debido a que la mejor transmisión de datos se da en la red 3G, ésta se ha escogido para este Proyecto como red para envío de SMS, visualización de videos en tiempo real y navegación de páginas de Internet.

Para el envío de mensajes, en este Proyecto se utiliza código PHP (Hypertext Pre Processor), este código de programación está enfocado a la creación de páginas web dinámicas, permite que se le incruste dentro de código HTML y se ejecuta desde el servidor tomando código como entrada y arrojando páginas web como salidas. El PHP es un lenguaje de programación que permite crear scripts (guión o archivo de órdenes) del lado del servidor (se ejecuta sobre el servidor Web. Además, es un lenguaje de programación que está pensado para realizar gran cantidad de aplicaciones web, entre la cuales se tienen: creación dinámica de imágenes, carga de archivos y envío de correos electrónicos.

Dentro de las capacidades más importantes del PHP, se cuenta la compatibilidad con bases de datos como MySQL, mSQL, Oracle, Informix y ODBC [21].

Es importante resaltar, que para el envío de mensajes con alertas, se ha tomado un servicio que actualmente maneja el operador Comcel, que esencialmente se basa en el envío de correos electrónicos a celulares del mismo operador, a través del servidor del operador y la red de telefonía celular 3G.

Entre los servicios que el operador Comcel ofrece encontramos los siguientes:

- Telefonía móvil sobre red 3G
- Transmisión de datos sobre red 3G
- Roming internacional
- Manejo de multimedia
- Manejo de mensajes MMS
- Manejo de mensajes SMS

La mensajería SMS que se encuentra en Comcel, es una mensajería especializada, entre las cuales están: mensajes de Comcel a Comcel, mensajes de Comcel a otro operador, SMS internacional, Internet a Comcel y Short Mail Correo eléctrico a Comcel).

El Short Mail permite a los usuarios Comcel recibir Mensajes de Texto en su celular enviados desde una cuenta de correo electrónico (email). Sólo enviando un email al número celular del usuario seguido de: @comcel.com, el mensaje es de \$363 IVA Incluido [22].

Este servicio ofrecido por Comcel puede ser permitido o denegado por el usuario. Al mismo tiempo, puede ser utilizado desde cualquier cuenta de correo electrónico. Hoy en día el operador de telefonía celular Tigo se encuentra implementando el servicio de Short Mail, mediante el envío de correos electrónicos al dominio @sms.tigo.com.co.

Diferencias básica entre SMS y Short mail.

SMS es el sistema de mensajes cortos y sistema de mensajería de texto estándar para la mayoría de proveedores de telefonía celular, son dos vías de mensajería desde y hacia los teléfonos. Los teléfonos celulares admiten redactar un mensaje y pulsar enviar, lo que genera el envío del mensaje a través de la red celular directamente al destinatario, el mensaje se descarga de manera automática en el teléfono del destinatario, lo que permite leerlo de manera rápida y fácil. Todo este proceso se realiza sin necesidad de acceso a la web.

Short Mail es una versión de mensaje basado en los mensajes de texto web, para redactar el mensaje, es necesario estar conectado a la web; luego de seleccionar la opción de enviar, el portal de correo envía los datos a un servidor web, para luego ser enviados al destinatario. Este tipo de mensajes son unidireccionales, a diferencia de los SMS.

La conversión de correos electrónicos a mensajes SMS (Short Mails) se realiza de dos maneras diferentes. A continuación se expondrán estas arquitecturas.

E-Mail a SMS mediante conexión a servidor SMTP (Simple Mail Transfer Protocol):

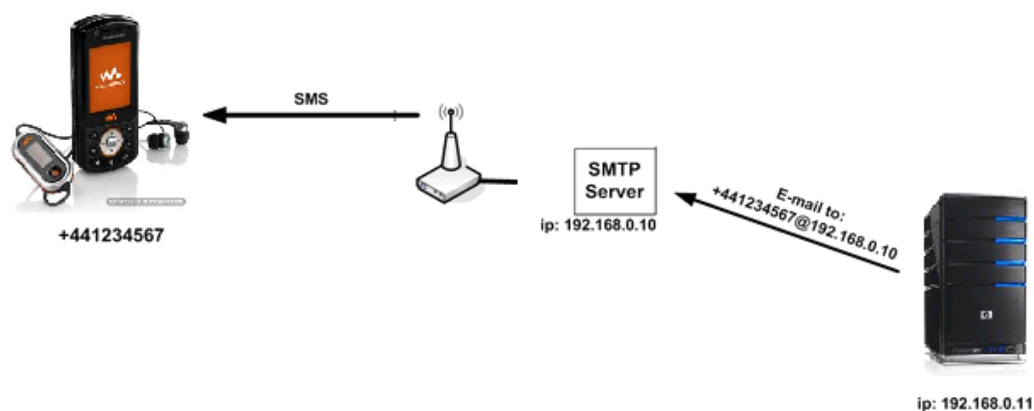


Figura 7. Envío E-mail a SMS directo

Esta arquitectura se basa en enviar un e-mail desde cualquier proveedor de cuenta de correo electrónico, a un servidor SMTP, el cual se encarga de generar el mensaje SMS a partir de un protocolo de transferencia que trabajan sobre redes de transmisión de datos.

E-Mail a SMS mediante conexión a servidor SMTP Corporativo (Simple Mail Transfer Protocol) a Servidor SMPT:

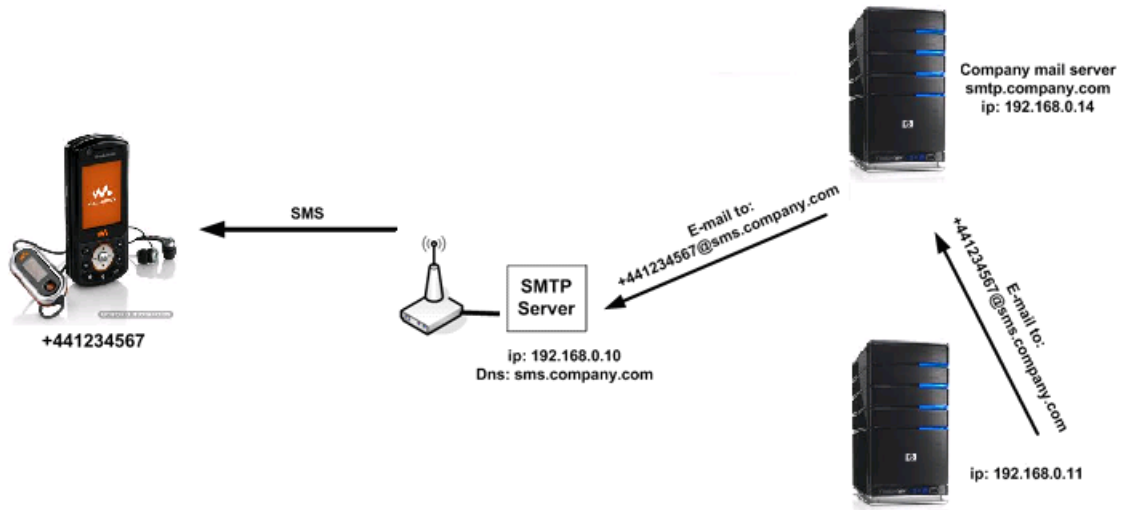


Figura 8. Envío E-mail to SMS con servidor corporativo

Esta arquitectura es una variación de la anterior ya que el e-mail se envía desde cualquier proveedor de cuenta de correo electrónico a un servidor SMTP corporativo, en donde primero se valida si los destinatarios cuentan con el servicio de recepción, luego éste se encarga de enviar la información a un segundo servidor SMTP para el cambio de e-mail a SMS mediante el uso de protocolos para transferencia de este tipos de archivos.

Ya que para el envío de la información desde los servidores hasta los celulares se utiliza la red de telefonía 3G se debe indicar que ésta, es esencialmente una tecnología inalámbrica que admite la transmisión de datos a velocidades de hasta 2 Mbps, lo que le permite ofrecer servicios multimedia inalámbricos de alta calidad y amplia movilidad. Estas cualidades permiten el uso de comunicación de voz de alta calidad, video y acceso a Internet.

El sistema de comunicación de esta red está diseñado para ser operativo, a costos razonables y en zonas de poca infraestructura.

## 2.6 Visualización de video en tiempo real

Para la visualización de videos en tiempo real, es necesario el uso de Streaming, este es la distribución o envío de video por Internet en cadenas de datos continuas con protocolos ligeros (UDP o RTSP), este envío se hace de manera continua y sin interrupción, lo que permite un almacenamiento en un buffer y, por ende, la no necesidad de descarga de archivos para visualización o escucha.

Ya que mencionamos los protocolos UDP y RTSP se hace necesario entender que el UDP (User Datagram Protocol) es un protocolo que se basa en el intercambio de información a través de una red, en la que no se necesita establecer una conexión previa o tener una confirmación de entrega de parte del receptor lo que permite el envío constante de información. El RTSP (Real Time Streaming Protocol) es un protocolo que establece y controla flujos sincronizados de datos, ya sean de audio o de video. Este protocolo actúa como un mando a distancia mediante la red para servidores multimedia [23]. Lo que permite generar mandos como play, forward, reward pausa o otras órdenes características de un reproductor de real audio.

Por otra parte, este Streaming es posible realizarlo mediante el uso de aplicaciones desarrolladas, exclusivamente, para la transmisión de video y audio en tiempo real. Este es el caso de USTREAMING página de internet que permite de manera gratuita crear un usuario para la transmisión de video y audio, en tiempo real.

El Streaming busca, principalmente, no tener que descargar los archivos de video o sonido al ordenador, lo que disminuye el uso de recursos del mismo; se basa en la arquitectura de cliente servidor, donde se tiene que este último es un servidor especializado para Streaming.

El uso de esta tecnología se ha venido incrementando, de una manera importante, en razón a que permite al usuario final ahorrar recursos de máquina y un acceso considerablemente más rápido a la información de audio y video. Otra de las razones para que el Streaming venga desarrollándose con tal crecimiento, es que éste permite grabar o capturar la información de manera local o remota, según la configuración implementada.

### **3. ESPECIFICACIONES**

#### **3.1 Dispositivos necesarios:**

- Laptop o Desktop con acceso a Internet.
- Cámara Web.
- Teléfono Celular de Tercera generación con acceso a red (3G o Wifi) y Códec de video.

#### **3.2 Programas y librerías utilizadas para crear la herramienta:**

- Windows XP, 7.
- DEV C++ 4.9 (SW libre que no requiere de licenciamiento).
- Open CV2.1 (SW libre que no requiere de licenciamiento).
- Java Eclipse (SW libre que no requiere de licenciamiento).
- Compilador Java JDK 1.5.
- Codec XVID.



### **3.3 Especificaciones básicas del PC donde se ejecutará el programa.**

- Procesador Intel o AMD doble núcleo 2 GHz mínimo.
- Memoria RAM 2 GB mínimo.
- Disco duro 80 GB mínimo.
- Códec Xvid
- Windows XP, 7.

### **3.4 Requisitos ambientales**

- Mínima luminosidad 46 ADC.
- Temperatura ambiente.

### **3.5 Formatos de video permitidos**

- .avi.

### **3.6 Otros requisitos.**

- Internet velocidad mínima 2 MB/seg.
- Plan de Datos Comcel red 3G (servicio de Short Mail activo).
- servidor de internet (24hosting).
- servidor de Streaming (Ustreaming).
- PC o laptop encendido durante el tiempo de monitoreo.

### **3.7 Parámetros variables Dentro del Código para detección de movimiento.**

- Distancia en pixeles máxima: Variable tipo Float; maxd=11;
- Distancia en pixeles mínima: Variable tipo float; mind=1;
- Variables que guarda las distancias: Variable tipo float: dx,dy;
- Variable que almacena el número de puntos: Variable tipo int: numep=0;
- Mínimo de puntos para tomar el video: Variable tipo int: minp=10;
- Número de cuadros por segundo para los videos: Variable tipo int: cps=20;
- Variable de tiempo de duración de los videos. Variable tipo Int: contuno.
- Variable de tiempo Post movimiento. Variable tipo Int: contdos.

- Variable de tiempo para detectar movimiento real. Variable tipo Long Int: mintiem\_vid.

### **3.8 Especificaciones de funcionamiento.**

- Detecta movimiento en algún lugar asignado por el usuario y graba clip de video durante el movimiento.
- Carga las imágenes a un servidor en Internet, en donde va a quedar guardado todo el historial.
- Envía un mensaje de texto a un teléfono móvil de Comcel cuando el sistema detecta movimiento.
- Permite el acceso a las imágenes captadas por la cámara Web, en tiempo real.
- Graba las imágenes obtenidas por la cámara Web durante todo el tiempo que está funcionando la herramienta.

### **3.9 Modos de funcionamiento.**

-Modo alarma.

En este modo el sistema monitorea el sitio de manera continua generando una alarma que le es enviada al usuario final a su teléfono celular, en el momento que se detecta movimiento en el lugar a monitorear. Alarma en la que se le informa al usuario sobre la detección de movimiento y se le indica el lugar en donde puede ingresar y descargar el video de alerta.

-Modo visualización.

En este modo el usuario puede acceder a una página Web desde un computador o desde un teléfono celular, donde se encuentra un cuadro de visualización de los videos que se están almacenando de manera constante en el servidor.

## **4. DESARROLLOS**

El Proyecto se trabajó inicialmente en fracciones diferentes, haciendo desarrollos y pruebas unitarias, para posteriormente unirlos e iniciar una fase de pruebas integrales. A continuación, se procede a explicar de forma clara y concisa qué se hizo, cómo se hizo y cómo se logró la integración de estos sistemas. La plataforma del sistema está montada en C++, y partiendo de eso se logró que este interactúe con los demás medios para así cumplir con los objetivos propuestos. Las fracciones en las que se dividió el desarrollo del proyecto fueron;

- Captura de imagen.
- Grabación de video.
- Análisis de video.
- Carga de archivos.
- Envío de mensajes.
- Visualización de video en tiempo real.
- Página de Internet. Servidor.

Seguidamente, se va a exponer el respectivo diagrama de cada parte del proyecto, como funciona y como se logró la integración.

#### 4.1 Captura de imagen



Figura 9. Diagrama de Bloques Captura de Imagen

En este bloque se captura video por medio del uso de una cámara web y un laptop o desktop, la imagen obtenida se procesa con Dev C++, mediante la implementación de un código que se basa en el uso de las librerías open CV y la ejecución de códec Xvid para la compresión del video.

La función que se utiliza en este código inicialmente pregunta si existe una cámara conectada en el equipo o si el video que se va a trabajar proviene de un archivo de video. Para esto se utilizó la función **cvCaptureFromCAM**. Luego de realizar la captura se le ordena al programa crear dos pantallas de visualización, la primera mostrando la captura de la cámara y, la segunda, marcando el flujo de puntos de la imagen. Se utilizó la función **cvNamedWindow**.

Después de la captura de la imagen del video, esta se almacena en un videoCuadro utilizando la función **cvQueryCuadro**. Posteriormente, se crea una variable de tipo imagen y se inicializa con el tamaño del videocuadro. Para inicializar la variable, se utilizó la función **cvCreateImage**.

Seguido a lo anterior, se definen los parámetros (ancho y alto) del tamaño de la imagen, para así mismo verificar que éstos son pares o impares. Este aspecto es muy importante a tener en cuenta, ya que OpenCv no funciona correctamente con imágenes que tengan ancho o alto de magnitud impar. Por esta razón, es significativo conocer y, dado el caso, corregir el tamaño, para así asegurar el correcto funcionamiento de la herramienta.

Para lograr esto se recurrió al algoritmo matemático modulo "%". Este algoritmo indica la paridad de un número si es utilizado de la siguiente manera;  $\text{numero} \% 2$ . Si el resultado es cero, significa que numero es par y si es 1 es impar. En caso de ser impar se le resta el valor de 1 al parámetro, y de esta forma se sabe cuál debe ser su tamaño, para posteriormente ajustarlo.

Luego de saber cuál es el tamaño de la imagen par, se crea otra variable tipo imagen y se inicializa con los parámetros ya definidos (tamaño par). Para esto se utiliza la función **cvCreateImage**. (Ver anexo1)

## 4.2 Grabación de video.

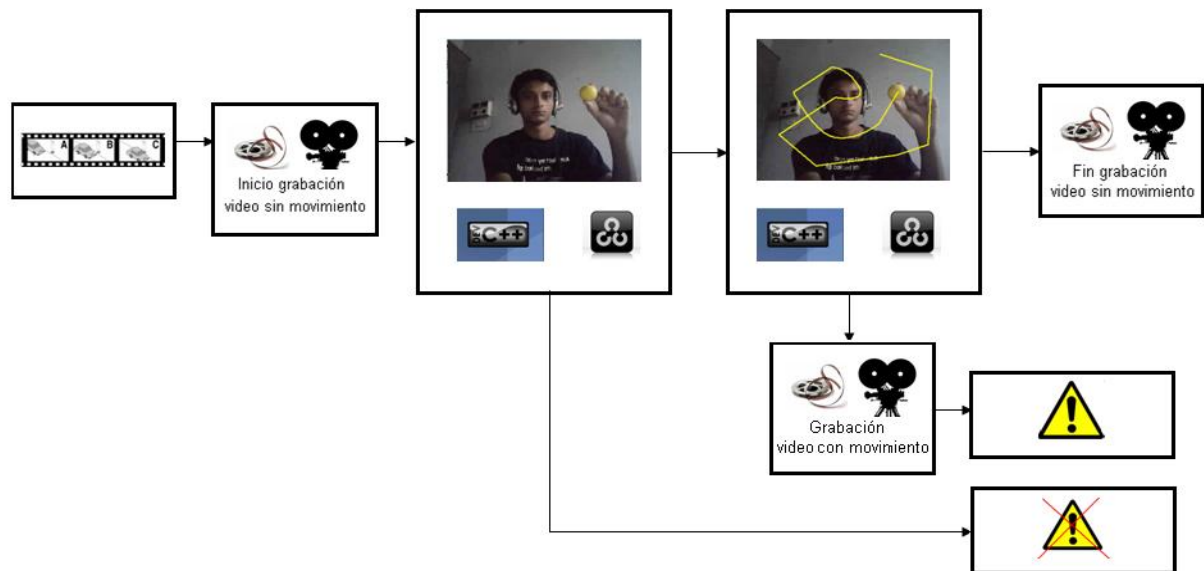


Figura 10. Diagrama de bloques Grabación de video

Para la grabación de video, es necesario crear un arreglo de caracteres en donde se va a almacenar la información del nombre del archivo. Para esto se declara una variable de tipo **char** y se inicializa con la ruta en donde se quiere que quede guardada en el disco duro y con el nombre que se le quiera asignar al video. En este caso se hizo de la siguiente manera; "**char** nombreuno[100]="C:\\TesisFTP\\video".

