UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Facultad de Informática

INSTITUTO UNIVERSITARIO
SIANI
INGENIERÍA COMPUTACIONAL

Proyecto fin de carrera:

SISTEMA DE VIGILANCIA AUTÓNOMO MULTIAGENTE

CARLOS M. FALCÓN TORRES

Las Palmas de Gran Canaria, Noviembre 2009

Proyecto fin de carrera de la Facultad de Informática de la Universidad de Las Palmas de Gran Canaria presentado por el alumno:

Carlos M. Falcón Torres

Título del Proyecto:

Sistema de vigilancia autónomo multiagente.

Tutor:

Modesto Castrillón Santana

José Daniel Hernández Sosa

Antonio Falcón Martel

DEDICATORIA

Los momentos más oscuros de nuestras vidas no deben ser ni enterrados ni olvidados; más bien son un recuerdo que debe permanecer para servir de inspiración y recordarnos la fortaleza del espíritu humano y nuestra capacidad para soportar lo intolerable"

A mis Padres y Hermano

AGRADECIMIENTOS

La culminación de ese proyecto no habría sido posible sin el apoyo incondicional de mi familia y amigos, a los que estaré eternamente agradecido.

De igual forma tengo que agradecer la ayuda de mis tutores que hasta el final del proyecto se han prestado a guiarme y aguantar mis quebraderos de cabeza.

Por ello quiero agradecer con especial atención a Modesto Castrillón Santana tutor de este proyecto, a José Daniel Hernández Sosa y Antonio Falcón Martel cotutores e impulsores de la idea.

Gracias por ser pacientes conmigo y alumbrarme el camino en todos aquellos días en los que no era capaz de ver la luz al final del túnel.

Contenido

Índice de figuras

Índice de tablas

1- Introducción.

En los últimos años el desarrollo de sistemas de vigilancia ha sido del interés de investigadores e industrias de todo el mundo, sumando esfuerzos para alcanzar el objetivo de incrementar la seguridad en aplicaciones del dominio de la seguridad nacional, seguridad en hogares, zonas comerciales y entidades financieras, monitorización del tráfico, turismo, aplicaciones militares, etc.

A día de hoy realizar un software genérico para la monitorización es un problema complejo por la cantidad de escenarios posibles y los costes que conlleva, por lo que no hay soluciones capaces de satisfacer todas las exigencias que se demandan. Con esta motivación surgió este proyecto con el objetivo de diseñar y crear una herramienta para la ayuda en la monitorización de zonas vídeo vigiladas. Hay que tener en cuenta que los sistemas de vídeo vigilancia autónomos no pretenden suplantar la labor de los vigilantes sino apoyar la labor de éstos y mejorar su rendimiento.

1.1- Sistemas de vigilancia actuales

Los sistemas de vídeo vigilancia actuales están orientados como herramientas de monitorización por parte de operadores humanos, Figura 1.1, en los que recae la eficiencia y la responsabilidad de estos sistemas. No es infrecuente que en este tipo de instalaciones se descuiden aspectos tan importantes como la adecuada selección de la tecnología usada o la correcta ubicación de las cámaras del sistema. De esta forma, tanto el número de cámaras, como el área vigilada están limitadas por el personal disponible, ya que incluso un experto vigilante no puede mantener su atención durante largos periodos de tiempo y contratar personal para monitorizar sistemas de vigilancia resulta caro.

Figura 1. 1 Vigilante monitorizando un sistema de cámaras

Por lo tanto, la práctica común en comercios y bancos es grabar vídeos en cintas que luego se estudiarán con herramientas forenses tras darse una situación anómala. Por ejemplo después de un crimen la grabación se usa para obtener las pruebas incriminatorias.

Sin embargo puede ser más ventajoso si tenemos un sistema que esté activo veinticuatro horas al día y siete días a la semana, con la capacidad de dar avisos en casos de anomalías, permitiendo al oficial de seguridad estar alerta ante una infracción antes de que ésta tenga lugar y poder realizar el protocolo de respuesta.

1.2- Sistemas de vigilancia autónomos

Para cubrir las carencias de los sistemas de vídeo vigilancia aparecieron los sistemas de vigilancia autónoma (S.A.V.), cuya historia comienza con la Defense Advanced Research Projects Agency (DARPA), que empezó a investigar y a desarrollar sistemas automáticos de vigilancia lanzando el programa Visual Surveillances y Monitoring (VSAM) entre 1997-99 y el programa Airbone Vídeo Surveillance (AVS) en 1998-2002. Actualmente el mayor esfuerzo de la comunidad de desarrollo en proyectos de visión es producir un sistema de vigilancia y seguimiento totalmente automático.

Con este propósito DARPA posee dos iniciativas de proyectos de vigilancia, Vídeo Verification of Identity (VIVID) y Combat Zones that See (CTS). Mientras que compañías como IBM, Siemens, Lockheed Martin y Boeing poseen programas de desarrollo de sistemas de vigilancia centrándose en sistemas totalmente automáticos.

Un sistema de vídeo vigilancia autónomo debe tener las siguientes características:

- Poder detectar y seguir objetos en movimiento en el campo de visión.
- Clasificar objetos y detectar actividades.
- Y como complemento también pueden ser capaces de describir eventos que ocurren dentro del espacio vigilado.

Los sistemas de vigilancia automáticos existentes se pueden clasificar atendiendo a los siguientes criterios:

- Según el medio a vigilar para el cual fueron diseñados (*indoor*, *outdoor* o *airborne*).
- El número de sensores que maneja el sistema de vigilancia (Sistema monocámara vs multicámara).
- La movilidad de los sensores (cámaras estacionarias vs cámaras móviles).

Además, hay que tener en cuenta que el número de cámaras requeridas para monitorizar un área se incrementa exponencialmente con la disminución de distancia entre cámaras. Por lo tanto éste es un requisito a tener en cuenta para el manejo del solapamiento de los campos de visión de las cámaras.

1.3- Objetivos del proyecto.

Aparte de los objetivos didácticos y formativos asociados a la elaboración de un proyecto fin de carrera, se desean alcanzar los siguientes ítems más específicos:

a) Estudiar el problema de la planificación sobre la ubicación óptima de dispositivos de seguridad en un recinto.
b) Estudiar los problemas de transformación de coordenadas y calibración en un sistema de adquisición de imágenes motorizado multicámara.
c) Desarrollar e implementar un sistema de adquisición de imágenes coordinado en un sistema multiagente.
d) Desarrollar e implementar técnicas de detección visual y seguimiento coordinado de trayectorias.
e) Implementar procedimientos de control del sistema de vigilancia que incluya monitorización de la actividad.
f) Desarrollar un conjunto de casos de aplicación sobre escenarios prototipo.
g) Introducir al alumno en los problemas de Investigación en Sistemas Inteligentes relacionados con las interfaces perceptuales hombre-máquina.

En este marco se ha diseñado y desarrollado un software multiplataforma para la gestión de sistemas de cámaras en las labores de vídeo vigilancias. Se trata de un software distribuido y centralizado desde una base de datos, como se describe en la Figura 1.2

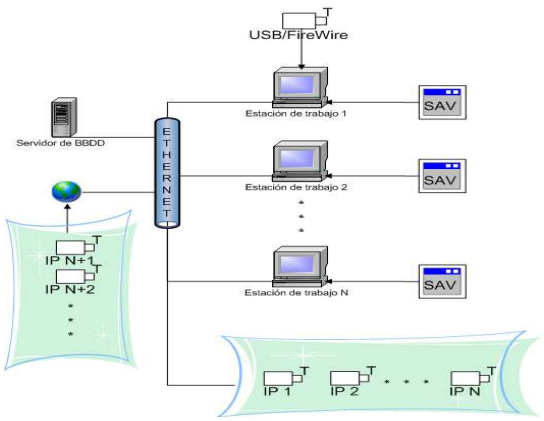

Figura 1. 2 Estructura del software

El software de vídeo vigilancia desarrollado está pensado para funcionar en medios interiores (*indoor*) aunque también puede funcionar en entornos abiertos (*outdoor*) con un conjunto de cámaras estacionarias. Además, el software es capaz de trabajar de manera cooperativa para controlar un mayor número de cámaras y poder abarcar más territorio. Dentro del desarrollo del proyecto se han ponderado los propósitos de los sistemas de vídeo vigilancia según las necesidades obtenidas en el análisis del software, teniéndose en cuenta otros objetivos que se consideraron importantes en éste. El desarrollo del software fue orientado a la explotación de recursos materiales de fácil acceso y bajo coste, así como a posibilitar la adquisición de imágenes desde cualquier tipo de cámara independientemente del modelo y marca de ésta.

1.4- Planificación temporal.

Para alcanzar los objetivos anteriores se ha seguido un plan de trabajo que se desarrolla según las siguientes etapas:

- **Etapa 1:** **120 Horas**

Recopilación de la documentación y estudio de los entornos en los que se trabajará:

- Sistemas multiagentes.
- Seguimiento de trayectorias.
- Control Visual.
- Realización de pequeñas tareas para familiarizarse con el entorno de trabajo.

- **Etapa 2:** **230 Horas**

Fase de ingeniería del software:

- Fase de metodología de desarrollo.
- Fase de análisis.
- Fase de diseño.

- **Etapa 3:** **30 Horas**

Proceso de montaje y calibración del sistema de monitorización.

- **Etapa 4:** **60 Horas**

Creación de la base de datos (BBDD) así como software para el manejo de las mismas:

- Diseño de la BBDD.
- Implementación de la BBDD.
- Implementación del software de gestión.
- Fase de test sobre la BBDD.

- **Etapa 5:** **90 Horas**

Desarrollo e implementación del sistema de adquisición de imagen remota y monitorización:

- Realizar la red de comunicación.
- Realizar software de adquisición remota.
- Realizar software de monitorización.
- Fase de pruebas de los módulos descritos.

- **Etapa 6:** **140 Horas**

Creación de un monitor de adquisición.

- **Etapa 7:** **140 Horas**

Realizar el proceso de implantación e integración de las técnicas de detección visual y seguimiento de trayectorias:

- Realización de técnicas de detección visual.
- Implementación de técnicas de seguimiento.
- Evaluación de las diferentes técnicas.
- Integración y test de los modelos escogidos.

- **Etapa 8:** **80 Horas**

Ejecutar las fases de pruebas de los distintos programas.

- **Etapa 9:** **90 Horas**

Desarrollo de manuales y documentos del software realizado.

1.5- Recursos necesarios

En este apartado se detallan los recursos tanto software como hardware que han sido necesarios para su realización y puesta en marcha. El desarrollo software del proyecto se ha enfocado bajo el uso de herramientas software libre y multiplataforma en la mayor medida posible, pretendiéndose conseguir la máxima portabilidad del software. Cuando esto no ha sido posible se ha recurrido a las herramientas ofertadas por la arquitectura de desarrollo siguiendo la premisa de que las herramientas usadas sean por lo menos gratuitas, no obstante el software se ha modularizado de tal manera que si migramos a otra plataforma la parte afectada pueda ser sustituida sin ocasionar grandes trastornos. Con estas premisas se ha realizado el proyecto bajo el SO Windows XP, utilizándose las herramientas software de esta plataforma que se citan a continuación.

Recursos software (Para la plataforma Windows):

- ***OpenCV*** es una librería libre de visión artificial originalmente desarrollada por Intel y publicada bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas, en el 2002. La librería es multiplataforma, y puede ser usada en Mac OS X, Windows y Linux. Está orientada al tratamiento de imágenes en tiempo real, lo que es posible si encuentra en el sistema las *Integrated Performance Primitives* de Intel (IPP).
- ***wxWidgets*** son unas bibliotecas multiplataforma y libres, para el desarrollo de interfaces gráficas programadas en lenguaje C++. Están publicadas bajo una licencia LGPL, similar a la GPL con la excepción de que el código binario producido por el usuario a partir de ellas, puede ser propietario, permitiendo desarrollar aplicaciones empresariales sin coste. Las wxWidgets proporcionan una interfaz gráfica basada en las bibliotecas ya existentes en el sistema (nativas), con lo que se integran de forma óptima y resultan muy portables entre distintos sistemas operativos. Están disponibles para Windows, MacOS, GTK+, Motif, Open VMS, OS/2.
- ***PostgreSQL*** es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (***PostgreSQL Global Development Group***).
- ***Visual Studio 2005 C++*** es la herramienta que se ha utilizado para realizar el desarrollo software del proyecto utilizándose parte de las bibliotecas que incorpora el Framework 2.0 del propio Visual Studio.

Recursos Hardwares y tangibles:

Como materiales tangibles para la realización del proyecto se ha utilizado:

- Un PC Intel(R) Core(TM)2 CPU T7200 @ 2.00GHz con 2.00 GB RAM.
- Varios tipos de cámaras USB e IP.
- Red LAN/WIFI.
- Cámaras fotográficas para la realización de los vídeos.

El entorno de desarrollo ha ido variado según las necesidades lo requerían empezando por el *aula de Proyecto* y una vez transcurrida la primera fase el desarrollo se traslado al laboratorio de investigación del grupo dentro de la facultad de Informática. Mientras que los escenarios mostrados en los vídeos corresponden a la tercera planta de la propia facultad de Informática.

1.6- Estructura del documento

El contenido del documento se divide en varios capítulos, empezando por un estudio de la viabilidad de los sistemas de vigilancia, capítulo donde se comentarán los aspectos más relevantes a la hora de acometer la implantación de un sistema de vigilancia. Seguido por el capítulo correspondiente a la fase de análisis de requisitos de usuario, viéndose en él las metodologías seguidas para el diseño y desarrollo de la aplicación. Éste viene a su vez precedido por el capítulo destinado a la fase de diseño arquitectónico construido a partir de los documentos obtenidos en el análisis. Tras concluir la etapa de ingeniería del software, se seguirá la memoria con las soluciones adoptadas para solventar las problemáticas del proyecto y una vez presentadas las soluciones, el siguiente capítulo está destinado al análisis de los resultados.

Terminando el grueso de la memoria con el capítulo de las conclusiones y trabajo futuro, donde se comentan los resultados obtenidos y algunas vías para continuar el proyecto. Para terminar se incluye un glosario con definiciones en más profundidad de algunos conceptos citados a lo largo de la memoria y la bibliografía utilizada en el documento.

2- Estudio y viabilidad de un sistema de vigilancia

En este capítulo se realiza un estudio sobre los aspectos a tener en cuenta a la hora de diseñar un sistema de vídeo vigilancia, centrándose en el análisis de los sistemas de vídeo vigilancia IP. El primer aspecto que se tratará en este capítulo será los puntos legales a tener en cuenta a la hora de diseñar un sistema de vídeo vigilancia con la normativa vigente. Concluida esta sección, se estudiarán distintos aspectos a tener en cuenta a la hora de montar un sistema de vídeo vigilancia a la vez que se introducen algunos conceptos usados en el desarrollo del proyecto.

2.1- Aspectos legales en vídeo vigilancia

A la hora de montar un sistema de vídeo vigilancia hay que tener en cuenta y respetar una serie de leyes vigentes, principalmente en el campo de la protección de datos, tanto para que las pruebas tengan validez jurídica, como para evitar posibles denuncias de los usuarios, ya que los usuarios de las zonas vigiladas deben conocer que están siendo monitorizados bajo su consentimiento. A continuación se citan los artículos y decretos más relevantes de la legislación vigente en materia de protección de datos de carácter personal relativa a los sistemas de vídeo vigilancia.

- Directiva 2002/58/CE del Parlamento Europeo y del Consejo, de 12 de julio de 2002, relativa al tratamiento de datos personales y a la protección de la intimidad en el sector de las comunicaciones electrónicas.
- Directiva 95/46/CE del Parlamento y del Consejo, de 24 de octubre de 1995, relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.
- Directiva 97/66/CE del Parlamento y del Consejo, de 15 de diciembre de 1997, relativa al tratamiento de los datos personales y a la protección de la intimidad en el sector de las telecomunicaciones.
- Convenio 108/81/CE del Consejo, de 28 de enero, para la protección de las personas con respecto al tratamiento automatizado de datos de carácter personal.
- Ley Orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal.

- Real Decreto 994/1999 de 11 de junio, por el que se aprueba el Reglamento de Medidas de seguridad de ficheros automatizados que contengan datos de carácter personal.
- Instrucciones dictadas por la Agencia Española de Protección de Datos desde el año 1995 y los diferentes Documentos del Grupo del Artículo 25 de la Directiva 95/46/CE.

Se puede encontrar más información en uno de los apéndices donde se adjunta la instrucción "21648 1/2006, de 8 de noviembre" referencia en el ámbito del cumplimiento de las leyes de protección de datos en vídeo vigilancia **[PW, Agpd]**. Mientras que los documentos expuestos en las zonas vídeo vigiladas para que los datos tengan valides judicial y cumplan con los decretos de la protección de datos son los siguientes:

- Modelo cláusula informativa.

  Formado por los siguientes ficheros según el art. 3, apartado B. Instrucción 1/2006, de 8 de noviembre, de la Agencia Española de Protección de Datos, sobre el tratamiento de datos personales con fines de vigilancia a través de sistemas de cámaras o vídeo cámaras.

  FICHERO PRIVADO

  De conformidad con lo dispuesto en el art. 5.1 LO 15/1999, de 13 de diciembre, de Protección de Datos, se informa:

  1. *Que sus datos personales se incorporarán al fichero denominado "....."y/o serán tratados con la finalidad de seguridad a través de un sistema de vídeo vigilancia.*
  2. *Que el destinatario de sus datos personales es:*
     (a) *La empresa de seguridad.....*
     (b) *El dueño del establecimiento.....*
  3. *Que puede ejercitar sus derechos de acceso, rectificación, cancelación y oposición ante el responsable del fichero.*
  4. *Que el responsable del fichero tratamiento es " (.....nombre o razón social.....)" o su representante D./Dª."....." ubicado en C/.....*

FICHERO PÚBLICO

De conformidad con lo dispuesto en el art. 5.1 LO 15/1999, de 13 de diciembre, de Protección de Datos, se informa:

1. *Que sus datos personales se incorporarán al fichero denominado "..... " del que es responsable ese organismo, creado por Resolución..... (BOE.....) y/o serán tratados con la finalidad de seguridad a través de un sistema de vídeo vigilancia.*
2. *Que el destinatario de sus datos personales es la empresa de seguridad.....*
3. *Que puede ejercitar sus derechos de acceso, cancelación y oposición ante el responsable del fichero.*
4. *Que el responsable del fichero tratamiento es ".....(nombre o razón social)....." ubicado en C/ .....*

- Rótulo de zona vídeo vigilada. Figura 2.1.

Figura 2. 1 Rótulo de notificación de zona vídeo vigilada

2.2- Las cámaras IP en la vídeo vigilancia

Antes de empezar, definimos una cámara IP como combinación de una cámara y un pequeño ordenador. Por lo tanto, cada cámara IP posee su propia dirección IP, un web server para gestionar la comunicación en la red e incluso algunos modelos incluyen entradas para alarmas y salidas de relé. Actualmente los modelos de gama alta proveen de funcionalidades para la detección de movimiento y la entrada/salida de vídeo analógico.

Debido a los avances en la materia, un usuario es capaz de realizar una monitorización remota de un número de cámaras, de manera segura y restringida y en tiempo real a través de una red interna LAN, WAN o Internet. Además, la mayoría de edificios cuenta con infraestructura de cableado lo que permite configurar una red IP, sin demasiado coste, facilitando la inserción de estos sistemas en detrimento de los modelos basados en tecnología analógica.

Si además de un conjunto de cámaras IP contamos con cámaras analógicas es posible aprovechar estas cámaras a través del uso de un servidor de vídeo que nos haga de pasarela para el acceso. Medida que reafirma la implantación de sistemas de vídeo vigilancia IP.

A la hora de adquirir imágenes de las cámaras existen dos vertientes, el uso de tecnología web y monitorizar las cámaras desde un navegador o el uso de software dedicado, camino seguido en el desarrollo de este proyecto.

2.3- Análisis del mercado actual

En el mercado actual existen dos clases de cámaras IP las creadas por fábricas procedentes del mundo de la Tecnología de Información y las creadas por fábricas procedentes del mundo de la seguridad electrónica. El primer grupo de cámaras está compuesto por elementos que buscan llevar vídeo a sitio remotos, casi siempre bajo condiciones favorables y a baja velocidad (~10-15 cps) y sin ninguna misión especial. Este grupo surgió a partir del boom de las tecnologías de videoconferencia y hoy en día, gracias al avance de la capacidad de proceso y almacenamiento, sumado al avance en técnicas de procesamiento digital y por supuesto a los bajos precios, en este grupo existen una gran variedad de marcas y modelos para satisfacer a todos los usuarios.

La segunda clase de cámaras IP, son un poco más recientes y fueron creadas por fábricas del gremio de seguridad electrónica. Son marcas reconocidas en la materia y que saben qué especificaciones deben ser exigentes para una misión tan crítica como es

la seguridad. Para ellos una cámara de vídeo, no es un accesorio más de un PC, es una herramienta vital para cuidar y velar activos, patrimonio y vidas humanas.
Pese a que el segundo grupo de cámaras es más conveniente por razones obvias para los sistemas de seguridad, en este proyecto nos centraremos en el uso de cámaras del primer grupo puesto que son más accesibles y están más extendidas.

2.4- Análisis del uso de tecnología IP

Las cámaras IP envían los datos digitales comprimidos a través de una tarjeta de red que dependiendo del modelo harán uso de alguno de los protocolos de transmisión de red, que en el caso del proyecto se optó por usar HTTP al tratarse de un protocolo extendido y con códigos de estado. Otro aspecto importante de las cámaras es el manejo de streams y el número de usuarios que es capaz de atender simultáneamente por *streams* o en varios *streams*, esto es conocido como *unicast* y *multicast* respectivamente.
Dado que las cámaras IP son un nodo más de la red, éstas son accesibles desde cualquier componente de la red. No obstante el acceso a la mayoría de las cámaras se puede configurar de dos modos, mediante el uso de un servidor web de vídeo o desde un software propio. Si se opta por el uso de un servidor web de vídeo al que tras acceder se puede monitorizar la cámara, además de poder configurar y ajustar algunos parámetros desde una web residente. Esta solución facilita la monitorización por parte de varios usuarios de la red a costa de comprometer la seguridad. La segunda alternativa es que la cámara no tenga servidor web por lo que para poder monitorizar y gestionar la cámara es necesario el uso de un software privativo dado por el fabricante. Esta opción añade mayor seguridad pero genera dependencia del fabricante a la hora de gestionar los accesos por varios usuarios.

2.4.1- Aspectos software

Con el uso de sistemas IP, la calidad del vídeo digital que se muestra depende de tres factores: sensor de la cámara, la tecnología de compresión y el software que gestiona el sistema de vídeo IP. Los dos últimos factores son aspectos software de gran importancia en los sistemas de vídeo vigilancia. Por un lado, la compresión de la información tiene como objetivo que las imágenes digitalizadas lleguen a su destino con buena resolución y velocidad, usando la mínima cantidad de recursos de la red de datos, es decir usando el menor ancho de banda (BW) posible.

El segundo aspecto a tratar es la importancia del software de gestión en los sistemas de vigilancia. Sin un buen software para la gestión de vídeo no se potencian ni las cualidades de las cámaras y por lo tanto no se ofrecen soluciones al usuario. Por lo que los software de gestión deben de posibilitar el control, administración y monitorizado de sistemas de cámaras en vivo o de vídeos. Premisas bajo las que se ha diseñado e implementado el software de este proyecto. Por otra parte el rendimiento general del software depende de la interacción con las cámaras IP, el hardware sobre el que corre y la red.

2.4.2- Análisis del consumo

Como se ha detallado con anterioridad uno de los factores más influyentes en el rendimiento de un sistema IP es el ancho de banda, que responde a la siguiente fórmula:

$$AB = Tamaño\ de\ la\ Imagen\ \times\ Cuadros\ por\ Segundos\ (cps) \times\ Canales \quad (2.1)$$

Así una cámara PAL en tiempo real (25cps) a compresión normal y tamaño normal consume tanto como: 8Kb × 25cps × 1 = 160Kbps. Mientras que una cámara IP en tiempo real suele trabajar en torno a 7cps suponiendo el mismo tipo de compresión y tamaño que en el caso anterior: 8Kb × 7cps × 1 = 45Kbps.

Otro de los aspectos críticos a considerar es el consumo de disco en las grabaciones de vídeo. Para poder calcularlo hay que hacer uso de la fórmula *(2.1)* para obtener el ancho de banda de cada cámara y una vez conocido éste, el consumo de disco se obtiene multiplicando el ancho de banda por la cantidad de segundos de las grabaciones a realizar más un pequeño margen del diez por ciento debido a la gestión de ficheros del S.O.. Por ejemplo una jornada de 8 horas de grabación continua necesitaría:

- Usando el cálculo de ancho de banda de una cámara IP:

  45Kps × 60seg × 60Min × 8Hs = 2181600 Kb = 2181,6Mb

  2,18 Gb × 1,10= 2,4Gb

- Usando el cálculo de ancho de banda de una cámara PAL:

  160Kps × 60seg × 60Min × 8Hs = 7756800 Kb = 7756,8Mb

  7,76 Gb × 1,10= 8,5Gb

Donde se constata la importancia de determinar cuál será el número de cuadros por segundo a transmitir y el tamaño y nivel de compresión de la imagen.

2.5- Conclusiones

Diseñar un sistema de vídeo vigilancia IP resulta ventajoso en muchos aspectos que se han comentado a lo largo de este estudio pero no obstante su implantación conlleva tener en cuenta los siguientes aspectos:

- La distribución de las cámaras a monitorizar.

  Cuando la distribución de las cámaras es mínima no se recomienda la implantación de un sistema de vídeo vigilancia IP que cargue la red existente. Únicamente es plausible su implantación si se dispone de una red de alta velocidad y se desea realizar la inversión en cámaras IP, aunque siempre es más recomendable el uso de una red dedicada exclusivamente a la vigilancia como ocurre en el caso de los sistemas analógicos.

- La disposición de una red de datos.

  Si no se dispone es de una red de datos lo suficientemente potente para aguantar la demanda de las cámaras, se requiere de un estudio de coste y capacidades de recepción de la red a montar. En muchas ocasiones el gasto en equipos IP es desproporcionado en comparación a la red que se dispone, teniéndose que sacrificar calidad y desempeño de las imágenes proporcionadas por las cámaras. Debido a todo ello, hace falta llegar a un consenso con el número de cámaras a utilizar atendiendo al ancho de banda disponible y a la configuración de las cámaras (cps y resolución de las imágenes).

- El ancho de banda disponible.

  Otro aspecto importante a considerar en las redes de datos que no son de uso exclusivo para la vigilancia son las condiciones externas que en ocasiones son desfavorables para el uso de sistemas IP, obligando a priorizar los eventos y ceder todo el ancho de banda al más importante.

- Existencia de material de vídeo vigilancia.

  Cuando ya existen equipos de vídeo analógicos la recomendación al implantar un sistema IP es integrar dichos dispositivos al sistema mediante el uso de codificadores y decodificadores en caso de que se desee realizar la operación contraria.

Por todo ello las recomendaciones a la hora de implantar una solución de vídeo vigilancia IP es ir haciéndolo progresivamente a la par de la mejora o creación de redes de comunicación de alta velocidad que garanticen la rentabilidad y aprovechamiento de la inversión y tener en cuenta todos los factores que se han mencionado en este capítulo.

3- Fase de análisis

En este capítulo se tratarán todos los aspectos relativos al análisis del proyecto. Se empezará definiendo la metodología donde se describirá la técnica de prototipado, para luego continuar con la descripción y motivación de los patrones de diseño elegidos. Posteriormente se pasará al análisis y especificaciones de los casos de uso y se concluirá el capítulo con el análisis para la construcción del prototipo.

3.1- Metodología de desarrollo.

El paradigma de desarrollo software elegido para la realización del proyecto ha sido el orientado a prototipos. Se ha elegido este paradigma debido a su versatilidad a la hora de diseñar la interfaz, ya que este paradigma implica un contacto continuo con el usuario final, al que se le proporciona un prototipo básico con en el flujo de la información, que irá evaluando y redefiniendo hasta que se cree un prototipo final que se ajuste a sus necesidades. Como única contramedida, este paradigma presenta una etapa de análisis algo extensa, que depende de las veces que se deban modificar los prototipos, por lo que es conveniente tener contacto directo con el usuario final y conocer sus necesidades desde un inicio.

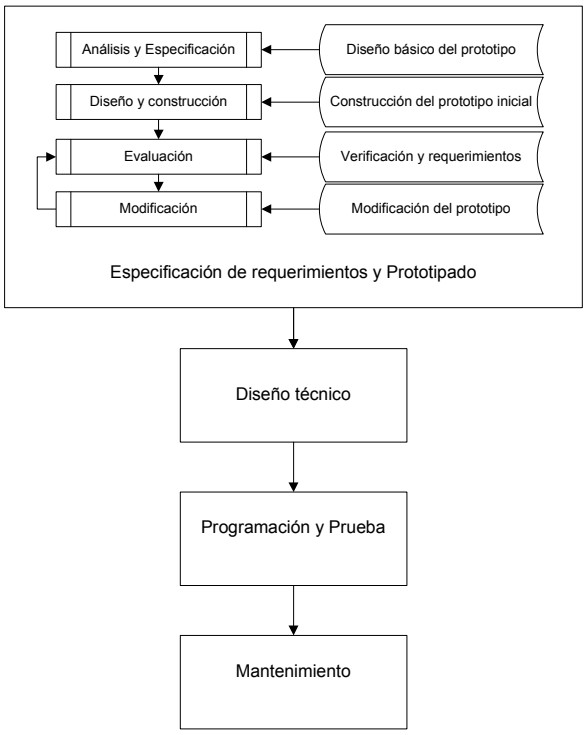

Figura 3. 1 Diagrama de desarrollo de prototipados **[Pressman, 1997]**

En la Figura 4.1 se detalla el diagrama de desarrollo de prototipos en el podemos observar las distintas fases de las que consta este tipo de desarrollo software. A lo largo del desarrollo del proyecto se han centrado los esfuerzos en los tres primeros puntos, dejándose la fase de mantenimiento como un refinamiento en la última etapa del desarrollo del Proyecto Fin de Carrera (PFC).

3.2- Patrones de diseño.

Para el diseño de aplicaciones se ha utilizado el patrón de diseño Modelo-Vista-Controlador (MVC), Figura 3.2, en combinación con el patrón Comando, Figura 3.3.
Esta unión de patrones nos permite por un lado separar los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes independientes evitando que las modificaciones de un grupo afecten a los otros y se produzcan menos errores al hacer uso del patrón MVC. Mientras que el patrón Comando nos proporciona un método para implementar una interfaz que permite invocar las acciones de forma uniforme y ampliar el repertorio de acciones de forma sencilla y facilitando su uso. Con la unión de estos dos patrones se logra por tanto un mayor desacople de los componentes, se aumenta la reusabilidad de los mismos y un menor impacto en la lógica de datos, siguiéndose las directrices del paradigma de desarrollo elegido.

Componentes del patrón MVC:

- Modelo: datos y reglas de negocio.
- Vista: muestra la información del modelo al usuario.
- Controlador: gestiona las entradas del usuario.

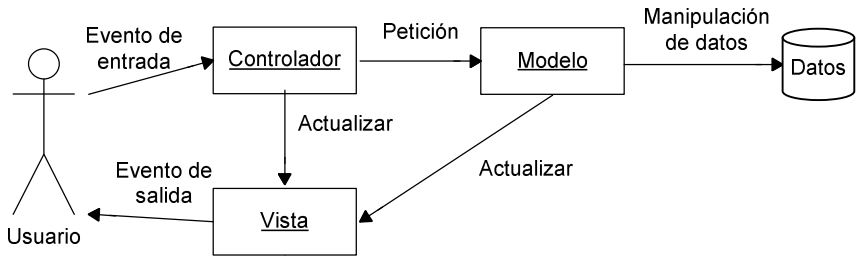

Figura 3. 2Patrón Modelo-Vista-Controlador **[PW, MVC]**

Dentro del MVC en la elección del diseño del software se optó por el uso de un MVC con un modelo pasivo (que es aquel que no notifica cambios en los datos) que responde a las entradas del usuario, pero no detecta los cambios en datos del servidor, ya que el software no realiza chequeo del sistema de datos una vez iniciada la aplicación.

Componentes del patrón Comando:

- *AbstractCommand*: Clase que ofrece un interfaz para la ejecución de órdenes.
- *ConcreteCommand*: Clase que implementa una orden concreta.
- *Invoker*: Clase que instancia las órdenes.
- *CommandManager*: Responsable de gestionar una colección de objetos orden creadas por el *Invoker*.