En seguida, se define el formato del video según los códec que se están utilizando. Se crea un apuntador de tipo **CvVideoWriter**, nombrado como **creavideouno**, en donde se va a almacenar el video antes de guardarlo en disco. En este punto inicia el análisis de video, que detecta el movimiento, y se hace el mismo procedimiento para crear el archivo de video movimiento y para almacenar las imágenes en una variable **creavideos** de tipo **CvVideoWriter** antes de guardarlas en disco. Una vez se cumplen los tiempos establecidos de grabación o se termina de grabar el video de movimiento, se procede a almacenar el video en disco local. Para esto se utiliza la siguiente función de la siguiente manera: Por ejemplo: **cvReleaseVideoWriter(&creavideouno);**

(Ver anexo1)

### 4.3 Análisis de video.

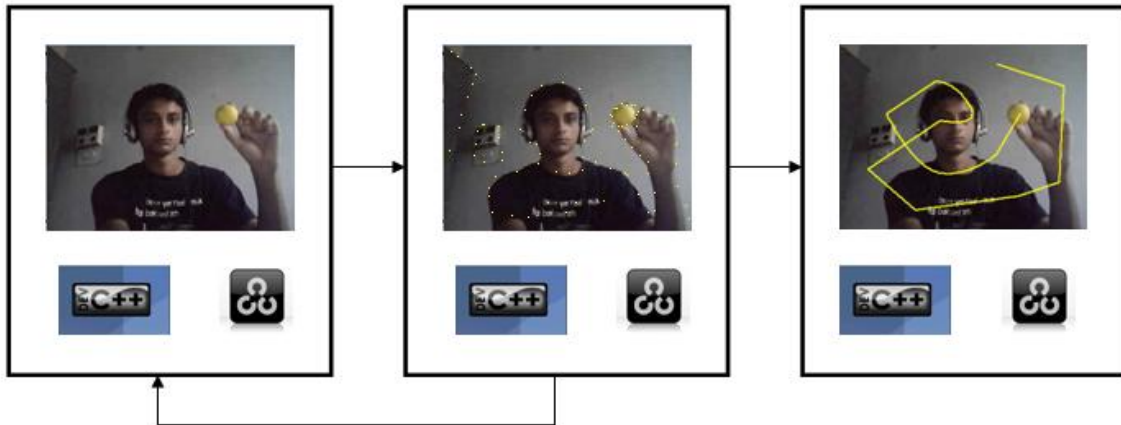


Figura 11. Diagrama de Bloques Análisis de Imagen

Una vez se tiene la variable **creavideos** que es en donde se va a almacenar el video antes de guardarlo en disco, se procede a hacer lo siguiente. Se crean variables tipo imagen para almacenar las imágenes temporalmente. Para detectar el movimiento, se está utilizando el método *good features to track*. Como ya se mencionó anteriormente, este método encuentra puntos determinantes en la imagen, para que posteriormente se le pueda hacer un seguimiento. Lo primero que se debe hacer es cambiar la gama de colores de la imagen a la escala de grises. Es decir, dejar la imagen en blanco y negro para que el rango de niveles de intensidad sea de 0-256.

Una vez realizado lo anterior, se definen los parámetros a tener en cuenta para la función GFTT. Estos parámetros corresponden a la distancia mínima que puede haber entre dos puntos, y a la división entre los valores propios de la matriz. Usando la función **cvGoodFeaturesToTrack**, se encuentran los mejores puntos para seguir. Con la función **cvFindCornerSubPix** se encuentran las esquinas. Luego de encontrar las esquinas se calcula el flujo óptico piramidal, mediante la función **cvCalcOpticalFlowPyrLK**. Esta función retorna los puntos que encontró de la imagen actual en la siguiente. Luego los puntos se pasan por un filtro para validar que cumplan con las condiciones máximas y mínimas de distancia, para que de esta forma, se pueda evitar seguir puntos que realmente no son y se pueda disminuir el ruido. Posteriormente a esto, se pasa por otro filtro para validar que los puntos que lograron pasar por el filtro anterior, sean suficientes en cantidad, según el parámetro de minp, que es el número mínimo de puntos que debe encontrar para que sea un movimiento válido. Esta función de flujo óptico se calcula siempre que se captura una nueva imagen. A medida que van llegando las siguientes imágenes y se calcula flujo óptico, se van perdiendo los mejores puntos para seguir, pues no siempre los encuentra todos. Por esto, se decidió volver a llamar la función **cvGoodFeaturesToTrack** cada ‘x’ número de imágenes.

Para mejorar la sensibilidad del programa, se estableció una barrera de tiempo, en donde la acción de detección de movimiento se activara únicamente si el movimiento supera esta barrera de tiempo. Es decir, si la barrera de tiempo es de 2 segundos y el movimiento duró 1 segundo, éste no lo toma como movimiento. Ahora bien, si el movimiento supera los 2 segundos, comienza a grabar el video en la variable predeterminada.

Para este logro, se creó un arreglo de tipo **IplImage**, en donde se almacena, de forma temporal, las imágenes captadas antes de superar la barrera de tiempo. Cuando el movimiento logra superar la barrera de tiempo, estas imágenes almacenadas se copian a la variable **creavideos**, y se continúa el proceso para grabar normalmente hasta que se termina el movimiento y se ha superado otra barrera de tiempo

final en la que ya no exista movimiento. Es decir, graba una cantidad “x” de imágenes, luego de que se ha terminado el movimiento. Si durante ese tiempo el programa detecta un nuevo movimiento, lo graba y se vuelve a iniciar el conteo del tiempo para terminar de grabar el video.

Adicionalmente, para grabar el Pre-evento, se creó otro arreglo también de tipo IplImage para almacenar las imágenes que hay antes del movimiento. Se almacenan constantemente en este arreglo (según el tamaño del arreglo es la duración del pre-evento.) para que cuando se detecte un movimiento que supere la barrera de tiempo nombrada anteriormente, se concatenan las imágenes que hay en ambos arreglos (arreglo para pre-alarma y para barrera de tiempo) y se graba el movimiento. De esta manera, en el video que se carga al servidor se puede ver cuando comienza el movimiento, el movimiento como tal, y como termina.

A continuación se muestra el diagrama de flujo del programa de análisis de movimiento en el que se pueden apreciar los ciclos de este. (Ver anexo 1)

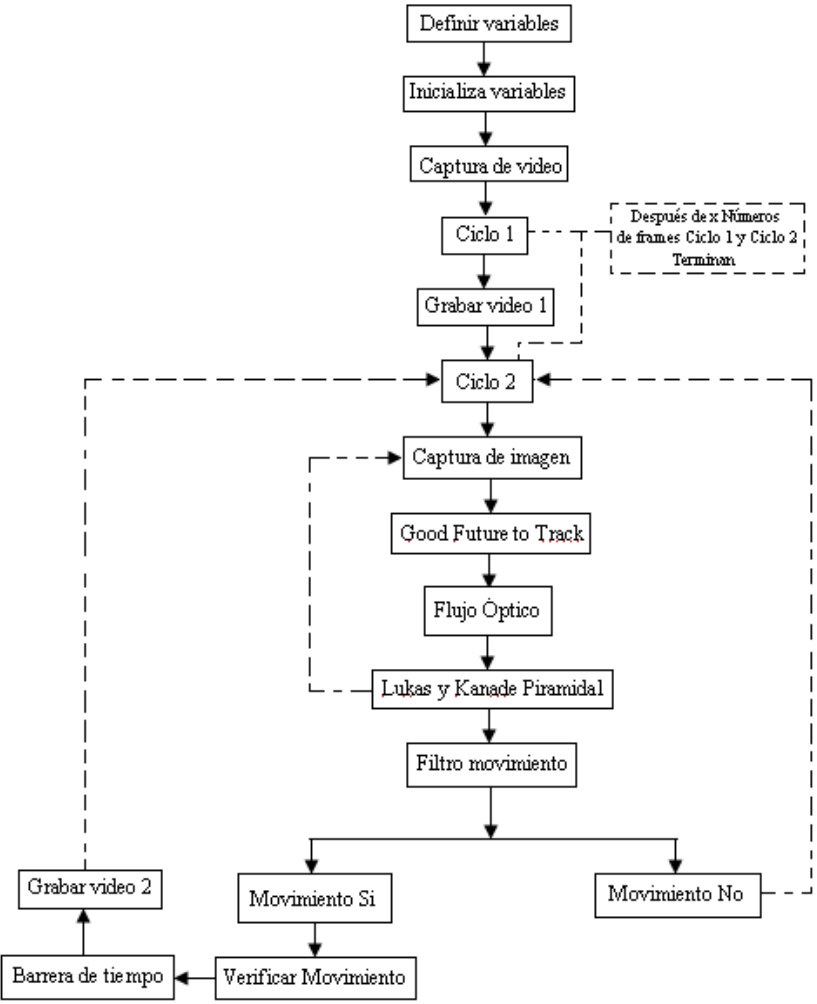


Figura 12. Diagrama de flujo programa análisis de video.

#### 4.4 Carga de archivos.

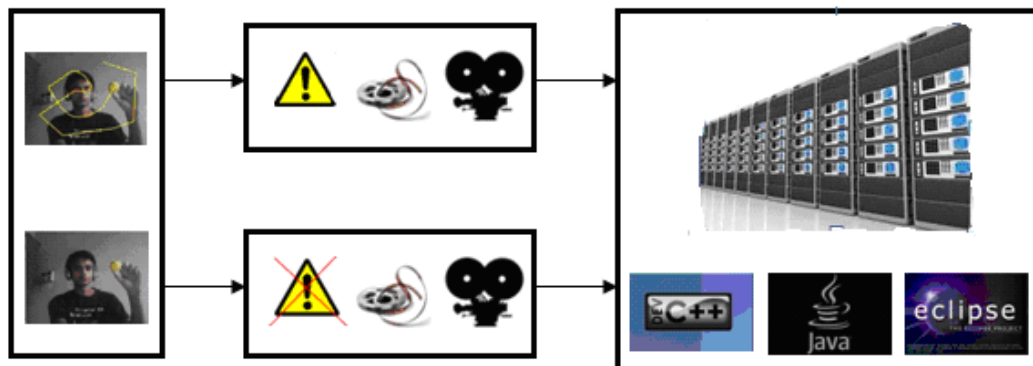


Figura 13. Diagrama de Bloques Carga de Archivos

Como se comentó anteriormente, para la carga de archivos se está utilizando el protocolo FTP. Este está hecho en el lenguaje Java, y utiliza la librería Simple FTP. Finalmente, el FTP quedó con una extensión .jar (ejecutable de Java), que será ejecutada desde C++ en el momento que se requiere. El FTP se configura principalmente en 4 partes importantes, a saber:

1. Conectarse al servidor.
2. Establecer tipo de datos.
3. Indicar ruta destino del servidor para cargar archivo
4. Indicar ruta y nombre del archivo que se va a cargar.

La primera parte es indicar el servidor al que se va a conectar, relacionando la dirección IP, el puerto por el cual se va a hacer la transferencia de datos y el nombre de usuario y contraseña. Para esto, se utiliza lo siguiente: `ftp.connect("dir ip", puerto, "nombre Usuario", "contraseña");`.

Seguido a esto, se debe establecer cuál es el tipo de datos que se va a transmitir. En este caso, se va a utilizar el tipo binario y se indica utilizando lo siguiente: `ftp.bin();`.

Luego, mediante `ftp.cwd("/httpdocs/tesis")`; se indica en qué ruta del servidor se van a cargar los archivos.

Una vez se tienen esos datos, el siguiente paso es indicar cuál es el archivo que se va a cargar al servidor. En este caso, no se tiene un archivo con nombre fijo para cargar, ya que el nombre del archivo es dinámico y siempre está cambiando. Se optó por enviar el nombre del archivo como un argumento que recibe la clase de java. Cuando se tiene definido el nombre del archivo que se va a cargar, se envía desde C++ como un parámetro que recibe el FTP. Una vez se tiene este parámetro como un argumento, se ejecuta el comando `ftp.stor(new File(args[0]));` para indicar cuál es el archivo que debe ser cargado. Este argumento se puede enviar desde C++ utilizando la siguiente sintaxis: `system("java -jar c:\RUTA\Argumento");`. (Ver anexo2).

## 4.5 Envío de mensajes

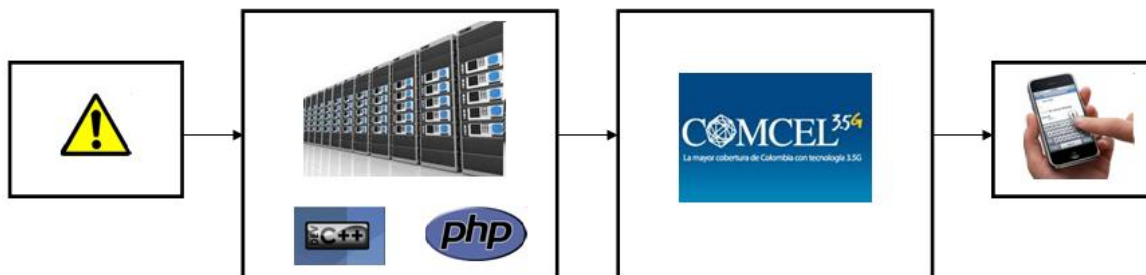


Figura 14. Diagrama de Bloques Envío de Mensajes

Cuando se genera una alerta de movimiento por el código de detección de movimiento, se envía un mensaje SMS de alerta al usuario, en el que se le indica la alerta y la dirección de internet a la que puede acceder para ver los videos de seguridad y videos de alerta de movimiento.

Esta alerta se genera desde el programa en C++, la cual abre automáticamente una página de internet, que está programada de tal forma que envíe el mensaje.

Este PHP básicamente es un código desarrollado en el block de notas, el cual nos permite validar y almacenar el teléfono de Comcel al que se le va enviar la alerta mediante la función (`$_GET['telephoneNumber'] == ""`), la cual se basa en la información enviada desde el programa de detección de movimiento de C++.

Este código está basado en 7 partes importantes, que son:

1. Valida que exista el número telefónico.
2. Adquiere número telefónico
3. Concatena el número telefónico de la siguiente manera. `# telefonico@comcel.com`
4. Ingresa el asunto del mensaje.
5. Ingresa el contenido del mensaje.
6. Envía el mensaje utilizando el comando "mail".
7. Valida si fue enviado el mensaje exitosamente.

Este código tiene un query que sirve para que el número telefónico que se adquiere, pueda ser diferente. Ahora, la dirección Url que se debe llamar desde C++ debe ser de la siguiente manera:

```
system("start http://videonlinemonitoring./sendMail-1.php?telephoneNumber=31387998xx");
```



1. Para validar que el número telefónico exista, se utiliza lo siguiente:

```
if($_GET['telephoneNumber'] == "").
```

2. Para adquirir el número telefónico se utiliza lo siguiente:

```
echo "Hola, " . $_GET['telephoneNumber'];
```

3. Para concatenar el número telefónico se utilizó lo siguiente:

```
$to = $_GET['telephoneNumber']."@comcel.com";
```

4. Para insertar el asunto del mensaje se hizo lo siguiente:

```
$subject = "ALERTA!";
```

5. Luego se deja un mensaje predeterminado, utilizando lo siguiente:

```
$body = "Hola,\n\n Alerta detección de movimiento en lugar predefinido";
```

6. Para enviar el mensaje se utiliza la función mail de la siguiente manera:

```
mail($to, $subject, $body)
```

7. Para validar que efectivamente el mensaje fue enviado y notificar al usuario, se hace lo siguiente:

```
if (mail($to, $subject, $body)) {  
    echo("<p>Mensaje enviado con éxito!</p>");  
}
```

Cuando se ejecuta el comando system() desde C++ conforme lo indicado en precedencia, automáticamente se abre un navegador de internet (el que tenga predefinido el usuario en la máquina) y se envía el mensaje instantáneamente. En la pantalla se puede ver el mensaje definido por el usuario, cuando se valida que el mensaje se envió exitosamente. (Ver anexo 3)

#### 4.6 Visualización de Video en Tiempo Real

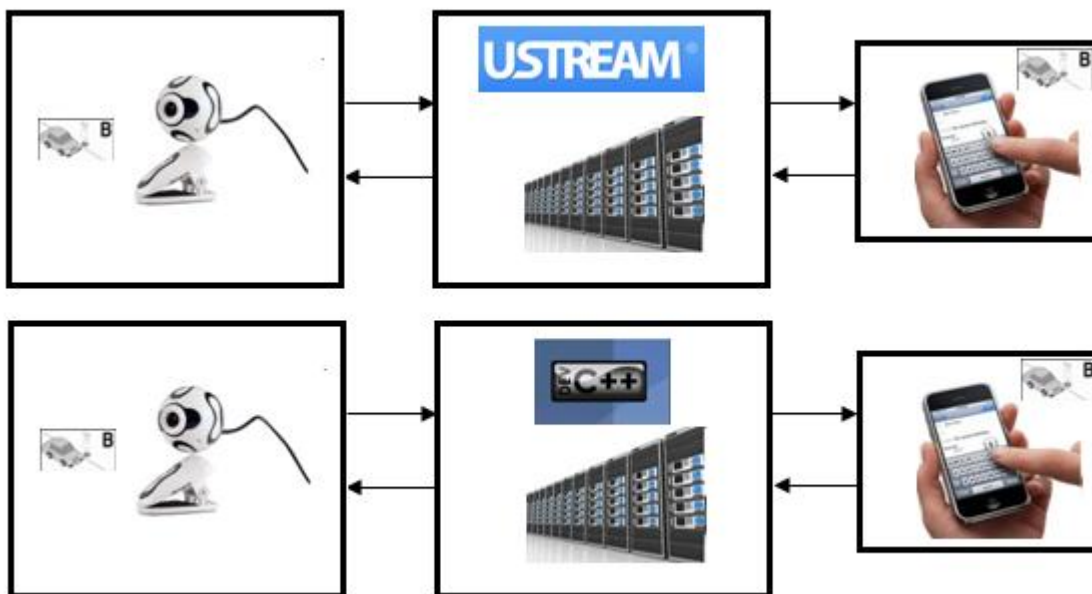


Figura 15. Diagrama de Bloques Streaming y Modo Visualización

Para realizar la visualización de video en tiempo real se tuvieron en cuenta dos opciones: la primera, la de generar un “*stream*” a partir de una herramienta ya existente en la página de Internet [www.ustream.com](http://www.ustream.com), que permite realizar un “*stream*” en tiempo real mediante el uso de una cuenta gratuita. Cuando se genera el “*stream*”, la página *ustream* genera un código HTML el cual se puede incrustar sobre el dominio [www.videonlinemonitoring.com](http://www.videonlinemonitoring.com). (Video tiempo real).

La segunda opción, consiste en realizar una grabación de video clips de intervalos cortos, los cuales son cargados en el servidor para ser reproducidos por el usuario. Este proceso es llevado a cabo por el código desarrollado en Dev C++. La idea en este método es crear 10 clips de videos secuencialmente, que serán cargados siempre en la misma ruta y que se sobrescribirán durante un ciclo que solo termina cuando se detiene el programa.

Se utilizaron 10 archivos con código HTML para lograr este ciclo. El principio es que cada uno de estos archivos ejecute uno de los clips de video, y que transcurrido un tiempo establecido, llame otro archivo HTML que ejecutara el siguiente video. Por ejemplo, el archivo `play_local_videos1.html` reproduce `stream1.avi`, y hace la recarga al archivo `play_local_videos2.html` que reproduce `stream2.avi`. Este a su vez hace la recarga llamando el archivo `play_local_videos3.html` que reproduce `stream3.avi` y así sucesivamente. (Ver anexo 4)

Los archivos HTML se cargan en el servidor. Para acceder al ver las imágenes de debe digitar la url del servidor indicando la ruta en la que se encuentran los archivos.

Por ejemplo; [www.corrientecomun.com/tesis/play\\_local\\_videos1.html](http://www.corrientecomun.com/tesis/play_local_videos1.html).

De esta manera, se puede recrear la sensación de streaming, con un retardo que depende de la duración que tiene la grabación del video y su carga. Se está hablando de tiempos dentro del rango de los 5 a 15 segundos, lo que genera un retardo normal comparado con los retardos que tienen los streaming que actualmente funcionan en la red. Ahora, el usuario debe actualizar la página cada intervalo de tiempo, para que se pueda reproducir el nuevo video y así poder ver continuidad en el video.

En el caso de utilizar la primera opción, es necesario tener exclusividad sobre las imágenes capturadas por la cámara, en razón a que si se ejecutan los dos programas al tiempo (streaming y detección de movimiento) se genera un conflicto en la captura de imagen, problema que no se ve en caso de ejecutar la segunda opción, pues ésta consiste en realizar la detección de movimiento y el streaming en el mismo programa.

#### **4.7 Página de Internet (servidor).**

Con el fin de presentar al usuario final una interfaz gráfica agradable y sencilla, que integra la mayor parte de desarrollos de este trabajo, se escogió el servidor de Internet [www.24hosting.com](http://www.24hosting.com), el cual permite generar un dominio de acceso público en Internet, en el que el usuario puede ingresar mediante la dirección de Internet [www.videonlinemonitoring.com](http://www.videonlinemonitoring.com). Ver Figura 14.

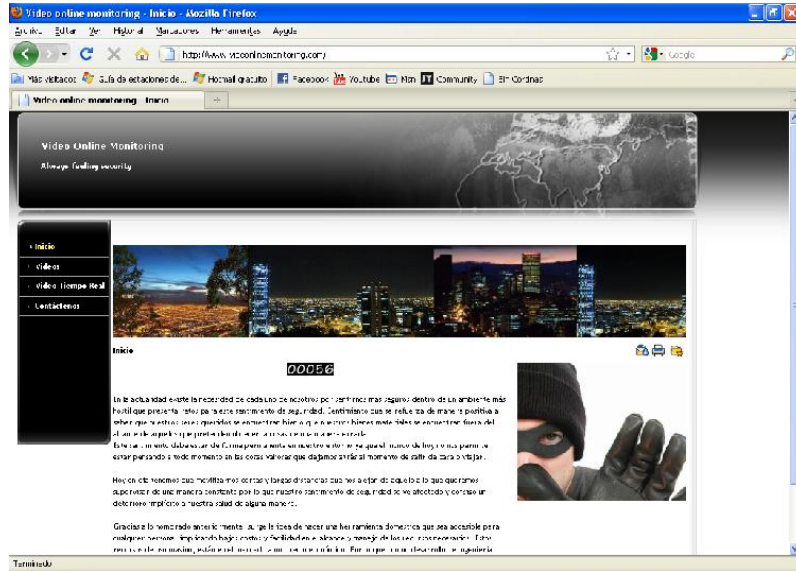


Figura 16. Site www.videonlinemonitoring.com Página de inicio

En esta pagina se presentan de manera simple los videos generados por el programa de captura de video, análisis de movimiento y carga de datos, cargados mediante los protocolos FTP.

El site consta básicamente de cuatro vínculos (Inicio, Videos, Video en tiempo real y Contáctenos). En el primer vínculo se observa una descripción básica de lo que es este proyecto, con el fin de que el usuario se relacione un poco con el tema y la herramienta. En el segundo se cuenta con dos opciones videos: de alertas y videos de seguridad. Estos son los links en donde el usuario podrá consultar los videos generados por el programa de detección de movimiento; además, allí se genera un listado de consulta de los videos con las horas de registro y en formato .avi. En el tercero, el usuario podrá realizar un monitoreo en tiempo real de la cámara Web previamente configurada para este fin. En el cuarto, el usuario encontrará un formato de contacto con el cual podrá contactar vía e-mail a los administradores de la herramienta.

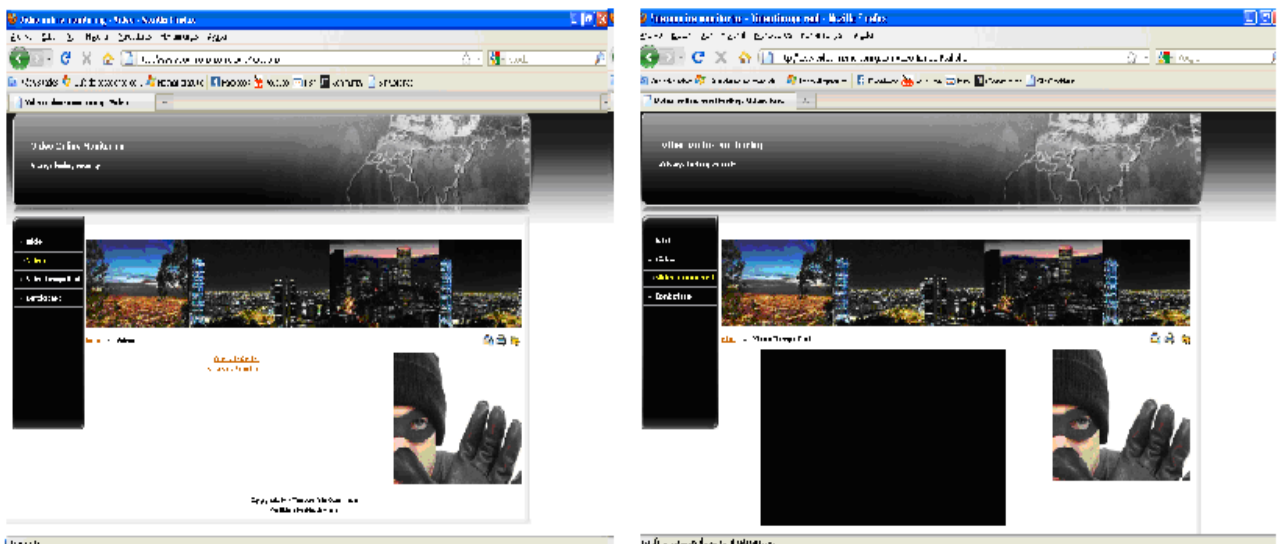


Figura 17. www.videonlinemonitoring.com Página de videos y Modo Visualización

La página fue diseñado a partir de una interfaz gráfica propia del servidor y programación HTML, mediante la herramienta Rv site Builder disponible en cPanel del servidor (Solo administrador). Ver figura 16.

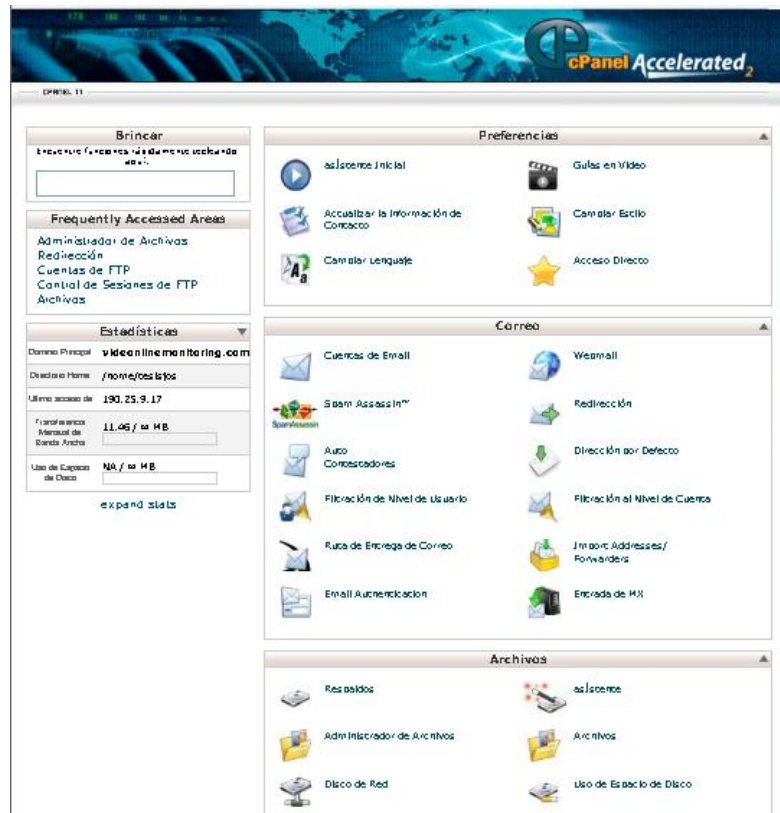


Figura 18. cPanel www.videonlinemonitoring.com

Para ver especificaciones y características del dominio [www.videonlinemonitoring.com](http://www.videonlinemonitoring.com) ver Tabla (Servicio Premium Hosting).

	» Free Hosting	» Premium Hosting
Price	\$0.00	\$4.84 / month
Disk Space	1500 MB	Unlimited Disk Space!
Data Transfer	100 GB / month	Unlimited Data Transfer!
Add-on Domains	5	Unlimited
Sub-domains	5	Unlimited
E-mail Addresses	5	Unlimited
MySQL Databases	2	Unlimited
Free domain yourname.COM, .NET, .ORG, .INFO, .CO.UK	✗	✓
Control Panel	Custom Panel	cPanel Pro, <a href="#">see demo</a>
Reseller Hosting Feature	✗	✓
	<a href="#">Order Now</a>	<a href="#">Order Now</a>
Simple Site Builder	✓	✓
Advanced Site Builder	✗	✓
Support by Phone, Live Chat	✗	✓
Assistance in Installing Scripts	✗	✓
Help in Developing Your Website	✗	✓
Backups	✓	✓
Automated Weekly Backups	Limited	✓
Uptime	99%	99.9%
FTP Accounts	1	Unlimited
Anonymous FTP Access	✗	✓
CGI Scripts	✗	✓
PHP	✓	✓
Ruby On Rails	✗	✓
SSH Access	✗	✓
Perl	✗	✓
Ruby On Rails	✗	✓
Python	✗	✓
Cron Jobs	✓	✓
FrontPage	✗	✓
Email Accounts	5	Unlimited
Web Mail	✓	✓
Mailing Lists	✗	✓
IMAP Support	✓	✓
SMTP Server	✗	✓
Autoinstaller	10 Scripts	50 Scripts (Fantastico)
Custom Error Pages	✓	✓
AWStats (Real Time Stats)	✗	✓
Webalizer Stats	✗	✓
Password Protect Directories	✓	✓
Shared SSL	✗	✓
Private SSL	✗	✓

Tabla 2. Características servidor [www.videonlinemonitoring.com](http://www.videonlinemonitoring.com)

## 5. ANÁLISIS DE RESULTADOS

Debido a que esta es una herramienta compuesta por varios sistemas, se optó por diseñar un protocolo de pruebas que incluyen pruebas unitarias y pruebas integrales. Para la pruebas unitarias se va a probar y a medir las diferentes variables que tiene cada sistema por separado y, de esta manera, encontrar ventajas y desventajas de cada uno de ellos, para que en el momento de hacer las pruebas integrales, se logre, en principio un óptimo funcionamiento de cada una de las partes. Las pruebas unitarias van a estar divididas de la siguiente manera:

- Captura y grabación de video.
- Análisis de video
- Carga de archivos
- Envío de mensajes.
- Visualización de los videos desde el browser. (Desde celular y PC).

Dentro de las pruebas integrales se va a mirar que el sistema funcione y cumpla con los objetivos descritos al inicio del Proyecto. En cuanto a especificaciones, se va a medir:

- Rendimiento.
- Tiempos.
  - Compilación.
  - Ejecución.
  - Respuesta.
- Veracidad y calidad de la información.

### 5.1 Pruebas Unitarias.

#### 5.1.1 Captura y grabación de video.

Para realizar las pruebas de captura y grabación de video, se utilizan, en principio, 2 cámaras web, cada una con diferentes especificaciones. La primera cámara que se utilizó fue una MARKVISION Focus Web Cam.

Especificaciones Web Cam 1.

- Lente CMOS de ajuste manual.
- Frecuencia de cuadro hasta 15 CPS/SVGA (1.3M).
- Compatible USB 2.0.
- Distancia de foco mínima 3cm.
- Brillo y control de exposición automático.
- Sensor de imagen 480k pixeles (800x600).

La segunda cámara que se utilizó, fue la cámara integrada del Portátil Toshiba L656.

## Especificaciones Web Cam 2.

- Lente CMOS de ajuste automático.
- Frecuencia de cuadro hasta 30 CPS/SVGA (2M).
- Compatible USB 2.0.
- Distancia de foco mínima 1cm.
- Brillo y control de exposición automático.
- Sensor de imagen 600k pixeles (1200x1000).

Lo primero que se hizo fue grabar utilizando OpenCv2.0. Cuando se capturaba la imagen con esta versión, se dañaba de la siguiente forma. La imagen se veía corrida en un pixel en la segunda fila, en 2 pixeles en la tercera, en 3 pixeles en la cuarta y así sucesivamente. Ver imagen 18.

1	2	3	4	5
5	1	2	3	4
4	5	1	2	3
3	4	5	1	2
2	3	4	5	1

Figura 19. Matriz de píxeles desplazada

Debido a esto se intentó hacer grabar utilizando OpenCv2.1, pero cuando se ejecutaba el programa no se veía imagen en las pantallas mostradas. Esto ocurrió debido a la versión de OpenCv que se estaba utilizando. OpenCv2.1 está pensado para trabajar en el sistema operativo Linux, y el archivo llamado `opencv_ffmpeg210.dll` era el que generaba el error. Para solucionar esto, fue preciso modificar un archivo que está en la versión anterior de OpenCv2.0 y reemplazarlo por el de la última versión. A continuación, vemos una imagen de lo que se podía ver sin reemplazar el archivo.

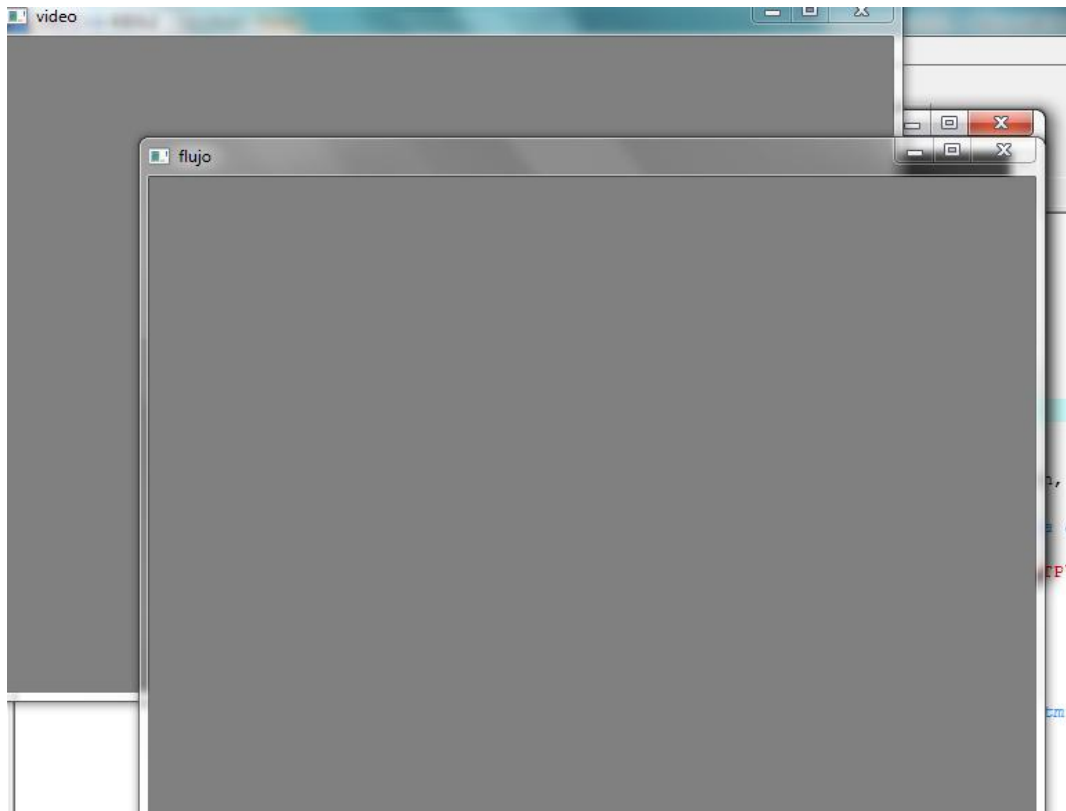


Figura 20. Pantallas desplegadas por OpenCv sin Imagen

Luego de reemplazar el archivo nombrado anteriormente, se puede ver que el programa funciona de manera correcta, en cuanto a la forma de la imagen y grabación de la misma. Ver imagen 21.