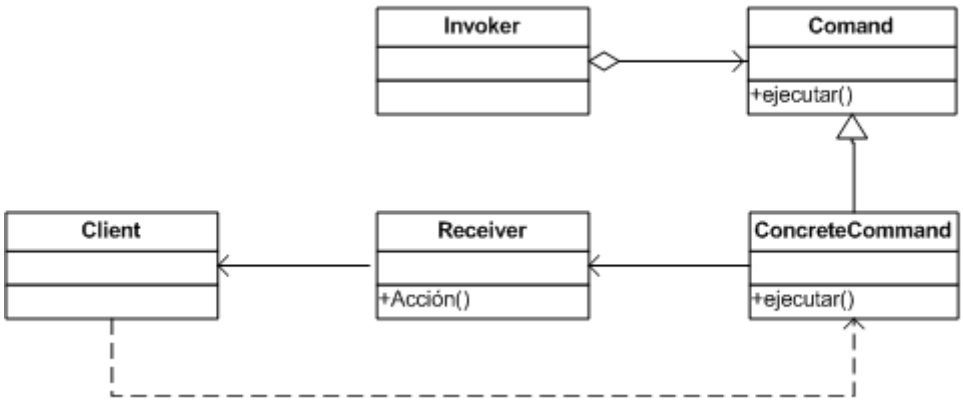

Figura 3. 3 Patrón Comando **[PW, Com]**

3.3- Análisis y especificación de requisitos.

En el análisis se ha realizado la especificación de los requisitos del software, es decir, que se ha documentado lo que tiene que hacer. Además, se ha llevado a cabo un enfoque alternativo del análisis de requisitos, consistente en el desarrollo de prototipos de interfaz. El objetivo de esta fase es la identificación y especificación de las operaciones que van a estar disponibles en la aplicación a desarrollar.

3.3.1 - Identificación y definición de los casos de uso.

La definición de casos de uso permite describir la manera en la que los diferentes tipos de personas, actores, usarán el sistema. Los actores identificados interactúan directa y frecuentemente con el sistema para lograr el funcionamiento requerido y obtener el beneficio propuesto.

Antes de detallar los casos de uso hay que recordar algunas de las características del Sistema Autónomo de Vigilancia (S.A.V.):

- S.A.V. es un sistema distribuido compuesto por varias cámaras y ordenadores.
- Es un software de escritorio configurable desde archivo.
- Posee un plano de la zona a monitorizar.
- Con acceso a un servidor de BBDD (PostgreSQL):
    - Base de datos de Usuarios.
    - Base de datos de Cámaras.
    - Base de datos de Log.

A continuación se definirá el diagrama de caso de uso para el software (Figura 3.4) en el que podemos encontrar dos tipos de actores que interactuarán con él: el administrador y el vigilante. El administrador es el encargado de gestionar y administrar el software, entre sus funciones destacan la configuración del sistema de vigilancia (añadir las cámaras y gestionarlas), gestionar las bases de datos y el plano de la instalación y configurar el software. Para ello el usuario se tendrá que registrar para poder realizar dichas operaciones. Dentro de este rol existen varios tipos de administradores, Figura 3.5, todos ellos con diferentes privilegios para operar con el software:

- Vigilante: Es el encargado de gestionar y controlar el funcionamiento del sistema. Con potestad para la gestión de las cámaras y el plano.
- Administrador del Sistema: Además de contener los privilegios del usuario *Vigilante* este tipo de usuario tiene acceso limitado para la gestión de la base de datos de las cámaras.
- Administrador de S.A.V.: Es el súper usuario del software con total acceso a las funcionalidades del software.

El vigilante tiene la potestad de definir los parámetros de configuración de la monitorización (Seleccionar el algoritmo, gestionar las salidas, etc.) además de responder a los mensajes devueltos en el log del sistema. Hay que tener en cuenta que ambos actores pueden ser representados por una misma persona, eso sí, ejerciendo los dos roles independientes, de manera no concurrente.

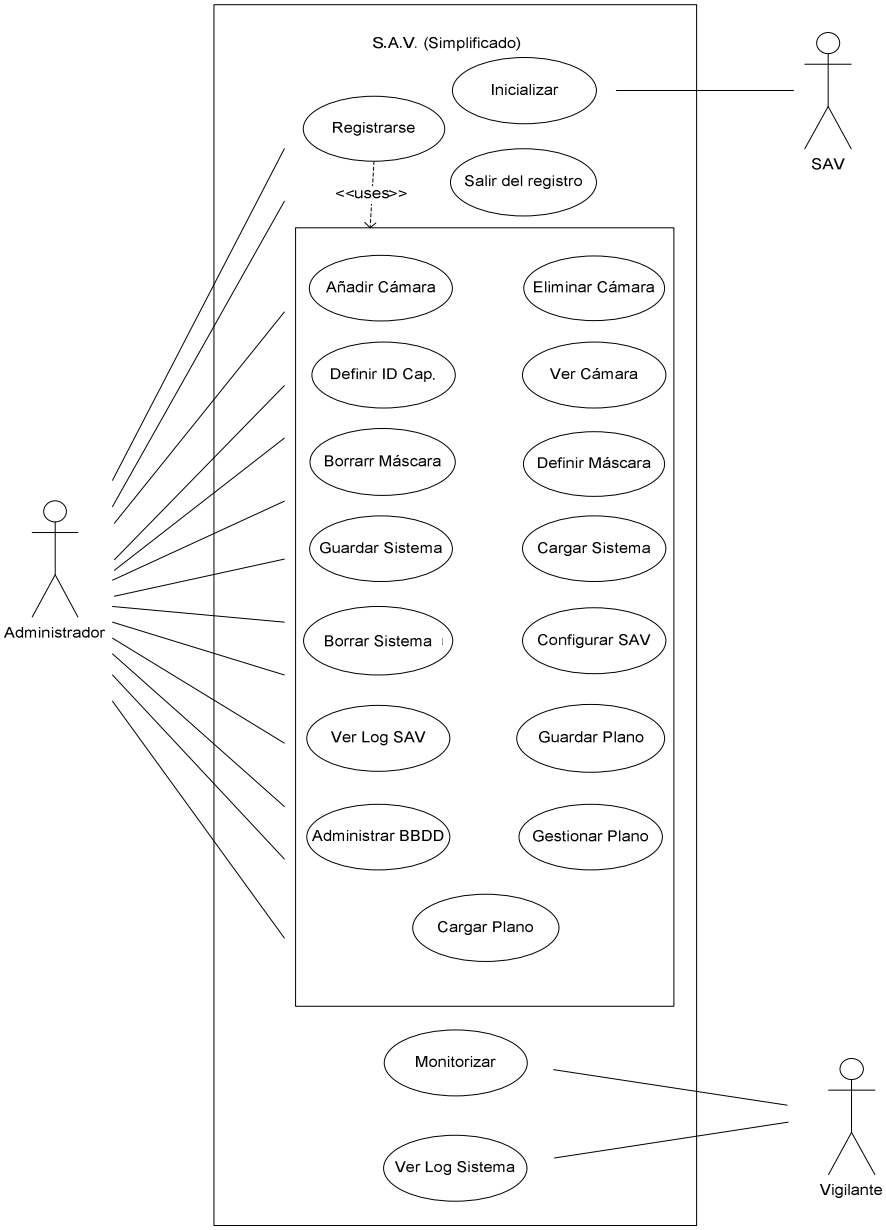

Figura 3. 4 Casos de uso S.A.V.

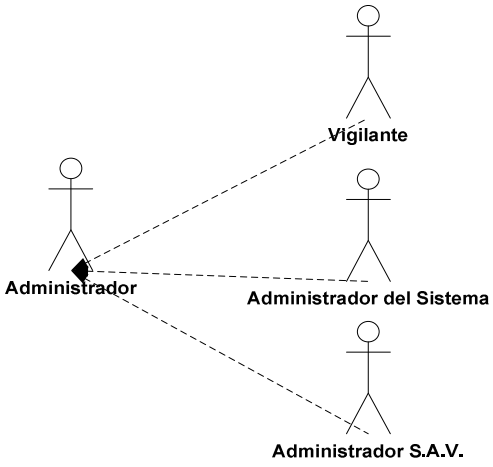

Figura 3. 5 Roles del Usuario Administrador

Registrarse

Esta actividad permite a un usuario identificarse en el software dándole acceso a las demás actividades. Para poder registrase un usuario deberá ir a la zona de administración e introducir su usuario y clave que se contrastará con la base de datos. En caso de transcurrir correctamente la operación se le dará la potestad para acceder a aquellas operaciones del software en las que tenga privilegios. La única excepción que puede producirse es al introducir un usuario o clave erróneo, originando que el usuario no pueda registrarse y por lo tanto no acceder a las operaciones de administración.

Nombre	Registrarse	Id	T001
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	El usuario activa su cuenta como administrador.
Condiciones previas	Existir el usuario en la base de datos.
Disparador	Realizar tareas de administrador.
Flujo Normal	1- Ir a la pestaña Administrar 2- Introducir usuario y clave 3- Registrarse
Excepción	1- Usuario no registrado
Actor Secundario	Gestor de BBDD.
Canal hacia el actor secundario	Gestor de BBDD: Comprobar usuario.
Notas	No hay

Tabla 3. 1 Caso de uso Registrarse

Salir del registro

Esta actividad es la contraria a la anterior operación, con ella el usuario dejará de estar registrado en el sistema y no podrá realizar ninguna actividad de administración que requiera la identificación de un usuario.

Nombre	Salir del registro	Id	T002
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	El usuario deja de estar registrado en el sistema.
Condiciones previas	Haber un usuario registrado.
Disparador	Dejar de estar registrado.
Flujo Normal	1- Ir a la pestaña Administrar 2- Salir del registro
Excepción	No hay.
Actor Secundario	No hay.
Canal hacia el actor secundario	No hay.
Notas	No hay

Tabla 3. 2 Caso de uso Salir del registro

Añadir Cámara

Esta actividad permite al administrador añadir una cámara al sistema actual. Para ello deberá de ejecutar la opción *Añadir* de la sección *Op. de cámaras* mostrándose un diálogo en el que se podrá añadir una o varias cámaras al sistema actual. Existen dos opciones con las que añadir cámaras al sistema, en la primera de ellas el usuario seleccionará la cámara desde una lista en la que estarán las cámaras existentes en la base de datos que no formen parte del sistema actual. Mientras que en la segunda opción el usuario tiene la posibilidad de completar una sentencia SQL.

Una vez elegida una de las opciones pueden ocurrir las siguientes excepciones:

- No existen cámaras en la BBDD.
- Fallo en la conexión con la BBDD.
- Sentencia SQL incorrecta.

Una vez añadida la cámara ya formará parte del sistema y se podrá operar sobre ésta. Esta operación se guarda en el registro de *Log del sistema*, y detiene el proceso de monitorización si estuviera activo.

Nombre	Añadir Cámara	Id	T003
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Añade una nueva cámara al sistema actual.
Condiciones previas	- El usuario debe de estar registrado. - Debe existir una cámara física.
Disparador	Se añade una nueva cámara al sistema.
Flujo Normal	1- Ir a la pestaña Administrar 2- Añadir Cámara 3- Seleccionar cámara 4- Confirmar operación 5- Respuesta del sistema a la acción 6- Registrar Log
Excepción	1- No existen más cámaras en el sistema 2- Fallo en la conexión con la BBDD 3- Sentencia SQL incorrecta
Actor Secundario	Gestor del sistema. Gestor BBDD.
Canal hacia el actor secundario	Gestor BBDD: Obtener Cámara. Gestor BBDD: Registrar Log. Gestor : Añadir Cámara
Notas	No se puede añadir una cámara con el modo de vigilancia activo.

Tabla 3. 3 Caso de uso Añadir Cámara

Eliminar Cámara

La operación eliminar cámara como su nombre indica elimina la cámara correspondiente al identificador seleccionado desde la lista de cámaras. Esta operación no cabe a error alguno ya que los identificadores mostrados son los de las cámaras que actualmente existen en el sistema. Si la cámara se está visualizando esta operación no estará habilitada.

Los pasos a seguir son ir a la zona de administración, seleccionar una cámara y eliminar dicha cámara mostrándose el resultado de la operación en la ventana de registro. Esta operación queda registrada en el *Log del sistema*. Si el proceso de monitorización está activo ocasionará que éste se pare.

Nombre	Eliminar Cámara	Id	T004
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Elimina una cámara del sistema.
Condiciones previas	- El usuario debe de estar registrado como administrador. - Seleccionar una cámara. - La cámara no debe de estar visualizándose.
Disparador	Deseo de eliminar una cámara en el sistema.
Flujo Normal	1- Ir a la pestaña Administrar 2- Selección de una cámara 3- Presionar el botón eliminar 4- Respuesta del sistema a la acción 5- Registrar Log
Excepción	No hay.
Actor Secundario	Gestor de Sistema. Gestor BBDD.
Canal hacia el actor secundario	Gestor: Eliminar Cámara. Gestor BBDD: Registrar Log.
Notas	No se puede eliminar una cámara que se está visualizando, para poder realizar esta operación debe cerrarse la ventana correspondiente al ID de la cámara a eliminar.

Tabla 3. 4 Caso de uso Eliminar Cámara

Ver Cámara

Función para verificar la recepción de la cámara seleccionada, su adquisición se mostrará desde una ventana creada con la OpenCV. Esta operación nos permite emular un sistema de vigilancia sin automatización.

Para ejecutar esta funcionalidad se debe de ir a la zona de administración y seleccionar la cámara a visualizar. Esta operación no requiere de registro alguno, basta con que existan cámaras en el sistema actual.

Existe la posibilidad de no poder monitorizar la cámara por problemas de la conexión o mal funcionamiento de ésta. En este caso se deberá comprobar el estado de la cámara y/o su definición en la base de datos ya que el sistema no ocasionará ninguna excepción y simplemente nos dará un mensaje de error de conexión.

Nombre	Ver Cámara	Id	T005
Creado	Carlos Falcón	Modificado	-

Actor Principal	Cualquiera.
Descripción	Muestra la cámara seleccionada desde una ventana de la OpenCV.
Condiciones previas	- Seleccionar una cámara.
Disparador	Verificar el funcionamiento de una cámara.
Flujo Normal	1- Ir a la pestaña Administrar 2- Selección de una cámara 3- Presionar el botón ver cámara 4- Se lanza una ventana de monitorización 5- Cuando se desee se cierra dicha ventana.
Excepción	No hay.
Actor Secundario	Gestor del sistema.
Canal hacia el actor secundario	Gestor: Ver Cámara.
Notas	El usuario es el responsable de cerrar la ventana de monitorización de la cámara para evitar el bloqueo de otras operaciones de la pestaña Administración.

Tabla 3. 5 Caso de uso Ver Cámara

Definir Máscara

Con esta operación el usuario puede definir una zona de interés dentro del campo de visualización de la cámara seleccionada. Para ello deberá ir a la zona de administración y seleccionar la cámara en la que se desea definir la zona de interés. Como consecuencia de esta acción se le mostrará una ventana emergente al usuario con el campo de visión de la cámara seleccionada, donde podrá definir una zona de interés rectangular. Tras seleccionar una zona de interés, ésta se le presentará al usuario y si está satisfecho con el resultado obtenido deberá cerrar la ventana pulsando la techa *Esc* sobre ésta.

Existe la posibilidad de no poder obtener el marco de visualización de una cámara si ésta no es accesible. Hay que recordar que si existiera una máscara ya definida en el sistema ésta será reemplazada por la nueva y en caso de que la cámara seleccionada se esté visualizando esta operación estará inactiva. Si el proceso de monitorización está activo ésta operación ocasionará que éste se pare.

Nombre	Definir Máscara	Id	T006
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Se define una zona de interés sobre una cámara.
Condiciones previas	- Seleccionar una cámara. - Cámara accesible. - Cámara no visualizada
Disparador	Establecer una zona de interés sobre una cámara.
Flujo Normal	1- Ir a la pestaña Administrar 2- Selección de una cámara 3- Presionar el botón Definir Máscara 4- Se lanza una ventana con el marco de visión de la cámara seleccionada. 5- Definir la máscara hasta quedar satisfecho. 6- Cerrar la ventana emergente. 7- Respuesta del sistema a la acción
Excepción	No hay conexión con la cámara.
Actor Secundario	Gestor del sistema. Gestor BBDD
Canal hacia el actor secundario	Gestor: Definir Máscara. Gestor BBDD: Actualizar Máscara.
Notas	El punto de interés definido sustituirá al anterior punto si este existiera. Si la cámara se está visualizando esta operación no estará activa.

Tabla 3. 6 Caso de uso Definir Máscara

Borrar Máscara

Se trata de la acción contraria a la anterior. Con esta acción se elimina la zona de interés de la cámara seleccionada si esta existiese. Para ello basta con seleccionar la cámara en la que se desea eliminar la máscara. Tras ejecutar la operación se borrará tanto del sistema como de la base de datos la máscara de dicha cámara. Hay que recordar que si la cámara se está visualizando esta operación no estará activa. Si el proceso de monitorización está activo esta operación ocasionará que éste se pare.

Nombre	Borrar Máscara	Id	T007
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Se borra la zona de interés de una cámara.
Condiciones previas	- Seleccionar una cámara. - Cámara no visualizada.
Disparador	Establecer una zona de interés sobre una cámara.
Flujo Normal	1- Ir a la pestaña Administrar 2- Selección de una cámara 3- Presionar el botón Borrar Máscara 4- Respuesta del sistema a la acción
Excepción	No hay conexión con la cámara.
Actor Secundario	Gestor del sistema. Gestor BBDD
Canal hacia el actor secundario	Gestor: Borrar Máscara. Gestor BBDD: Actualizar Máscara.
Notas	Si la cámara se está visualizando esta operación no estará activa.

Tabla 3. 7 Caso de uso Borrar Máscara

Definir ID del capturador

Se trata de la operación para establecer el identificador del capturador. Esta operación está orientada para las cámaras USB/FireWire puesto que su conexión es directa al ordenador donde se ejecute S.A.V. y el reconocimiento de éstas se ve influido por el sistema operativo. Con ella el usuario puede modificar el índice del capturador de las cámaras que le fue asignado de manera automática según se añadieron al sistema.

Para ejecutar esta operación hay que dirigirse a zona de administración y seleccionar la cámara de la lista en la que se desea *Definir ID del capturador*. Esta acción se llevará a cabo desde un diálogo donde se seleccionará el índice deseado, siendo responsabilidad del usuario definir un índice válido.

Tras las modificaciones de los identificadores puede que el sistema sea incongruente y algunas cámaras no sean accesibles por lo que es vital realizar esta operación correctamente en la totalidad de las cámaras USB/FireWire. Si el proceso de monitorización esta activo esta operación ocasionará que éste se pare.

Nombre	Definir ID del capturador	Id	T008
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Modifica el ID del capturador de una cámara USB/FireWire
Condiciones previas	- Seleccionar una cámara.
Disparador	Verificar el funcionamiento de una cámara.
Flujo Normal	1- Ir a la pestaña Administrar 2- Selección de una cámara 3- Presionar el botón Definir ID del capturador 4- Se lanza una ventana emergente 5- Seleccionar el ID deseando 6- Respuesta del sistema a la acción
Excepción	No hay.
Actor Secundario	Gestor del sistema.
Canal hacia el actor secundario	Gestor: Definir ID.
Notas	El usuario es el responsable de la asignación de los IDs de las cámaras. Por lo que debe de asegurarse de realizar esta operación correctamente. Si la operación se realiza en una cámara IP no tendrá ningún efecto en el sistema.

Tabla 3. 8 Caso de uso Definir ID del capturador

Guardar Sistema

La operación guardar sistema nos permite crear un fichero donde se guardarán todos los identificadores de las cámaras que actualmente componen el sistema. Permitiéndonos poder cargar dicho fichero en otras ocasiones y tener un sistema de cámaras sin necesidad de estar añadiendo una a una para obtener el mismo sistema que teníamos anteriormente, evitando la pérdida de tiempo que esto conlleva.

Esta operación sólo es posible llevarse a cabo desde la zona de administración del sistema y como condición debe existir un número de cámaras mayor que uno. Una vez cumplidas estas precondiciones únicamente se debe ejecutar la operación *Guardar sistema*, dar el nombre y la ruta de salida del fichero. Puede ocurrir un error de fichero en la generación de éste por lo que la operación quedaría abortada. En cualquiera de los casos el sistema nos avisará del resultado de la operación. Esta operación produce un mensaje de log que queda registrado en la base de datos.

Nombre	Guardar Sistema	Id	T009
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Guarda en un archivo la configuración actual del sistema de cámaras.
Condiciones previas	- El usuario debe de estar registrado. - Que exista un conjunto de cámaras (más de una cámara).
Disparador	Realizar una copia de seguridad de la configuración del sistema actual.
Flujo Normal	1- Ir a la pestaña Administrar 2- Presionar el botón guardar 3- Indicar nombre del archivo de salida 4- En caso de existir reemplazarlo o variar el nombre 5- Confirmación de la acción 6- Respuesta del sistema a la acción 7- Registrar Log
Excepción	1- Fallo en el salvado del sistema.
Actor Secundario	Gestor. Gestor BBDD.
Canal hacia el actor secundario	Gestor: Guardar Sistema. Gestor BBDD: Guardar Log
Notas	En caso de coincidir el nombre del fichero en la ruta seleccionada se requerirá confirmación por parte del usuario para reemplazar el fichero existente.

Tabla 3. 9 Caso de uso Guardar Sistema

Cargar Sistema

Es la operación es inversa a la comentada con anterioridad, en este proceso se cargará un fichero que contiene la configuración del sistema realizando la demanda de los datos a la BBDD oportuna. Si existiera ya una configuración previa a la carga de un sistema como resultado obtendríamos la combinación de ambas. Para ejecutar dicha opción debemos ir a la zona de administración y ejecutar la operación de carga de un sistema, seleccionar el archivo y se procederá automáticamente a la carga del sistema seleccionado. Hay que asegurarse de que el fichero contenga un sistema ya que en caso contrario abortará la ejecución del software. Si alguna de las cámaras del fichero a cargar existe en el sistema actual ésta no se añadirá. La única excepción que se puede producir es un fallo en la conexión con la BBDD. Esta operación produce un mensaje de log que queda registrado en la base de datos y en caso de que el proceso de monitorización estuviese activo ésta operación ocasionará que éste se pare.

Nombre	Cargar Sistema	Id	T010
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Carga una configuración de cámaras desde un archivo.
Condiciones previas	- El usuario debe de estar registrado. - Es necesario que exista un fichero con la configuración de un sistema.
Disparador	Cargar un sistema creado con anterioridad.
Flujo Normal	1- Ir a la pestaña Administrar 2- Presionar el botón cargar 3- Seleccionar ruta 4- Confirmación de la acción 5- Carga y Actualización del sistema 6- Respuesta del sistema a la acción 7- Registrar Log
Excepción	1- Error total o parcial al cargar el sistema. 2- Fallo en la conexión con la BBDD.
Actor Secundario	Gestor. Gestor BBDD.
Canal hacia el actor secundario	Gestor: Cargar Sistema. Gestor BBDD: Obtener Cámara. Gestor BBDD: Registrar Log.
Notas	Si una cámara ya está definido en el sistema esta no se cargará

Tabla 3. 10 Caso de uso Cargar Sistema

Borrar Sistema

La función de esta operación no es otra que eliminar todas las cámaras del sistema. Dicha operación ocasionará la inoperatividad de la pestaña de monitorización en su modo *On line*. Para realizar esta operación los pasos a seguir son los siguientes, ir a la zona de administración y ejecutar la operación *Borrar sistema*. Esta operación se podrá realizar siempre y cuando no haya ninguna cámara visualizada, quedando registrada en la base de datos y finalizará el proceso de monitorización en caso de que estuviese activo.

Nombre	Borrar Sistema	Id	T011
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Borra el sistema actual.
Condiciones previas	- El usuario debe de estar registrado. - Debe existir un sistema que borrar. - No debe de haber ninguna cámara visualizándose.
Disparador	Desear borrar el sistema actual.
Flujo Normal	1- Ir a la pestaña Administrar 2- Apretar el botón Borrar 3- Respuesta del sistema a la acción 4- Registrar Log
Excepción	No hay.
Actor Secundario	Gestor. Gestor BBDD.
Canal hacia el actor secundario	Gestor: Borrar Sistema. Gestor BBDD: Registrar Log
Notas	Si la configuración no se ha guardado con anterioridad se perderá tras su borrado.

Tabla 3. 11 Caso de usos Borrar Sistema

Guardar Plano

La operación guardar plano nos da la opción de salvar mediante un fichero la configuración de las conexiones de las cámaras dentro del área vigilada. Esta operación sólo es posible llevarse a cabo desde la zona de administración del sistema cuando previamente ha sido creado o modificado un plano. Una vez cumplidas estas precondiciones únicamente hace falta ejecutar la operación *Guardar plano*, dar el nombre de salida del fichero y su ruta. En caso de que en la ruta seleccionada exista un fichero con el mismo nombre se le pedirá al usuario si desea reemplazarlo. Puede ocurrir un error en la generación del fichero que no permita realizar correctamente la operación.

Nombre	Guardar Plano	Id	T012
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Guarda en un archivo la configuración actual del plano.
Condiciones previas	- El usuario debe de estar registrado. - Que exista un plano y este se haya modificado.
Disparador	Realizar una copia de seguridad del plano actual.
Flujo Normal	1- Ir a la pestaña Administrar 2- Presionar el botón guardar plano 3- Indicar nombre del archivo de salida 4- Confirmación de la acción 5- Respuesta del sistema a la acción
Excepción	1- Fallo en el salvado del plano.
Actor Secundario	Plano.
Canal hacia el actor secundario	Plano: Guardar Plano.
Notas	No hay.

Tabla 3. 12 Caso de uso Guardar Plano

Cargar Plano

Es la operación inversa a la comentada con anterioridad, en este proceso se cargará desde un fichero la configuración del plano del sistema. Para poder cargar un plano en el sistema hace falta que haya como mínimo dos cámaras y si existiera un plano con anterioridad, éste sería sustituido por el cargado.

Para ejecutar dicha opción debemos ir a la zona de administración y usar la operación *Cargar sistema*, que mostrará un diálogo con el que poder seleccionar el archivo y tras hacerlo se procederá a la carga del plano. Las excepciones que se pueden producir son debidas a errores en la gestión o formulación del fichero o en la carga de éste. En caso de que el proceso de monitorización estuviese activo se pararía.

Nombre	Cargar Plano	Id	T013
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Carga una configuración de cámaras desde un archivo.
Condiciones previas	- El usuario debe de estar registrado. - Es necesario que exista un fichero con la configuración de un sistema. - Debe de haber dos o más cámaras en el sistema
Disparador	Cargar un sistema creado con anterioridad.
Flujo Normal	1- Ir a la pestaña Administrar 2- Presionar el botón cargar 3- Seleccionar ruta 4- Confirmación de la acción 5- Borrar el sistema anterior 6- Carga del sistema 7- Respuesta del sistema a la acción
Excepción	1- Error total o parcial al cargar el sistema. 2- Fichero mal construido.
Actor Secundario	Plano.
Canal hacia el actor secundario	Plano: Cargar Plano.
Notas	No hay.

Tabla 3. 13 Caso de uso Cargar Plano

Calcular Frames/Segundos

Mediante esta operación el sistema puede obtener una estimación de los cuadros por segundo a los que las cámaras responden. Debe de existir al menos una cámara para poder llevar a cabo el cálculo. Para ejecutar dicha opción debemos ir a la zona de administración y ejecutar la operación *Calcular frames/segundos*, la cual bloqueará durante diez segundos el sistema, tiempo en el que se realiza la operación. Si la cámara seleccionada para el cálculo se encuentra inoperativa el sistema dará un valor por defecto medio y como en las operaciones anteriores si el proceso de monitorización está activo, éste se parará.

Nombre	Calcular Frames/Segundos	Id	T014
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Calcular una estimación de los cuadros/segundos del sistema.
Condiciones previas	- El usuario debe de estar registrado. - Debe de haber una o más cámaras en el sistema
Disparador	Calcular los cuadros/segundos
Flujo Normal	1- Ir a la pestaña Administrar 2- Presionar el botón calcular cuadros/segundos 3- Respuesta del sistema a la acción
Excepción	No hay.
Actor Secundario	Gestor del sistema.
Canal hacia el actor secundario	Gestor: calcular Frames/segundos.
Notas	No hay.

Tabla 3. 14 Caso de uso Calcular Frames/Segundos

Configurar S.A.V.

Configurar S.A.V. es la operación para modificar y establecer la configuración del software al arrancar. Esta operación se encarga de producir un fichero de configuración que el software usará a posteriori. En él podremos seleccionar por ejemplo que se realice una carga predefinida desde archivo y establecer los parámetros de acceso a la base de datos, etc. Pequeños detalles de configuración que agilicen la puesta en marcha del software. Para realizar esta operación debe dirigirse a la zona de administración y ejecutar la operación *Configurar*, mostrándose una ventana emergente en la que introducir/modificar los datos del software y al guardar éstos generaremos el fichero de configuración. La única excepción que puede ocurrir es que se produzca un fallo en la creación del fichero.

Nombre	Configurar S.A.V.	Id	T015
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Permite configurar los parámetros del programa
Condiciones previas	- El usuario debe de estar registrado.
Disparador	Desear configurar el sistema.
Flujo Normal	1- Ir a la pestaña Administrar 2- Apretar el botón configurar 3- Realizar las configuraciones 4- Aceptar 5- Respuesta del sistema a la acción
Excepción	Error en la configuración.
Actor Secundario	No hay.
Canal hacia el actor secundario	No hay.
Notas	Establece la configuración que se ejecutará al arrancar el sistema. Si esta es modificada para que los cambios surjan efectos es necesario reiniciar el software.

Tabla 3. 15 Caso de uso Configurar S.A.V.

Ver Log

Mediante esta operación podremos obtener y visualizar los Log del sistema o alertas de un sujeto concreto entre dos fechas. Para poder ejecutar esta operación hay que ir a la zona de administrar y ejecutar la operación *Ver log*, seleccionar el usuario si se desea obtener los registros de un sujeto concreto e introducir las fechas en el que se quieren ver. El resultado de esta operación podrá verse en la *Ventana de log* del programa y como posible excepción se puede producir un fallo de conexión con la BBDD.

Nombre	Ver Log	Id	T016
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Muestra las actividades registradas en el log entre las dos fechas indicadas.
Condiciones previas	- El usuario debe de estar registrado. - Introducir las fechas correctamente. - En caso de introducir un usuario/cámara esta debe de existir
Disparador	Querer ver el registro de log.
Flujo Normal	1- Ir a la pestaña Administrar 2- Apretar el botón del log 3- Realizar las configuraciones 4- Aceptar 5- Visualizar
Excepción	1- Fallo al conectar con la BBDD.
Actor Secundario	Gestor BBDD.
Canal hacia el actor secundario	Gestor BBDD: Obtener log.
Notas	No hay.

Tabla 3. 16 Caso de uso Ver Log

Gestionar BBDD

Gestionar BBDD es la operación que hace de interfaz con una herramienta de gestión de bases de datos PostgreSQL, donde se podrá realizar la puesta a punto de las bases de datos del sistema. Por lo tanto al ejecutar la operación *Gestionar BBDD* se obtendrá un submenú para administrar las bases de datos del sistema (añadir cámaras, editarlas, eliminarlas, añadir usuarios, eliminarlos y borrar el log entre otras). Se trata de una operación restringida y sólo tendrán acceso a ella los administradores. Con privilegios de administrador del sistema se tendrá un acceso restringido a las operaciones de gestión de la base de datos mientras que con privilegios de administrador de S.A.V. la accesibilidad a las operaciones será total.

Como excepción puede ocurrir que la base de datos esté caída y los cambios que se realicen en las tablas de la base de datos que afecten al sistema actual no tendrán efecto hasta que el software sea reiniciado. Las operaciones de gestión de la base de datos producen mensajes de log que quedan registrados en la base de datos.

Nombre	Gestionar BBDD	Id	T017
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador (con privilegios de administrador de sistema o de S.A.V.)
Descripción	Función para gestionar las bases de datos del sistema.
Condiciones previas	- El usuario debe de estar registrado. - Que existan las BBDD y tener acceso a estas.
Disparador	Necesidad de gestionar una base de dato.
Flujo Normal	1- Ir a la pestaña Administrar 2- Apretar el botón de gestionar BBDD 3- Seleccionar la operación a realizar 4- Rellenar campos si fuese necesario 5- Respuesta del sistema a la acción
Excepción	1- Fallo en la conexión con la BBDD.
Actor Secundario	Gestor BBDD.
Canal hacia el actor secundario	Gestor BBDD: Operaciones sobre las tablas.
Notas	Cualquier modificación en la base de datos no afectará al sistema actual en caso de a tañarle en algo hasta que este se reinicie.