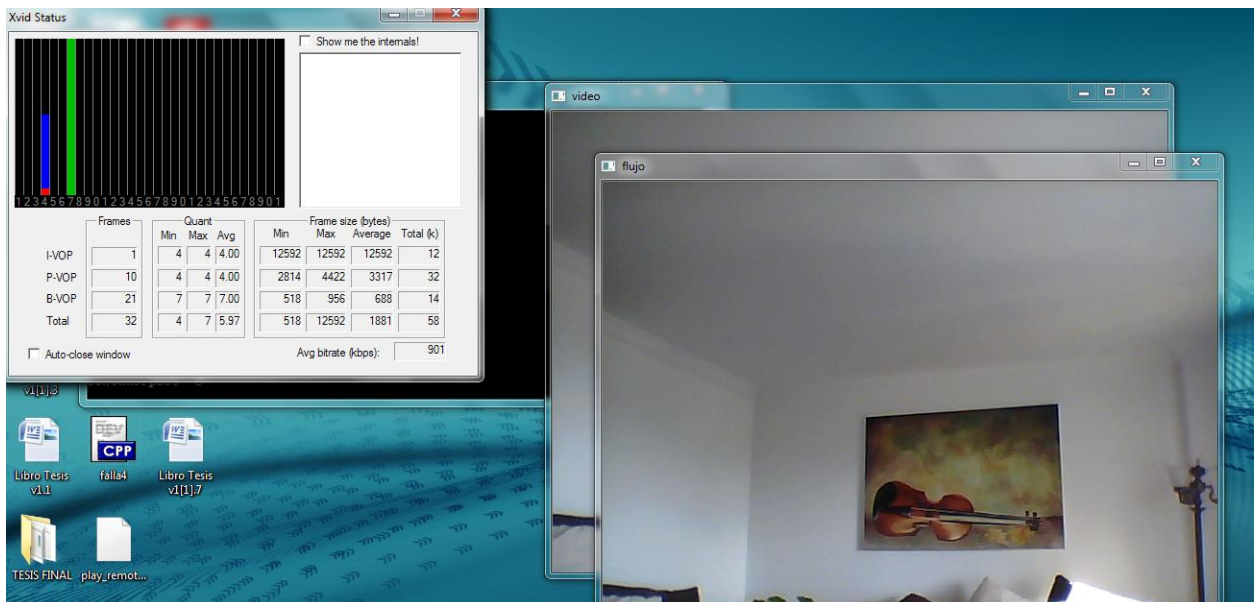
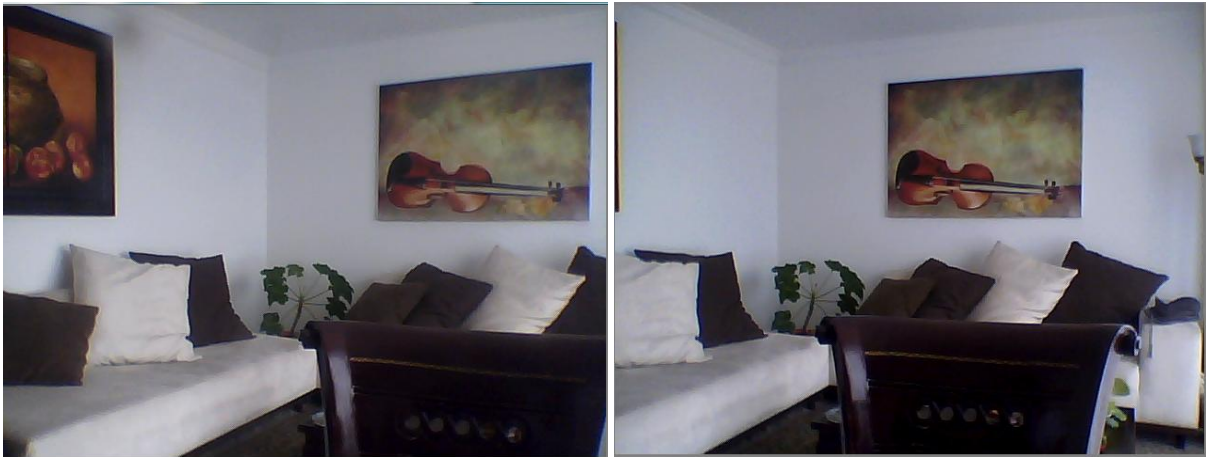


Figura 21. Pantallas desplegadas por OpenCv con Imagen

Luego de lograr la captura de la imagen correctamente, se probó con las dos cámaras web citadas con anterioridad. A continuación. Se puede ver la imagen capturada por cada una de ellas.





(a)

(b)

Figura 22. Capturas de imagen en dos cámaras Web diferentes (a) Cámara 1(b) Cámara 2.

Como se puede observar, la cámara 2 cuenta con mejor resolución y mejor contraste, lo que ayuda a que el análisis de imágenes sea más acertado. Debido a esto, se decidió utilizar la cámara 2 para el desarrollo de la herramienta.

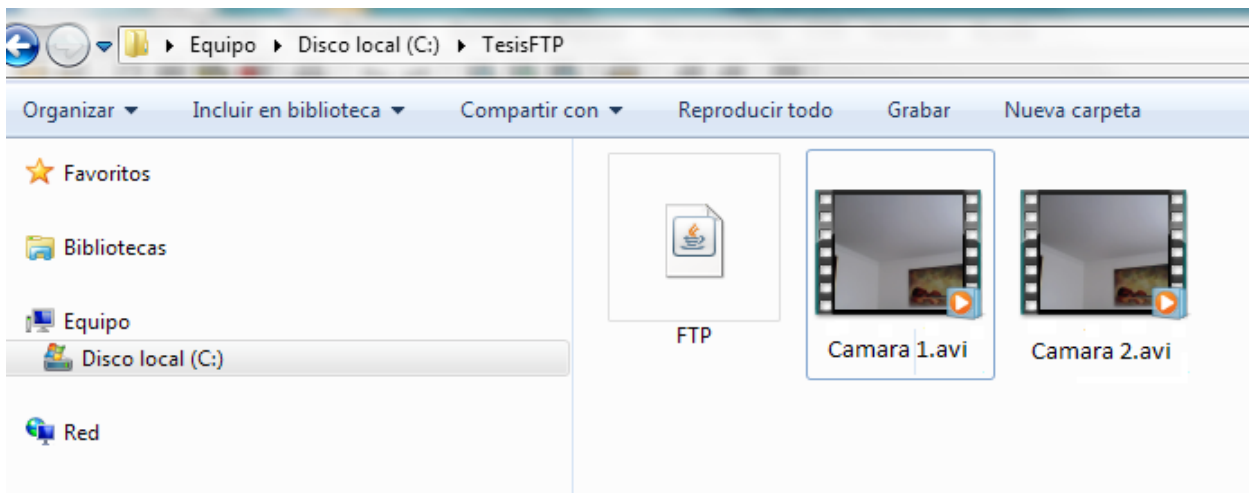


Figura 23. Archivos almacenados en formato .avi en disco duro.

En la imagen anterior se pueden observar los archivos .avi de video generado por cada una de las cámaras.

### 5.1.2 Análisis de video.

Para realizar las pruebas unitarias del programa de detección de movimiento, se buscó analizar el comportamiento del sistema, cambiando el valor de las variables que impactan más en la sensibilidad del programa. Se definieron 3 parámetros de movimiento diferentes, para así mismo establecer el rango de las variables en que el programa funciona mejor.

Los parámetros de movimiento son:

#### Cambio en la distancia

- Movimiento uniforme a distancia de 3 m.
- Movimiento uniforme a distancia de 2 m.
- Movimiento uniforme a distancia de 1 m.

#### Cambio en la velocidad.

- Movimiento uniforme a 1m/s
- Movimiento uniforme a 0.5m/s
- Movimiento uniforme a 0.2m/s

#### Duración de movimiento.

- Movimiento uniforme con barrera de tiempo 13 cuadros
- Movimiento de cámara brusco con barrera de tiempo 1 cuadro
- Movimiento de cámara brusco con barrera de tiempo 20 cuadros.

Para analizar el comportamiento del programa, cuando hay cambios en estos parámetros, se van a manipular las siguientes variables del código.

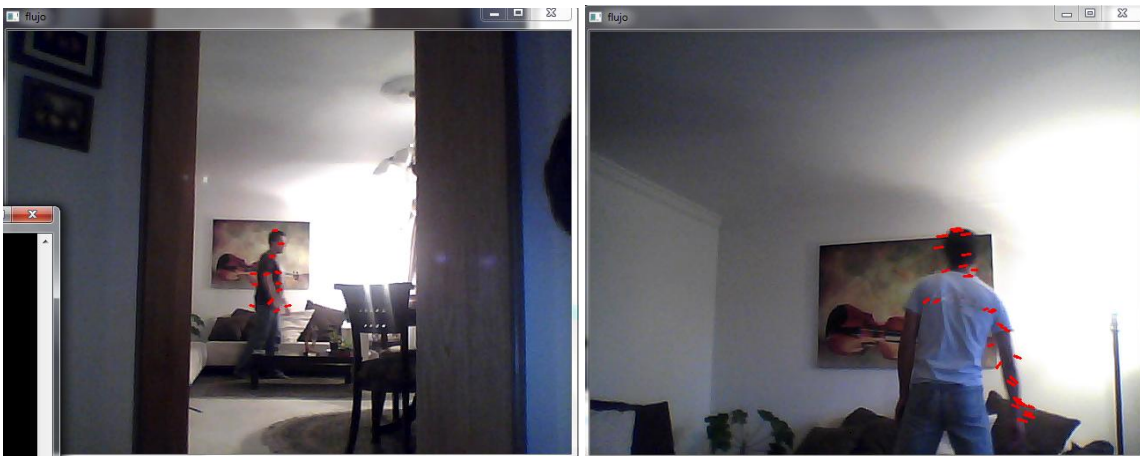
cps: Ciclos por segundo.

cal\_g: Número de cuadros que tienen que pasar para que vuelva a buscar nuevos puntos para seguir.

minp: Número mínimo de puntos que debe encontrar en el siguiente cuadro para que detecte movimiento.

mintiem\_vid: Número mínimo de cuadros que tiene que detectar movimiento para que se active la alerta.

#### Cambio en la distancia



(a)

(b)

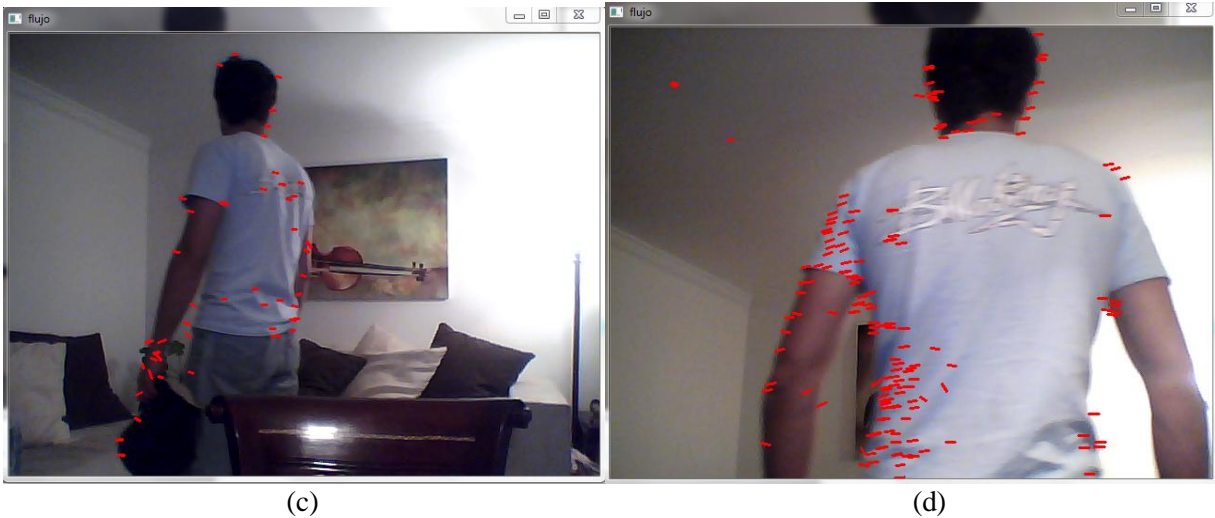


Figura 24. Captura de imagen con cambios en distancia (a) 6m (b) 3m (c) 2m (d) 1m. Con variables fijas en Cal\_g=10, Minp=12, Cps=20 y tamaño de la persona aproximado 180x50 cm.

Imagen	Distancia del movimiento a la cámara (m)	Número de puntos	Porcentaje de Imagen (%)
A	6	13	40
B	3	30	80
C	2	47	130
D	1	102	200

Tabla 3. Prueba de variación de movimiento

Como se puede observar en las imágenes anteriores, para distancias entre 1 y 6 metros no fue necesario modificar ninguna de las variables para mejorar la sensibilidad. Se puede notar claramente que entre más lejos está el movimiento, el programa detecta menos puntos. A medida que el movimiento se hizo más cerca, se detectan más puntos y la detección de movimiento se hace más sensible. A 6 metros con un movimiento uniforme a velocidad constante, el programa detectó en promedio 14 puntos. La variable minp, está en 7, por lo tanto, el programa en ésta configuración está en capacidad de detectar movimiento en cualquier punto dentro de este rango. En el caso que se necesite detectar movimiento a una distancia mayor de 6 metros, se debe bajar la variable minp, para hacer más sensible la detección. Ver imagen

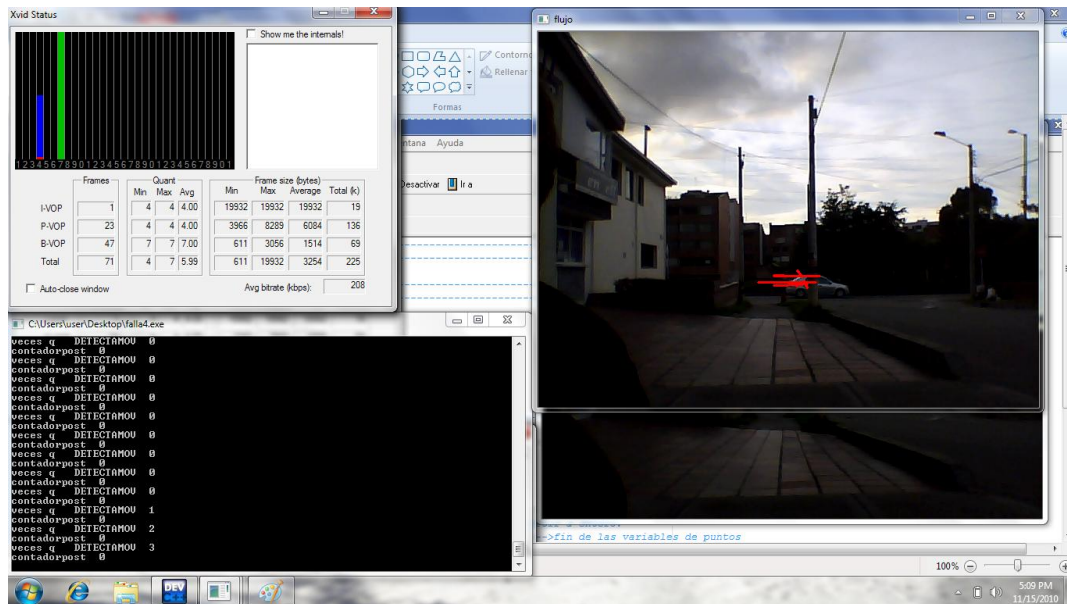


Figura 25. Prueba de distancia con movimiento a 60 m.  
Porcentaje de imagen 5%

En este caso se definió  $minp=5$ , y fue también necesario aumentar el ancho de banda del filtro dejando  $maxd=40$ , pues los autos pasaban a una velocidad promedio de 40 km/h. En la siguiente sección se va a analizar cómo se debe configurar las variables para detectar movimientos a diferentes velocidades.

### Cambio en la velocidad

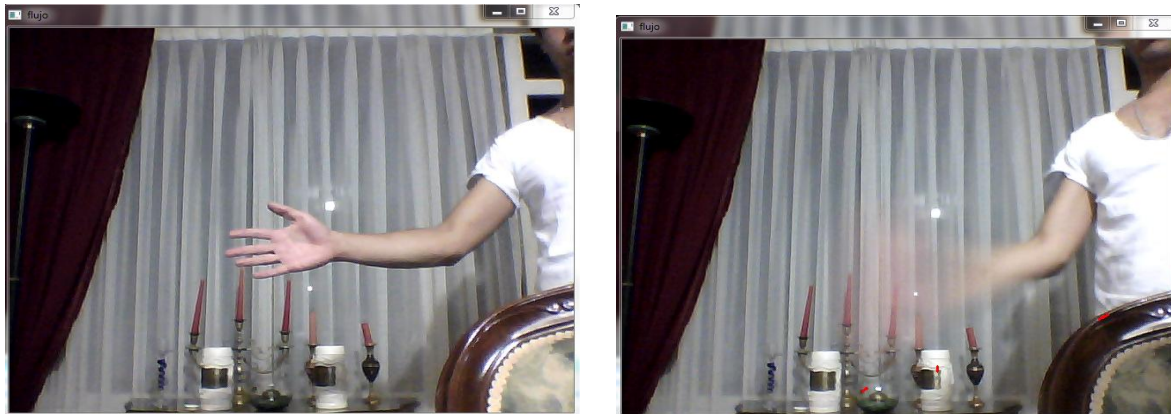


Figura 26. Cambios de velocidad con variables fijas en  $minp = 7$ ,  $maxd = 11$ ,  $mind = 5$ . Puntos encontrados 0.  
Porcentaje de imagen 20%

Las imágenes anteriores fueron capturadas durante un movimiento muy lento, y también durante un movimiento muy rápido. Como se puede ver, en ninguna imagen se detectaron puntos, es decir, el programa no detectó movimiento. Esto ocurrió debido a los filtros que involucran las variables  $minp$ ,  $maxd$ ,  $mind$ . Las pruebas que se van a mostrar a continuación se hicieron cambiándole los valores a estas variables, hasta encontrar un parámetro que definió qué valor se le debe dar a cada una para que el filtro funcione correctamente, según las necesidades del usuario.