Tabla 3. 17 Caso de uso Gestionar BBDD

Gestionar Plano

La siguiente operación es la encargada de gestionar la representación de las cámaras en el plano de la zona monitorizada. Muestra la representación del plano mediante una matriz de adyacencia con las cámaras conectadas entre sí, además de permitir conectar/desconectar nuevas cámaras o las que ya estaban. Es necesario que al menos exista un mínimo de dos cámaras para poder llevar a cabo esta operación. Para realizar la gestión del plano debemos dirigirnos a la zona de administración y ejecutar la operación *Gestionar plano*, obteniéndose un diálogo en el que se mostrará la situación actual del plano, además de dar la posibilidad de conectar/desconectar las cámaras existentes en el sistema actual. Al conectar dos cámaras se puede seleccionar el peso que esta tendrá al añadirse y la dirección. El peso de las cámaras conectadas a una cámara es siempre constante (100) por lo que según se conectan/desconectan las cámaras los pesos se recalculan. Si el proceso de monitorización está activo esta operación ocasionará que éste se pare.

Nombre	Gestionar Plano	Id	T018
Creado	Carlos Falcón	Modificado	-

Actor Principal	Administrador. (En cualquiera de sus roles)
Descripción	Operación que muestra una representación del plano actual y muestra una serie de operaciones sobre este.
Condiciones previas	- El usuario debe de estar registrado.
Disparador	Gestionar las BBDD.
Flujo Normal	1- Ir a la pestaña Administrar 2- Apretar el botón de gestionar plano 3- Lanzar diálogo 4- Gestionar 5- Salida del sistema
Excepción	No hay.
Actor Secundario	Plano.
Canal hacia el actor secundario	Plano: Conectar/Desconectar.
Notas	El administrador es el responsable de la correctitud de este.

Tabla 3. 18 Caso de uso Gestionar Plano

Monitorizar

El proceso de monitorización es el más complejo de todos, ya que nos permite múltiples configuraciones en la tarea de monitorizar sistemas. Existen dos estilos de monitorización, en vivo (*Online*) y basada en el análisis de vídeos (*Offline*). Dentro de la monitorización en vivo existen tres tipos de monitorización, *Manual*, *Automática* o *Por eventos* que necesitan que haya definido un sistema con al menos una cámara. Las dos primeras opciones realizan tratamiento de las imágenes capturadas y es posible definir una serie de parámetros a aplicar, mientras que la tercera se vale de los datos recogidos en la base de datos del software. A continuación se describen con más detalles los tres tipos de monitorización:

- Manual. El vigilante es el encargado de seleccionar la cámara a monitorizar en cualquier momento, haciendo que el software se centre únicamente en la cámara elegida, ejecutándose la configuración realizada y generando las alertas de la cámara seleccionada.
- Automática. Tras el paso previo de configuración del sistema de monitorización se procederá a una monitorización continua de las cámaras (sistema seleccionado) hasta que se produzca una alarma que ponga en marcha el algoritmo de seguimiento que se haya seleccionado. Si este se pierde se continuará con el proceso de monitorización continua. En todo este proceso el vigilante tendrá la función de supervisión del sistema e ir viendo las alertas que se generan en el sistema.
- Eventos. Esta opción lo que nos permite es realizar configuraciones distribuidas del procesamiento del sistema. Ya que si seleccionamos esta opción, el software no hará procesamiento de las imágenes y la selección de ésta dependerá de los log de las cámaras registrado en la base de datos. Este modelo permite lanzar la vista de todas las cámaras del sistema de las cuales no realizará ningún procesamiento.

Mientras que en la monitorización de vídeos se necesita de un fichero que contenga como mínimo una ruta válida y accesible donde se encuentre definido al menos un vídeo. Este tipo de monitorización sirve para observar el comportamiento del software ante escenarios simulados.

Nombre	Monitorizar	Id	T019
Creado	Carlos Falcón	Modificado	-

Actor Principal	Vigilante
Descripción	Operación para monitorizar el sistema.
Condiciones previas	Debe de existir al menos una cámara
Disparador	Ejecutar el proceso de monitorización de una cámara o un sistema.
Flujo Normal	1- Administrar previa del SAV (sistema) 2- Ir a la pestaña Monitorizar 3- Configuración de monitorización 4- Ejecutar monitorización 5- Interacción con el software
Excepción	No hay.
Actor Secundario	Capturador.
Canal hacia el actor secundario	Capturador: Seleccionar Imagen.
Notas	Es necesario tener el sistema con el que se va a interactuar, para las cámaras IP hace falta que exista una red con conexión.

Tabla 3. 19 Caso de uso Monitorizar

Inicializar

Se trata de una operación totalmente transparente para el usuario final. Dicha operación es la encargada de configurar el software al iniciarse éste, evitando y/o facilitando la labor al usuario en la tarea de configuración del software. Esta operación puede no llevarse a cabo en caso de que no exista un fichero de configuración en el directorio raíz de la aplicación y como única excepción puede ocurrir que el fichero esté mal formado y provoque problemas de lectura.

Nombre	Inicializar	Id	T020
Creado	Carlos Falcón	Modificado	-

Actor Principal	S.A.V..
Descripción	Inicializar el software con la configuración establecida en anteriores ejecuciones.
Condiciones previas	Que este operativa la base de datos y se pueda acceder a esta.
Disparador	Al ejecutarse S.A.V..
Flujo Normal	1- Ejecutar S.A.V.. 2- Cargar configuración inicial
Excepción	1- Problema en la lectura del fichero.
Actor Secundario	Gestor, Gestor de BBDD, Gestor de Plano.
Canal hacia el actor secundario	Gestor: Añadir Cámaras. Gestor de BBDD: Obtener Cámaras Gestor de Plano: Cargar Plano
Notas	No hay.

Tabla 3. 20 Caso de uso Inicializar

3.3.2- Modelado de análisis.

En los diagramas de colaboración de S.A.V. Figuras 3.6, 3.7 y 3.8, se muestran las interacciones entre los objetos, creando enlaces entre ellos y añadiendo mensajes a esos enlaces. Se ha tratado que el nombre de los mensajes denote el propósito del objeto invocante en la iteración con el objeto invocado. En la Figura 3.7 se representan las colaboraciones en las tareas de monitorización, mientras que en la Figura 3.8 se representan las colaboraciones en las actividades de inicialización de la aplicación y por último en la Figura 3.8 se representan las colaboraciones en las actividades de administración.

Monitorizar:

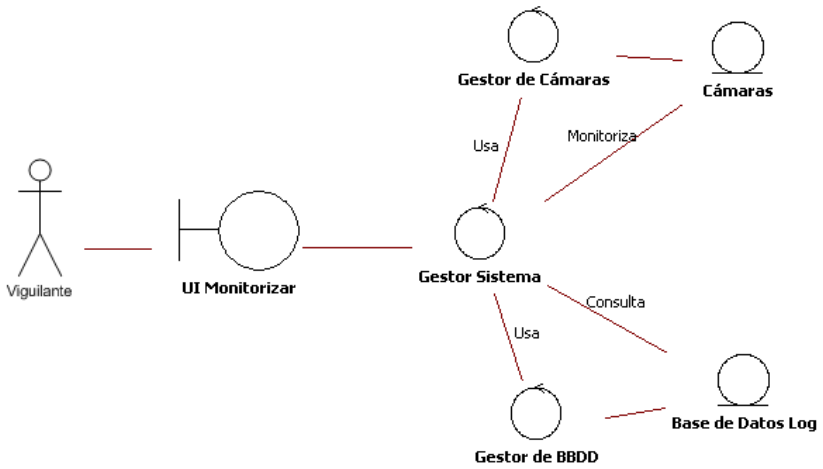

Figura 3. 6 Diagrama de colaboración para la función Monitorizar

Inicializar:

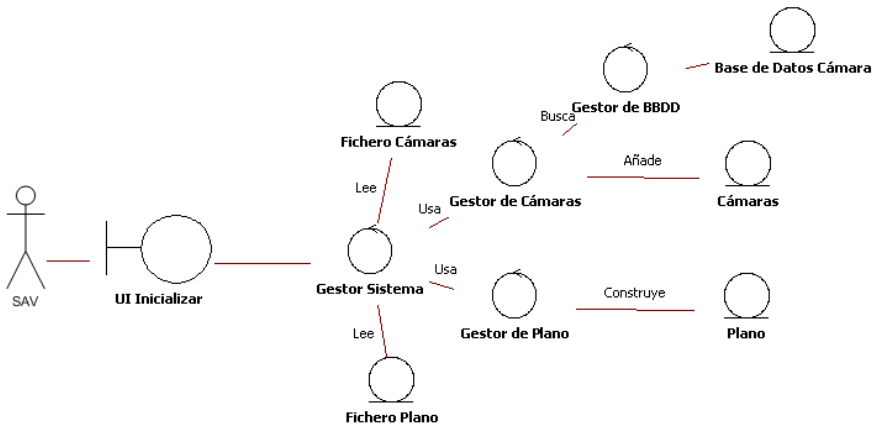

Figura 3. 7 Diagrama de colaboración para la función Inicializar

Administrar:

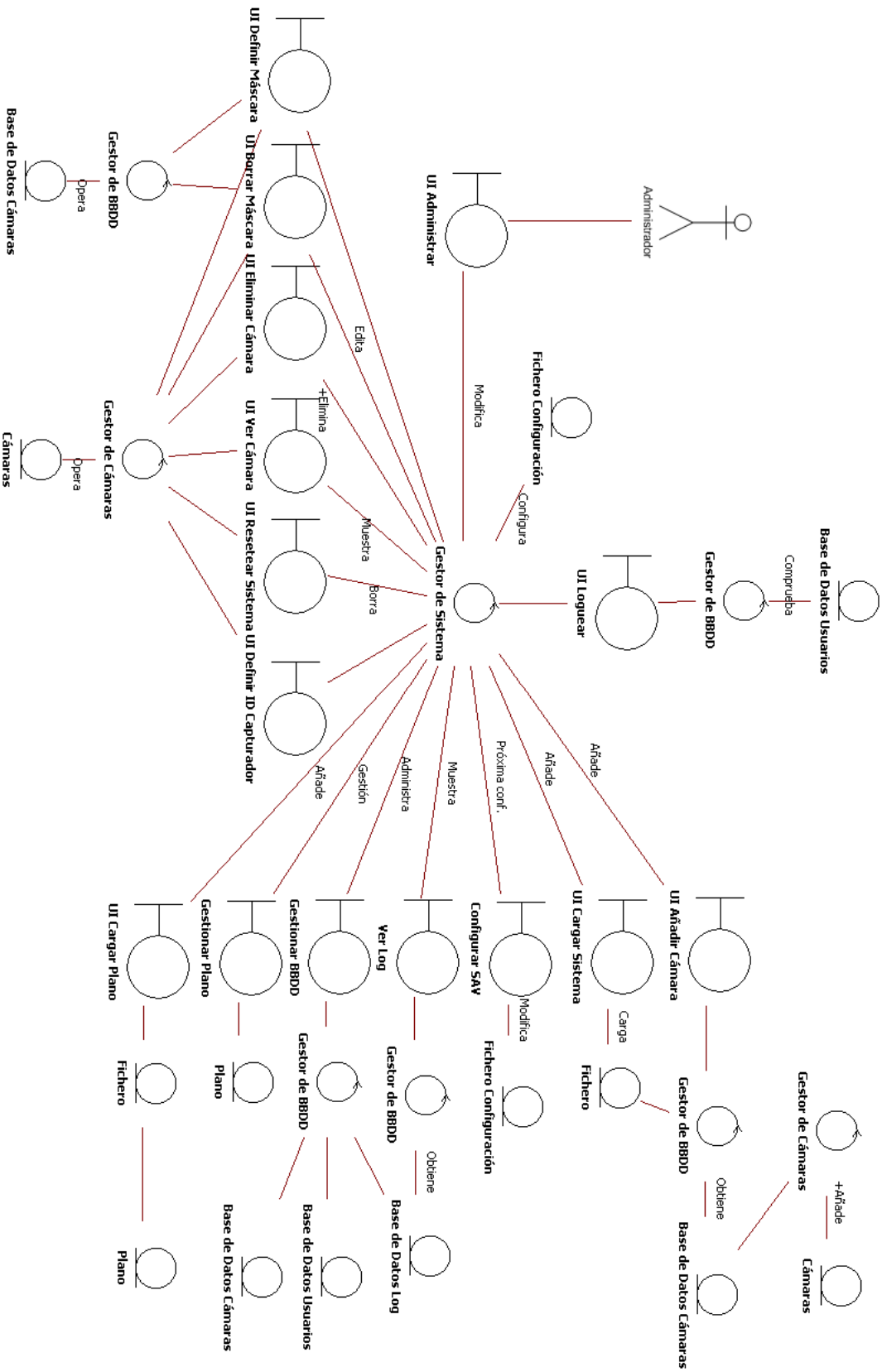

Figura 3. 8 Diagrama de colaboración para la función Administrar

3.3.3- Prototipo de la interfaz.

El desarrollo de prototipos es una herramienta útil para comunicar, distribuir y definir las ideas entre las partes responsables, así como para clarificar los requisitos. Por lo tanto, el uso de esta técnica, en conjunción con la definición de los casos de uso, nos ha permitido completar el análisis de requisitos. El enfoque utilizado para el desarrollo del presente PFC ha sido el prototipo evolutivo, que si bien ha requerido más tiempo para su creación, nos proporciona un resultado parecido al producto final.

3.3.4- Construcción del prototipo

Para la construcción de prototipos se seleccionó la tecnología wxWidgets (wxMSW bajo Windows) que es una librería para realizar interfaces multiplataforma. Hay que aclarar que presenta cierta dificultad para realizar las interfaces (al usar esta librería con el Visual Studio) ya que carece de una herramienta *Guide* por lo que el desarrollo de las interfaces es algo costoso y lento. No obstante se ha continuado usando la librería por sus prestaciones y su posibilidad multiplataforma, únicamente se ha tenido que realizar un esfuerzo más exhaustivo en el apartado del diseño de las interfaces para posteriormente implementarlas. Como cabía de esperar el ciclo de desarrollo de las interfaces ha sido evolutivo y ha sufrido bastantes variaciones a lo largo de su desarrollo hasta que se alcanzó una plantilla casi definitiva. La interfaz que se muestra en la Figura 3.9 es el primer prototipo que se realizó para conocer las posibilidades de la herramienta y hacernos una idea inicial de las necesidades del software a realizar.

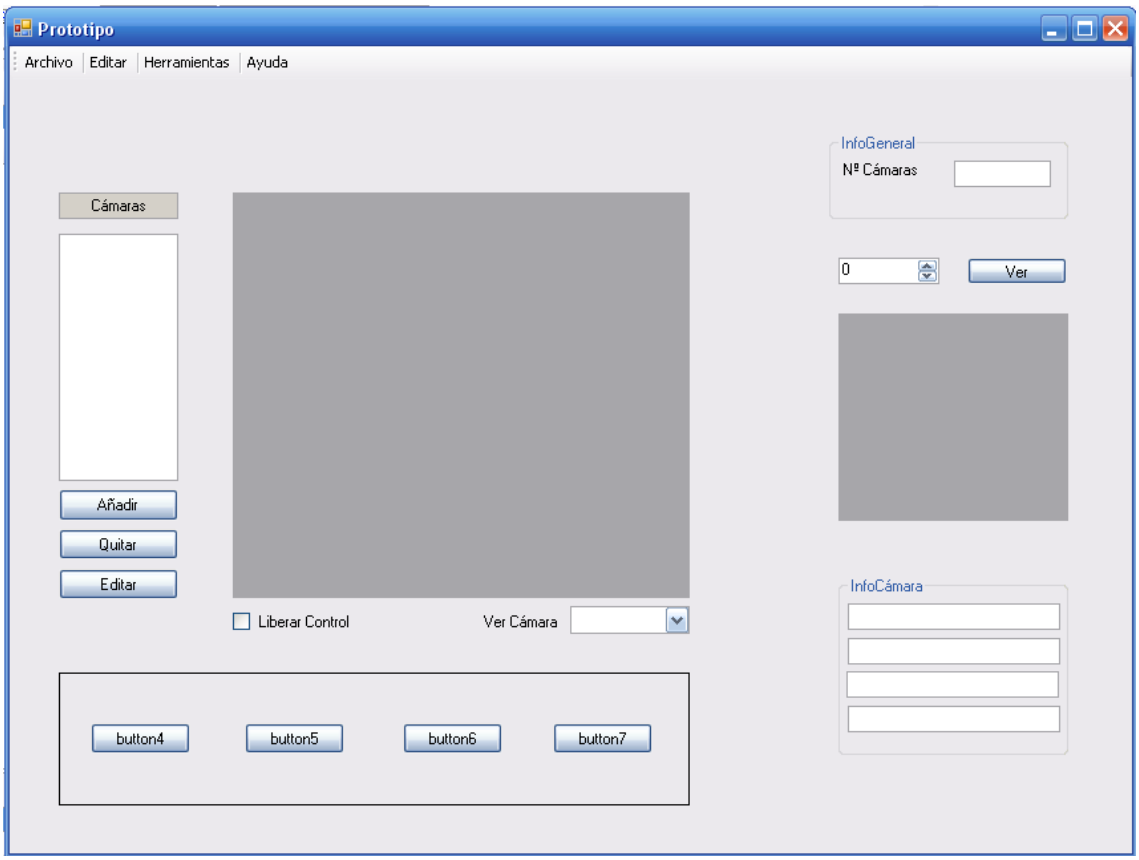

Figura 3. 9 Diseño inicial de la interfaz

Tras estudiar las posibilidades que ofrecía esta interfaz y observar sus carencias, entre la que destacó la necesidad de separar las tareas de administración y monitorización debido a un cambio en la gestión de los sistemas. Pasándose de un software basado en ficheros a uno centralizado en una base de datos, por lo que se realizó otro análisis que conllevaría realizar cambios importantes en el diseño.

Al desecharse este modelo de interfaz y se construyó uno nuevo que permitía una mayor usabilidad y amigabilidad del software, separándose las vistas según las necesidades del tipo de usuario (rol) que interactuaba con el software.

Diseñándose las dos plantillas que se muestran en la Figura 3.10 y 3.11 que satisfacían las demandas del usuario, por lo que se siguió con su diseño y se fueron implementando a lo largo del desarrollo del PFC.

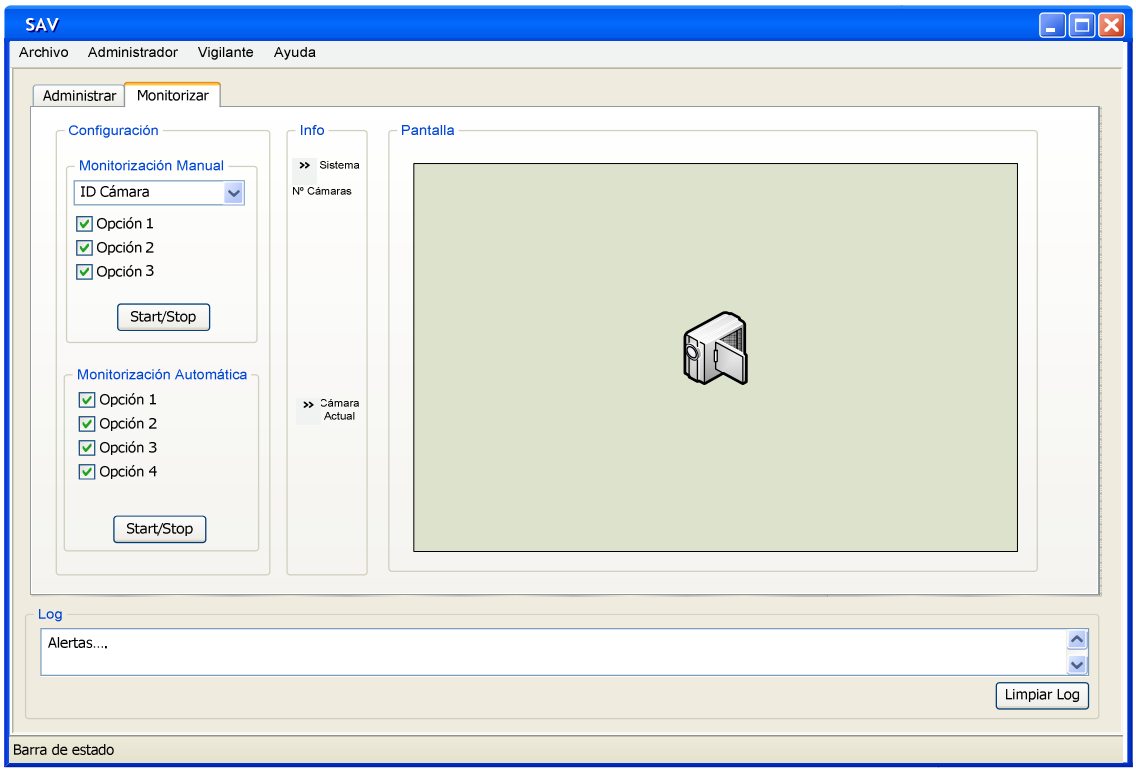

Figura 3. 10 Prototipo Ventana de Monitorizar

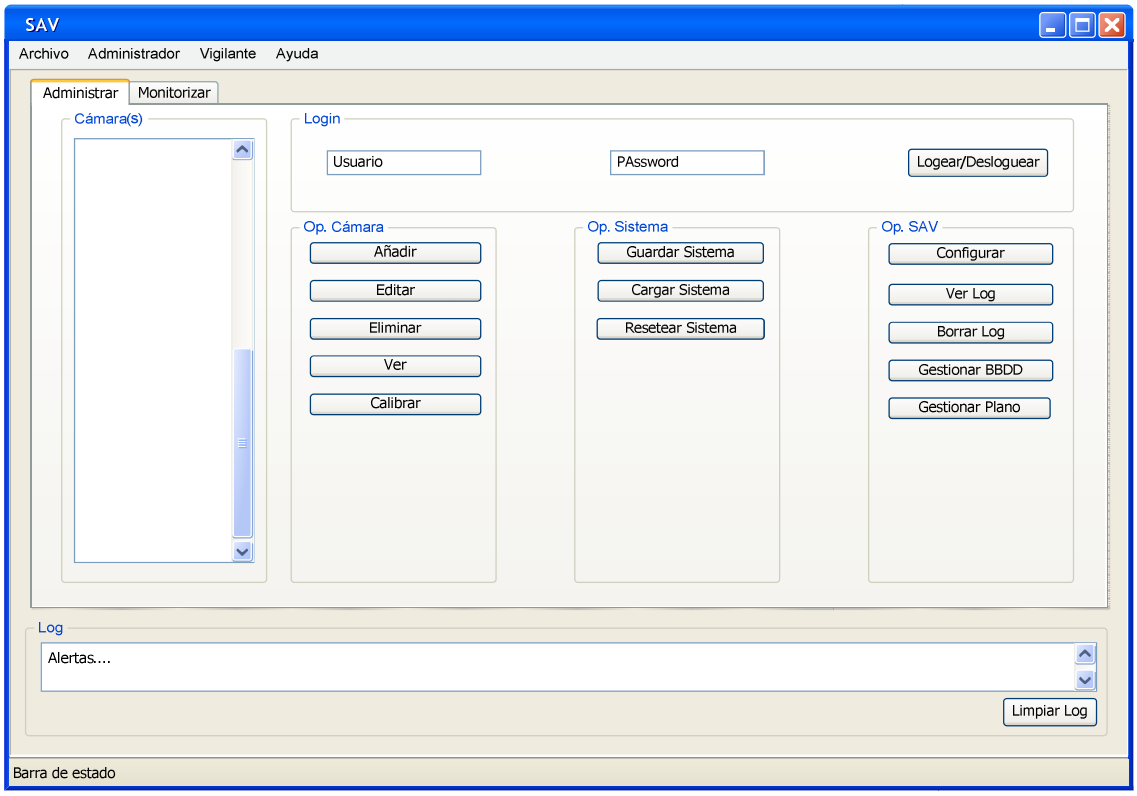

Figura 3. 11 Prototipo Ventana de Administrar

En el paso de plasmar las ideas de la plantilla al software final no fue directo y este se fue transformando al surgir nuevas necesidades, no obstante el esqueleto del programa se mantuvo hasta la ampliación de la aplicación final. Por ello podemos destacar como partes integrantes del software su barra de menú, sus dos pestañas *Administrar* y *Monitorizar* y un *Log del sistema* que siempre está visible. Llegados a este punto se comentarán los aspectos de análisis que se tuvieron en cuenta a la hora de detallar la interfaz.

Barra de menú

En este hito se decidió incluir los siguientes tres submenús *Archivo*, *Opciones* y *Ayuda*.

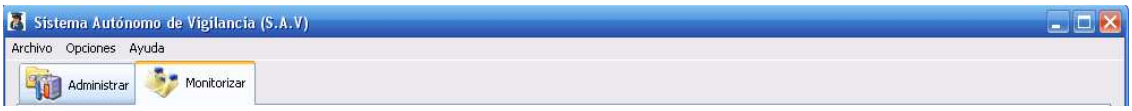

Figura 3. 12 Diseño barra de menú

En el primero de ellos, Figura 3.13, encontramos las operaciones relativas al software y en un principio se ha optado por incluir solamente la opción de abandonar el software.

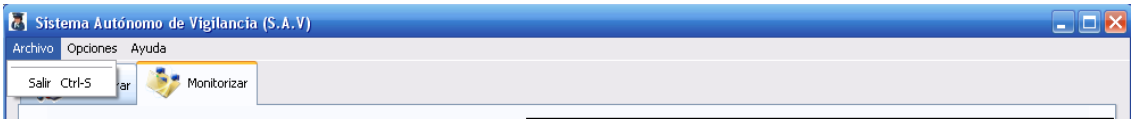

Figura 3. 13 Menú Archivo

El menú de Opciones, Figura 3.14, va destinado a la configuración del software y a las operaciones globales de éste. Entre las que destacan las operaciones de configuración y ver todas las cámaras**.** La opción de *Configurar*, Figura 3.15*,* permite al usuario establecer los parámetros del software. Mientras que la operación de *Mostrar todas las cámaras* deberá lanzar ventanas emergentes que permitan visualizar individualmente todas las cámaras del sistema. Por último en el menú de ayuda, Figura 3.16, se dispondrá de la operación para lanzar la información del software que al pulsar sobre éste, se lanzará una ventana emergente, Figura 3.17.

Figura 3. 14 Menú Opciones

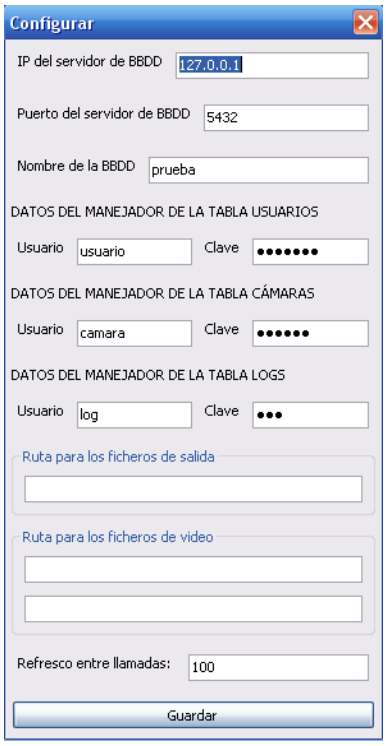

Figura 3. 15 Diálogo Configurar

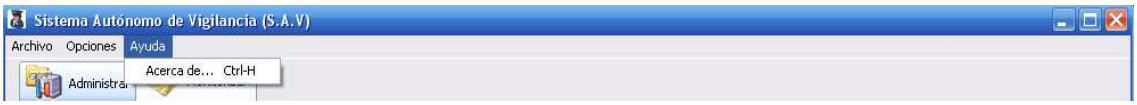

Figura 3. 16 Menú Ayuda

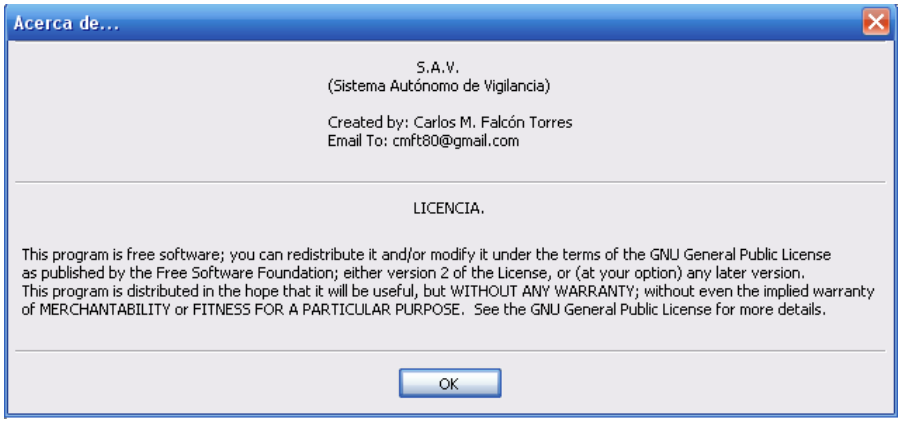

Figura 3. 17 Diálogo de ayuda

Ventana del administrador del sistema

En la ventana del administrador podemos encontrar todas las herramientas que nos permiten configurar y gestionar la aplicación. En esta pestaña es necesario registrarse, por lo que una vez realizado este paso y según los privilegios que se posea se dará acceso a los diferentes menús.

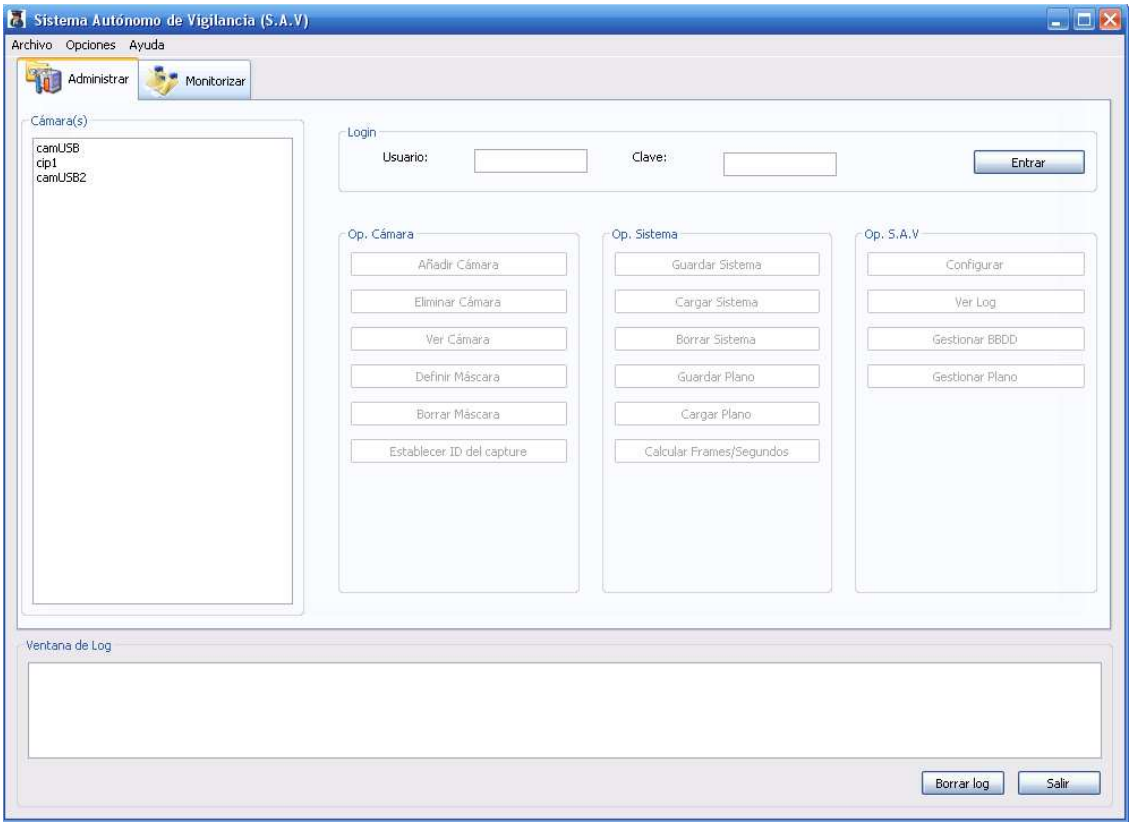

Figura 3. 18 Pestaña Administrar (I)

En la Figura 3.18 se muestra la pestaña *Administrar* no existe un usuario registrado en el sistema. Mientras que en la Figura 3.19 se muestra lo que vería un usuario con privilegios de administrador. En esta misma ventana si se selecciona una cámara del sistema se activarán más opciones, relativas a funcionalidades con respecto a la cámara seleccionada.

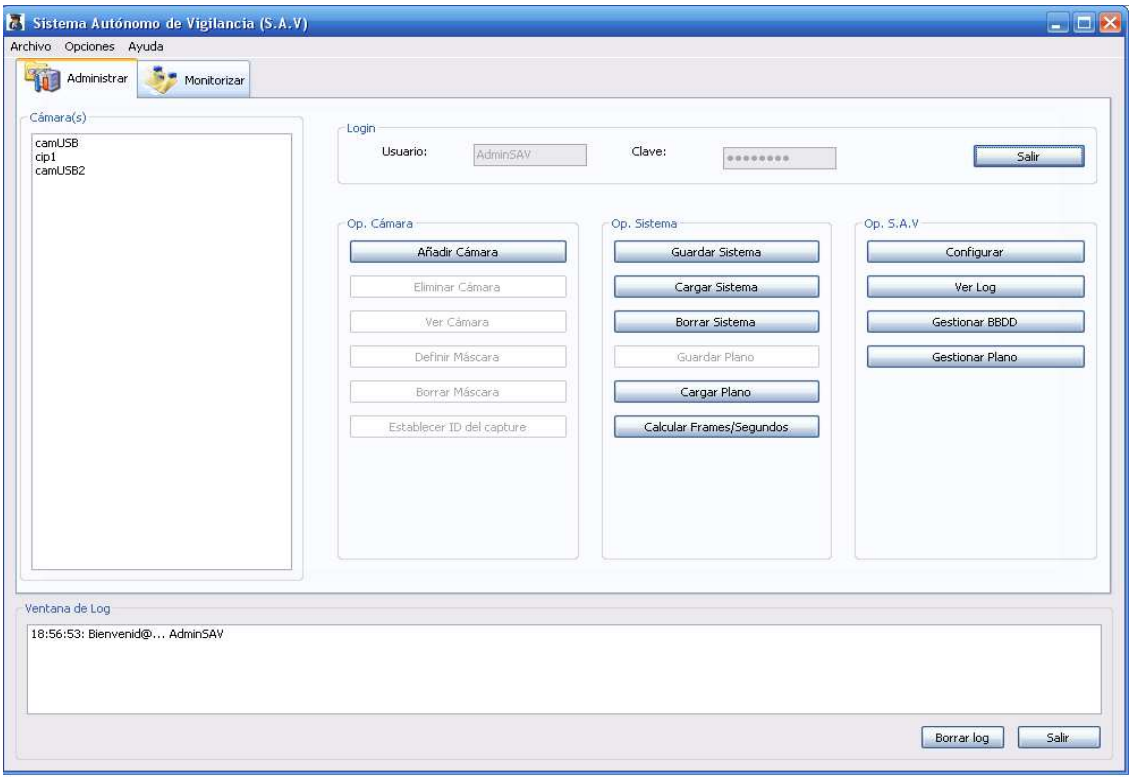

Figura 3. 19 Pestaña Administrar (II)

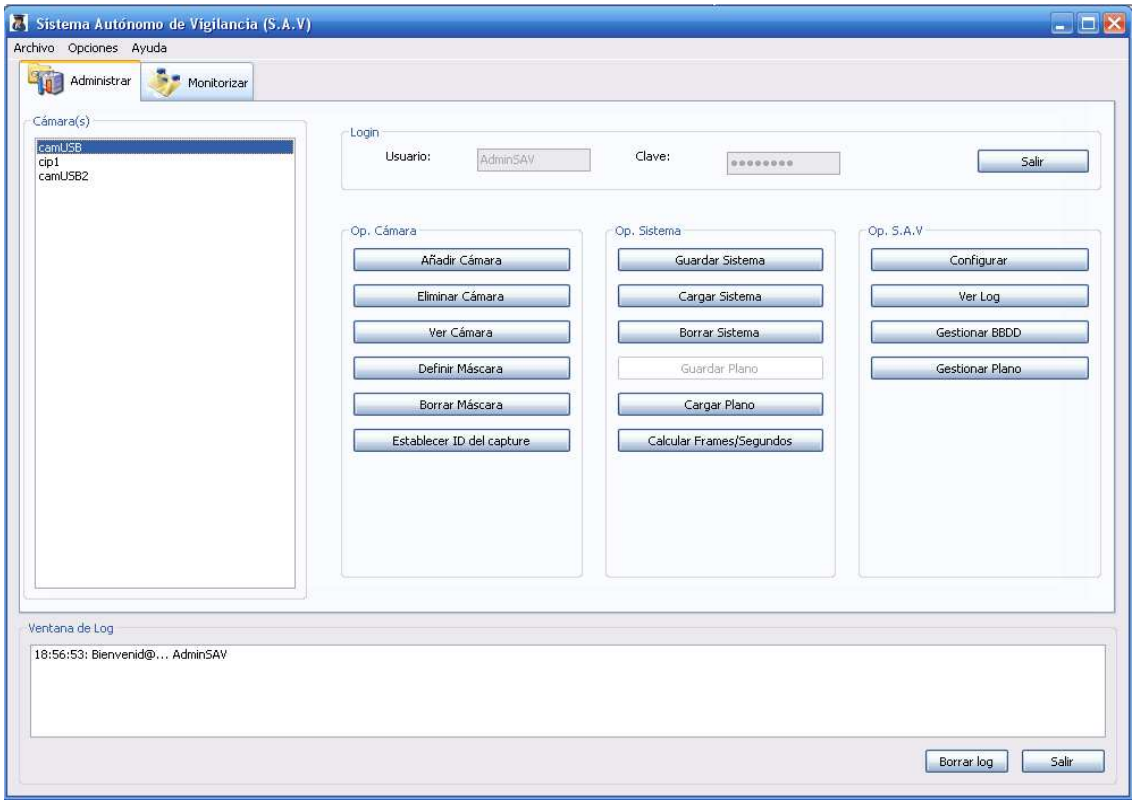

Figura 3. 20 Pestaña Administrar (III)

Una vez vista la ventana en términos generales pasaremos al estudio de los distintos submenús dentro de la pestaña *Administrar* en la que se pueden observar tres agrupaciones de botones: *Op. Cámara*, *Op. Sistema* y *Op. S.A.V.*.

Submenú Op. Cámara

Como su nombre hace referencia, en el se contienen las operaciones relativas a las cámaras del sistema:

- *Añadir Cámara(s)* al sistema actual.
- *Eliminar la Cámara* seleccionada del sistema actual
- *Ver Cámara* donde se puede observar la operatividad de la cámara seleccionada.
- *Definir Máscara*, posibilita la definición de puntos de interés en las cámaras del sistema
- *Borrar Máscara*, su objetivo es eliminar la máscara definida en un sistema.
- *Establecer ID del capturador*, operación para configurar la conexión de las cámaras USB/FireWire.

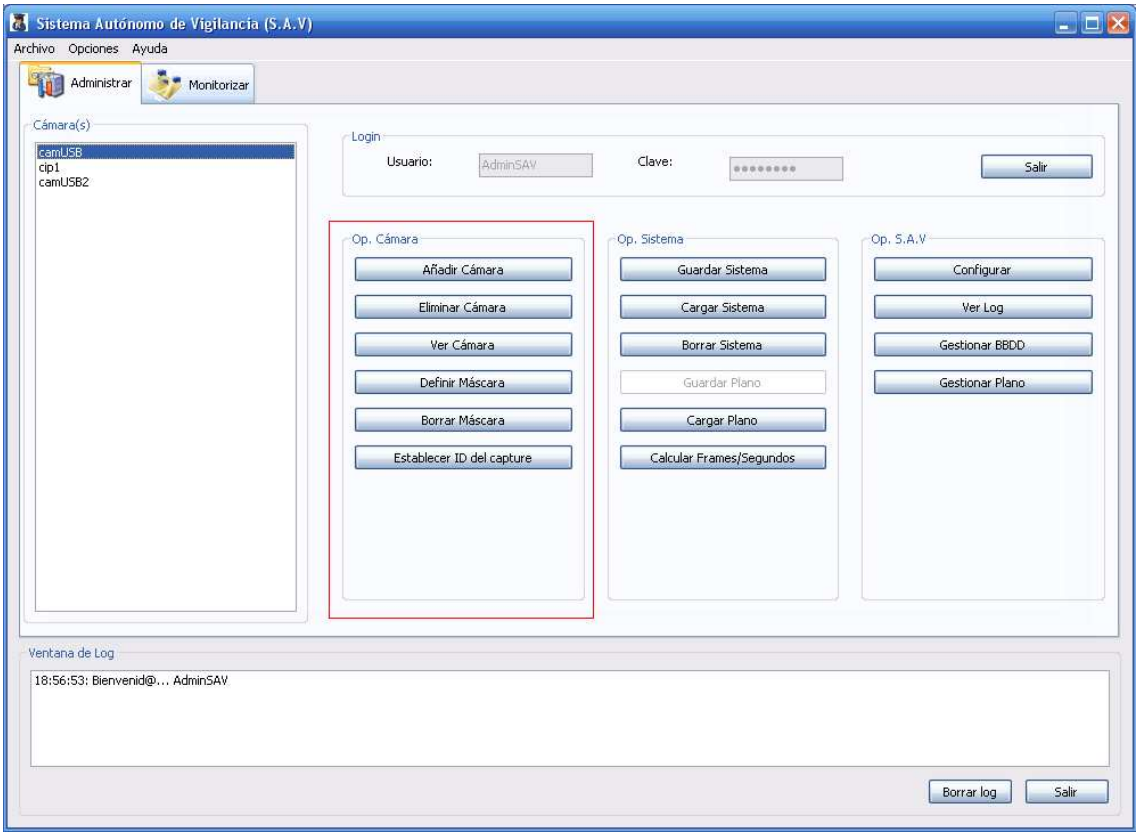

Figura 3. 21 Pestaña Administrar. Submenú Op. Cámara

Los únicos diálogos de este submenú, que presentarán ventanas emergentes son *Añadir cámara* y *Establecer ID del capturador*.

La ventana emergente de añadir cámara se muestra en la Figura 3.22. Como se puede observar en la imagen existen dos opciones, seleccionar una cámara desde la base de datos o añadiendo un subgrupo de cámaras a partir de una sentencia SQL. Mientras que la estructura de la ventana emergente del botón *Establecer ID del capturador* es la mostrada en la Figura 3.23.

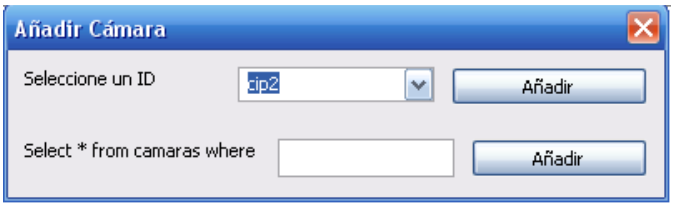

Figura 3. 22 Diálogo Añadir cámara

Figura 3. 23 Diálogo Establecer ID del capturador

Submenú Op. Sistema

Este submenú está orientado a las distintas operaciones que afectan al sistema, entre ellas tenemos:

- *Guardar Sistema*, que generará un fichero con los identificadores de las cámaras que componen el sistema actual.
- *Cargar Sistema*, mediante esta operación podemos cargar las cámaras desde la base de datos que se hayan leído del fichero indicado.
- *Borrar Sistema*, operación que borra el sistema actual.
- *Guardar Plano* operación que se activará cuando el plano sufra una modificación que nos permite realizar un fichero con el plano actual.
- *Cargar Plano*, esta opción permite cargar desde un fichero la configuración de las cámaras actual.
- *Calcular Frames/Segundos*, calculo de una estimación de la recepción de cuadros/segundos del sistema.

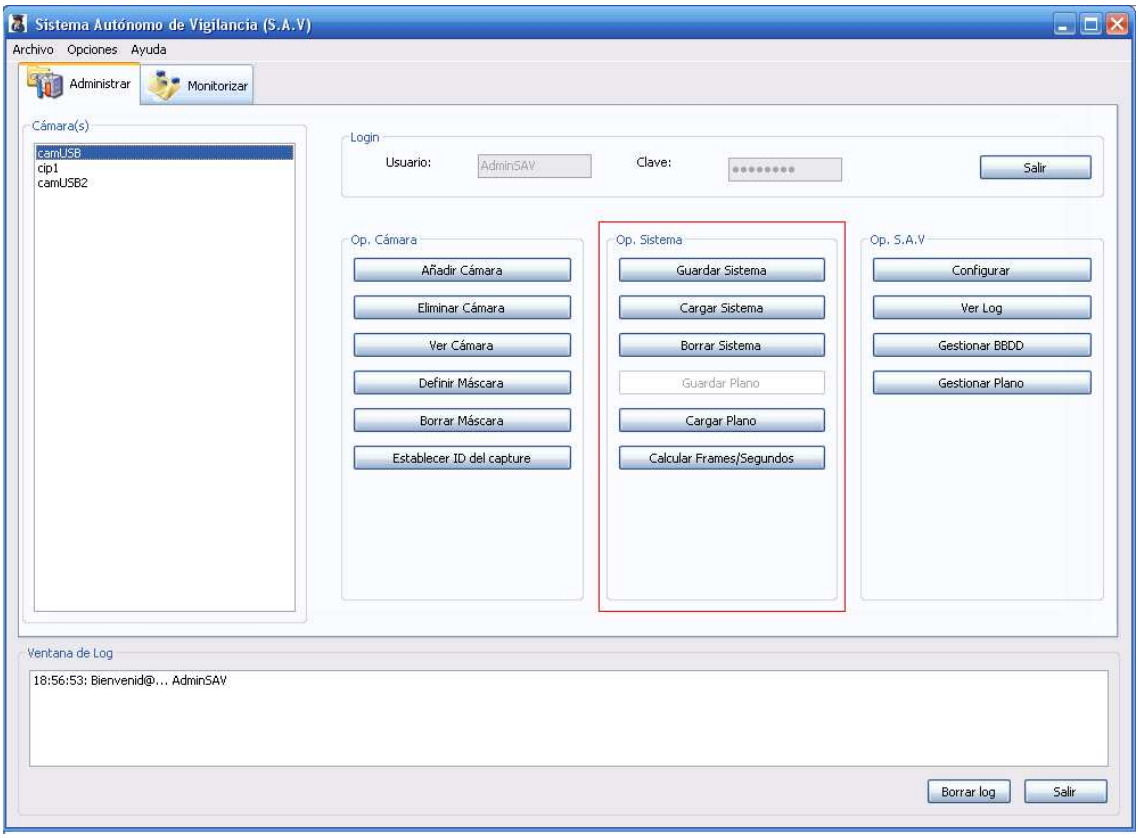

Figura 3. 24 Pestaña Administrar. Submenú Op. Sistema

Como en el caso anterior las siguientes capturas nos mostrarán los diálogos correspondientes a las distintas opciones que nos permite este submenú. Empezaremos mostrando la estructura del diálogo lanzado por las operaciones de *Guardar Sistema y Plano,* Figura 3.25. En él se nos debe dar la posibilidad de seleccionar la ruta y el nombre del fichero con el que queremos nombrar al nuevo sistema o al plano y en caso de que el nombre ya exista se pedirá al usuario confirmación para sobre escribirlo, Figura 3.26.

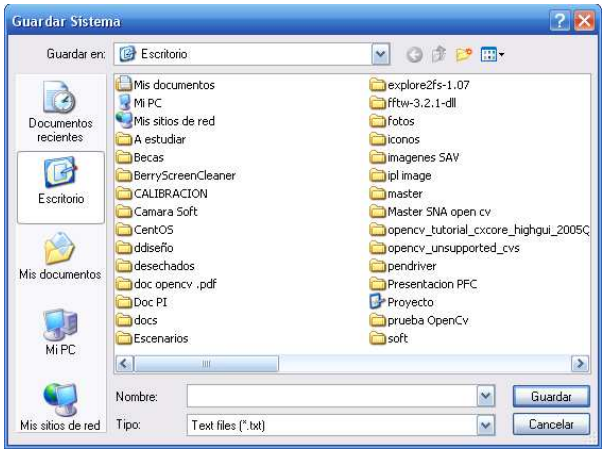

Figura 3. 25 Diálogo Guardar Sistema

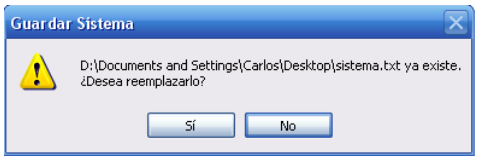

Figura 3. 26 Diálogo Guardar Sistema

El diálogo correspondiente a las operaciones de *Carga Sistema* y *Cargar Plano* debe de presentar la estructura que se muestra en la Figura 3.27.

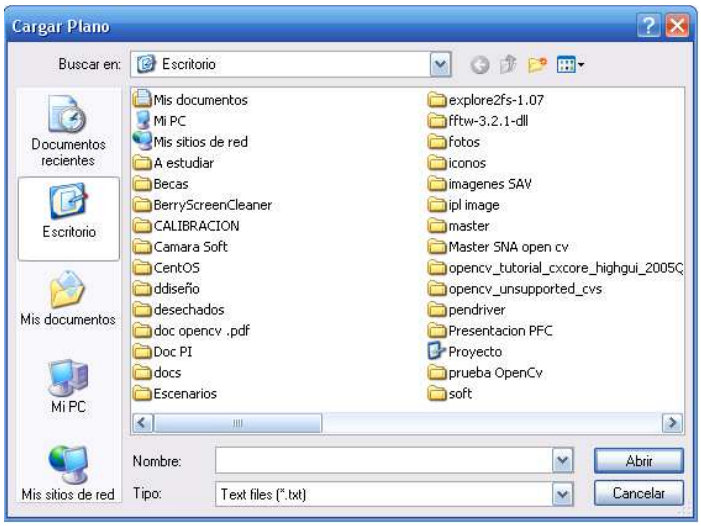

Figura 3. 27 Diálogo Cargar Plano

Las demás operaciones no presentan diálogos y simplemente se limitan a modificar los contenidos del sistema como en el caso de *Borrar Sistema* o la configuración del software como en el caso de *Calcular Frames/Segundos.*

Submenú Op. S.A.V.

En este último submenú se muestran las operaciones relevantes a la aplicación:

- *Configurar*, en esta opción podemos definir los parámetros de configuración de S.A.V..
- *Ver Log*, esta operación nos permite obtener los log entre dos fechas.
- *Gestionar BBDD*, en este botón podemos configurar y administrar las bases de datos.
- *Gestionar Plano*, por último este botón nos permiten gestionar el plano actual.

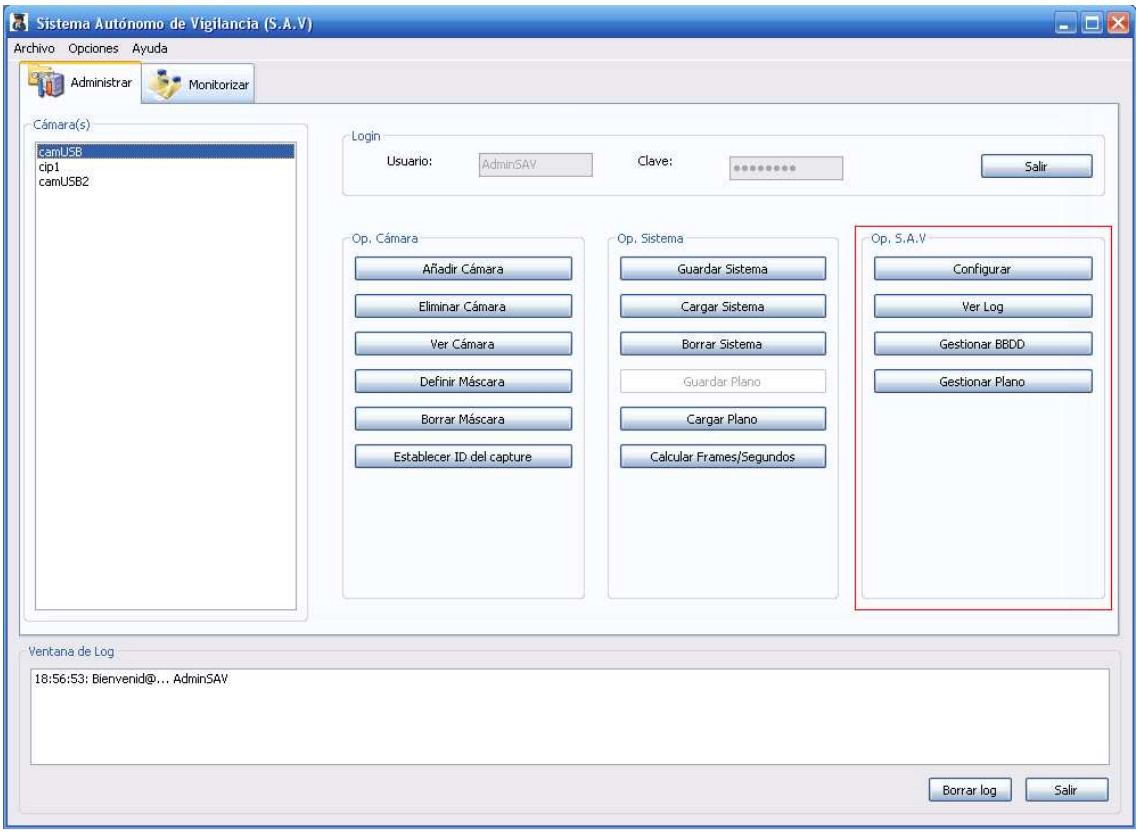

Figura 3. 28 Pestaña Administrar. Submenú Op. S.A.V.

Como en las otras ocasiones a continuación se pasará a describir los diálogos del submenú. Se empezará describiendo el diálogo del botón *Configurar, Figura 3.29*, en el que como se puede observar en la siguiente imagen se da la posibilidad al usuario de configurar los parámetros del software que serán cargados al iniciar el software en futuras ocasiones.

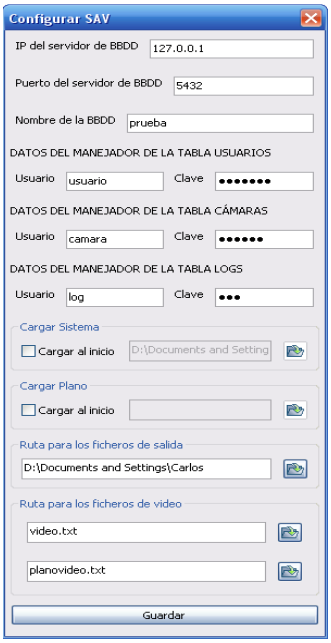

Figura 3. 29 Diálogo Configurar

Ahora seguiremos analizando el diálogo a presentar por la operación *Ver Log*, donde podemos indicar las fechas entre las que queremos obtener los registros del sistema e incluso se nos da la posibilidad de seleccionar un usuario o una cámara en concreto, como se muestra en la Figura 3.30.

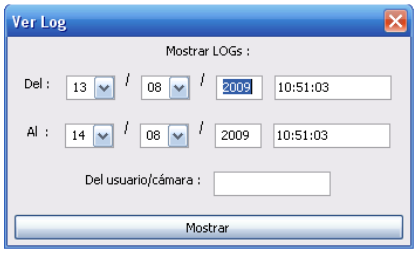

Figura 3. 30 Diálogo Ver Log

Los mensajes de log que produce la operación serán mostrados en la *Ventana de Log* del sistema. Otra de las ventanas que analizaremos es la generada por la operación de *Gestionar BBDD,* que se muestra en la Figura 3.31, con la que se administrarán las tablas de la base de datos del sistema, estas operaciones será accesible siempre y cuando se tenga privilegios para administrarlas dentro del S.A.V..

Figura 3. 31 Diálogo Gestionar BBDD

Dentro del diálogo de gestión de BBDD analizaremos sus botones empezando por los correspondientes para la gestión de la tabla cámaras. La primera operación que podemos observar, es la de *Añadir* una cámara a la base de datos y su diálogo se muestra en la Figura 3.32.

Figura 3. 32 Diálogo Añadir Cámara

El siguiente diálogo, Figura 3.33, que se muestra a continuación es el de *Editar* los parámetros de una cámara, el cual tiene la siguiente estructura.

Figura 3. 33 Diálogo Editar Cámara

Para terminar con este bloque de diálogos, nos queda el perteneciente a *Eliminar* una cámara de la base de datos que se corresponde con la Figura 3.34.

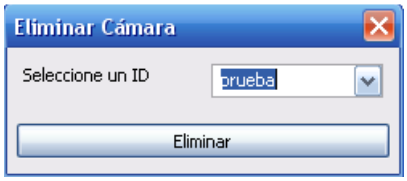

Figura 3. 34 Eliminar Cámara

Una vez terminado con el grupo de gestión de la tabla cámaras se pasará a detallar el correspondiente a la tabla Usuarios, donde encontramos la operación *Añadir* usuario, Figura 3.35, y *Eliminar* usuario, Figura 3.36.

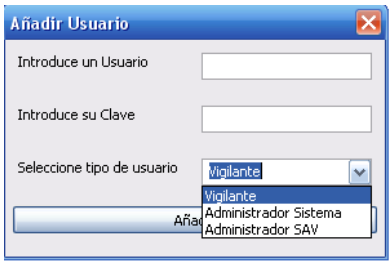

Figura 3. 35 Diálogo Añadir Usuario

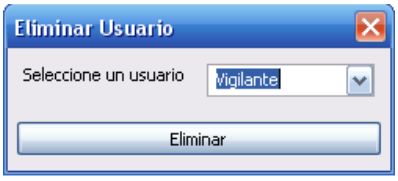

Figura 3. 36 Diálogo Eliminar Usuario

La operación para borrar la tabla Log no lanza ningún diálogo. Para terminar con el diálogo de *Gestionar BBDD* nos queda el botón *Comando SQL* cuyo formato es el que podemos observar en la Figura 3.37.

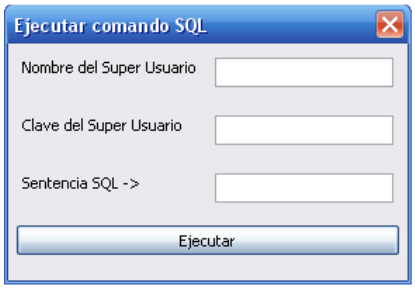

Figura 3. 37 Diálogo Ejecutar comando SQL

Para terminar con este grupo tenemos el botón *Gestionar Plano*, Figura 3.38, que lanzará el siguiente diálogo.

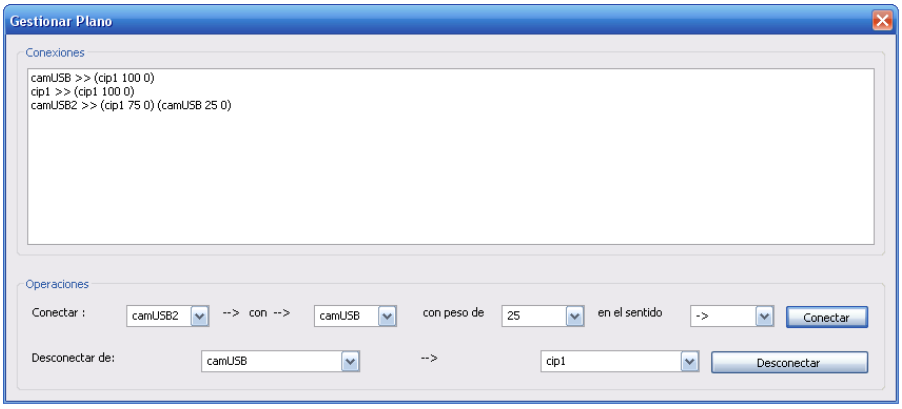

Figura 3. 38 Diálogo Gestionar Plano

Ventana del vigilante

Ahora analizaremos la pestaña *Monitorizar,* Figura 3.39*,* destinada para un usuario sin grandes conocimientos del software. En ella se podrá monitorizar un sistema de cámaras utilizando cualquiera de las configuraciones posibles.

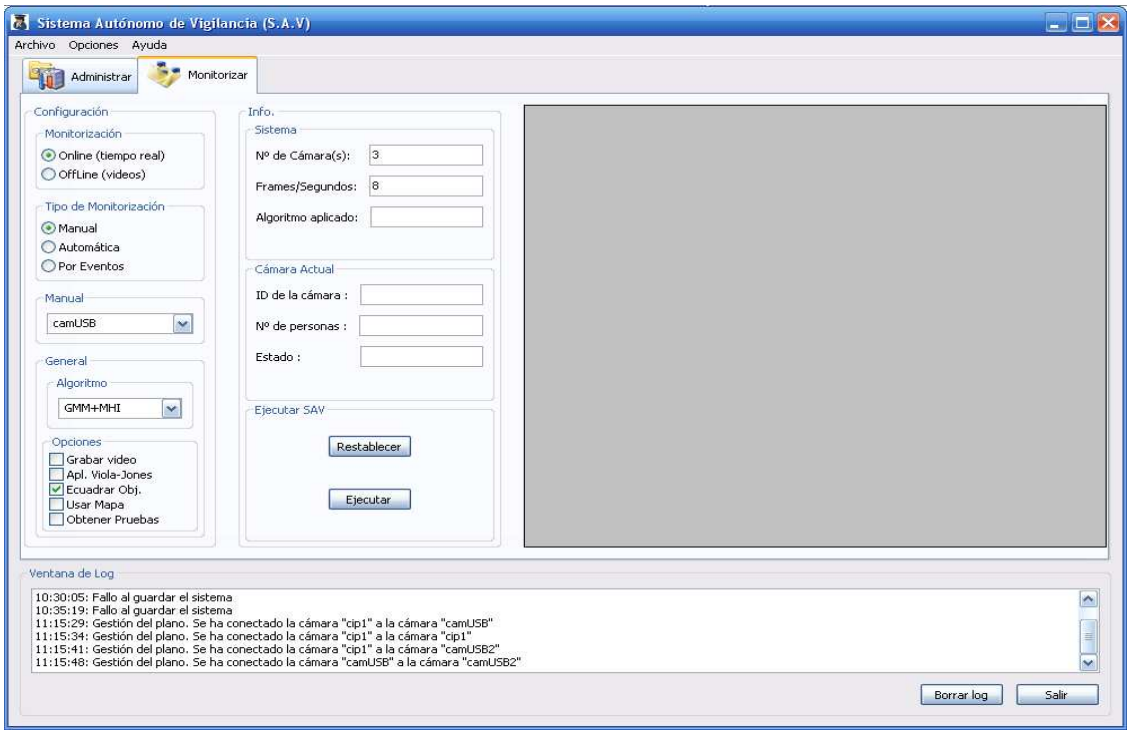

Figura 3. 39 Pestaña Monitorizar

En esta pestaña no existen diálogos emergentes y todos los controles y opciones de monitorización son accesibles desde ésta. Por lo que su diseño sencillo hace que el rasgo a destacar sea su amigabilidad.

Ventana de Log

La *Ventana de Log*, Figura 3.40, es el canal de comunicación del software con el usuario, por lo que siempre estará visible para el usuario en la parte inferior de la ventana de la aplicación.

Figura 3. 40 Ventana de Log

4- Fase de diseño

En este capítulo se describen los distintos flujos del programa así como las estructuras de este. En él podremos ver el diseño arquitectónico de la aplicación, los diagramas de secuencia de las operaciones del software, la descripción de sus usuarios y el diseño de las bases de datos.

4.1- Diseño arquitectónico

El diseño arquitectónico es una de las fases más importantes dentro del desarrollo de un proyecto porque es en este donde se establecen los cimientos que se utilizarán para implementar el software, tomándose como referencia las especificaciones que nos han llegado de la etapa de análisis.

4.1.1- Relaciones de las clases del software

En la Tabla 4.1 se muestra la relación existente entre las entidades que componen el software. Dada la metodología de desarrollo usada, la interfaz y los diálogos son las entidades que más clases poseen, esto es debido a que estas entidades se encargan de la interacción con los usuarios en las tareas de la gestión del software y los componentes hardware.

	INTERFAZ	DIÁLOGOS	HILO	CAMVIEW	GESTCAM	CÁMARA	PLANO	CAPTURADOR	GBBDD
INTERFAZ		X	X	X	X		X	X	X
DIÁLOGOS					X		X		X
HILO				X				X	
CAMVIEW									
GESTCAM	x					X			
CÁMARA									
PLANO	x								
CAPTURADOR					X	X	X		X
GBBDD									

Tabla 4. 1 Relación de las entidades

Las casillas que contienen una "*x*" indican que existe interacción directa entre las clases ya sea por inclusiones o cooperaciones a la hora de compartir datos.

4.1.2- Estructura y descripción de las clases

En este punto se debería mostrar la estructura y funcionalidad de las clases del software. Debido a la cantidad de clases que componen éste, se remite al usuario al manual de programación construido con *DoxyGen* incluido en el DVD entregado con el proyecto, en el que podremos encontrar una descripción de todos los componentes de las clases. Por este motivo, en este punto sólo se hará una descripción teórica de las clases que gestionan los componentes del software cuya implementación se ha intentado desacoplar totalmente de la interfaz para posibilitar su reusabilidad.