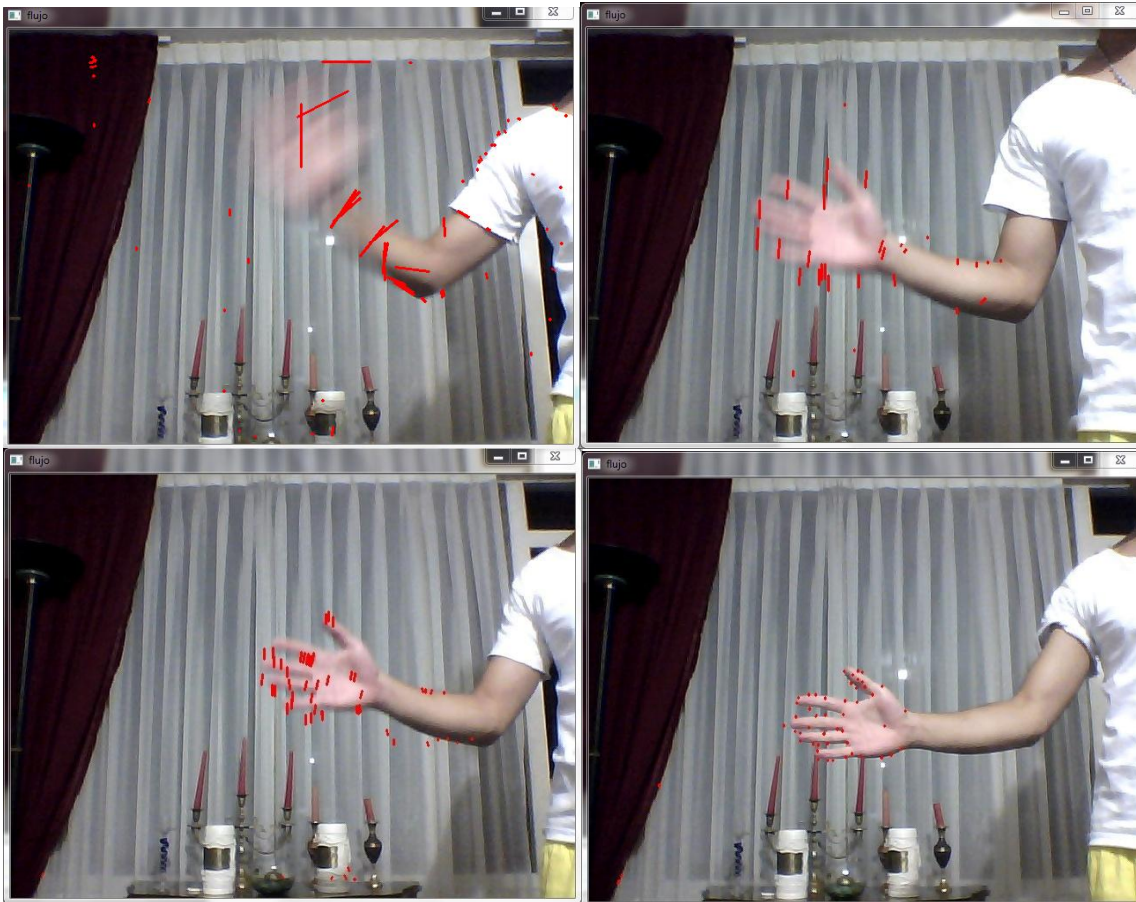


Figura 27. Pruebas para fijar filtro en el valor mas recomendado, con variación en velocidad, variables fijas en  $minp = 7$ ,  $maxd = 80$ ,  $mind = 1$ .  
Porcentaje de imagen. 25%

Velocidad Movimiento (m/s)Aprox	Número de puntos
2	15
1	30
0.5	40
0.1	50

Tabla 4. Prueba de variación de velocidad en movimiento

Para poder cambiar la sensibilidad cuando se está modificando el parámetro de velocidad en el movimiento, la variable más importante a tener en cuenta para esto son los filtros.  $Minp$ ,  $mind$  y  $maxd$ .

$Mind$  y  $Maxd$  hacen referencia a la distancia máxima y mínima (medida en pixeles) que puede existir entre el punto anterior y el punto siguiente (si lo encuentra), para que éste sea contado como movimiento e incremente el contador  $minp$ . Si  $mind$  es muy pequeño, el programa está en capacidad de detectar movimientos muy lentos, ya que un movimiento muy lento detecta el punto más cerca de su ubicación anterior cuando busca el punto en el siguiente cuadro. Ahora bien, si  $mind$  está muy alto, los movimientos lentos no los va a detectar. Para los movimientos que son rápidos, se tiene que el punto anterior y el punto siguiente (si lo encuentra) van a estar separados por una distancia (medida en pixeles) grande, por lo que si se necesita detectar movimientos de alta velocidad, se debe dejar el valor  $maxd$  en un valor grande. Ahora, dependiendo de lo que necesite el usuario, se puede configurar el filtro de diferentes formas, para registrar movimientos a determinadas velocidades únicamente. También es posible, si el usuario así lo desea,

implementar más filtros para mejorar la personalización de la herramienta. Como por ejemplo, si se desea detectar movimientos muy lentos o movimientos muy rápidos únicamente, se podría implementar un filtro para tipo de movimiento.

### Barrera de tiempo.

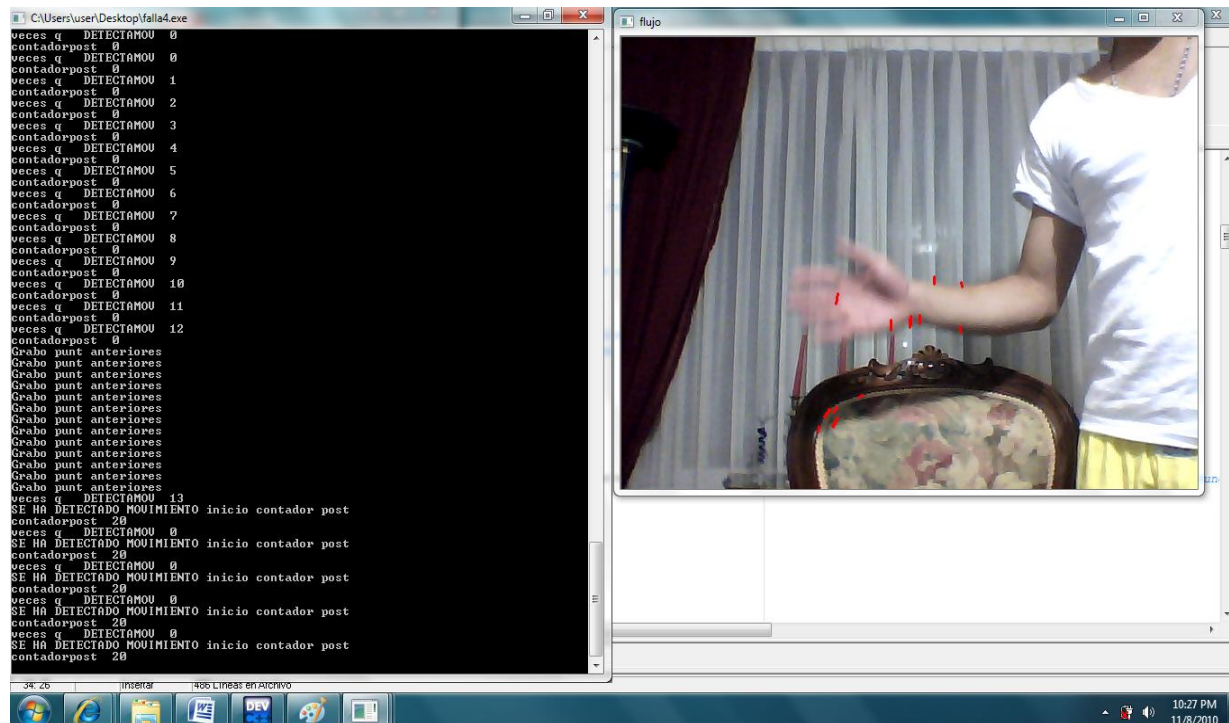


Figura 28. Con variables definidas en  $minp = 7$ ,  $maxd = 11$ ,  $mind = 5$ .  $Mintiem\_vid=13$   
Porcentaje de imagen 23%

Como la mayoría de sistemas de seguridad hacen énfasis en evitar las falsas alarmas, en este Proyecto también se tomaron medidas para que la herramienta cuente con un sistema que ayude a evitarlas y que le garantice al usuario, en un alto grado de confiabilidad, para tener la certeza de que la información que está reportando es verdadera. Este sistema también le da a la herramienta una ventaja en cuanto a acciones que pueda ocurrir con la cámara, como por ejemplo, un movimiento brusco e inesperado, un cambio de luz repentino (suponiendo que vuelve a su estado inicial), o simplemente acciones como caída de objetos, que no deben ser tomados como alarmas (cabe aclarar que el usuario podría configurar la herramienta para detectar también los hechos nombrados anteriormente). Este sistema funciona como una barrera de tiempo que puede ser configurada dependiendo de las necesidades. El sistema no genera alarma hasta que exista movimiento durante un número de cuadros predefinido. Como se puede observar en la imagen anterior, la barrera de tiempo está en 13 cuadros. Si en el cuadro número 13 ya no detecta movimiento, no se genera alarma y vuelve a comenzar el conteo. Si en el cuadro número 13 existe movimiento, quiere decir que ha superado la barrera de tiempo y se genera la alarma.

## Pruebas de movimientos bruscos.

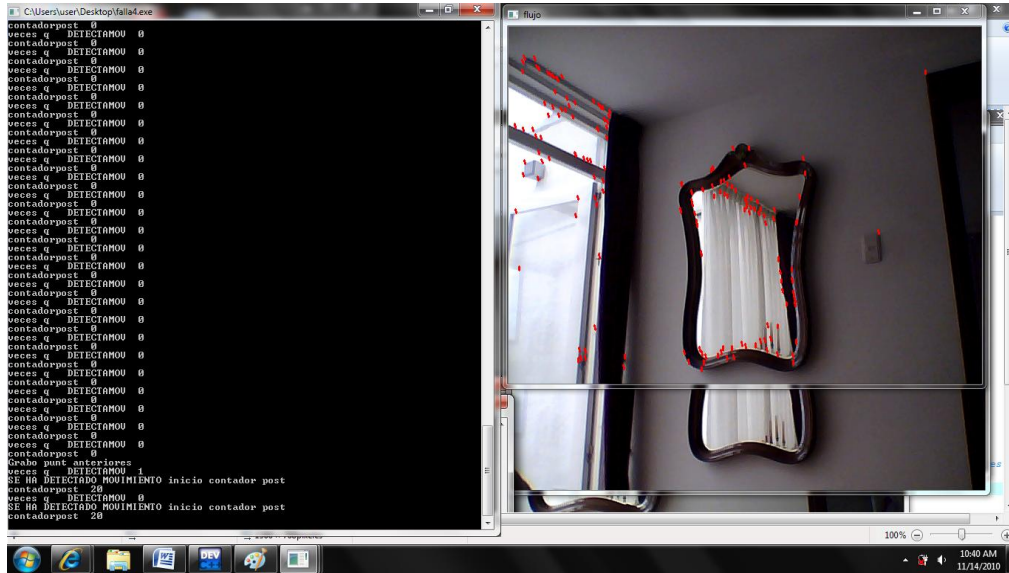


Figura 29. Pruebas de barrera de tiempo con variables fijas en minp = 7, maxd = 11, mind = 5. Mintiem\_vid=1.

En la imagen anterior se configuró una barrera de tiempo de 1 cuadro, es decir, que para que se genere la alarma debe detectar movimiento en tan solo un cuadro después. Esto lo hace sensible a movimientos bruscos que no deben ser tomados como una alarma. Para mejorar esto, solo se debe aumentar la barrera de tiempo dependiendo de las necesidades del usuario. A continuación se procura mostrar qué pasa si la barrera de tiempo es más grande y la cámara es sometida a este tipo de movimientos.

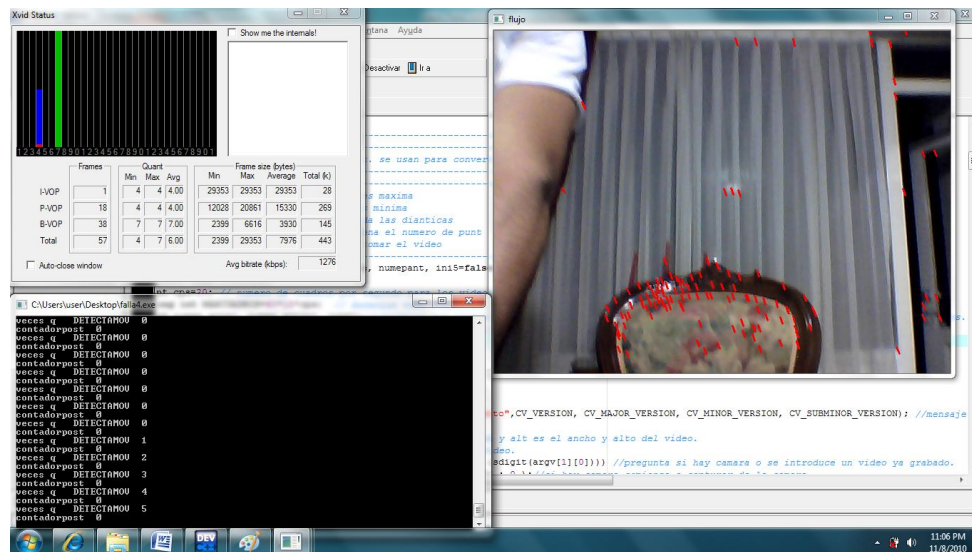


Figura 30. Pruebas con barrera de tiempo mayor, con variables fijas en minp = 7, maxd = 11, mind = 5. Mintiem\_vid=20.

En la imagen anterior se puede evidenciar claramente que la cámara tuvo un movimiento brusco y que apenas habían pasado 5 cuadros en los que detectó movimiento. Si el movimiento finaliza en ese instante, la alarma no se activa. Para que la alarma se active, el movimiento tiene que ser detectado durante 20 cuadros, (puede ser más o menos, según la configuración que se tenga dependiendo de las necesidades del usuario).

## Cambios bruscos de luz

Para la prueba se vario la intensidad de luz de forma extrema (La prueba se realizo con cambios de luz natural o artificial de intensidad normal, a oscuridad parcial y total. Del 0% al 100% donde 0% corresponde a 10, y 100% a 256). Dependiendo de la barrera de tiempo se puede o no generar una falsa alarma. En caso de que la luz vuelva a su intensidad normal antes de que la barrera de tiempo sea superada no existe alarma. Si la luz cambia y permanece así durante un tiempo mayor a la barrera de tiempo, se puede generar una falsa alarma.

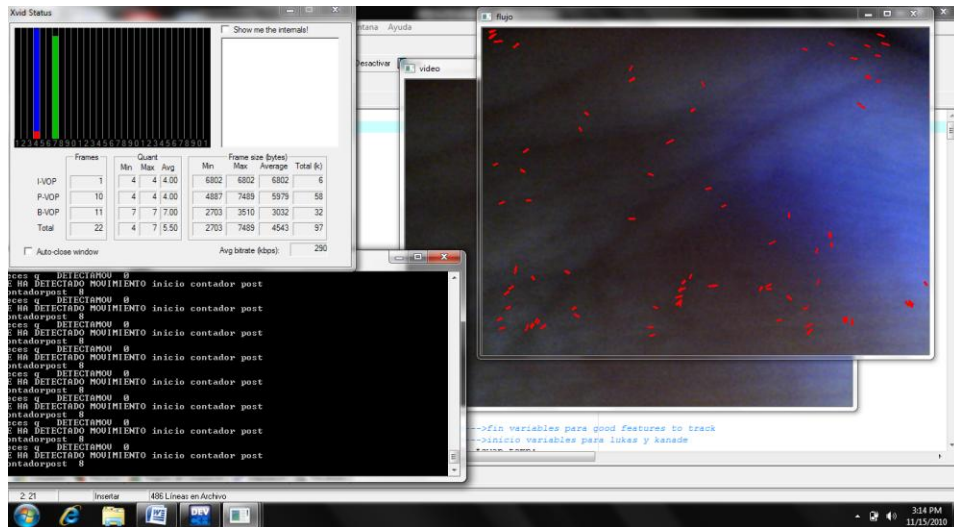


Figura 31. Pruebas de cambio de luz  
Intensidad de luz 15%

Luego de realizar las pruebas mencionadas anteriormente, se obtuvieron los siguientes resultados en cuanto a la sensibilidad. En la siguiente tabla se puede observar cómo se comporta la sensibilidad de la herramienta ante la variación de las variables.

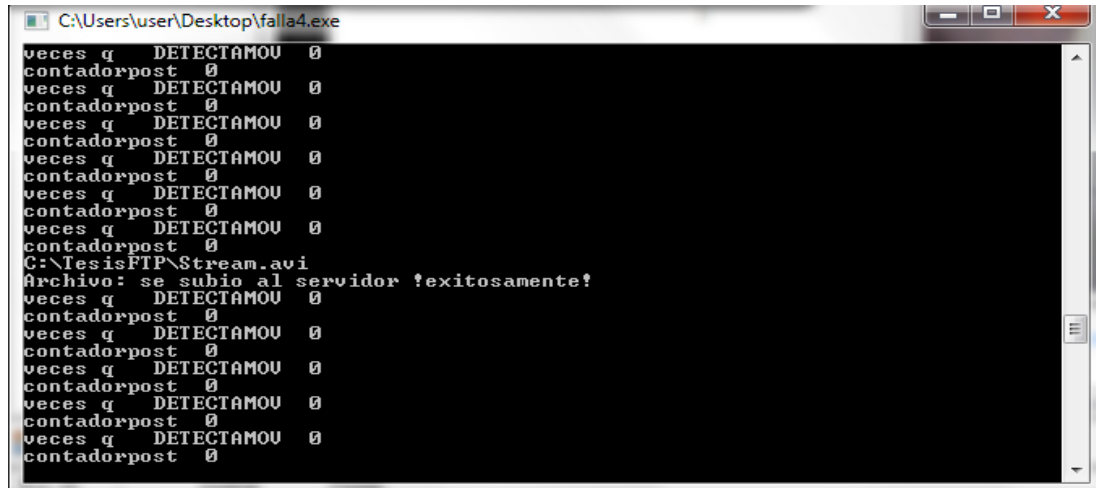
			Sensibilidad
Variable	--	++	Descripción
Minp	Aumenta	Disminuye	Mínimo de puntos que debe encontrar para que exista movimiento.
maxd	Disminuye	Aumenta	Estas variables hacen referencia filtro. Si el ancho de banda del filtro es más grande, la sensibilidad es mayor. Si mind es muy pequeño, se hace extremadamente sensible.
Mind	Aumenta	Disminuye	
mintiem_vid	Aumenta	Disminuye	Se controla la duración de la Barrera de tiempo.
Cps	Disminuye	Aumenta	Variable para controlar los cuadros por segundo que la herramienta analiza. Si se aumenta el valor de cps, la sensibilidad mejora, pero se hace más lento
cal_g	Aumenta	Disminuye	Numero de cuadros que deben pasar para que busque nuevos puntos para seguir. Si aumenta es más rápido y pero menos sensible.

Tabla 5. Análisis de sensibilidad



### 5.1.3 Modo Visualización

El primer punto a tener en cuenta para hacer la pruebas, es que el ftp que se ejecuta desde c++, este cargando los videos exitosamente al servidor. La imagen que esta acontinuacion muestra que el video stream.avi se cargo exitosamente.



```
C:\Users\user\Desktop\falla4.exe
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
C:\TesisFTP\Stream.avi
Archivo: se subio al servidor !exitosamente!
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
```

Figura 32. Carga de archivos con FTP ejecutado desde C.