Los componentes a destacar son los siguientes:

- *Cámara*: Esta clase es la representación software de las cámaras, con ella se realiza la descripción y modo de acceso de las cámaras.
- *Gestor de Cámaras*: En esta clase se gestiona el sistema de cámaras declaradas en el software, permitiendo configurar los parámetros que atañen a este.
- *Gestor de BBDD*: Su función es realizar las operaciones de gestión y adquisición de datos desde las bases de datos.
- *Plano*: Es la clase encargada del diseño y manejo del plano.
- *Capturador*: Se encarga del manejo de la monitorización del sistema y los procesos para el análisis de las imágenes.
- *Gestor de Pruebas*: Es la clase encargada de gestionar las imágenes obtenidas por las cámaras y en caso de captarse una intrusión en alguna de las zonas vídeo vigiladas guardar una secuencia de hasta diez imágenes de ésta.

Para encontrar los detalles de implementación de estas clases y las demás que componen el software, una vez más se remite al lector al manual de programación. Las otras clases no se han comentado ya que están centradas en la creación y manejo de la interfaz del software y sus eventos.

Dentro del DVD también se dispone del proyecto que contiene la implementación del software en el que podemos encontrar los distintos ficheros que se clasifican según la metodología de desarrollo seguida en:

- Modelo: son todos aquellos archivos que empiezan por I de interfaz
- Vista: son los ficheros con el acrónimo página ya que el diseño es tipo libro.
- Controlador: caracterizadas por empezar por una G de gestor.
- Diálogos: son todos aquellos ficheros que empiezan con una D.

4.1.3- Diagrama de contexto

En la Figura 4.1 se muestra el diagrama de contexto arquitectónico de S.A.V. en él se muestra una vista muy abstracta de la interacción de componentes con el software.

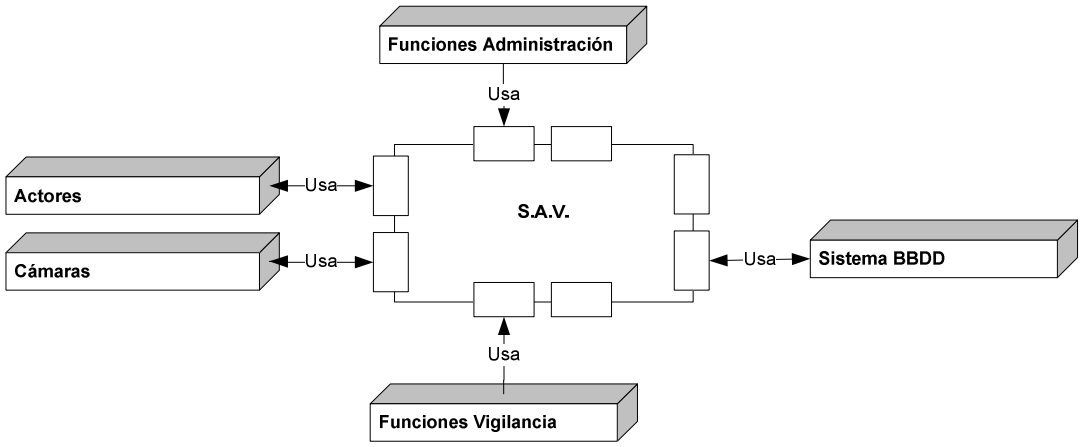

Figura 4. 1 Diagrama arquitectónico

4.1.4- Estructura general de la arquitectura de S.A.V.

En este punto se definen el conjunto de componentes de nivel superior que atienden a la siguiente funcionalidad:

- Administración de comunicaciones externa: coordina la comunicación de la función de seguridad con entidades externa.
- Procesamiento del panel de control: maneja toda la funcionalidad de S.A.V..
- Manejo de detector: coordina el acceso a todos los detectores conectados al sistema.
- Procesamiento de alarmas: verifica todas las condiciones de la alarma y actúa sobre ellas.

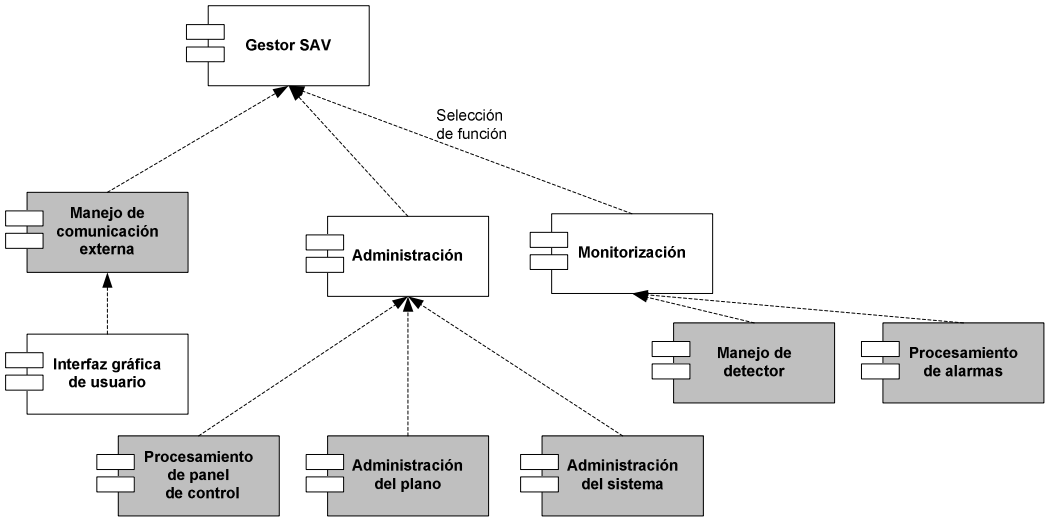

Figura 4. 2 Estructura arquitectónica del software

En la Figura 4.2 se muestra la estructura arquitectónica general (diagrama de componente en UML), donde el componente *Manejo* de la comunicación externa adquiere las transacciones provenientes de los componentes que procesan la interfaz gráfica de usuario de S.A.V.. El componente *Gestor de S.A.V.* maneja esta información y selecciona la función apropiada en cada caso. El procesamiento de *Panel de control* interactúa con el usuario para configurar S.A.V. mientras que el componente *Manejo de alarma* produce una salida cuando se detecta la alarma.

4.1.5- Correlación de transacciones

En este punto se analizará el modelo de análisis global de S.A.V. sin llegar a pasar por cada una de las etapas, ya que esto requiere un gran nivel de profundidad. En la Figura 4.3 se describe el flujo de datos con el fin de correlacionar estos diagramas en una arquitectura (DFD de nivel 1).

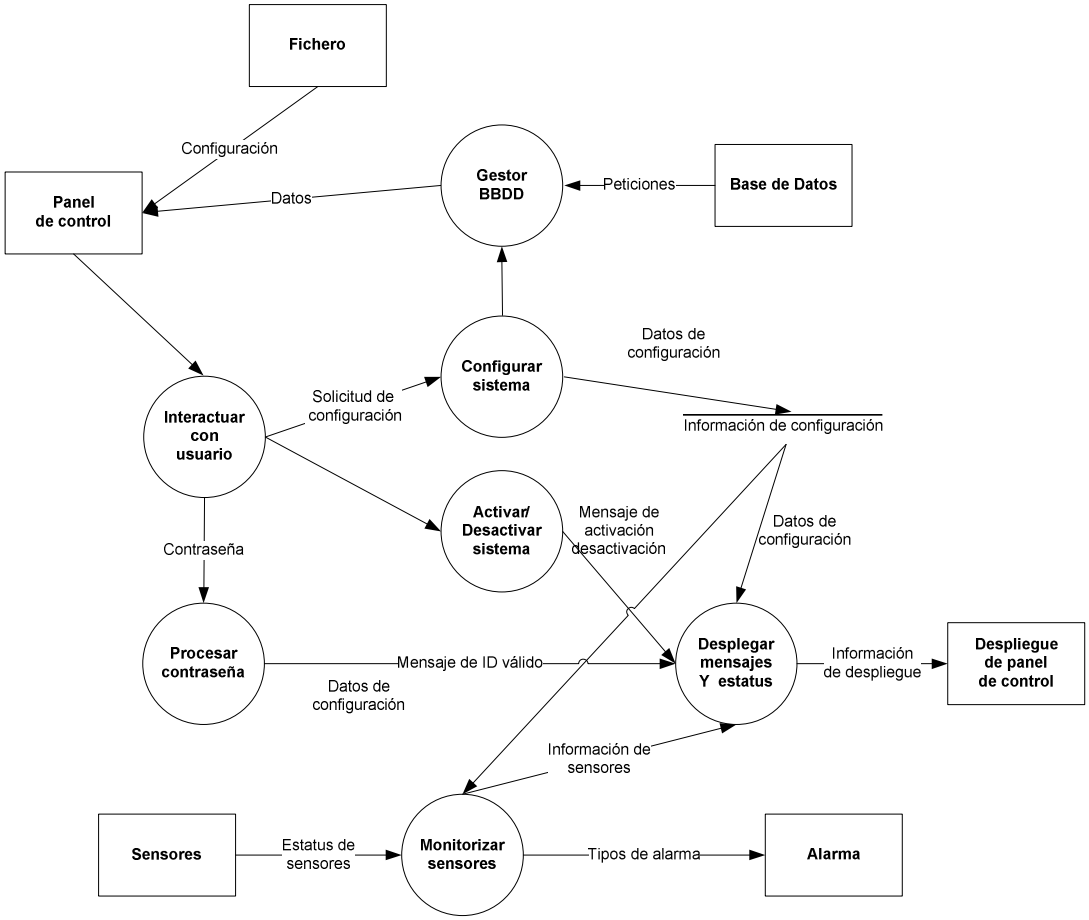

Figura 4. 3 DFD de nivel 1

4.1.6- Diagrama de actividades UML

En el siguiente punto se presentarán los diagramas de actividades UML de las principales operaciones de S.A.V., discriminando entre las operaciones según el rol del usuario que las ejecuta y dentro de ella si se trata de una tarea hacia el sistema, propia del S.A.V. o de vigilancia.

Diagrama de actividad para las operaciones de registro y salida del sistema.

La primera actividad que se comentará es la de registrarse en el sistema, actividad previa para todas las tareas de administración y la acción de salir del registro, Figura 4.4.

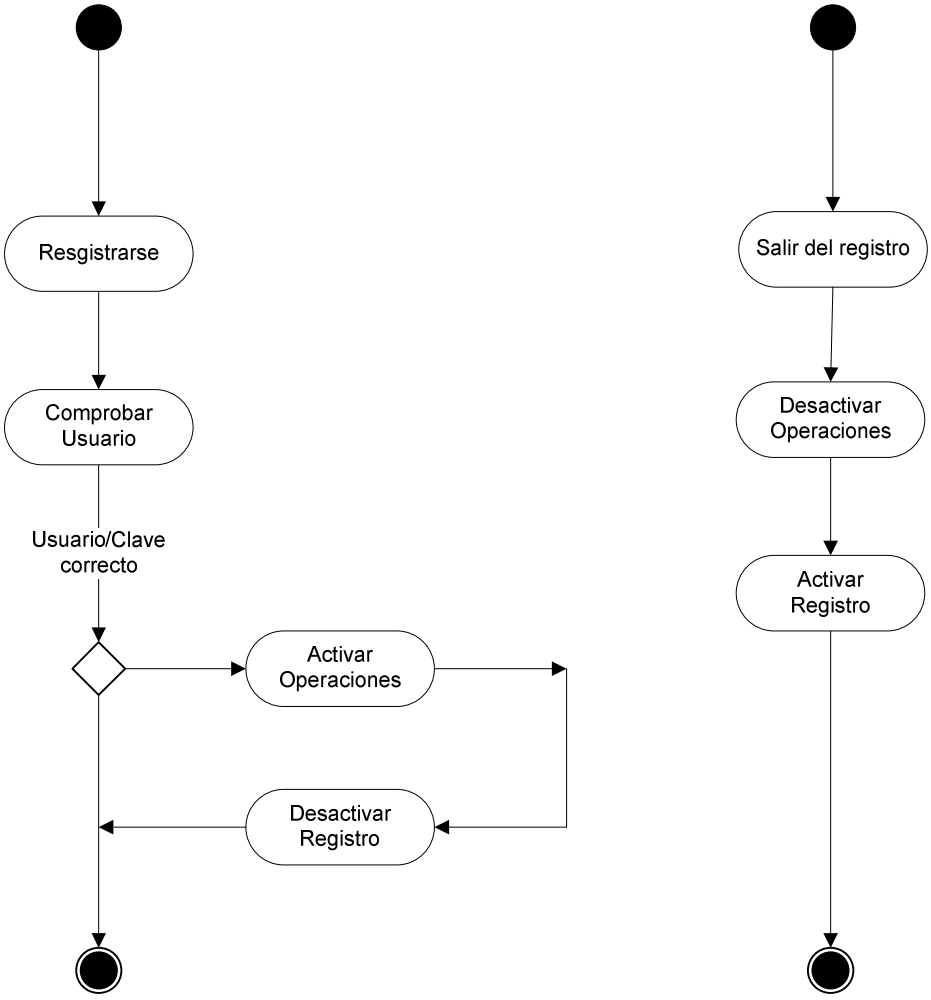

Figura 4. 4 Diagrama UML de la actividad de entrada y salida del registro

Diagrama de actividad para las operaciones sobre el sistema.

En la Figura 4.5 se muestran las actividades que se pueden realizar sobre el sistema. Estas actividades están orientadas a que las realice un administrador.

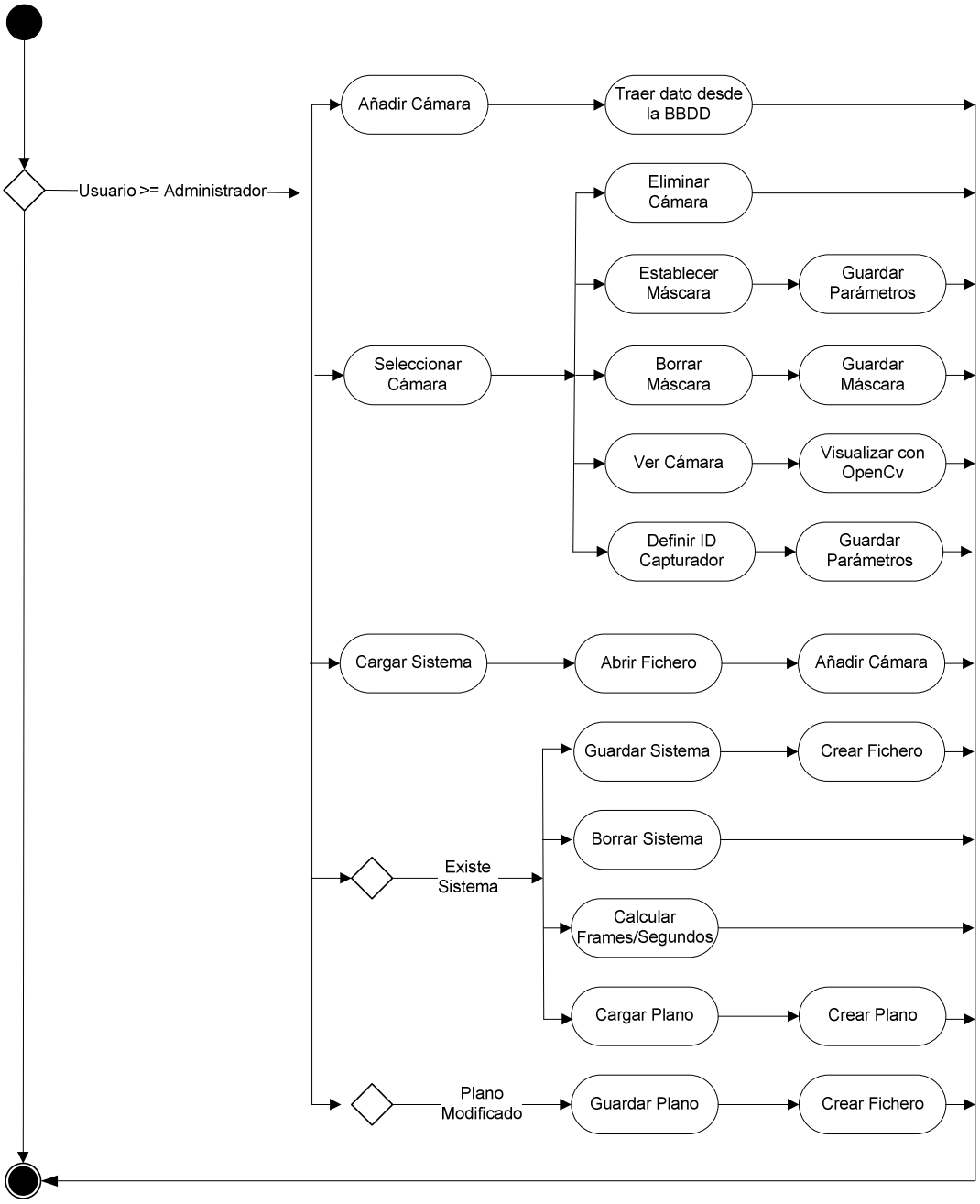

Figura 4. 5 Diagrama UML para las actividades del sistema

Diagrama de actividad para las operaciones sobre S.A.V.

El siguiente diagrama, Figura 4.6, se muestra el flujo de las actividades que se pueden realizar sobre S.A.V.. Dichas actividades son realizadas por un administrador y en caso de algunas actividades sobre la base se necesita tener privilegios de administrador S.A.V..

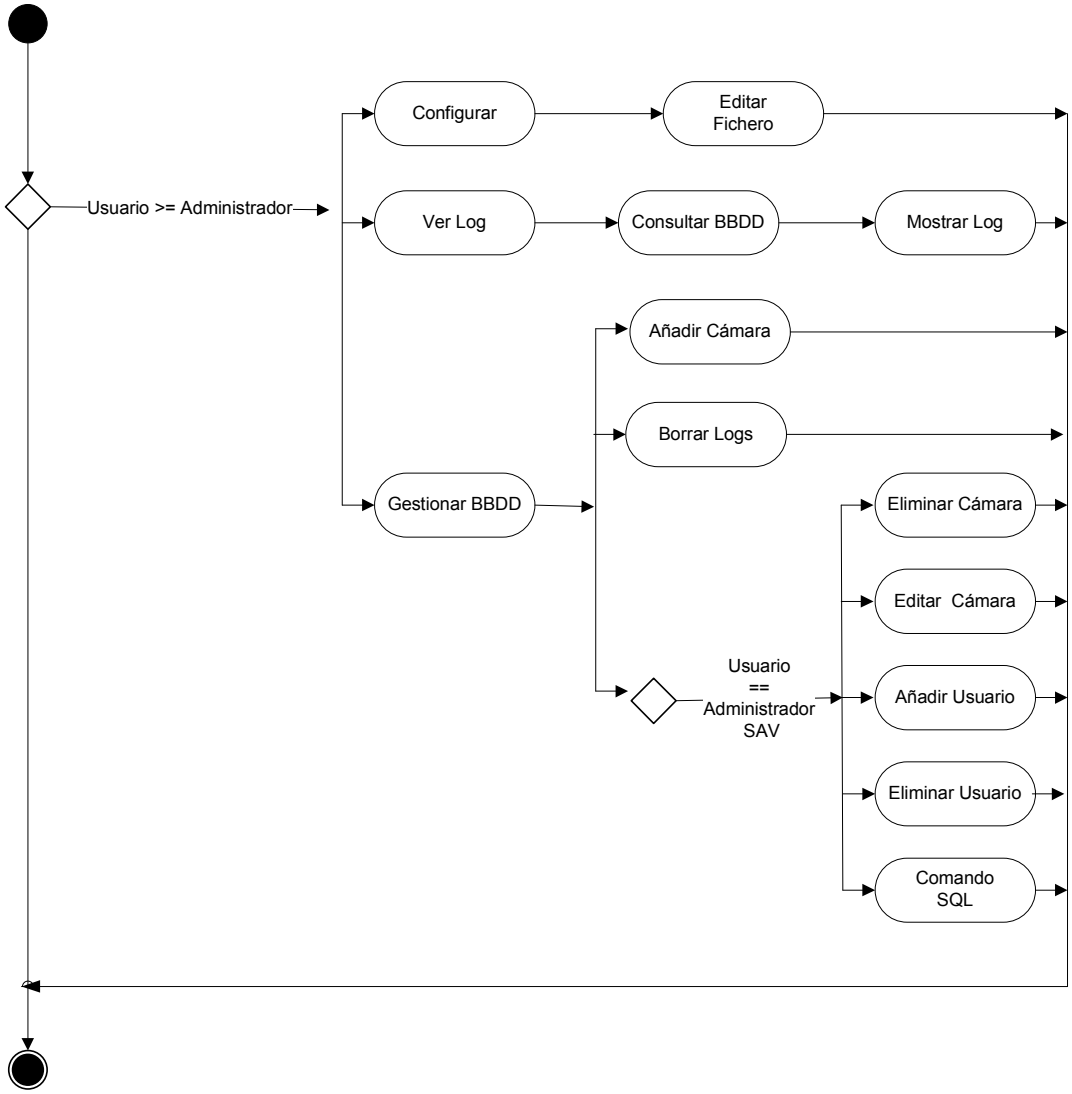

Figura 4. 6 Diagrama UML para las actividades S.A.V.

Diagrama de actividad de Monitorización.

Este diagrama está orientado al usuario con rol de vigilante y muestra el funcionamiento de las actividades de vigilancia. Figura 4.7.

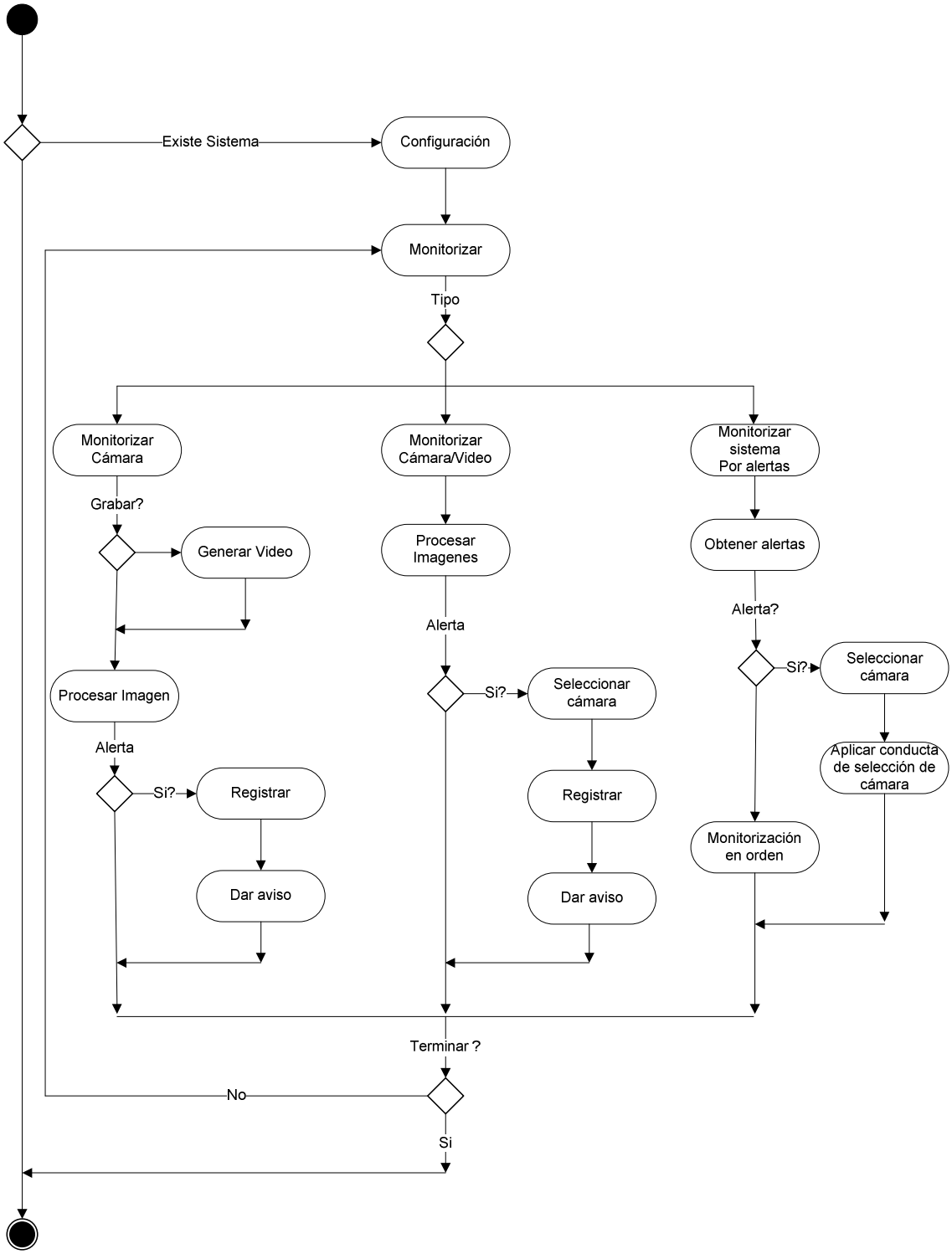

Figura 4. 7 Diagramas UML para las actividades de Monitorización

4.2- Diseño de ficheros

En este punto se muestra el diseño y estructura de los ficheros que manejará S.A.V..

Fichero de Configuración

El fichero de configuración es usado por la aplicación a la hora de iniciarse, ya que este fichero contiene los valores de las variables del sistema. Su funcionalidad es la de evitar al usuario estar configurando el software cada vez que este se inicie.

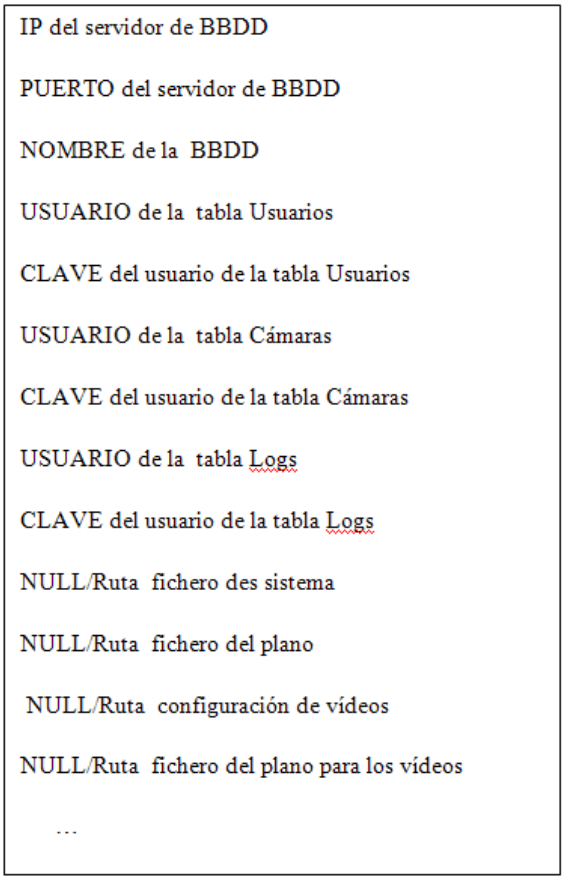

Figura 4. 8 Estructura del Fichero de configuración

Fichero de Sistema

En el fichero de sistema indicaremos los identificadores de las cámaras a cargar desde la base de datos por el software.

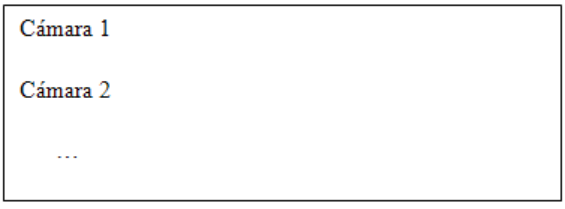

Figura 4. 9 Estructura del Fichero de Sistema

Fichero de Plano

El fichero de plano contiene las relaciones entre las cámaras. Gracias a él S.A.V. es capaz de conocer la conectividad entre las cámaras y formular el plano del sistema.

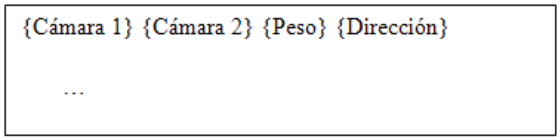

Figura 4. 10 Estructura del Fichero de Plano

Fichero de vídeos

Este fichero contiene las rutas y los nombres de los vídeos a cargar así como la prioridad que poseen las cámaras simuladas.

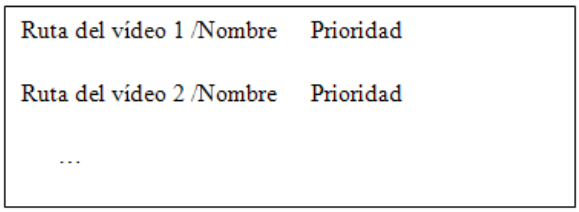

Figura 4. 11 Estructura del Fichero de Sistema para vídeos

Fichero de Plano

Este fichero posee la misma estructura y utilidad que el fichero de Plano anterior, salvo que en lugar de usar los identificadores de la cámara en las conexiones se usan las rutas de los vídeos.

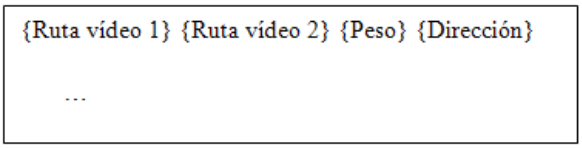

Figura 4. 12 Estructura del Fichero de Plano para vídeos

4.3- Diseño de usuarios

En este punto se muestran el diseño de los tipos de usuarios que existirán tanto en el software como en la base de datos PostgreSQL.

En el caso de los usuarios de S.A.V. la diferenciación entre usuarios permite controlar la accesibilidad a las zonas de administración del software. Mientras que en el caso de los usuarios PostgreSQL son los que deben existir actualmente para poder comunicarse con las tablas desde el software.

Usuarios S.A.V.

TIPO	NIVEL DE PRIVILEGIOS	FUNCIONALIDADES
Vigilante	0	Acceso al modulo de vigilancia y administración del sistema
Administrador SAV	1	Posee los privilegios del Vigilante además de acceso limitado a la administración de BBDD.
Administrador	2	Acceso total en el software.

Tabla 4. 2 Descripción de usuarios S.A.V.

Usuarios PostgreSQL

USUARIO	CLAVE	PRIVILEGIOS
Postgres	carlos	Súper Usuario (Interfaz gráfica)
Admin	admin	Súper Usuario
Usuario	Usuario	Dueño de la tabla Usuarios
Camara	Camara	Dueño de la tabla Cámaras
Log	Log	Dueño de la tabla Log

Tabla 4. 3 Descripción de usuarios PostgreSQL

4.4- Diseño de bases de datos

En este punto se detallan los enclaves del diseño de las bases de datos con los que trabajará el software. Indicando las estructuras de las bases de datos, detallando los nombres de los campos así como su tipo y su tipo de nulidad.

Descripción de la tabla Usuarios

CAMPO	TIPO
Nombre	Text (NOT NULL; PRIMARY KEY)
Clave	Text(NOT NULL)
Privilegios	Integer(NOT NULL)

Tabla 4. 4 Descripción de la tabla Usuarios

Descripción de la tabla Camaras

CAMPO	TIPO
ID	Text(NOT NULL; PRIMARY KEY)
Tipo	Integer (NOT NULL)
Colocacion	Integer(NOT NULL)
Edificio	Text(NOT NULL)
Planta	Integer(NOT NULL)
Url	Text(NOT NULL)
User	Text(NOT NULL)
Pass	Text(NOT NULL)
Calibrada	Integer(NOT NULL)
Prioridad	Integer(NOT NULL)
Mascara	Text(NOT NULL)
Eliminar Fecha	Integer(NOT NULL)
Pintrinsecos	Text(NOT NULL)
Cdistorsion	Text(NOT NULL)
Errorrepro	Text(NOT NULL)
Pextrinsecos	Text(NOT NULL)

Tabla 4. 5 Descripción de la tabla Camaras

Descripción de la tabla Log

CAMPO	TIPO
ID	Serial(NOT NULL; PRIMARY KEY)
Fecha	Timestamp without time zone(NOT NULL)
Alerta	Text(NOT NULL)
Actor	Text(NOT NULL)

Tabla 4. 6 Descripción de la tabla Log

4.5- Diagramas de secuencia

En el siguiente punto se muestran los flujos de las operaciones e interacciones entre componentes del software y los usuarios. La mayoría de las operaciones serán iniciadas por una interacción del usuario con el Sistema Autónomo de Vigilancia que intentará responder a las acciones de manera rápida y eficiente, evitando las intervenciones del usuario en las tareas de confirmación. Se ha elegido una representación horizontal por facilitar el entendimiento de los diagramas.

Registrarse en S.A.V

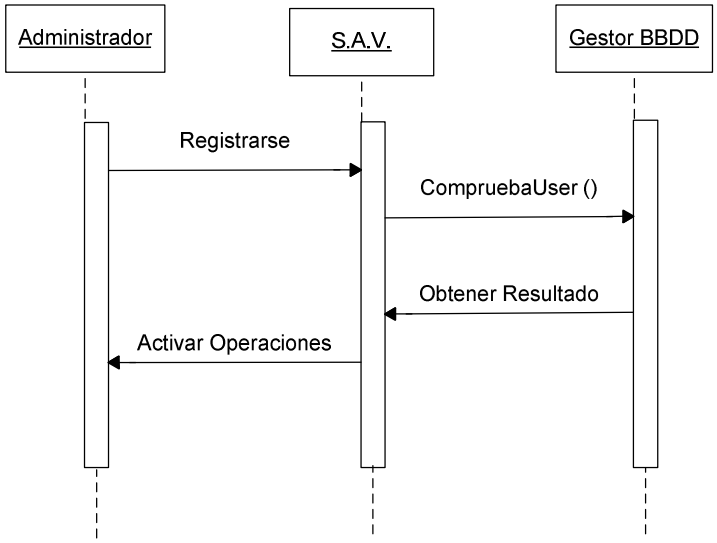

Figura 4. 13 Diagrama de secuencia para el registro

Salir del SAV

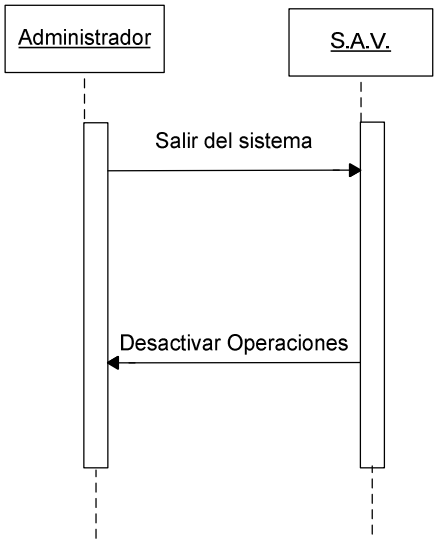

Figura 4. 14 Diagrama de secuencia para abandonar el registro

Añadir cámara

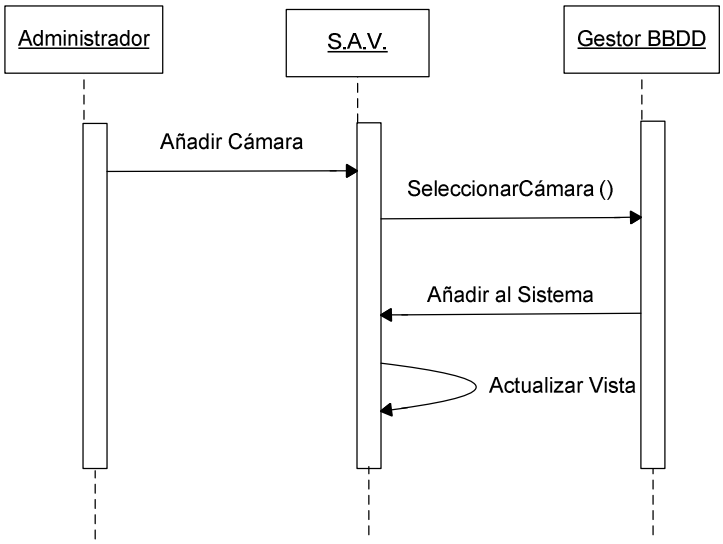

Figura 4. 15 Diagrama de secuencia para añadir una cámara

Eliminar cámara

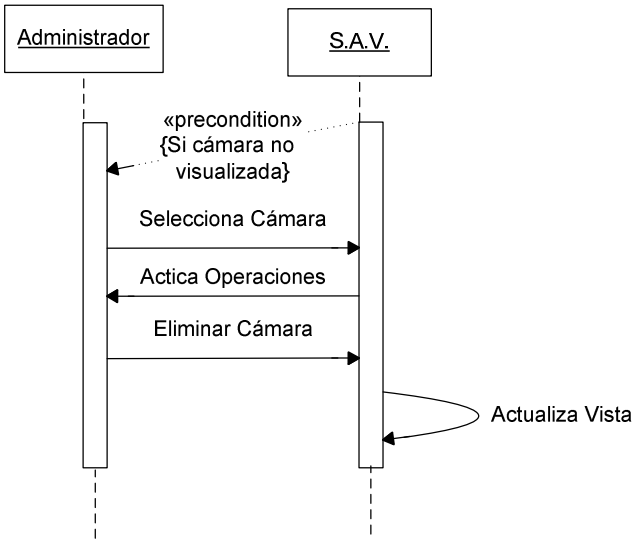

Figura 4. 16 Diagrama de secuencia para eliminar una cámara

Ver cámara

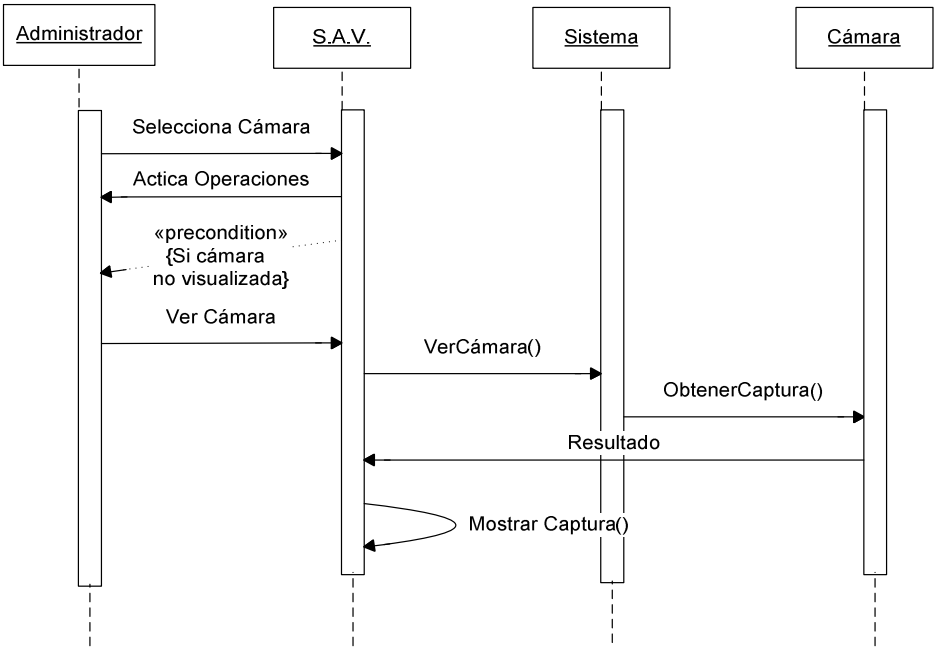

Figura 4. 17 Diagrama de secuencia para monitorizar una cámara

Definir Máscara

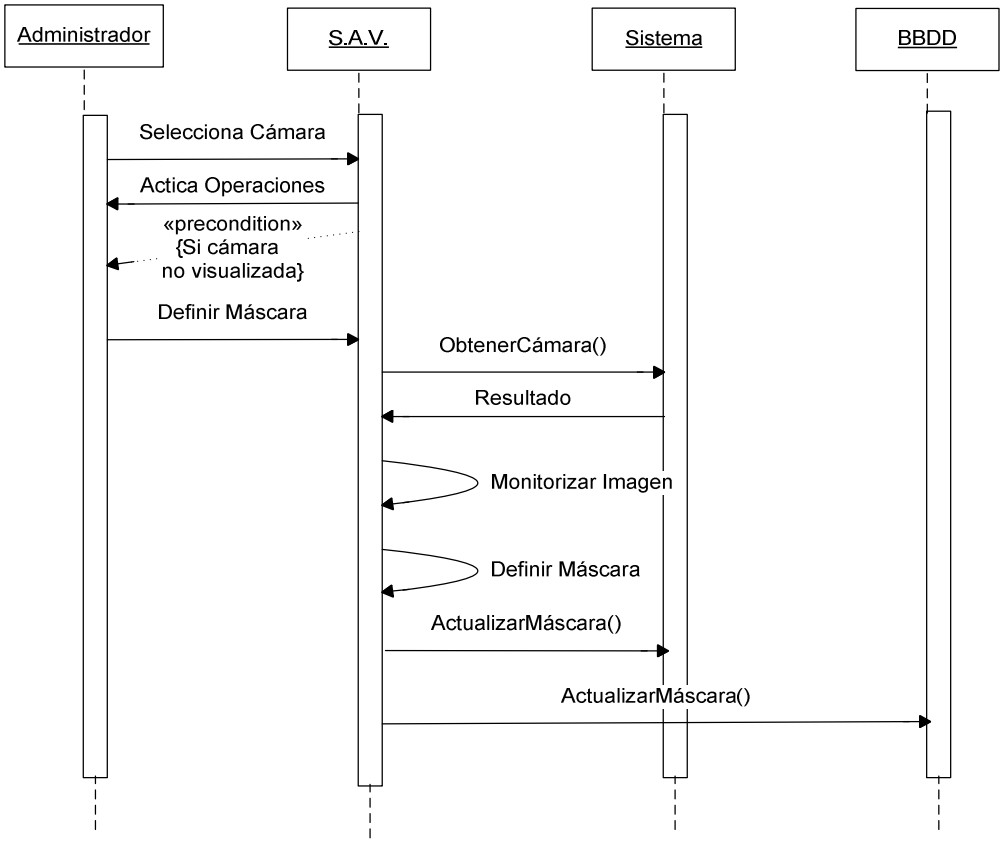

Figura 4. 18 Diagrama de secuencia para definir un área de interés

Borrar Máscara

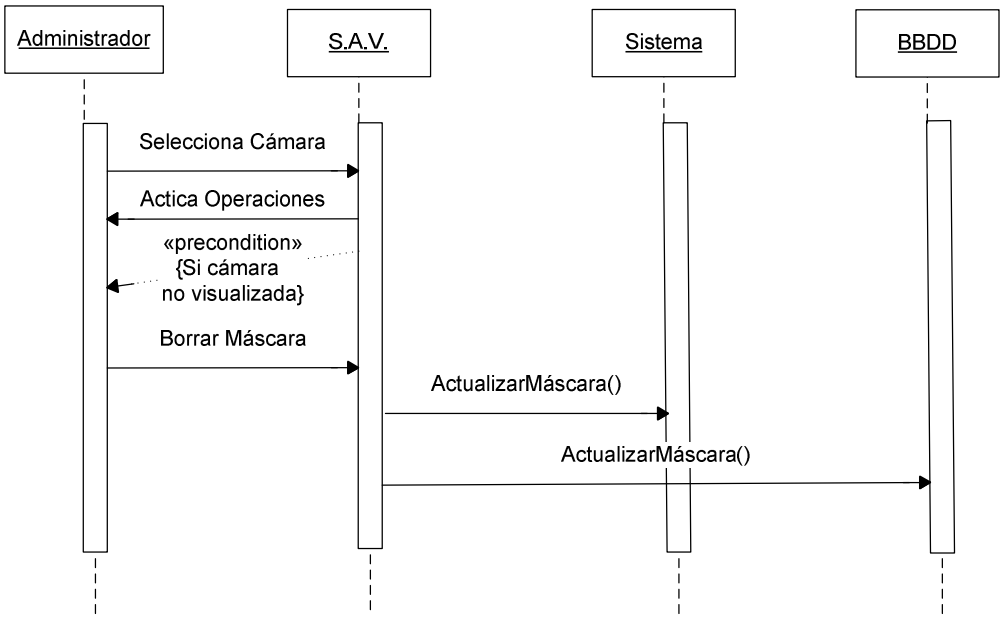

Figura 4. 19 Diagrama de secuencia para borrar una máscara

Definir ID del capture

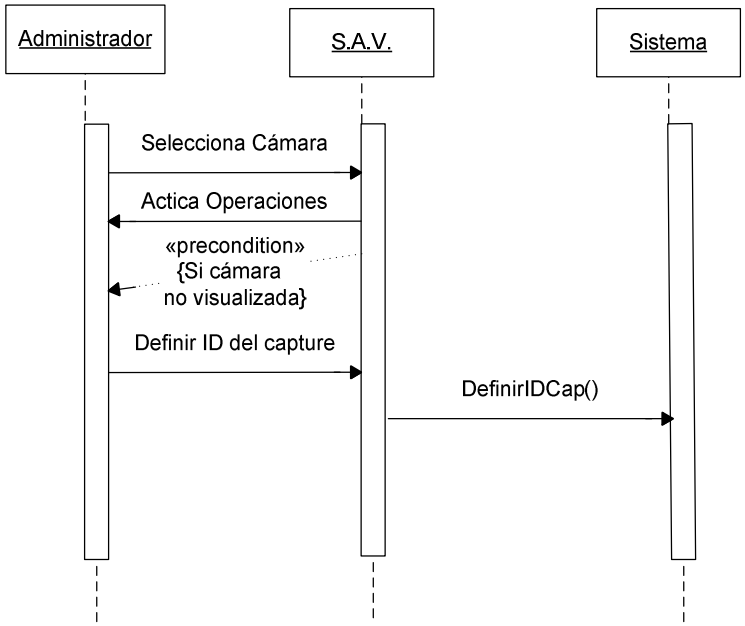

Figura 4. 20 Diagrama de secuencia para definir un capture

Guardar sistema

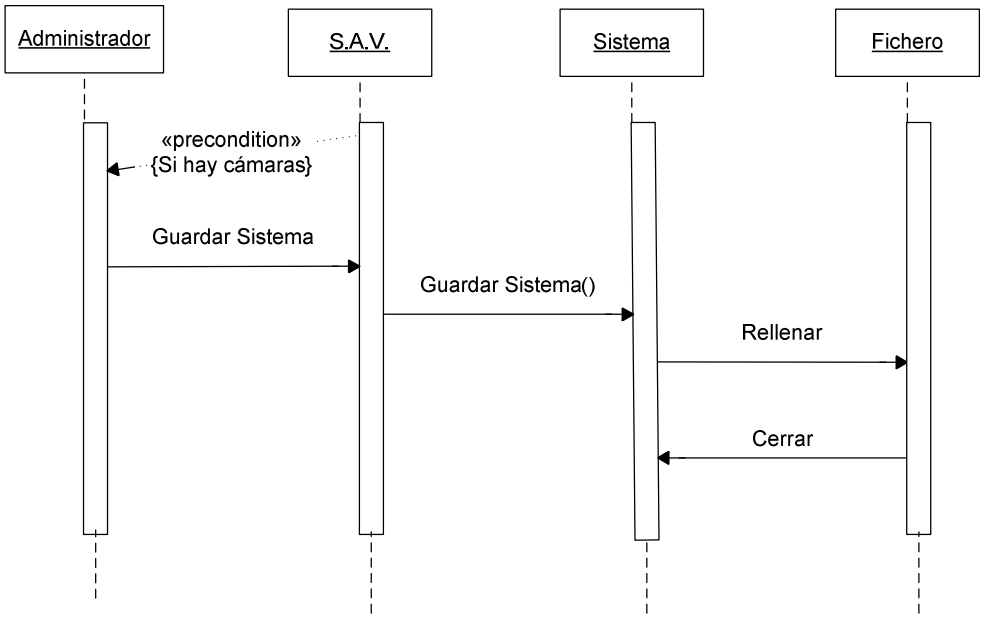

Figura 4. 21 Diagrama de secuencia de la operación guardar sistema

Cargar sistema

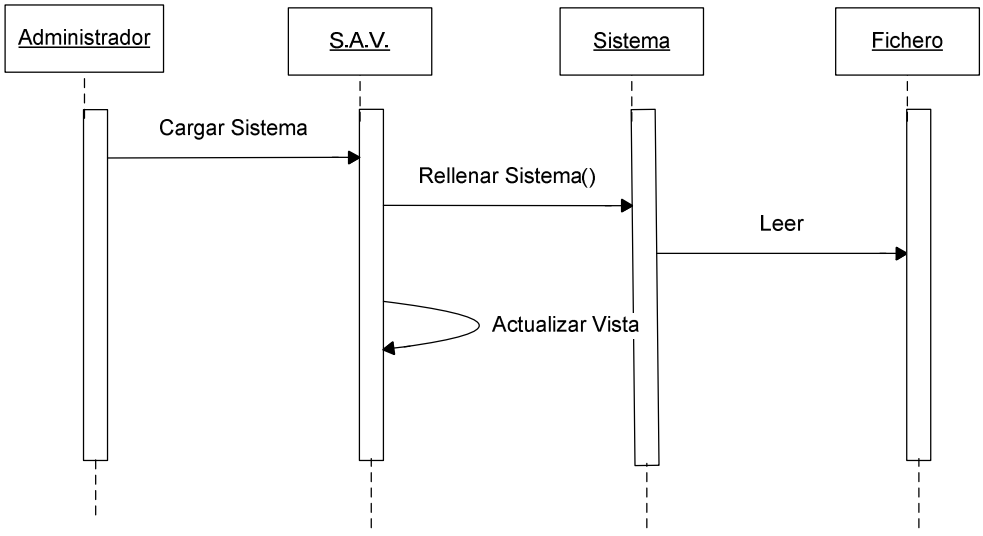

Figura 4. 22 Diagrama de secuencia de la operación cargar sistema

Borrar sistema

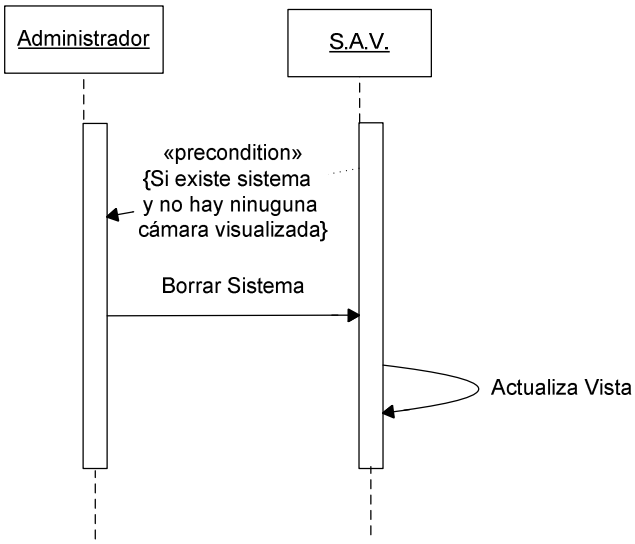

Figura 4. 23 Diagrama de secuencia de la operación borrar sistema

Guarda plano

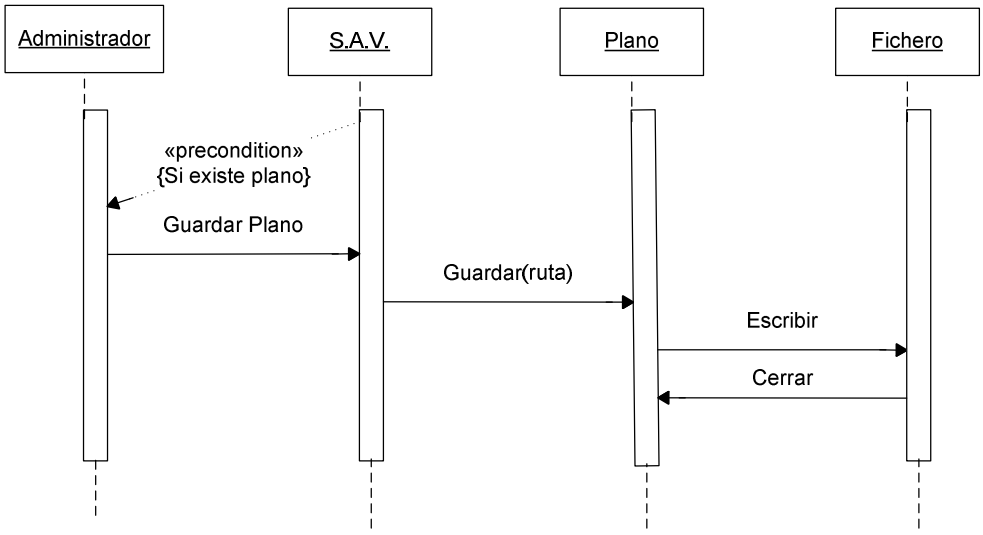

Figura 4. 24 Diagrama de secuencia de la operación guardar plano

Cargar plano

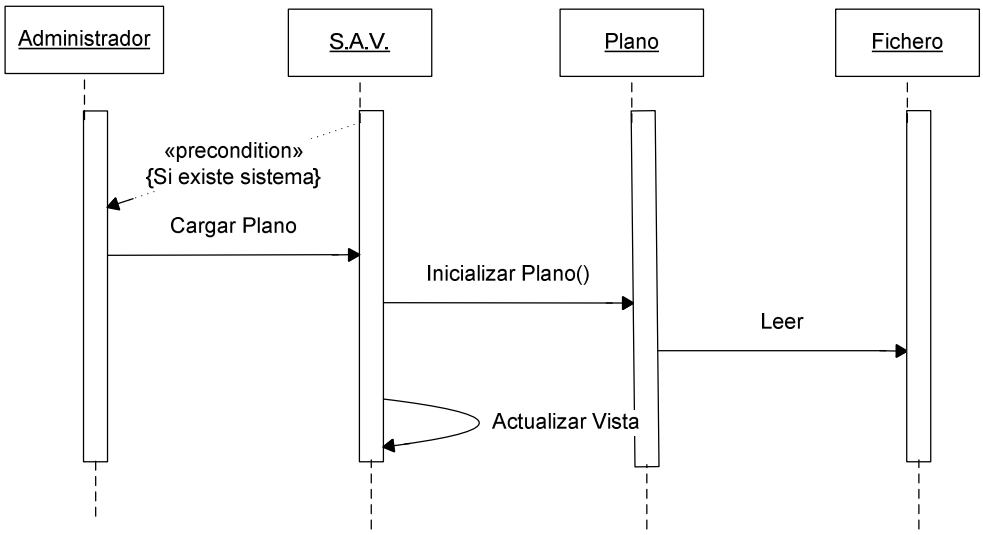

Figura 4. 25 Diagrama de secuencia de la operación cargar plano

Calcular Frames/Segundos

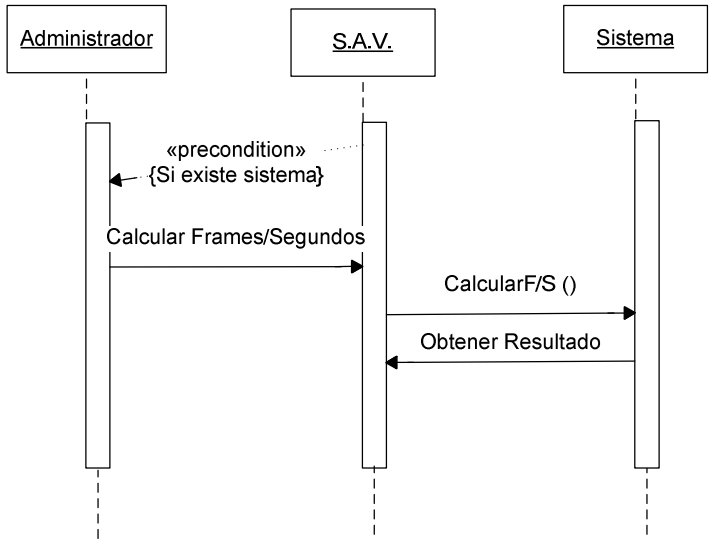

Figura 4. 26 Diagrama de secuencia de la operación para calcular los cps

Configurar S.A.V.

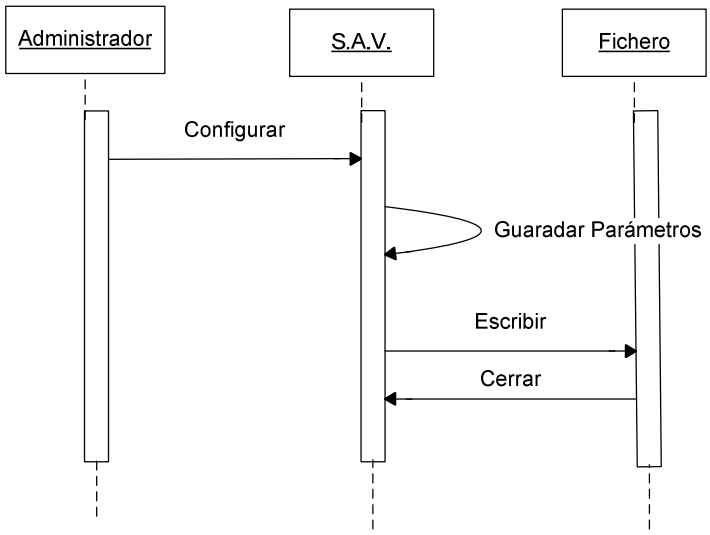

Figura 4. 27 Diagrama de secuencia de la operación para configurar el software

Ver log

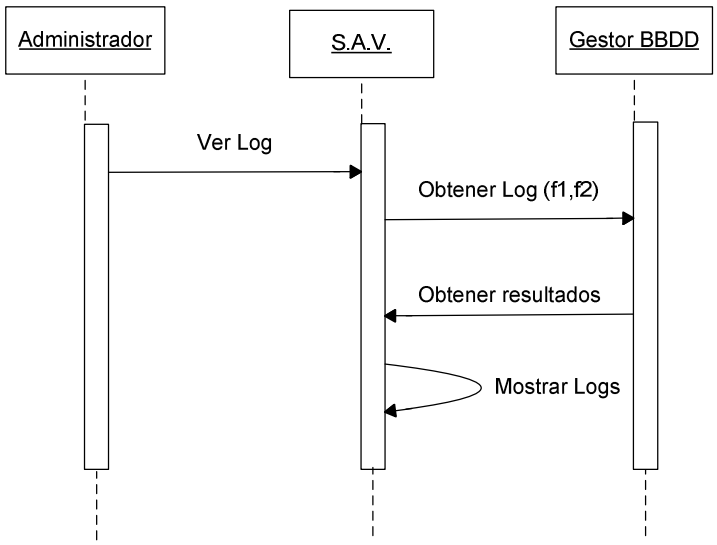

Figura 4. 28 Diagrama de secuencia de la operación ver log

Gestor de BBDD

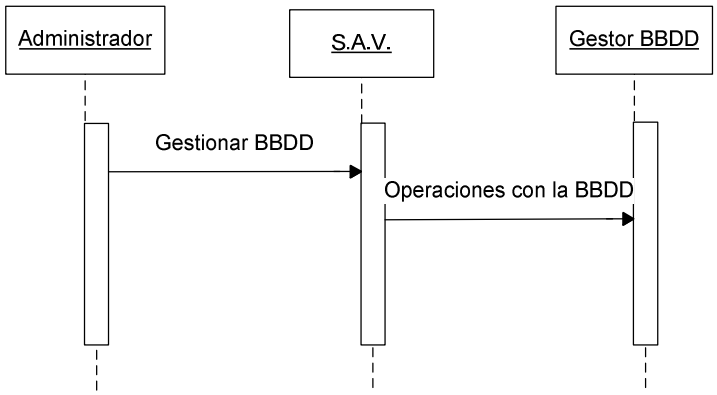

Figura 4. 29 Diagrama de secuencia de la operación para la gestión de la BBDD

Gestor de plano

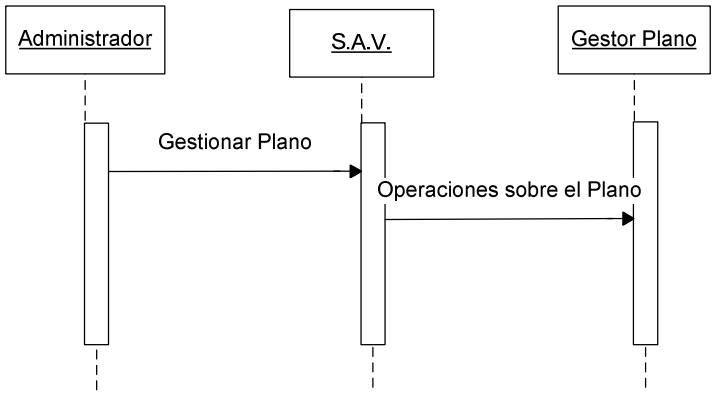

Figura 4. 30 Diagrama de secuencia de la operación para gestionar el plano

Monitorizar

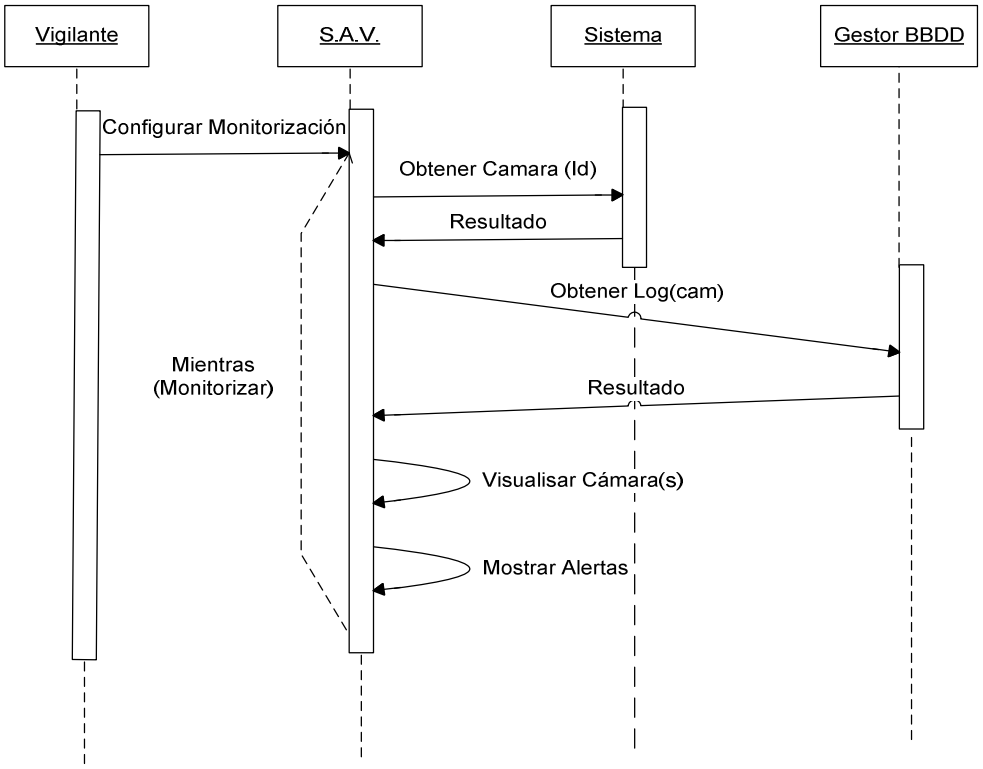

Figura 4. 31 Diagrama de secuencia de la función monitorizar

Inicializar S.A.V.

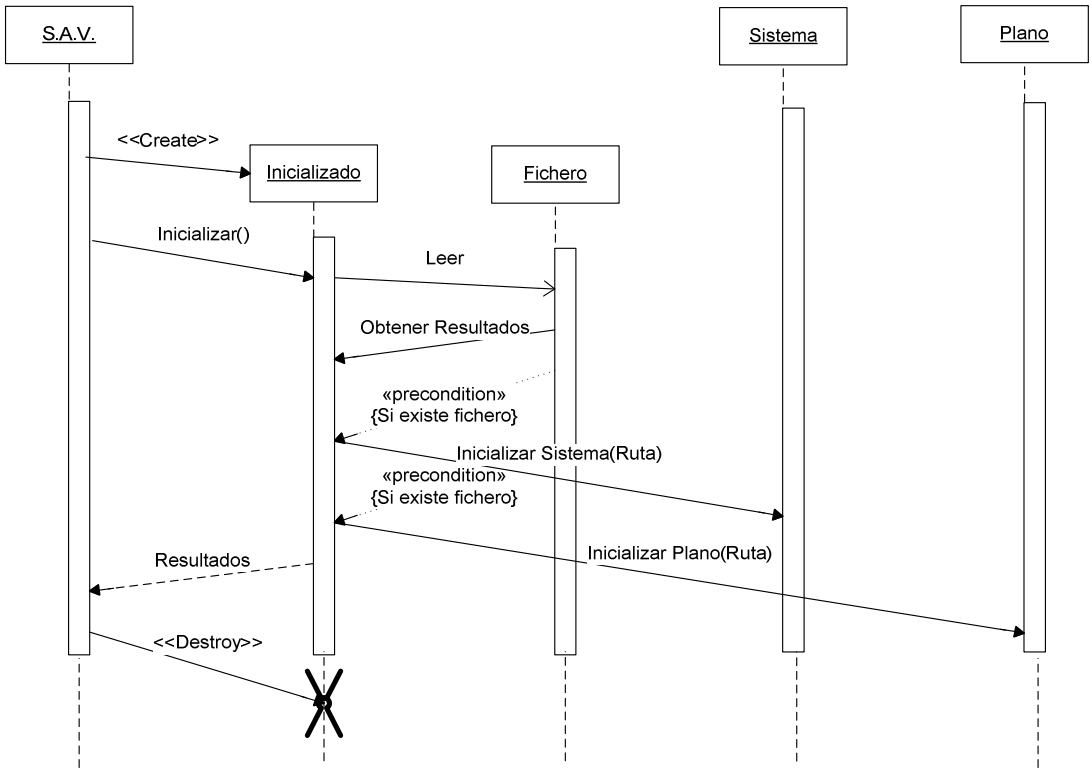

Figura 4. 32 Función para inicializar el software

5- Solución adoptada

A lo largo de esta sección se abordarán y documentarán las soluciones adoptadas para alcanzar los objetivos planteados. Debido a la temática del proyecto se abordarán aspectos de adquisición y tratamiento de imágenes centrándose en los siguientes puntos.