En el mismo instante que se carga el archivo se puede ingresar la siguiente url, en donde se puede observar el video cargado recientemente. Ver figura 33.

[www.corrientecomun.com/tesis/play\\_local\\_videos.html](http://www.corrientecomun.com/tesis/play_local_videos.html)

ver anexo para ver codigo fuente html.

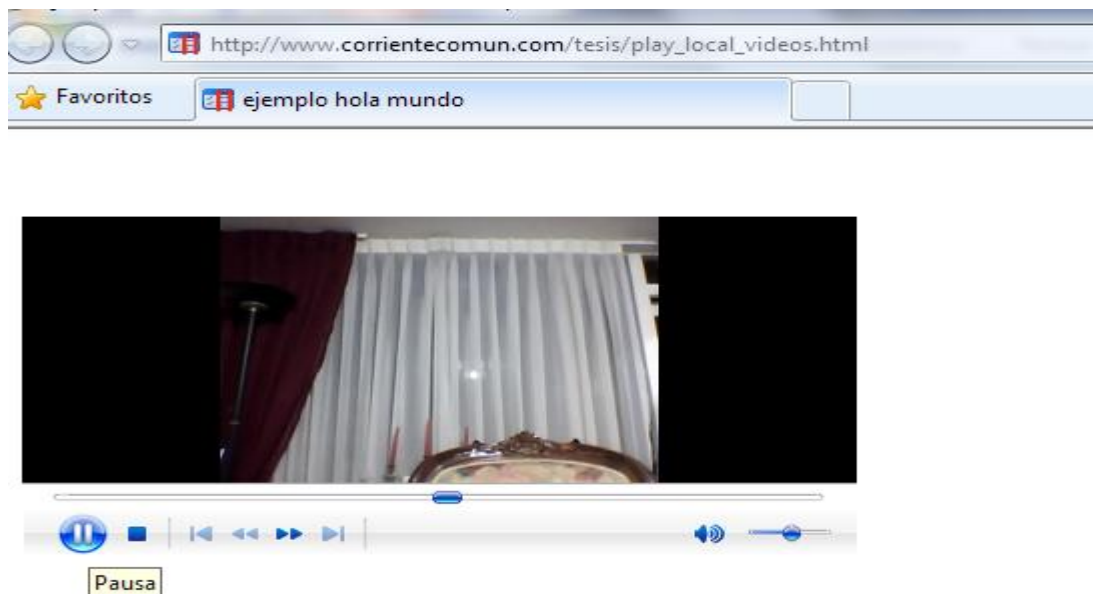


Figura 33. Prueba de modo visualizacion.

En la imagen mostrada anteriormente, se puede observar el modo visualización funcionando. Para estas pruebas se utilizo un tiempo de recarga de la pagina de 30 segundos, y se cargaron videos de 14 segundos.

Si pasados los 30 segundos, no se ha cargado el nuevo video, la pagina reproducirá el mismo video luego de la recarga. Una vez detecta que el video fue reemplazado, reproduce el nuevo video, recreando así la sensación de streaming. La idea de dejar un tiempo muerto de 16 segundos es evitar que la carga del archivo sea al mismo tiempo que la descarga del archivo para reproducirlo, pues, cuando esto ocurre, el FTP apenas puede conectarse con el servidor, pero no puede reemplazar el archivo, ya que esta en uso (ver imagen). Este tiempo en determinadas ocasiones no es suficiente, pues la velocidad de conexión para cargar los videos no es constante, y puede que demore más o menos tiempo. Para el caso en el que se demora mas, una solución podría ser aumentar el tiempo muerto, pero quedarían intervalos de corte muy largos y se pierde la sensación de continuidad en el video.

La imagen expuesta a continuación muestra claramente el mensaje que imprime el FTP cuando intenta cargar el archivo y no lo puede reemplazar.



```
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
C:\TesisFTP\Stream.avi
java.io.IOException: SimpleFTP was not allowed to send the file: 550 The process
cannot access the file because it is being used by another process.
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
contadorpost 0
veces q DETECTAMOU 0
```

Figura 34. Error carga de archivo cuando está en uso

Para corregir este problema se decidió crear 10 videos para el modo visualización (stream1, stream2,..., stream10), para que de esta forma se pueda evitar que el archivo que se esta reproduciendo sea el mismo que se va a reemplazar. Adicionalmente con un “búfer” de 10 videos se está mejorando notablemente la sensación de *streaming*, con la contra del retardo que se hace más grande. El retardo máximo es cuando se inicia la visualización y el programa está cargando el archivo stream10.avi.

El tiempo de descarga del video oscila entre 15 y 16 segundos con conexión banda ancha de 2 MB. Este es el motivo por el cual se debe esperar 30 segundos para hacer la recarga y no 14 segundos que es lo dura el video.

#### 5.1.4 Carga de archivos

Para probar la correcta carga de archivos en el servidor, se realizaron 4 pruebas de manera progresiva. La primera prueba se basó en cargar los archivos a través de la herramienta propia del servidor, la cual se encuentra dentro del cPanel (panel de administrador del servidor). Esta permite cargar archivos de diferentes tipos dentro de las carpetas del servidor, usando una aplicación en la que se indica el lugar del archivo dentro del PC y la carpeta de destino dentro del servidor. Esta prueba nos permitió demostrar que es posible cargar, borrar o administrar cualquier tipo de archivo dentro del servidor, por medio de esta herramienta propia del servidor.

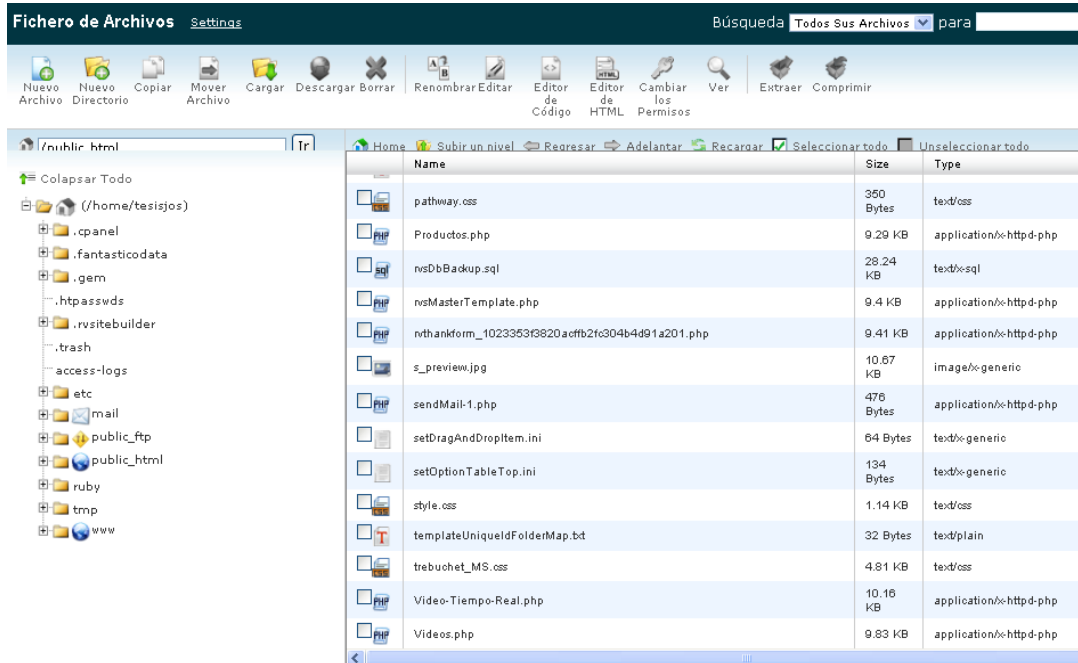


Figura 35. Archivos cargados en el servidor

Luego de validar que el servidor permite la carga de archivos por medio del cPanel, se decidió seguir con la carga de archivos al servidor mediante un programa externo (Filezilla), el cual es un programa al que se le indican la direcciones IP del servidor, puertos de transferencia, rutas de destino y origen del archivo. Filezilla posee una interfaz gráfica en la que es posible validar los archivos existentes en el servidor y en el PC, por lo que con esta interfaz, fue posible confirmar que el servidor también permite la carga de archivos con el uso de un programa externo.

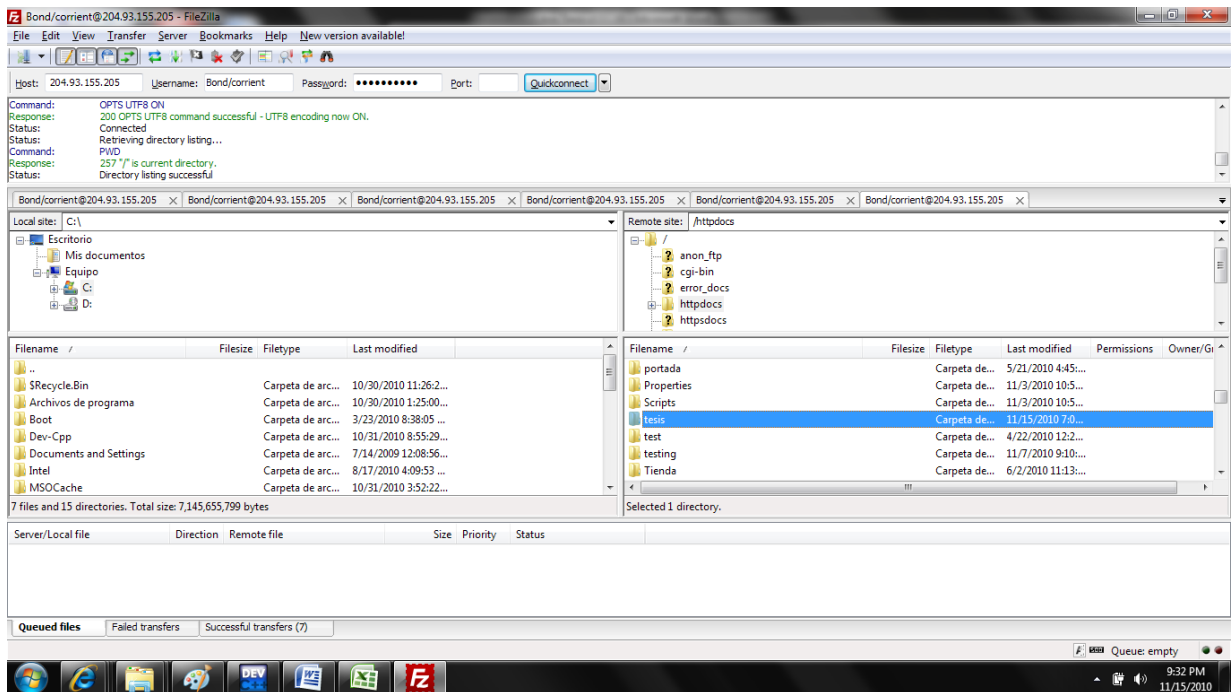
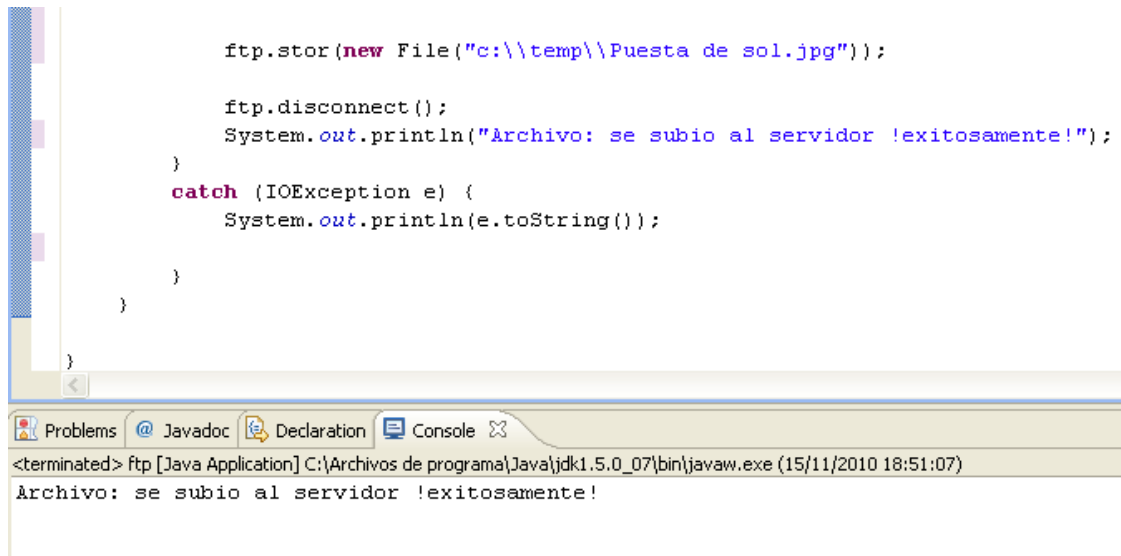


Figura 36. Pantalla de Filezilla (Carga y descarga de archivos)

Como tercera prueba, se ejecutó el FTP modificado sobre la consola de Java en la cual se pudo revisar que los archivos nuevamente fueran cargados en el servidor. Para ello, se utilizó el compilador JDK 1.5, y la consola de java Eclipse. Inicialmente, los archivos cargaban con éxito, ya que se estaba cargando siempre el mismo archivo. Para probar la carga del archivo se utilizaron 3 estrategias diferentes. La primera, fue indicar la ruta exacta con el nombre del archivo. Esta estrategia funcionó con éxito.



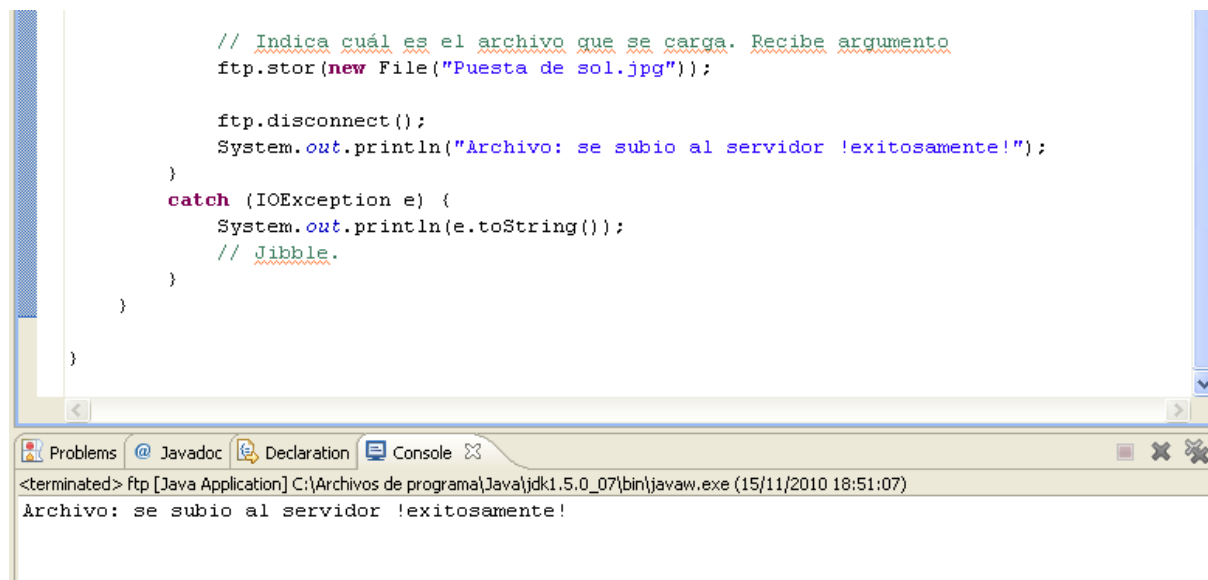
```
ftp.stor(new File("c:\\temp\\Puesta de sol.jpg"));

ftp.disconnect();
System.out.println("Archivo: se subio al servidor !exitosamente!");
}
catch (IOException e) {
    System.out.println(e.toString());
}
}
}

<terminated> ftp [Java Application] C:\Archivos de programa\Java\jdk1.5.0_07\bin\javaw.exe (15/11/2010 18:51:07)
Archivo: se subio al servidor !exitosamente!
```

Figura 37. Prueba de carga de archivos con FTP ejecutado en consola Java.

La segunda, fue incluir el archivo que se iba a cargar dentro de las carpetas del proyecto. De esta forma, no era necesario indicar ninguna ruta, solo era necesario indicar el nombre del archivo. Esta método de prueba también funcionó con éxito.



```
// Indica cuál es el archivo que se carga. Recibe argumento
ftp.stor(new File("Puesta de sol.jpg"));

ftp.disconnect();
System.out.println("Archivo: se subio al servidor !exitosamente!");
}
catch (IOException e) {
    System.out.println(e.toString());
    // Jibble.
}
}
}

<terminated> ftp [Java Application] C:\Archivos de programa\Java\jdk1.5.0_07\bin\javaw.exe (15/11/2010 18:51:07)
Archivo: se subio al servidor !exitosamente!
```

Figura 38. Prueba de carga de archivos con FTP ejecutado en consola Java y con inclusión de archivo dentro de proyecto.

Luego de probar estas 2 estrategias y de saber que efectivamente el FTP funciona para los 2 casos anteriores, se intentó cargar archivos haciendo que el FTP tenga el nombre del archivo como un argumento. Esto se hizo pensando en que el nombre del archivo nunca va a ser el mismo. En la primera

imagen se muestra como se define el argumento desde consola. En la segunda se ve que el archivo fue cargado con éxito al servidor, recibiendo el nombre del archivo como un argumento.

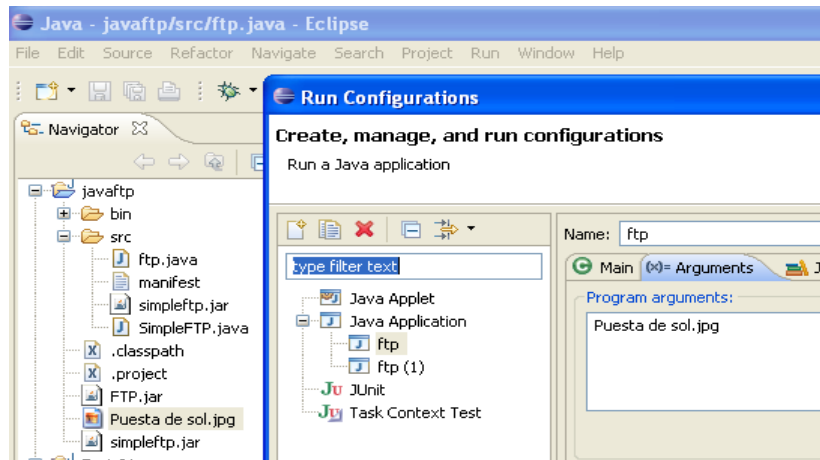


Figura 39. Prueba de carga de archivos con FTP con argumento.