5.1- Adquisición de imágenes

La actual tendencia en los sistemas de vídeo vigilancia son las soluciones basadas en tecnología IP, cuya popularidad está creciendo rápidamente. Existe una infinidad de fabricantes en el mercado, que proporcionan gran variedad de cámaras y servidores de vídeo IP preparados para funcionar en sistemas de vídeo vigilancia a nivel profesional usando tecnología IP es por ello que se decidió incluir esta tecnología en el proyecto.

A lo largo de este capítulo se describirá la experiencia obtenida al trabajar con las diferentes cámaras de vídeo, estudiándose la adquisición de imágenes tanto de cámaras accesibles desde la red (IP, Web Servers) como de cámaras conectadas directamente al ordenador donde se ejecuta la aplicación (USB/FireWire).

También se describirán las herramientas estudiadas y las utilizadas para resolver la problemática de la adquisición de imágenes de cualquier tipo de cámara sin importar su marca y/o modelo, mostrándose los resultados del estudio en la aplicación diseñada y desarrollada para el PFC.

Las herramientas estudiadas para resolver esta problemática las podemos encontrar en el DVD adjuntado con la memoria dentro de la carpeta *Software Usado,* en ella podemos encontrar los siguientes programas:

- TVídeoGrabber
- VídeoMan
- Icimagingcontroltrial
- Adquisición JPEG con actualización continua de la fuente
- WMEncoder
- OpenCV

5.1.1- Adquisición de imágenes en cámaras USB/FireWire

Para la adquisición de imágenes de cámaras USB/FireWire, es decir, cámaras que necesitan estar conectadas a un ordenador físicamente para funcionar, existen varias técnicas en el mercado actual. Una opción es usar una librería de tratamiento de imágenes que permiten la comunicación con las cámaras, o bien usar algún software

para transformar la adquisición de las cámaras a vía IP. En el estudio del proceso de adquisición de imágenes del proyecto se decidió explorar ambas vías.

En la correspondiente al uso de la librería se decantó inmediatamente por el uso de la OpenCV debido a las recomendaciones de los tutores ya que cumplía con los requisitos impuesto en la realización del proyecto, software multiplataforma y gratuito y es una herramienta avalada por sus experiencias. Por lo que pese a existir otras herramientas en el mercado para el tratamiento de imágenes, éstas no se estudiaron.

En cuanto a la segunda alternativa, tras realizar búsquedas de herramientas que permitieran transformar las cámaras USB/FireWire en cámaras IP con la ayuda de un ordenador, se encontró la herramienta WMEncoder que es una aplicación gratuita desarrollada por Microsoft para la captura de vídeos desde varios dispositivos.

Para la importación de imágenes desde el dispositivo de captura, el programa incluye bastantes opciones de configuración que van desde el tipo de dispositivo elegido y el sistema de codificación del vídeo a la elección del formato de audio y *bitrate*. Por último, también ofrece opciones bastantes útiles para la retransmisión de vídeos o de imágenes que se obtengan de la webcam.

Tras el estudio de ambas técnicas se consideró la elección para la adquisición de imágenes desde cámaras USB/FireWire a partir de los siguientes puntos:

- Limitación en la comunicación.
- Software multiplataforma.
- Infraestructura disponible.
- Necesidad de una herramienta para el tratamiento de imágenes.

Del cual si enfrentamos ambas técnicas nos sale como victorioso el uso de la librería OpenCV, ya que es la única técnica que cumple las precondiciones. Además hace que el software desarrollado admita cámaras conectadas directamente al ordenador, es una librería multiplataforma capaz de funcionar con multitud de marcas de cámaras USB/FireWire y satisface la necesidad de tener un software para el tratamiento de imágenes sin requerir el uso de software adicional.

No obstante la opción de usar el WMEncoder como medio de transformación de cámaras USB/FireWire en cámaras IP sigue siendo viable ya que el software realizado también admite el uso de cámaras IP en el conjunto de cámaras que componen el sistema a monitorizar.

5.1.2- Proceso de adquisición de imágenes en cámaras IP

La adquisición de imágenes desde cámaras IP presenta una gran problemática ya que los métodos que ofrecen los distribuidores, APIs o SDKs, pese a ser gratuitos son exclusivos para cada fabricante y en ocasiones incluso para cada modelo. De aquí que la principal desventaja de estos sistemas, es cuando se quiere diseñar su propia aplicación de escritorio y necesita acceder a las cámaras IP. Este problema se debería ir solventando con la llegada de los nuevos dispositivos puesto que a finales de 2008 los principales fabricantes acordaron seguir y utilizar un protocolo universal por lo que cualquier API o SDK debería funcionar con cualquier máquina.

Este hecho no se ha podido constatar al no disponer de cámaras modernas y las anteriores a 2008 presentan arquitecturas de codificación propias para cada tipo de cámaras, pese a todo ello se intentó probar algunas API sin obtener ningún resultado satisfactorio. Por lo tanto esta vía fue desechada al no conseguir llegar a buen puerto y se centraron todos los esfuerzos en la búsqueda de otros medios de adquisición de imágenes que permitieran la máxima compatibilidad entre equipos y dispositivos, objetivo primario de este proyecto.

A continuación se comentarán los pros y contras de las herramientas estudiadas y la elección utilizada en la implementación del proyecto para dicho propósito:

- TVídeoGrabber:

  Es un componente de captura y reproductor de medios, capaz de capturar fuentes de vídeo análogas o digitales, cámaras FireWire, vídeo grabadoras DV, webcams USB, cámaras IP, tarjetas de TV, tarjetas PCI compuestas, dispositivos USB. Además el software posibilita grabación de vídeos, compresión en vuelo, detección de movimiento, entre otras características. Su principal problema es que es un software privativo cuya licencia es excesiva, evitando ceder derechos para trabajos de investigación por lo que al no cumplir el requisito fijado para el proyecto donde todas las herramientas que se usen deben de ser software libre y/o gratuito en el peor caso. No obstante los detalles constatados fueron probados y los resultados obtenidos de la demo fueron satisfactorios.

- VídeoMan:

  Es otra de las APIs universales que existen en el mercado, algo más deficiente que la anterior ya que daba menos prestaciones, pese a ello se consiguieron resultados con la versión de prueba (licencia evaluación). Fue descartada por ser

una herramienta privativa aunque su coste era del orden de diez veces inferior que el de TVídeoGrabber.

- Icimagingcontroltrial:

  Fue otras de las herramientas estudiadas, en este caso no se consiguió obtener resultado de las cámaras y tras comprobar que se trataba de otra herramienta privativa que ofrecía una versión de pruebas de sesenta días fue descartada.

- Adquisición JPEG con actualización continua de la fuente :

  Es el formato de vídeo más simple, que incluyen casi todas las cámaras de vídeo IP / servidores, no debe ser incluido como un formato de vídeo ya que sólo se obtienen una imagen por petición HTTP. La URL para acceder a una cámara debe estar documentada por el fabricante.

  Este enfoque para la adquisición de imágenes tiene sus ventajas y desventajas, la principal desventaja es que está obligado a enviar una petición HTTP al servidor de la cámara Web cada vez que se necesite capturar una imagen, lo que añade algo de pérdida de velocidad a causa de los datos adicionales (cabeceras HTTP) que se envían / reciben. Mientras que la principal ventaja que ofrece esta técnica es que desde la aplicación puedes controlar fácilmente la velocidad de captura.

Debido a que adquirir información con la técnica de *Adquisición JPEG* resulta muy fácil y a las limitaciones que presentaban las otras aplicaciones estudiadas, se optó por el uso de esta técnica para solventar la problemática de la adquisición de imágenes de cámaras IP. Este método tan sólo requiere realizar peticiones HTTP, obtener la respuesta y de ésta extraer la imagen, por lo que se optó por la búsqueda de una herramienta que permitiera realizar peticiones HTTP. Al principio se intentó hacer uso de la librería de comunicación de wxWidgets, pero el tratamiento de la conexión que ofrece es a muy bajo nivel y sin apenas control de errores. Debido a las limitaciones presentadas por la anterior opción, se optó por hacer uso del FrameWork de Visual Studio, puesto que aporta operaciones a nivel HTTP y un completo control de errores. Siendo la única pega de esta herramienta la limitación que añade a la aplicación, ya que ésta no es multiplataforma, quedando como trabajo futuro la búsqueda de una herramienta o librería que cumpla con dichos objetivos bajo otras plataformas.

5.2- Substracción de fondo y detección de objetos

Los métodos de detección de presencia analizados y empleados en el proyecto se ha orientado a la detección de personas. Del estudio y la puesta en práctica de los distintos

algoritmos se han seleccionado los modelos a utilizar para la detección de movimiento en el software. En este estudio sólo se tendrá en cuenta el comportamiento de cada algoritmo en una sola cámara estacionaria, no obstante en el software se adaptaron estos métodos para que pudiesen procesar sistemas de cámaras. Los puntos a evaluar en los distintos modelos han sido la estabilidad del método, comportamiento ante puntos críticos y el coste computacional del método.

Para empezar se establece un punto de partida dentro de los sistemas de vigilancia, donde se presuponen dos aspectos importantes, que se dispone de un buen ancho de banda para la transmisión de imágenes y que las cámaras posean una buena calidad de imagen. Como el marco de desarrollo de este proyecto no ha sido un entorno profesional se ha tenido que adaptar a las condiciones y a los medios que se poseen. Realizándose el sobre esfuerzo de simular condiciones de escenarios con menos calidad que la que se dispondrían en entornos reales, no obstante dichas condiciones deberían de robustecer a los sistemas creados y en ningún caso que éstos se especialicen en escenarios poco dotados.

5.2.1- Método 1: Imagen$_x$ frente a un modelado del fondo

Se trata del algoritmo más sencillo de implementar en el proceso de detección de movimiento. Basado en el modelado de un fondo (representado por *imf*) a partir de n imágenes (*img*) con el que se contrastará frente a una imagen de cada instante.

$$imf(x,y) = \sum_1^n \frac{img_n}{n} \qquad (5.1)$$

A nivel computacional se trata de un método poco exigente, no obstante requiere un proceso de adquisición y modelado del fondo, que dependiendo de la exigencia que se le dé a esta tarea, es uno de los principales cuellos de botella para aplicar este método en tiempo real y más aún con alta variación de los parámetros (contraste, luz, etc.).

Un fondo aceptable se obtiene a partir de un modelado (del promediado del valor de los píxeles (5.1)) de unas veinte imágenes, siendo recomendable modelar fondos de cien imágenes. Además hay que tener en cuenta la variación de luz que sufre un escenario a lo largo del periodo de funcionamiento, por lo que el fondo debería actualizarse periódicamente, teniéndose en cuenta los muchos aspectos que influyen a la hora de realizar este proceso.

$$imf_{nueva}(x,y) = \alpha \cdot imf(x,y) + \beta \cdot img_{n+1}(x,y), donde\ \alpha + \beta = 1 \qquad (5.2)$$

$$imf(x,y) = imf_{nueva}(x,y) \qquad (5.3)$$

Por otro lado este método necesita que el objeto a seguir presente mayor contraste con respecto al fondo del escenario ya que si no pasaría casi desapercibido. Además este método se ve gravemente afectado por las condiciones ambientales (luz ambiente, ruidos en la imagen, etc.).

$$img_f(x,y) = |\, imf(x,y) - img_{n+1}(x,y)| \qquad (5.4)$$

Las Figuras 5.1 y 5.2 corresponden a la evaluación de uno de los algoritmos implementados para la sustracción de fondo basado en la diferencia de dos imágenes (5.4), donde una de ellas es la perteneciente a un fondo evolutivo obtenido al aplicarse las fórmulas (5.2) y una vez evaluado se actualiza el fondo aplicándose la fórmula (5.3).

Figura 5. 1 Resultado del algoritmo con un objeto de alto contraste

Como se puede observar en la imagen 5.1 el comportamiento del algoritmo para un fondo más o menos estable y un elemento que ofrezca gran contraste es bastante bueno, siendo fácilmente reconocible el objeto en el fondo. Sin embargo cuando el fondo y el objeto poseen colores similares el resultado obtenido deja mucho que desear, resultando prácticamente imposible diferenciar entre objeto y ruido.

Figura 5. 2 Resultado del algoritmo con un objeto de bajo contraste

Estos resultados se han obtenido tras someter al resultado de la fórmula (5.3) a una serie de filtrados (5.5), (5.6), (5.7), con el objetivo de resaltar las detecciones y eliminar ruidos en la imagen de salida, estos filtros se han configurados mediante la experiencia

de la prueba y error, estableciéndose una configuración genérica que sea válida para cualquier escena.

$$img_n(x,y) = \begin{cases} 1, para\ img_n(x,y) \geq T \\ 0, para\ img_n(x,y) < T \end{cases} \quad (5.5)$$

$$x \ominus B = \{ p \in \mathcal{E}^2 : p = x + b\ para\ cada\ b \in B \quad (5.6)$$

$$x \oplus B = \{ p \in \mathcal{E}^2 : p = x + b, si\ x \in X\ y\ b \in B \quad (5.7)$$

El comportamiento de dicho algoritmo con objetos en movimientos siempre y cuando el fondo presente contraste con el objeto es bastante aceptable, pero se descartó su utilización ya que los escenarios a los que se orienta el funcionamiento del proyecto no tienen porqué cumplir dichas precondiciones.

Tras la primera prueba se optó por estudiar un método que modelase el fondo sin actualizarse a lo largo del tiempo, es decir sin aplicar las fórmulas (5.2) y (5.3). Como en el caso anterior tras la implementación del algoritmo hubo que realizar pruebas para configurar los parámetros de filtrado aplicándose las fórmulas (5.5), (5.6), (5.7), tras la operación de la diferencia entre imagen y fondo (5.4). Los resultados que se muestran a continuación se han realizado en un entorno ideal (fondo estable a lo largo del tiempo y de color uniforme) entre elementos con y sin alto contraste del objeto respecto al fondo.

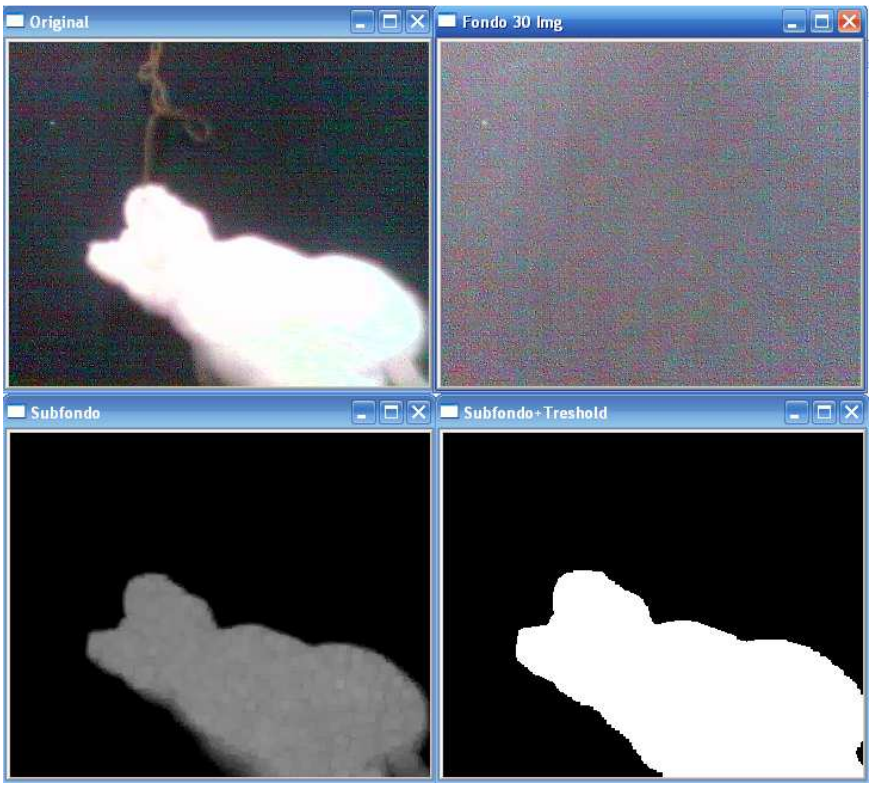

Figura 5. 3 Prueba del algoritmo con modelado de fondo de 30 imágenes

En la Figura 5.3 podemos observar el resultado obtenido tras aplicarse el algoritmo. En él se ha fijado que el modelo se calcula a partir de las primeras 30 imágenes aplicándose

técnicas estadísticas (5.1), dicho proceso tarda unos treinta segundos en ejecutarse. Como se observa el resultado dado por el algoritmo funciona bastante bien con un objeto con alto contraste con respecto al fondo, como cabía de esperar.

Con la Figura 5.4 se evalúa los resultados que se obtienen cuando el objeto ya no presenta un contraste alto. En este caso se ha sido un poco extremista, ya que se ha esperado a que el algoritmo se estabilice obteniéndose una salida más degradada. No obstante se quería hacer hincapié en la calidad de una posible salida. El comportamiento del algoritmo en condiciones ideales es bastante bueno pero como se ha comentado a lo largo de este punto la aplicación no se va a usar en entornos ideales.

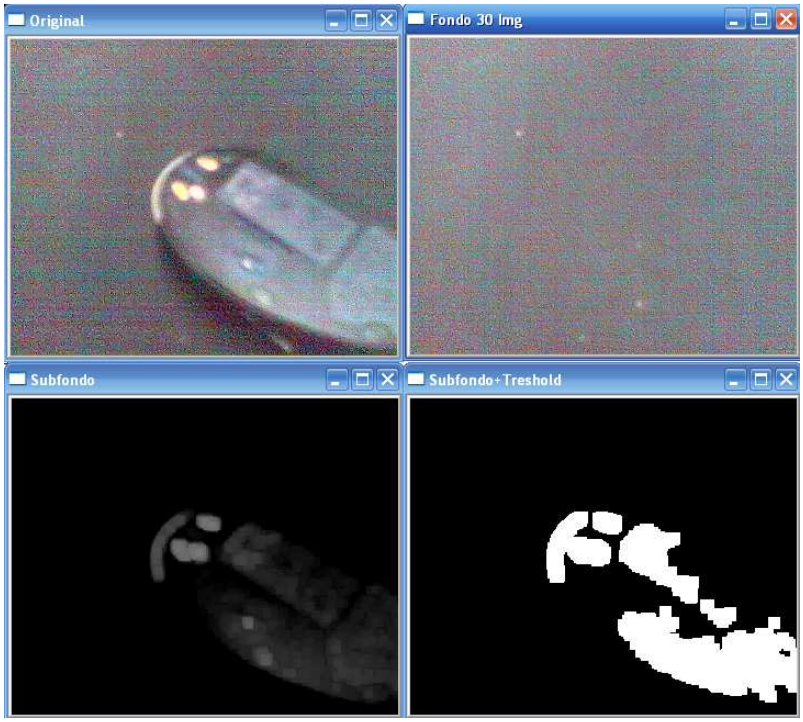

Figura 5. 4 Prueba del algoritmo con modelado de fondo con objeto con bajo contraste

Con las siguientes imágenes de la Figura 5.5 se muestra un ejemplo de los resultados obtenidos tras el cálculo del modelado del fondo a partir de 100 imágenes. Estos resultados nos permiten concluir que para un fondo uniforme, realizar este proceso que conlleva un tiempo de cómputo superior a los treinta segundos en el equipo de referencia (citado en el punto de componentes hardware del proyecto), es un costo elevado que no aporta nada nuevo en relación a lo obtenido en el caso anterior.

Figura 5. 5 Prueba del algoritmo con modelado de fondo de 100 imágenes

En las imágenes de la Figura 5.6 se representa la salida obtenida con la misma configuración que en el proceso anterior con respecto a un objeto con poco contraste.

Se puede observar como la salida obtenida no nos permite diferenciar con exactitud el objeto tras el procesado de la substracción del fondo.

Figura 5. 6 Prueba del algoritmo con modelado de fondo de 100 imágenes y objeto de bajo contraste

Por último se muestra qué ocurre cuando un objeto se superpone a un fondo no homogéneo.

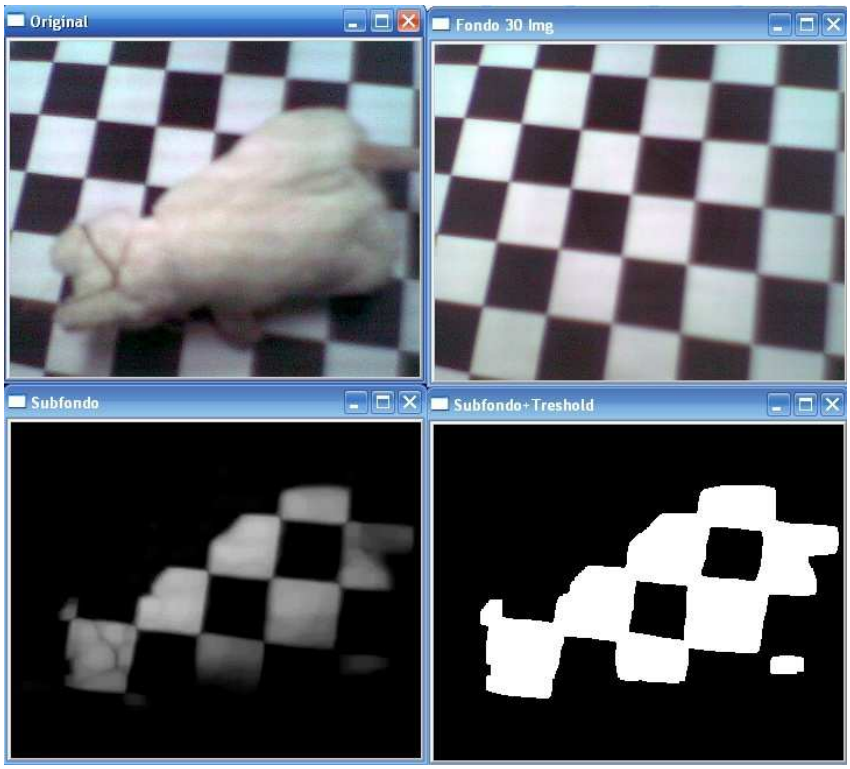

Figura 5. 7 Prueba del algoritmo con un fondo no homogéneo

Del estudio de la Figura 5.7 se puede determinar que en los valores de los píxeles del fondo son superiores que los del objeto provocando ocultaciones parciales o totales del objeto, principal causa por la que dicho algoritmo fue descartado.

5.2.2- Método 2: Fondo/objeto *Codebook*

Este segundo método también se basa en la creación de un modelo del fondo, aunque algo más elaborado que la anterior técnica, no obstante sigue siendo un método de modelado estacionario basado en técnicas de segmentación en el que para cada píxel se estudia los valores de sus componentes RGB y la 6-tupla formada por los valores máx. y mín. del brillo, la frecuencia, MNRL (máx. tiempo sin repetición de la consigna) y el primer y último tiempo de acceso. Este algoritmo se ha realizado con operaciones proporcionadas por la OpenCV (librería de tratamiento de imagen), con las que se va construyendo un modelado del fondo basado en técnicas de Codebook, usando las medias de los promedios de las diferencias para obtener el fondo. Usándose como referencia el modelo propuesto por Kyungnam Kim en el artículo **[PW, Kim]** y mostrado en la Figura 5.8.

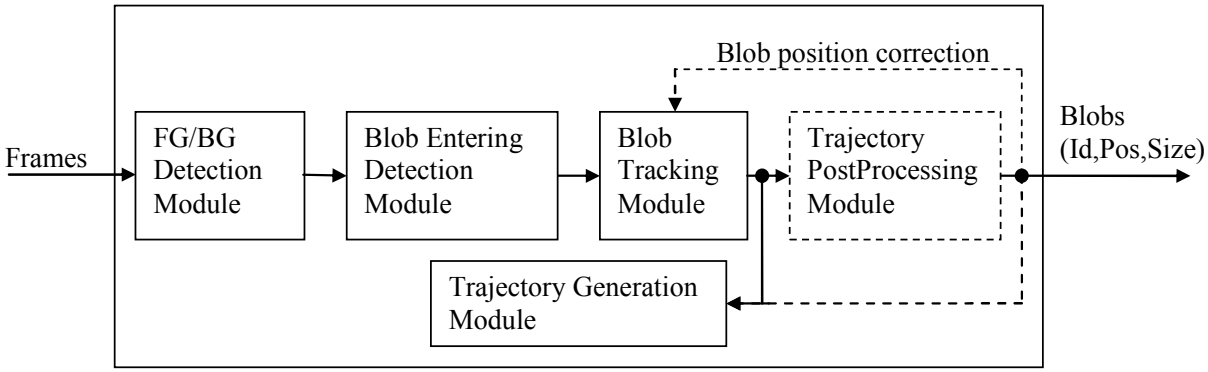

Figura 5. 8 Diagrama de los módulos de Blob Tracking ***[PW, Blob]***

Como contrapartida esta técnica necesita un número de iteraciones bastante alto para obtener un buen modelo de fondo (en torno al doble de imágenes iniciales que en el anterior método, siendo recomendable usar unas trescientas imágenes). No obstante presenta mejores resultados que el método anterior al ser más adaptativo a los medios (fondo, luz, etc.), como podemos observar en la Figura 5.8. En ella el modelado del fondo se ha realizado a partir de 50 imágenes y como se ve en la imagen superior derecha, el modelo del fondo se construye a partir de imágenes transformadas al modelo YUV, es decir en términos de una componente de luminancia y dos componentes de crominancia.

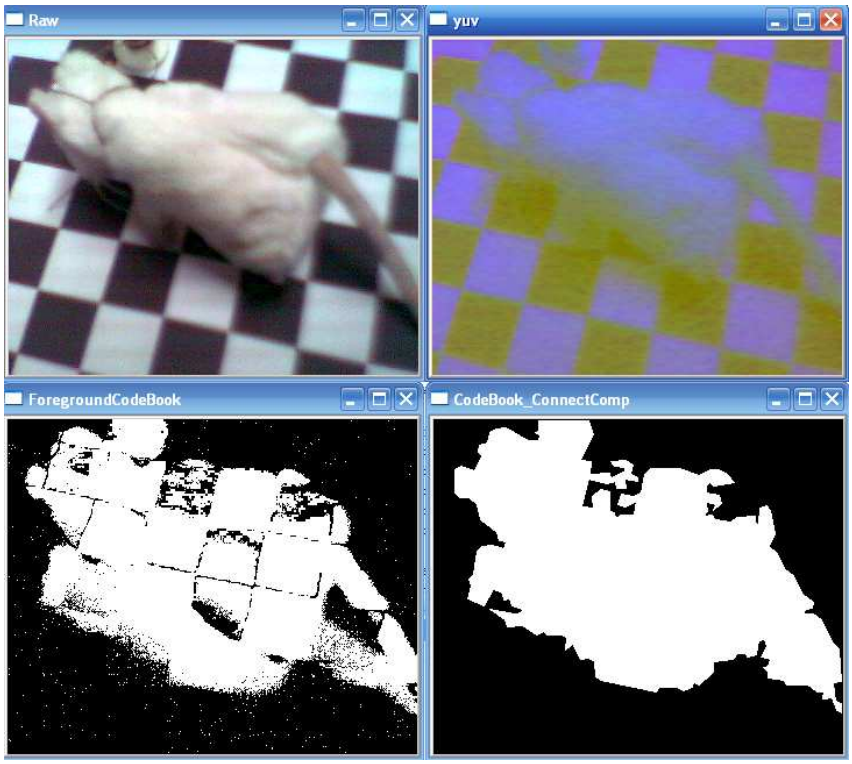

Figura 5. 9 Prueba del algoritmo Background/foreground Codebook

Aún sin ser ideal el resultado obtenido tras aplicar el filtro del método *Codebook*, la imagen obtenida es algo mejor que la obtenida en la Figura 5.8 donde el resultado no era un objeto bien definido. Hay que comentar que cuantas más imágenes compongan el modelado del fondo mejor será el resultado obtenido y no obstante mayor el tiempo de cálculo de éste. Este método se ve menos afectado en sistemas luminosos, como ocurría en los métodos anteriores, siempre y cuando no aparezcan sombras en el escenario ya que afectan considerablemente en el resultado obtenido.

La técnica de filtrado usada en este método es bastante sensible al ruido de la imagen por lo que la calidad de la imagen obtenida en el preprocesado influye significativamente en el resultado. Por lo que se llegó a probar alguna técnica de filtrado de las imágenes de entrada (Mezcla de Gaussianas) pero aún así a veces el ruido provocaba falsos positivos.

Como conclusión se determinó que se trata de un método que aporta resultados aceptables siempre y cuando las imágenes se presenten sin mucho ruido y sombras en el escenario. Se trata de un método que requiere una inicialización (construcción del modelo del fondo) muy costosa, lo que unido a lo comentado con anterioridad se terminó de descartar.

5.2.3- Método 3: Imagen$_{x-1}$ frente a Imagen$_x$

Con esta técnica de diferencia entre imágenes se estudiarán varios métodos de procesado que ofrece la librería de la OpenCV. Dichos modelos son incapaces de detectar objetos que se encuentren inmóviles en el escenario ya que sólo se centran en la detección de movimientos entre imágenes y como ocurría en los métodos anteriores no modelan el fondo. Por lo tanto son usados como apoyo a la construcción de modelos que nos permiten detectar movimiento entre secuencias de imágenes. La diferencia entre las imágenes es un proceso cuyo coste computacional es escaso, estando disponibles multitud de métodos. Para el presente estudio se han tenido en cuenta los siguientes:

Histórico de Movimiento

Este método se construye a partir de una sucesión de imágenes que componen un modelado evolutivo, llamado histórico de imágenes, que nos permiten detectar movimiento y la dirección de éste, dada una secuencia de más de dos imágenes. Este método en combinación con otros procesos nos permite realizar encuadre de los objetos dentro del plano monitorizado. Un ejemplo de este algoritmo se puede observar en la Figura 5.9 y su modelo de construcción es el siguiente:

$$mhi(x,y) = \begin{cases} timestamp, si\ silhouette(x,y) \neq 0 \\ 0, si\ silhouette(x,y) = 0\ y\ mhi < (timestamp - duration) \\ mhi(x,y),\ en\ otro\ caso \end{cases} \quad (5.8)$$

Los parámetros que más afectan en los resultados finales son los relativos al tiempo de duración de la secuenciación del modelado (HMI), ya que la sección crítica del algoritmo se encuentra en ajustar dichos parámetros. Otra de las claves fundamentales es el ajuste del encuadre, pero dicho aspecto se comentará al final del estudio de los métodos ya que es compartido por varios de ellos.

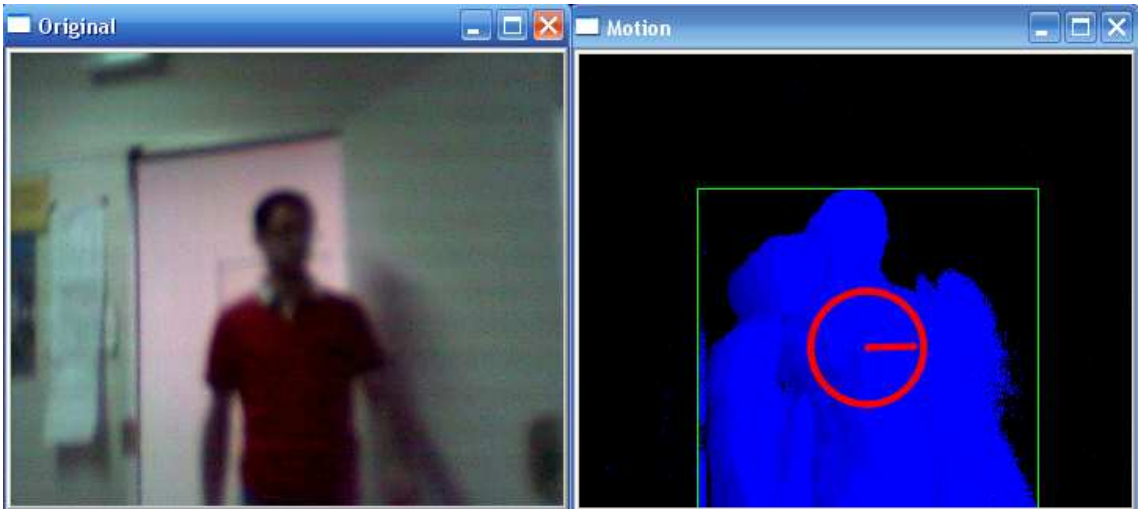

Figura 5. 10 Resultados al aplicar el algoritmo de HMI

Este método por si sólo se ha descartado debido a la complejidad que tiene el cálculo y establecimiento de los parámetros para construir un HMI estable, además del problema común a todos los métodos comentados dentro de este punto de no detectar objetos que permanezcan estáticos en el escenario. No obstante se trata de un método bastante interesante cuando se pretende realizar el seguimiento de objetos (*traking*) por lo que, junto a una de las técnicas que veremos más adelante (Método probabilístico con mezcla de gaussianas), que cubre las carencias de este método se ha implementado un algoritmo que hace uso de esta técnica.