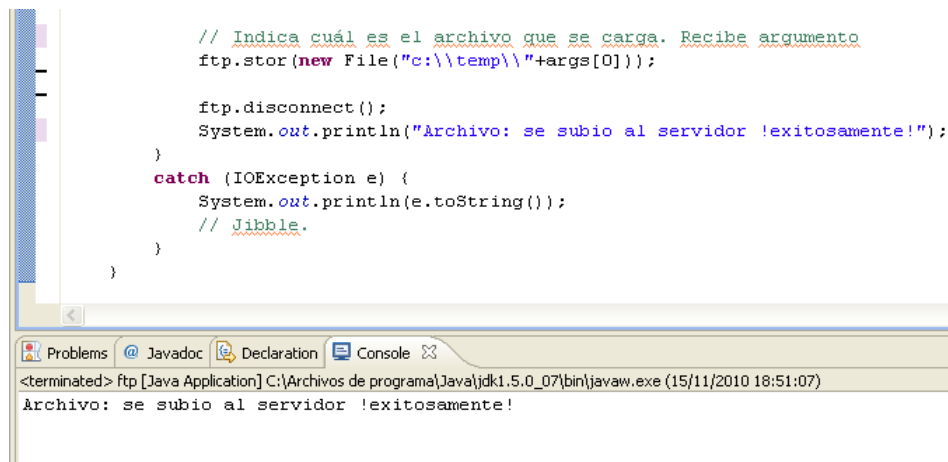
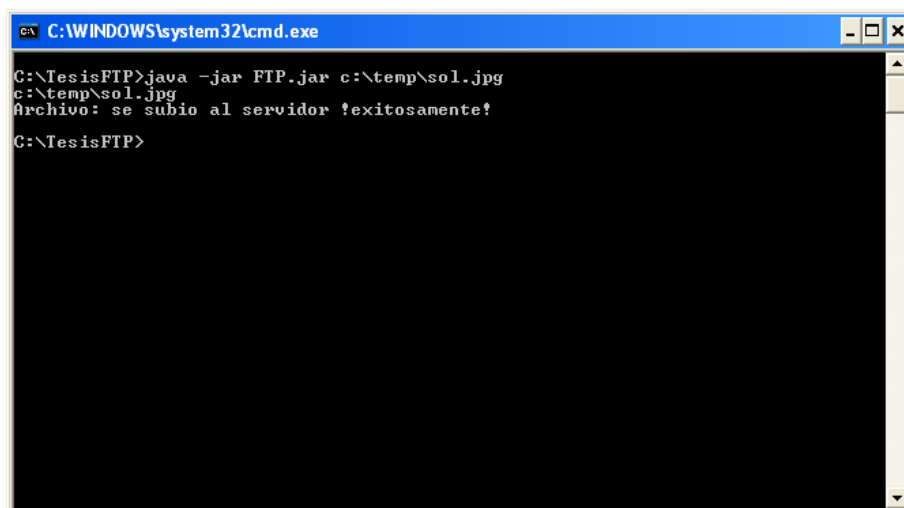


Figura 40. Prueba de carga de archivos con FTP con argumento con carga exitosa.

Se procedió a crear el .jar, dejando especificado que el archivo que se va a cargar se recibe como un argumento. Cabe aclarar, que el argumento contiene la ruta y nombre del archivo que debe ser cargado. La prueba para esto se hizo desde DOS y el resultado fue el siguiente.



```
C:\WINDOWS\system32\cmd.exe
C:\TesisFTP>java -jar FTP.jar c:\temp\sol.jpg
c:\temp\sol.jpg
Archivo: se subio al servidor !exitosamente!
C:\TesisFTP>
```

Figura 41. Prueba de carga de archivos ejecutando .jar desde DOS.

### 5.1.5 Envío de mensajes

Las pruebas que se realizaron en los teléfonos celulares se dividieron en cuatro partes, la primera parte consistió en enviar un Short Mail desde correos electrónicos normales a diferentes teléfonos celulares y operadores para validar que estos recibieran el mensaje. De esta prueba, se destacan los siguientes resultados:

- Es posible enviar e-mails a teléfonos celulares desde correos electrónicos normales.
- Es necesario trabajar con el operador de telefonía celular Comcel, en razón a que este es el único operador celular que posee el servicio de Short Mail disponible a un costo moderado; se probaron tanto Tigo como Movistar y en el momento no proveen el servicio.
- Se probó el envío de e-mails a otros teléfonos móviles con el servicio de Short Mail activo, siendo fundado afirmar, que todos los teléfonos celulares que se ofrecen en el mercado actualmente, se encuentran en capacidad de recibir este tipo de mensajes (SMS) siempre y cuando esté dentro del rango de cobertura del operador.
- Es necesario tener activado el servicio de Short Mail con Comcel para poder recibir estos mensajes (es necesario estar en un plan postpago).

La segunda parte de la prueba consistió en crear un código PHP en el que se especificó un teléfono celular de Comcel, para luego ser cargado sobre el servidor, este código luego fue llamado mediante la apertura de un navegador de Internet y la introducción de una URL ([www.tesisjoseoscar.com/le/enviomensaje.php](http://www.tesisjoseoscar.com/le/enviomensaje.php)) directa al PHP cargado sobre el servidor. Con esto se confirma nuevamente la recepción de los mensajes Short Mail en los diferentes teléfonos celulares utilizados cambiando la simcard con el número predeterminado.

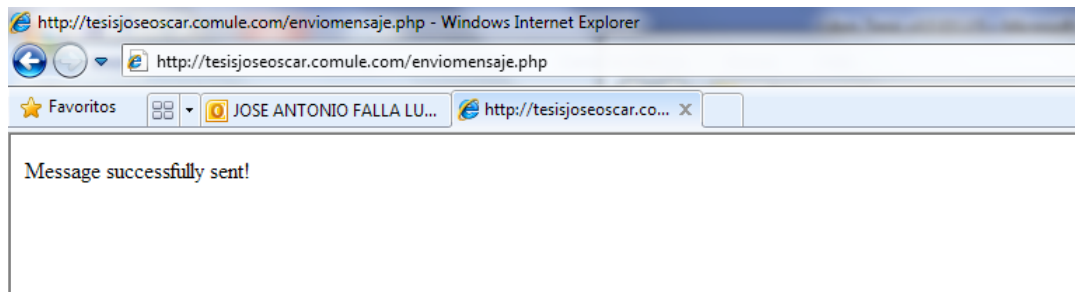


Figura 42. Pantalla de Mozilla con mensaje de envío de mensaje exitoso, mediante el uso de PHP sin Query.

La tercera parte de la prueba consistió en cargar nuevamente el PHP sobre el servidor, pero esta vez con la inclusión de un Query para generar el envío de mensajes a diferentes números celulares. Con esta parte se confirma que es posible el envío de Short Mails a diferentes números de celular de Comcel, usando un PHP cargado sobre el servidor.

La cuarta prueba se basó en el enviar el Short Mail a un celular con Roaming Internacional, prueba con la que se corrobora que el envío del mensaje de alerta (Short Mail) puede funcionar a un nivel global.

A continuación, se muestra la pantalla que se abre cuando se envía un mensaje desde el programa sin información del número telefónico.

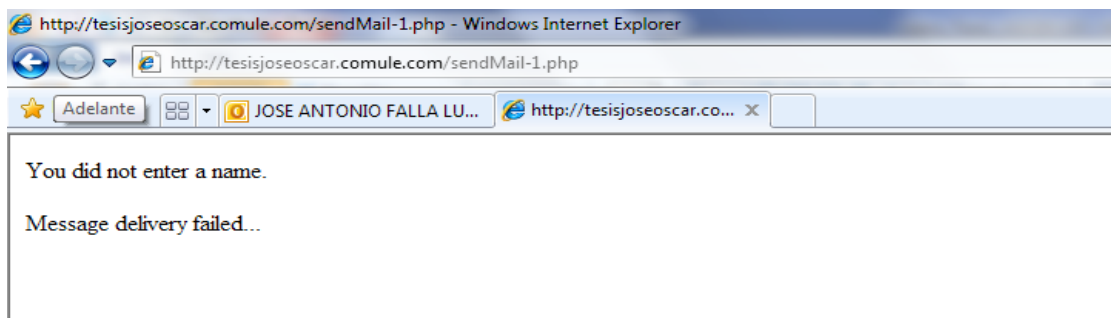


Figura 43. Pantalla Mozilla FireFox con mensaje “Message Delivery Failed”

Ahora, indicando el número telefónico en la Url, se tiene que el mensaje se envía satisfactoriamente. Ver imagen a continuación.

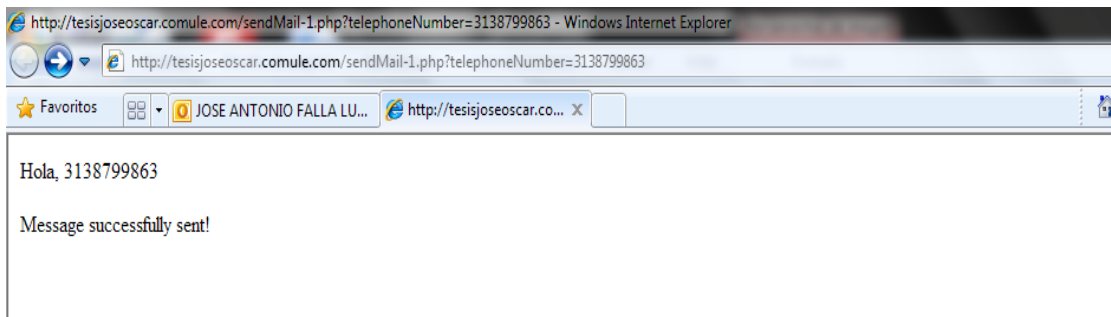


Figura 44. Pantalla de Mozilla con mensaje de envío de mensaje exitoso, mediante el uso de PHP con Query.

### 5.1.6 Visualización de los videos desde el browser. (Desde celular y PC).

A continuación se puede ver que los videos se reproducen con Windows media player. Cabe aclarar que no es necesario descargar los videos para ser visualizados.



Figura 45. Prueba de reproducción de video desde el Site.

Para probar que la página de Internet *www.videonlinemonitoring.com* y los videos cargados en esta página puedan ser visualizados o reproducidos por diferentes celulares. Se tomaron algunos teléfonos celulares con planes de datos o acceso a redes Wifi, en los que se ingresó a la página antes mencionada y se intentaron reproducir los videos. Como resultados de esta prueba, se obtuvieron las siguientes conclusiones:

- Es necesario tener activado el servicio de plan de datos del operador para poder ingresar a las páginas de Internet desde el teléfono.
- De no tener disponible el plan de datos del operador se hace necesario tener acceso a redes Wifi
- No todos los teléfonos celulares permiten o soportan la navegación en Internet.
- No todos los teléfonos celulares reproducen el formato de video .avi.
- Para poder reproducir el formato de video .avi en algunos teléfonos, es necesario la previa instalación de códec de video.

De acuerdo con las especificaciones de los diferentes teléfonos evaluados se generó la siguiente tabla, en la que es posible observar si algunos de los teléfonos móviles evaluados, poseen o no la función de reproducción de video en diversos formatos, acceso a Internet (navegación), recepción de SMS y el sistemas operativo que poseen los teléfonos.



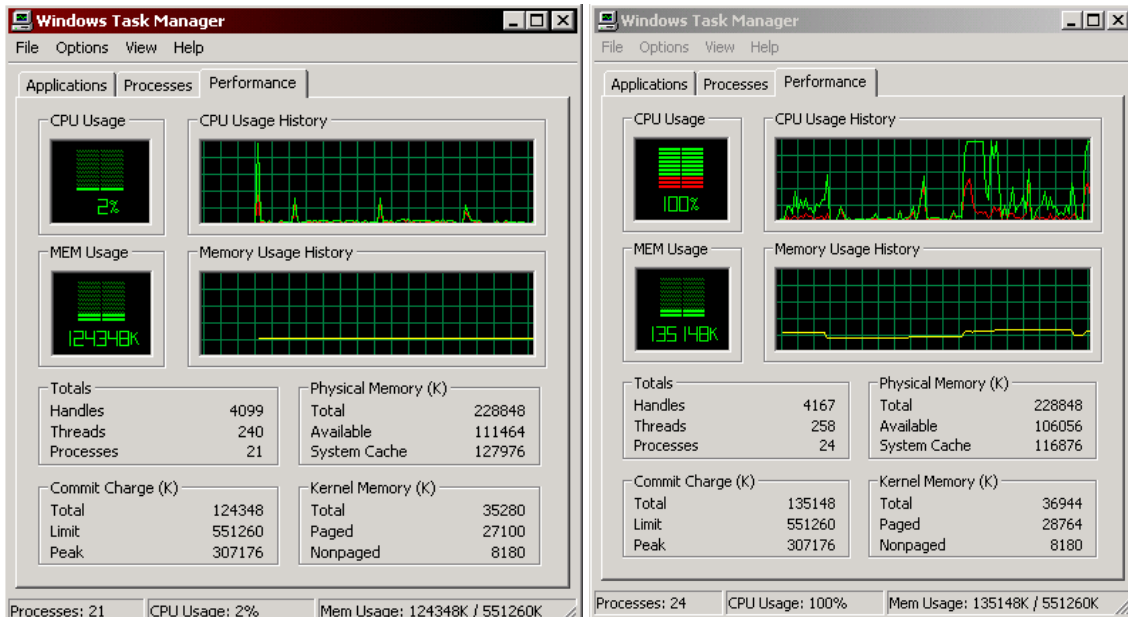
Celulares	IPHONE 4 16GB	Motorola WX290	Sony Ericsson W380-Morado	BlackBerry 9100 Pearl	Sony Ericsson W100 Spiro	HTC	Sony ericsson Vivaz
SMS	SI	SI	SI	SI	SI	SI	SI
Videos	SI	NO	SI	SI	SI	SI	SI
Internet	SI	NO	SI	SI	SI	SI	SI
SO	Vios 4	N/A	N/A	Black Berry	Windows Mobile 6.1	Android	Simbyan

Tabla 6. Características teléfonos celulares

Es necesario resaltar, que el hecho de que los teléfonos reproduzcan video, no implica que soporten el formato de video .avi manejado en este Proyecto. Por lo que se debe mencionar que para el funcionamiento de esta herramienta de monitoreo, es indispensable instalar los códec adecuados.

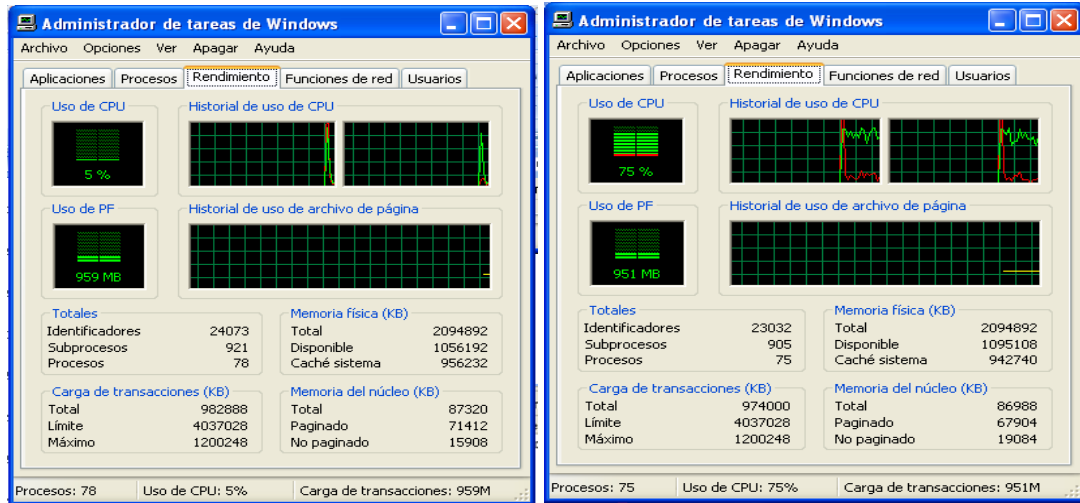
## 5.2 Pruebas Integrales

### 5.2.1 Rendimiento.



(a)

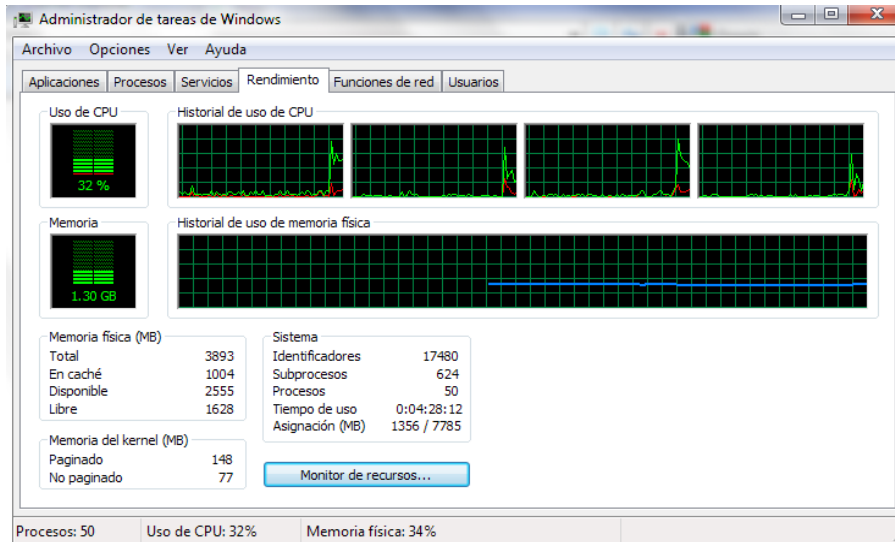
(b)



(c)

(d)

Figura 46. Pruebas de rendimiento de maquina, en equipo Celeron (a)(b) y equipo Pentium 4 (c)(d) con variables fijas en 15cps. Minp=7 cal\_g=10.



(a)

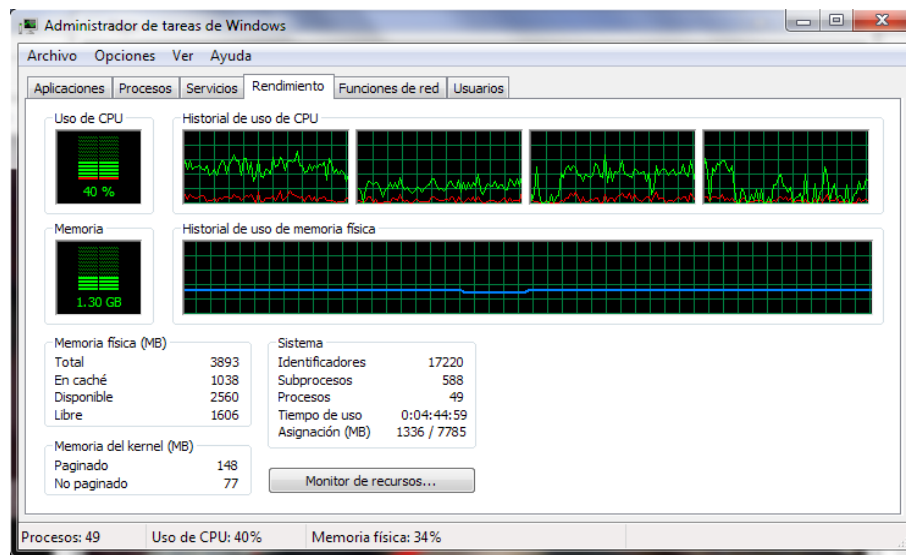


Figura 47. Pruebas de rendimiento de máquina, en equipo Core i5 con variables fijas en 15cps. Minp=7 cal\_g=10 (a)  
(b).

Procesador	Nucleos	Velocidad de Reloj (GHz)	Ram (GB)	Uso Procesador %	Sistema operative
Celeron	1	2.8	0.256	100	Windows 2000
Pentium 4 HT	2	3.0	2	75	Windows xp
Core i5	4	2.4	4	32	Windows 7

Tabla 7. Equipos en los que se realizo prueba de rendimiento.

Se realizaron las correspondientes pruebas de rendimiento de la herramienta en diferentes maquinas. Inicialmente se realizo en las maquinas con procesador Celeron y Pentium 4. El procesador se mantuvo al 100% y al 74% respectivamente la mayor parte del tiempo que se estuvo ejecutando el programa. La memoria RAM no fue impactada de forma notable, pues se mantuvo en 190mb y 1GB. El análisis de video en esta máquina “Celeron” (ver tabla) no fue eficiente, pues cada vez que detecta movimiento se satura el procesador en 100%.

Esto implica que la maquina debe tener exclusividad para el análisis de video y no estaría en la capacidad de llevar a cabo otras tareas. Cabe aclarar que esta máquina se demora en analizar el video cuadro a cuadro, lo que hace que no se analice la información completa. Es decir, si la maquina captura 30 cps, y el procesador alcanza a analizar únicamente 15, los 15 restantes no serán analizados. Esto hace que el programa pierda confiabilidad. Esta falencia también se ve reflejada al momento de ver el video capturado en la pantalla y al ver los videos guardados, pues no se ve de forma continua.

En la maquina “Pentium 4” (ver tabla), el rendimiento mejora notablemente, pues no necesita exclusividad para ejecutar la herramienta. De igual forma, se aclara que no fue suficiente el rendimiento del procesador para analizar el video capturado. Esto se evidencia en los videos obtenidos en las pruebas, pues dependiendo de la cantidad de puntos que se encuentran cuadro tras cuadro según la magnitud de movimiento, pierde continuidad en el análisis.

El mejor rendimiento se obtuvo en la maquina “Core i5” (ver tabla). Durante el tiempo que la herramienta se mantuvo en ejecución, el procesador en ningún momento sobrepaso el 40% de uso. Esto permitió ver que la herramienta es confiable.

También es de notar, que independientemente de la maquina en la que se ejecute el programa, cuando se está cargando el video al servidor, el consumo del procesador cae a un valor mínimo. Esto es debido a que OpenCv no permite ejecutar 2 procesos al tiempo. En la Figura 47 (b) se puede ver claramente que el procesador pasa de estar en un consumo mínimo a un consumo del 32%, luego de terminar la carga de un video al servidor.

### 5.2.2 Tiempos.

#### Compilación.

Procesador	Núcleos	Velocidad de Reloj (GHz)	RAM (GB)	Tiempo de Compilación (seg)	Sistema operativo
Celeron	1	2.8	0.256	15	Windows 2000
Pentium 4 HT	2	3.0	2	6	Windows xp
Core i5	4	2.4	4	1.7	Windows 7

Tabla 8. Prueba tiempos de compilación.

#### Ejecución.

Procesador	Núcleos	Velocidad de Reloj (GHz)	RAM (GB)	Tiempo de Ejecución (seg)	Sistema operativo
Celeron	1	2.8	0.256	22	Windows 2000
Pentium 4 HT	2	3.0	2	10	Windows xp
Core i5	4	2.4	4	3.2	Windows 7

Tabla 9. Prueba tiempos de ejecución.

#### Respuesta.

Procesador	Núcleos	Velocidad de Reloj (GHz)	RAM (GB)	Tiempo de Respuesta (seg)	Sistema operativo	Ancho de Banda (MB)
Celeron	1	2.8	0.256	35	Windows 2000	2
Pentium 4 HT	2	3.0	2	30	Windows xp	2
Core i5	4	2.4	4	20	Windows 7	2

Tabla 10. Pruebas tiempos de respuesta.

### 5.2.3 Veracidad y calidad de la información

Para realizar estas pruebas, se integraron todos los sistemas que contiene la herramienta. La integración se hizo bajo la plataforma de C++ en principio, y otros en el servidor. Desde C++ se logró que se ejecute el

FTP para la carga de archivos en el momento correspondiente y se consiguió que se pudiera ejecutar el código para el envío del mensaje de texto al celular.

La prueba realizada, consistió en dejar la herramienta en ejecución durante un largo periodo de tiempo. La cámara está fija, enfocando un lugar en el que inicialmente no hay movimiento. Para medir el porcentaje de veracidad de la información que da como respuesta la herramienta, se hicieron 100 movimientos periódicos en el lugar que está enfocando la cámara. Lo ideal es que lleguen 100 mensajes de texto al celular y se carguen 100 videos de movimiento al servidor, sin que se generen falsas alarmas. Los resultados fueron los siguientes.

Ancho de banda (MB)	Tiempo Ejecucion (h)	Cantidad de Movimientos	Periodo del movimiento (seg)	Tiempo de pre-alarma (seg)	Duracion Movimiento (seg)	Tiempo post-alarma (seg)	Alarmas generadas	Videos de movimiento Cargados	Efectividad
2	2	0	NA	NA	NA	NA	0	0	100%
2	2	100	63	2	5	2	85	85	85%

Tabla 11. Resultados de pruebas de calidad de la información

A partir de la tabla anterior, se puede notar que el sistema no generó falsas alarmas. Es de resaltar, que esta herramienta es confiable, es decir, si el sistema generó una alerta, se puede tener la certeza de que el programa detectó un movimiento válido. Ahora bien, la herramienta, desde este punto de vista es confiable, si se refiere a la cantidad de veces que se produjo movimiento y las alarmas generadas, solo se tiene una efectividad del 85%.

Es significativo exponer de manera consecuente con los resultados expuestos en la tabla anterior, que sólo 85 de 100 alertas se notificaron exitosamente. El mensaje de texto llegó al usuario en el momento indicado. Conjuntamente, los videos de movimiento se cargaron al servidor con el nombre adecuado y fue posible su visualización.

Para el caso de las 15 alarmas que no fueron notificadas al usuario, este problema se originó porque cuando se estaba cargando un archivo al servidor, la maquina dejó de capturar video y, en consecuencia, dejó de analizarlo. Esto se evidencia más claramente en la figura 46 (d), en donde se puede advertir que el uso del procesador registra un cambio brusco de un valor muy pequeño a un 75%. Este cambio surgió justo cuando se terminó de cargar un archivo al servidor. Este error se puede disminuir, utilizando una red con mejor ancho de banda que permita a la herramienta cargar los archivos rapidamente.

Para probar el modo de visualización, se dejó la cámara capturando imágenes en un sitio con movimiento constante. Los videos se visualizaron en el orden correcto, pero con un tiempo de latencia que osciló entre 2 a 3 minutos, según el ancho de banda que se manejó para cargar los archivos. Cabe aclarar, que los videos de alerta generados en esta prueba, no fueron cargados de manera exitosa y en algunas ocasiones no fueron grabados en el disco, por el mismo motivo expuesto anteriormente. Esto pone de presente la necesidad de implementar los diferentes modos de funcionamiento de forma independiente, o que el usuario pueda escoger ejecutar uno de los 2 modos únicamente.

## 6. CONCLUSIONES

Durante el desarrollo del Proyecto, se creó una herramienta que permite realizar un monitoreo remoto a un lugar determinado, a partir de la integración de tecnologías existentes y desarrollos de software. Esta herramienta permite al usuario final percibir una mayor sensación de seguridad y, por ende, una mejor calidad de vida. El sistema, que en un principio se implementó por partes y luego se integró, tuvo un funcionamiento adecuado, cumpliendo así con las metas planteadas inicialmente. Sin embargo, este Proyecto es susceptible de mejorar y de integrarle nuevos desarrollos, que hagan de ésta una herramienta cada vez más funcional.

Entre las posibles mejoras que aplican para futuros desarrollos, se encuentran las siguientes opciones: Adquisición de sonido junto con la captura de video; implementación de la aplicación sobre un servidor para muchos usuarios, analizando el video sobre el mismo y evitando que el usuario necesite tener un computador prendido en el sitio que desea monitorear. Esto, además, implica el desarrollo de cuentas de servicio personalizadas para cada usuario; cambiar el formato de compresión de los videos, para optimizar los recursos de máquina. No es necesario cargar videos de alta resolución, si en principio, sólo se necesita detectar movimiento. Esto mejora los tiempos de carga y descarga de los archivos, el rendimiento y uso del procesador. Además, ayuda a mejorar el sistema de visualización, desarrollado en el Proyecto.

Adicionalmente, se puede optimizar el sistema, integrando más cámaras a la herramienta, para que de esta forma, se pueda monitorear varios sitios a la vez. Cabe aclarar, que las cámaras deberían ser inalámbricas, con algún tipo de conexión a la red para la transferencia de imágenes. Por ejemplo, se podría pensar en utilizar cámaras que tengan dirección IP fijas que se conecten directamente al servidor que va a analizar el video.

En cuanto al análisis de video, se pueden hacer varias mejoras, pues este es un tema que está en pleno progreso, por lo que a futuro, con nuevos desarrollos, se podría llegar a tener más aplicaciones, como por ejemplo, detección de movimientos específicos o identificación de personas. En lo que se refiere al código desarrollado para el Proyecto, se puede perfeccionar estableciendo variables con memoria estática, para mejorar la interacción del usuario con la herramienta. Es decir, que el usuario pueda personalizar el sistema, almacenado la configuración establecida, hasta que éste lo desee. Se puede diseñar una interface gráfica que ayude al fácil manejo de la herramienta y la haga más agradable.

Una vez finalizado el Proyecto y luego de haber realizado las pruebas integrales, se detectaron falencias en el sistema. La principal causa se presenta básicamente durante la carga de los archivos al servidor. Vale la pena destacar, que los archivos se cargan con éxito, la falla se produce en razón a las características de funcionamiento de la plataforma utilizada para la implementación del Proyecto y a la velocidad de transferencia de datos. El problema consiste en que el sistema sólo puede realizar una tarea a la vez. Cuando se están cargando archivos de video, se deja de capturar y de analizar, por lo tanto, existe una notable pérdida de información y baja la efectividad de la herramienta. La carga del archivo toma un tiempo considerable, generando inexactitudes en el modo visualización. Con este problema, es difícil lograr completa sincronización entre C++ y HTML para disminuir considerablemente el tiempo de latencia, entre lo que ocurre en tiempo real y lo que se está visualizando.

Como tema para tratar en posibles mejoras, se podría independizar la carga de los archivos del análisis de video, para trabajarlos como 2 procesos paralelos.

Como tema que quedó inconcluso durante la implementación del Proyecto, se tiene el uso de protocolos UDP para la transferencia de datos de manera continua y sin confirmación. Este protocolo no fue utilizado en razón a que los servidores usados para la carga y almacenamiento de los videos, no lo soportan, pues son servidores que están prestando un servicio de bajo costo, que al mismo tiempo, tienen determinadas restricciones por parte del administrador. Este tema puede ser un proyecto para tener en cuenta para futuros desarrollos que se le puedan hacer a la herramienta, pues en caso de implementarse, se podría diseñar, por ejemplo, una aplicación para ver imágenes en tiempo real “streaming”, entre otras.

Para el desarrollo del Proyecto se realizó un análisis e investigación acerca de las redes de comunicación celular existentes, los cuales permitieron escoger la red de comunicación celular 3G de Comcel, como la mejor alternativa para el envío de Short Mails y la red de soporte para el acceso a los videos, a través de los teléfonos celulares adecuados para este fin. Debido a que el Proyecto se desarrolló con ayuda de aplicativos de la citada empresa de telefonía celular, se podría plantear en un futuro su implementación, sobre la plataforma de otros proveedores de telefonía celular, permitiendo así una mayor cobertura para el servicio y posiblemente a menor costo.

Al mismo tiempo, al desarrollar e integrar dentro del Proyecto el envío de mensajes Short Mail, la carga de archivos a través de protocolos FTP y la visualización de video en tiempo real, mediante el uso de la red de comunicación celular 3G, se logró cumplir con el objetivo de realizar transferencia de datos entre PC-Celular-Servidor.

Se tomaron desarrollos, servicios y tecnologías existentes, como la red de telefonía celular 3G (Comcel), OpenCV, servidores (hosting24), teléfonos celulares con transmisión de datos, que integrados permitieron el desarrollo de una herramienta de monitoreo remoto de un hogar.

Para este Trabajo de Grado, se tuvo en cuenta investigaciones sobre detección de movimiento y desarrollos sobre el mismo tema, trabajos que permitieron obtener un mejor soporte y guía para el avance del Proyecto y el resultado final del mismo. Por otro lado, se contribuyó con una aplicación que apenas se está implementando en el tema de tratamiento de imágenes e integración de herramientas desde la programación en C++.

## **7. BIBLIOGRAFÍA**

[1] Calderón Francisco, Urrego Germán. “Conteo Automático de Vehículos”. Trabajo de Grado. Departamento Ing. Electrónica, Pontificia Universidad Javeriana. Bogotá 2008.

[2] Néstor Romero y Carolina García, “Detección y Seguimiento de personas en un cruce peatonal”. Trabajo de Grado. Departamento Ing. Electrónica, Pontificia Universidad Javeriana. Bogotá 2009.

[3] Quiroga Julián, Mora David y Páez Andrés, “Detección de Objetos Móviles en una Escena Utilizando Flujo Óptico”, Trabajo de Grado. Departamento Ing. Electrónica, Pontificia Universidad Javeriana. Bogotá 2010.

[4] Jean-Yves Bouguet “Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm” Intel Corporation Microprocessor Research Labs.

- [5] B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*, vol 17, pp 185-203, 1981.
- [6] Jianbo Shi and Tomasi C. "Good features to track". IEEE Computer Society Conference on 21-23 June 1994 Page(s):593 – 600.
- [7] Gary Bradski E Adrian Kaebler, *Learning OpenCV: Computer Vision whit the OpenCV Library*, O'Reilly, Software That Sees.
- [8] Quiroga Julián, Mora David y Páez Andrés, "Detección de Objetos Móviles en una Escena Utilizando Flujo Óptico", XIV simposio de tratamiento de señales, imágenes y visión artificial – STSIVA 2009.
- [9] Página de XVID. Xvid.org <http://www.xvid.org>, consultada Septiembre de 2010.
- [10] Página de DivX. DivX.com <http://www.divx.com/divx> Consultada Septiembre de 2010.
- [11] Página The OpenCV Library: Computing Optical Flow.pdf [http://robots.stanford.edu/cs223b05/notes/CS %20223-B%20T1% 20stevens\\_opencv\\_optical\\_flow.pdf](http://robots.stanford.edu/cs223b05/notes/CS%20223-B%20T1%20stevens_opencv_optical_flow.pdf) Consultada Julio 2010.
- [12] Pagina Flujo Óptico [http://www.worldlingo.com/ma/enwiki/es/Optical\\_flow](http://www.worldlingo.com/ma/enwiki/es/Optical_flow) Consultada Octubre 2010.
- [13] Página Feature Points Detector en OpenCV <http://www.assembla.com/wiki/show/robotics/FeaturePoints> Consultada Octubre 2010.
- [14] Página Tracking colored objects in OpenCV <http://www.aishack.in/2010/07/tracking-colored-objects-in-opencv/> Consultada en Julio 2010.
- [15] Página Wikipedia.com <http://es.wikipedia.org/wiki/Archivo:Activo.svg> Consultada Septiembre 2010.
- [16] Pagina Wikipedia.com [http://es.wikipedia.org/wiki/File\\_Transfer\\_Protocol](http://es.wikipedia.org/wiki/File_Transfer_Protocol) Consultada Octubre 2010
- [17] Página Wikipedia.com [http://es.wikipedia.org/wiki/Archivo:Modelo\\_ftp.jpg](http://es.wikipedia.org/wiki/Archivo:Modelo_ftp.jpg) Consultada Octubre 2010 diagrama FTP +
- [18] Página Monografías.com <http://www.monografias.com/trabajos15/telefoniacelular/telefoniacelular.Shtml> Consultada Septiembre 2010.
- [19] Pagina Tutorial OpenCV [http://docencia-eupt.unizar.es/ctmedra/tutorial\\_opencv.pdf](http://docencia-eupt.unizar.es/ctmedra/tutorial_opencv.pdf) Consultado Octubre 2010.
- [20] Pagina Trabajo Visión [http://rommel.cuevano.org/trabajo\\_vision/node7.html](http://rommel.cuevano.org/trabajo_vision/node7.html) Consultado Octubre 2010.
- [21] Página de desarrollos web <http://www.desarrolloweb.com/articulos/392.php> Consultado Junio 2010.
- [22] Pagina Ideas Comcel <http://www.ideascomcel.com/index.aspx?banuse=OHw1MHwxMzN8bml2ZWwyLTUwTWVu> Consultado Julio 2010.
- [23] Pagina Wikipedia definición RTSP [http://es.wikipedia.org/wiki/Real\\_Time\\_Streaming\\_Protocol](http://es.wikipedia.org/wiki/Real_Time_Streaming_Protocol) Consultado Junio 2010.



## **8. ANEXOS**

Anexo 1. Captura y análisis de video.

Anexo 2. FTP (Protocolo para carga de archivos)

Anexo 3. PHP (Protocolo para envío de mensajes)

Anexo 4. HTML Modo visualización.