Promedio del Movimiento

Como en el caso anterior tras obtener las secuencias de diferencias entre imágenes lo que se hace es realizar un modelo, en este caso formado por el movimiento promedio entre las secuencias de imágenes. Según el siguiente modelo:

$$acc(x,y) \leftarrow (1-\alpha) \cdot acc(x,y) + \alpha \cdot img(x,y) \quad Si\ mask(x,y) \neq 0 \qquad (5.9)$$

Donde:

acc: es el acumulador.

α (alpha): es la velocidad de regulación.

Detectándose los elementos que varían su posición dado un umbral y realizándose procesos de escalados entre las imágenes. Con dicha técnica se obtienen resultados satisfactorios en los escenarios probados, uno de ellos se muestra en la Figura 5.10, siempre y cuando la calidad de la imagen tenga un mínimo aceptable. Entre sus características destaca su robustez ante cambios de luz entre imágenes, por lo que las sombras no son un problema en el resultado del procesamiento.

Figura 5. 11 Resultados al aplicar el algoritmo del movimiento promedio

El método destaca entre los demás por ser uno de los más estables y adaptativos en la detección de movimiento en escenarios con distintas condiciones y configuraciones. Como detrimento este método tiende a perder calidad en los resultados mostrados cuando el nivel de imágenes por segundo es escaso, por lo que para aplicaciones de

procesado de vídeos es sin duda uno de los que mejores resultados ofrece. Además uno de los aspectos a destacar de este método es su bajo costo computacional por lo que esta técnica fue seleccionada para formar parte del repositorio de algoritmos implementados en el software.

Mezclas de Gaussianas

Otro de los métodos que se han estudiado con el objetivo de la sustracción de fondo es el propuesto en el artículo **[PW, Fatih]** del que se ha implementado una función basada en la aplicación de mezcla de gaussianas que nos aporta la OpenCV. Dichas gaussianas no se usan para modelar un fondo, sino que como todos los métodos de este punto se centra en calcular la sustracción de fondo a partir de dos imágenes.

Es un método bastante útil para la detección de objetos en distintos escenarios, ya que las operaciones de la librería nos permiten inicializar los parámetros de las gaussianas para así adecuar el algoritmo al escenario. Un ejemplo de su funcionamiento se puede observar en la Figura 5.11.

Figura 5. 12 Resultado al aplicar el algoritmo de mezclase de gaussianas

Las grandes carencias que presenta este método son el efecto negativo que provoca la luz ambiental y en caso de tratarse de fluorescentes su frecuencia. Además las propias sombras que aparecen en el escenario generan una gran cantidad de ruido por lo que es un aspecto crítico el realizar una adecuada inicialización del algoritmo. Asimismo, y al no modelar el fondo, se necesita una alta frecuencia de imágenes por segundo para

poder presentar un resultado definido, ya que sino el resultado obtenido parece ruido en vez de un objeto detectado. Por lo tanto, tan sólo es válido para el procesado de vídeos con una alta frecuencia de cuadros por segundos, y siempre que se ajusten bien los parámetros iniciales. Por todo ello esta técnica ha sido descartada.

5.2.4- Método 4: Modelado probabilístico del fondo

Por último se estudió una técnica basada en el modelado del fondo con métodos probabilísticos y actualizado progresivo, para esta técnica partimos de una librería externa centrada en la utilización de mezcla de gaussianas como técnica de substracción de fondo. Esta librería fue desarrollada y tomada del estudio llevado a cabo por Alper Yilmaz, Omar Javed y Mubarak Shah **[PDF, OT],** posteriormente modificada por Luis Antón y Modesto Castrillón investigadores del instituto universitario SIANI de la ULPGC, y cedida y configurada para su utilización en este proyecto. Los resultados de su funcionamiento se visualiza en la Figura 5.12. Se trata de un método parametrizable, por lo que permiten ajustar el preprocesado de imágenes y construcción de fondo, tras un estudio exhaustivo y realización de pruebas, la prioridad al usar esta técnica consiste en la reducción al máximo del ruido que se genera en la salida. Por lo que hay que realizar filtrados en ésta, llegándose a probar diferentes métodos y configuraciones. Dentro de dichas pruebas se usaron los filtros aplicados por la técnica de *CodeBook* llegándose a la conclusión que los aportado por técnicas más sencillas, como el *Smooth*, *Erode* y el *Dilate,* son mucho más efectivos.

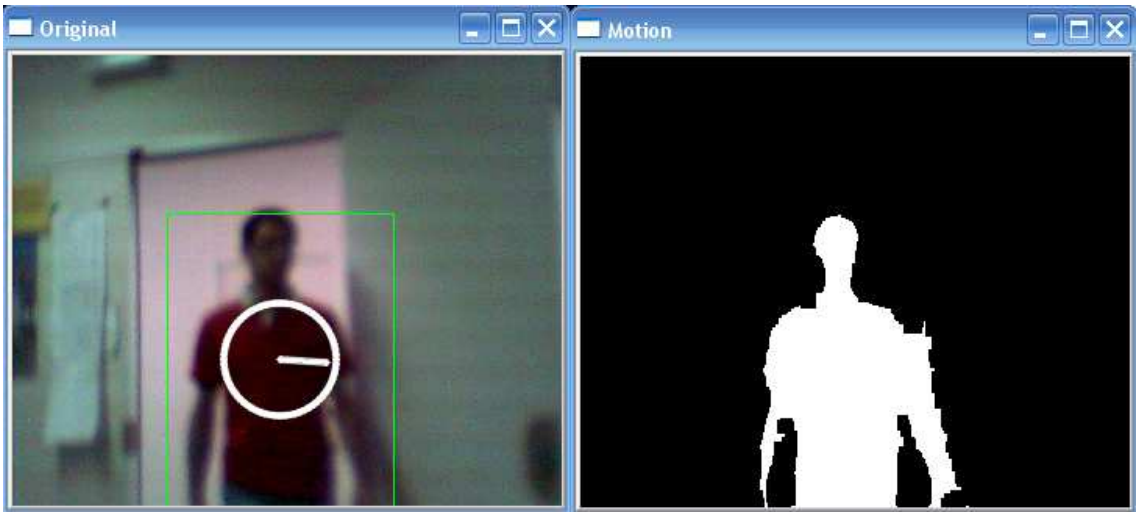

Figura 5. 13 Resultados al aplicar el algoritmo de modelado probabilístico

De este estudio se ha podido determinar que el número de gaussianas necesarias para modelar un escenario con variaciones de luz y calidad de imagen mínima debe ser igual o superior a cuatro, siendo el ideal unas diez para entornos muy extremos.

En el caso de disponer de imágenes de alta calidad y por lo tanto menor el ruido en las imágenes se puede llegar a reducir hasta el mínimo el número de gaussianas (dos gaussianas por píxel), no obstante hay que tener un valor de sensibilidad del movimiento dentro de las gaussianas bastante alto.

En lo relativo al cómputo necesario, dicha configuración tiene un coste medio, cierto es que a medida que decrece el número de componentes de la gaussianas tenderá a ser más rápido, pero en términos generales se podría decir que funciona a tiempo real, tomándose como referencia el ordenador usado para el desarrollo del proyecto.

Se trata de uno de los métodos más eficientes en la detección de movimiento de objetos. Los principales inconvenientes en la utilización de este método son la necesidad de obtener una imagen limpia del escenario, sin que aparezca ningún objeto susceptible a ser discriminado como objeto móvil, aunque es un método bastante adaptativo es sensible a la variación y frecuencia de la luz en el escenario, lo que genera ruido que distorsiona la salida. Debido a esto la aparición de sombras en el escenario es un aspecto crítico en el procesado de la salida obtenida, por lo que se debe de realizar un filtrado de ésta, en detrimento de la pérdida de información en la imagen. Se puede ver una comparativa del método con y sin filtros de las sombras en la Figura 5.13. Sin embargo en escenarios oscuros el funcionamiento del algoritmo suele ser bastante crítico, aunque esta situación no suele ser la más común en escenarios vídeo vigilados por cámaras comunes.

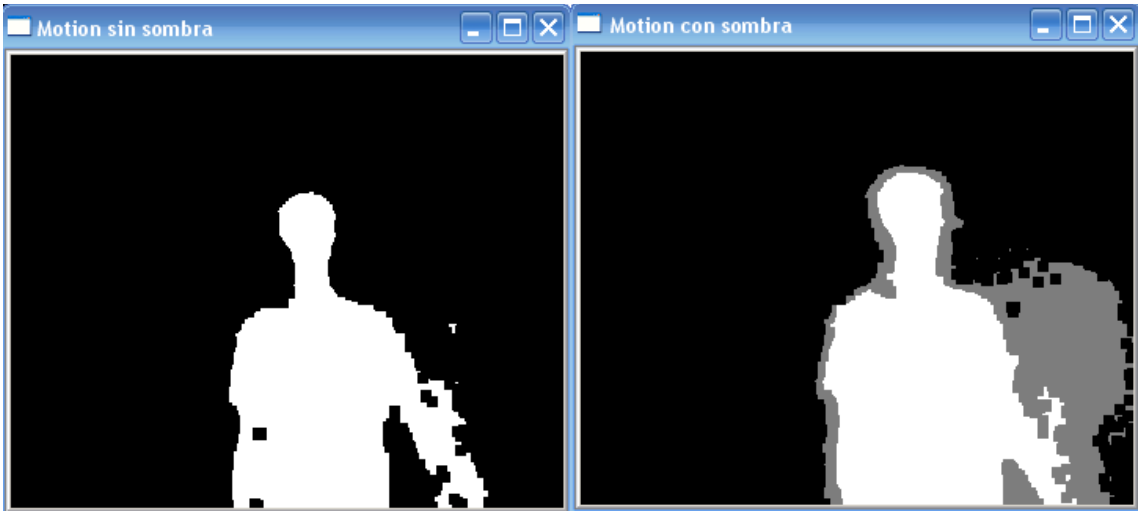

Figura 5. 14 Comparativa de resultados de la salida con y sin eliminación de sombras.

Pese a todo esto, este es uno de los métodos elegido para la detección de movimiento dentro de la aplicación generada. Se ha tratado de realizar un filtrado lo más genérico posible que intente cubrir la mayoría de escenarios posibles y en cuanto a la detección de personas se ha probado combinar dicho proceso con los comentados en el siguiente punto.

5.2.5- Técnicas de detección de personas

Las técnicas de detección de persona descritas en este punto parten de los resultados obtenidos por los métodos de substracción de fondo. Desarrollándose métodos a partir de la OpenCV basados en las siguientes técnicas:

- Búsqueda de contornos en la secuencia de imágenes procesadas que se ajusten a unos tamaños acordados según el posicionamiento de las cámaras.
- Segmentación del historial del imágenes que como en el caso anterior se considera persona todo aquello que cumpla las precondiciones de tamaño dadas.
- Uso de reconocedores de parámetros basados en el método de Viola-Jones **[PW, IJCV]**.

Los dos primeros métodos comentados se tratan de procesos de bajo coste computacional que sólo atienden a la discriminación de los resultados según una serie de condiciones dadas. Actualmente las cámaras con las que debe operar el software deberán estar colocadas a unos 2-2.50 metros de altura para el funcionamiento óptimo del algoritmo, posibilitando reconocimientos de objetos en un rango de 5-7 metros de radio dentro del campo de visión de la cámara. Ambos métodos están basados en la segmentación del modelo del fondo obtenido en los que se considerarán personas a aquellos elementos que superen un área superior al 10% de la imagen y su componente en altura sea mayor que su ancho. Por otro lado al usar el método de HMI y si se dispone de un suficiente ratio de imágenes en la construcción del modelo del fondo se puede predecir la dirección que sigue dicho objeto.

En cuanto al uso de reconocedores basado en Viola-Jones el factor determinante consiste en lograr un consenso entre la detección de falsos positivos y el coste computacional que supone disminuir éstos, al aumentar el nivel de exploración del reconocedor. Para ello se decidió no aplicar el algoritmo directamente a la imagen adquirida sino a la salida de la técnica de substracción de fondo aplicada. Consiguiéndose reducir el coste computacional que este método añade al disminuir los niveles a explorar y lográndose una tasa cercana a cero para los falsos positivos.

En el algoritmo de Viola-Jones se ha estudiado el comportamiento de éste usando tres patrones a reconocer: cabeza y hombro, parte inferior del cuerpo y cuerpo entero, según se han comentado los podemos clasificar atendiendo a la relación falsos positivos coste computacional. Al aplicar directamente el algoritmo sobre una imagen sin tratar este proceso nos da como resultado en torno al 50-70% de tasa de acierto y cerca de un 50% de falsos positivos, valores bastantes pobres para la cantidad de cómputo que requiere, por lo que se estudió realizar preprocesado de las imágenes analizadas. Dicho preprocesado consiste en la aplicación de sustracción de fondo quedando una imagen prácticamente sin ruido, por lo que los falsos positivos provocados por éstos se eliminan en más del 99% para una configuración media (nivel de profundidad 15-20) del algoritmo de Viola-Jones. Sin embargo, al distorsionar la imagen a procesar la tasa de acierto disminuye a un 30-40% dependiendo de la agresividad de las técnicas de filtrados aplicadas en el método de sustracción de fondo.

5.2.6- Problemas y limitaciones

En todos los métodos comentados a lo largo de este punto nos enfrentamos a dos grandes tesituras que en su mayor o menor medida afectan a los resultados obtenidos al tener que priorizar uno más que otro. Los dos problemas que se abordan en este punto son la substracción de fondo y la detección de movimiento.

En el primero de ellos, el gran escollo es lograr la configuración ideal de los filtros que nos permitan obtener los resultados más óptimos dentro de los distintos escenarios que nos hemos propuesto, tratándose de evitar con ello la especialización en un solo tipo de escenario. Por lo que se llegó a un compromiso de la calidad de las imágenes a estudiar, la cantidad de imágenes por segundo que se obtienen y la cantidad de ruido (variación de luz, apariciones de sombras, etc.) que afectan a las imágenes procesadas. Debido a estos aspectos se prefirió sacrificar calidad en los modelos que se obtenían para poder abarcar mayores escenarios.

En cuanto a lo relacionado con el seguimiento de personas, la principal dificultad es determinar qué se considera una persona, ya que en este proceso afecta la bondad del resultado obtenido en la fase anterior y el posicionamiento de las cámaras (principalmente la altura) además del área monitorizada. En este aspecto los métodos implementados en el software son bastantes primarios, ya que si multiplicamos la complejidad de procesamiento necesario para la substracción y modelado del fondo más

la detección de personas por el número de cámaras que componen el sistema nos encontramos ante el verdadero cuello de botella del software.

Por estos motivos, se han intentado realizar métodos robustos y genéricos a la hora de modelar el fondo y usar técnicas más básicas en la detección de objetos en el escenario este último que influye en el resultado de la detección de personas por parte del software que se ha intentado compensar con las técnicas de seguimiento implementadas en éste.

5.3- Algoritmos de selección de imagen y seguimiento

La monitorización de los sistemas al admitir distintas configuraciones requiere de técnicas de tratamiento y selección de la imagen a monitorizar que se adapten a las necesidades de éstos. Por ello uno de los puntos en la realización de este proyecto se ha dedicado al estudio implementación de procesos de selección de la cámara a monitorizar con y sin seguimiento en el escenario. Antes de entrar más a fondo en la descripción de éstos describiremos las diferentes situaciones que se han contemplado:

- Visualización de una sola cámara. La monitorización manual es el proceso más sencillo ya que sólo se realiza procesamiento y visualización sobre las imágenes de una cámara.
- Visualización de un sistema de cámaras. La monitorización de un sistema de cámaras admite diferentes configuraciones pero todas ellas se comportan de la misma manera mientras no haya eventos registrados en el sistema. Siempre se visualizan las cámaras de forma rotatoria durante un cuanto de tiempo realizando un barrido de las cámaras según fueron añadidas al sistema. Las otras configuraciones que se pueden encontrar son las siguientes:
    - Monitorización de manera automática. Dentro de este tipo de visualización el sistema puede funcionar de dos formas totalmente distintas atendiendo a los componentes declarados en el software. Por lo que la monitorización del sistema puede funcionar por prioridades de las cámara o guiada por el plano del sistema.
    - Monitorización por eventos. Es la única monitorización que no requiere de procesamiento de las imágenes de las cámaras ya que está guiada por los eventos de las cámaras que componen el sistema recogidos en la base de datos del software.

- Monitorización simulada con vídeos. El sistema empleado para esta configuración ha sido una adaptación del empleado en los sistemas monitorizados automáticamente.

A continuación se hará una descripción más detallada de los procesos de monitorización realizados en el software.

5.3.1- Monitorización manual

Es el proceso más simple de monitorización de un sistema, en el que sólo se ve afectado una cámara del sistema. Por lo que todo el gasto computacional de esta operación será empleado a una sola cámara. El proceso de monitorización responde a la Figura 5.15.

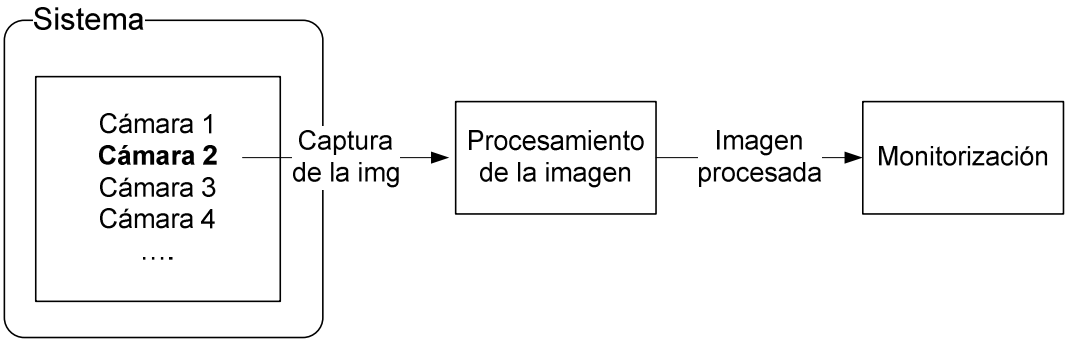

Figura 5. 15 Esquema de la técnica de monitorización manual

5.3.2- Monitorización rotatoria

En este tipo de monitorización ya se ve envuelto todo el sistema de cámaras y, como se comentó anteriormente, este tipo de monitorización transcurre cuando no hay eventos por parte de ninguna de las cámaras del sistema. En la Figura 5.16 se refleja el funcionamiento del sistema de selección de la imagen a monitorizar.

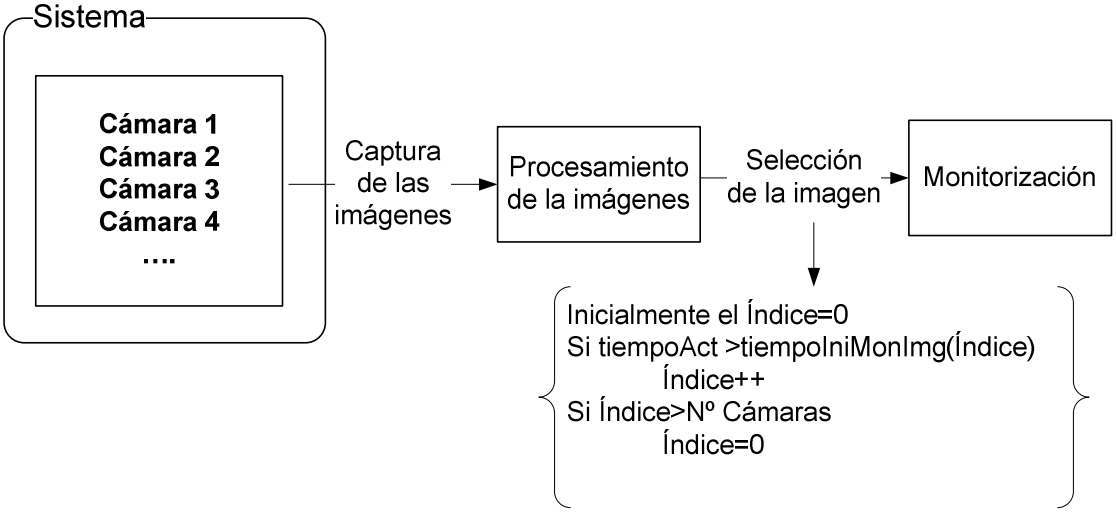

Figura 5. 16 Esquema de la técnica de monitorización rotativa

5.3.3- Monitorización por prioridad

La monitorización de un sistema por prioridad responde a la necesidad de atender unas cámaras antes que otras cuando ocurran alertas simultáneas dentro del sistema. Por lo tanto cuando se define una cámara en la base de datos se le establece una prioridad que va desde uno (máxima prioridad) a tres (menor prioridad). A partir de este sistema el software se apoya en un algoritmo para introducir y atender las alertas producidas por las cámaras basado en colas de prioridad con los mismo tres niveles que el de las cámaras. De esta forma, primero se atenderán las alertas producidas por las cámaras de mayor prioridad llegando a poder producir inanición a la hora de monitorizar un sistema donde las cámaras con mayor prioridad están fijas con alertas. Esta posibilidad se estudió y es un caso poco probable que suceda en un sistema vídeo vigilado, no obstante en caso de ocurrir lo que se recomienda al usuario es dar la misma prioridad a las cámaras solucionándose el problema de inanición.

A continuación se detalla el comportamiento del algoritmo reflejándose las diferentes situaciones en las que puede ver envuelto en la Figura 5.17. Tras producirse una detección de un objeto dentro del campo de visualización se consulta la prioridad de la cámara que produjo la detección del objeto, y si ésta no está ya en la cola de prioridad de su nivel se introducirá en ella.

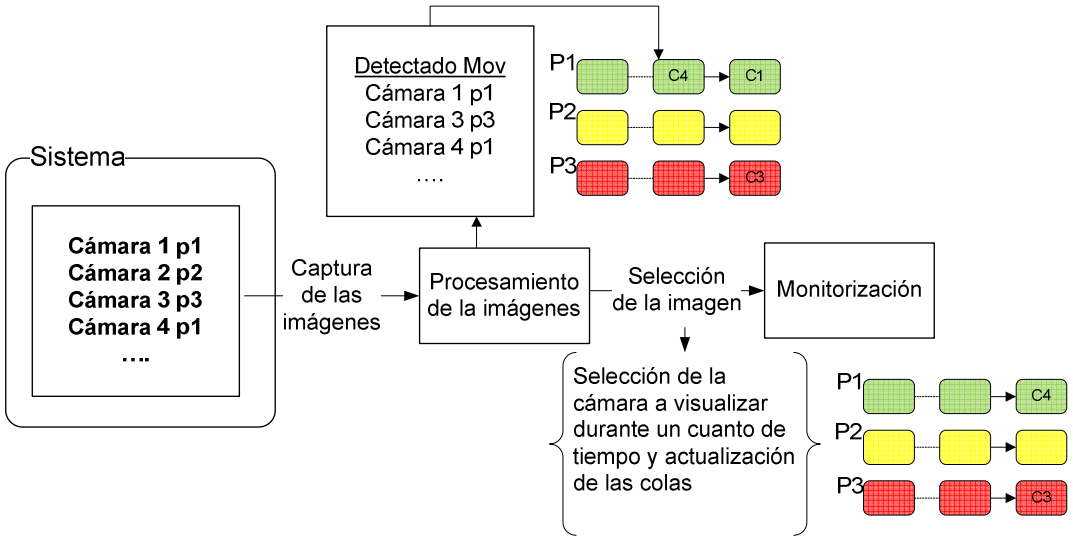

Figura 5. 17 Esquema de la técnica de monitorización por prioridad

En el caso de que se produzcan alertas en dos cámaras con la misma prioridad en un mismo instante se añadirá primero la cámara cuya imagen haya sido procesada antes (actualmente el procesamiento funciona de manera secuencial). Y si en caso contrario pertenecen a distintas prioridades, cada una irá a su cola.

Si a la hora de monitorizar una cámara que estaba en alguna de las colas de prioridad no muestra actividad en su campo de visión, se la sacará de la cola de su prioridad y se atenderá la siguiente petición de su cola, y en caso de estar vacía se atenderá las alertas de los siguientes niveles de prioridad. Si tras atender todas las peticiones las colas se encuentran vacías saltará automáticamente el algoritmo de monitorización rotatoria por donde lo había dejado.

5.3.4- Monitorización guiada por plano

La monitorización guiada por plano requiere que exista un plano bien formado del sistema declarado en el software. Empezaremos por definir un plano de un sistema de cámaras como una representación topológica de la situación de éstas, y que puede modelarse mediante una estructura de grafo representado por lista de adyacencia, como se muestra en la Figura 5.18.

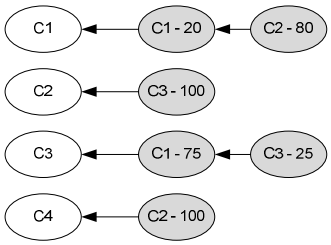

Figura 5. 18 Representación de la estructura de un plano

Dentro de este plano, que podemos construir en el software, destacamos varios aspectos importantes en su composición. Lo primero que observamos es el identificador de la cámara a la que se conectan *n* cámaras y a continuación es posible encontrar una serie formada por la dupla identificador de la cámara conectada y su peso. El peso viene a representar la probabilidad de que una persona detectada por una cámara *A* (nodo principal de la lista) sea monitorizada al cabo de un tiempo por una cámara *B*, la suma de las probabilidades de las cámaras "conectadas" a un nodo principal en caso de que existan conexiones debe ser 100. Además la configuración del plano actual admite representar conexiones de una cámara consigo misma. A continuación se mostrarán un plano y su configuración para facilitar el entendimiento de los aspectos que se han comentado.

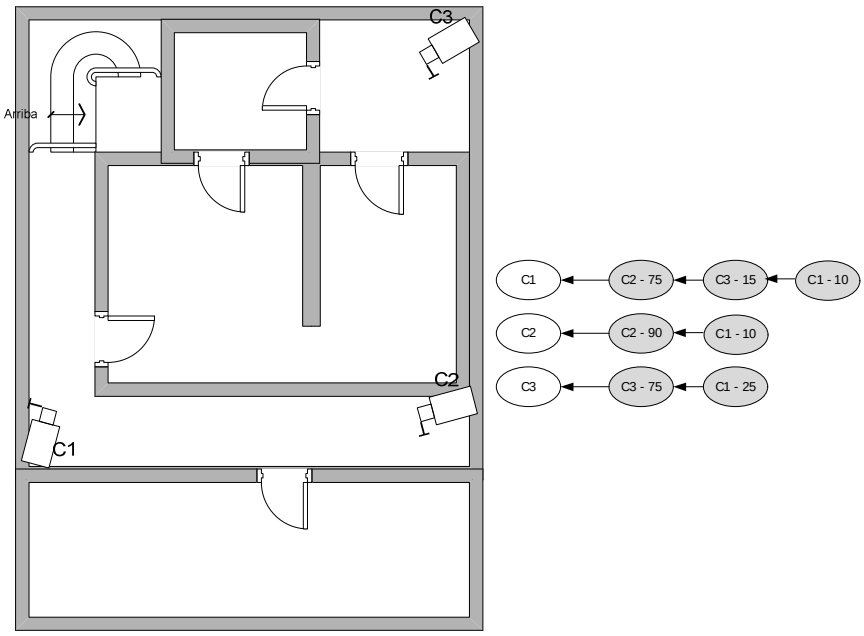

Figura 5. 19 Ejemplo de la correspondencia topológica de un plano

En la Figura 5.19 se muestra un plano de una planta y su representación topológica en el software, en ella se muestra todas las posibles interconexiones entre las cámaras existentes. Como ya se ha comentado, el número que acompaña al identificador de la cámara representa la probabilidad que dicha cámara tiene para detectar un individuo procedente de la cámara indicada (nodo blanco). En la Figura 5.19 se puede observar la existencia de habitaciones no monitorizadas que interconectan zonas controladas permitiendo que un individuo pueda realizar múltiples rutas que el sistema debe conocer y prever. Por ejemplo desde la zona controlada por la cámara 1, se puede acceder a la controlada por la cámara 2, a la zona de la cámara 3 o incluso retornar a la zona controlada por la cámara 1 sin pasar por ninguna de las otras zonas monitorizadas. El caso de los bucles de una cámara consigo misma cobra más sentido en este ejemplo, donde la cámara 2 y 3 tienen altas probabilidades de ser ella quienes se activen una vez se pierda al objetivo, posibilitando la detección al instante de éste si aparece antes de que pase un cuanto de tiempo.

Una vez puesto en situación se describirá el funcionamiento de la monitorización guiada por plano. Cuando se monitoriza con este método y no existen alertas en el sistema, como en los anteriores casos, toma el control de la monitorización el algoritmo rotatorio hasta que ocurra alguna alerta donde ya entraría en juego la monitorización guiada por plano. Cuando se capta la primera alerta en el sistema, la decisión de la cámara a

visualizar se deja en manos del algoritmo de selección por prioridad, lo que permite cubrirse las espaldas en caso de que se reciban varias alertas de cámaras a la vez.

Una vez detectado un intruso y fijado una cámara de referencia dentro del plano lo lógico es que dicho intruso siga un camino predecible, es decir se mueva de manera probabilística por determinadas zonas del plano vigilado, para poder adaptarse al entorno y mejorar el rendimiento del algoritmo se ha hecho que el peso de los nodos del plano se vayan ratificando cuando su elección sea la correcta, haciendo que el plano sea evolutivo a lo largo del tiempo.

En la Figura 5.20 se ha intentado ilustrar el funcionamiento del algoritmo poniéndonos en la situación de que una cámara deja de visualizar a un individuo, y se pasa a consultar a sus vecinas donde una de ellas logra captar al individuo y por lo tanto se reafirma su peso en detrimento del resto.

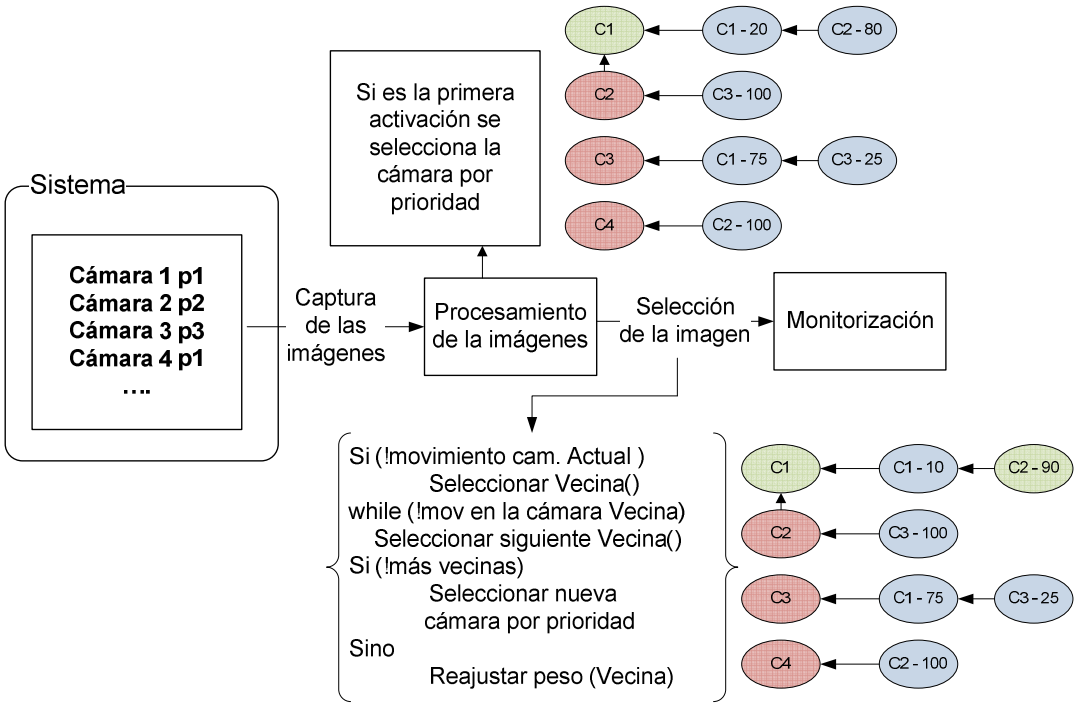

Figura 5. 20 Esquema de la técnica de monitorización por plano

Hay que comentar que las cámaras no comparten información por lo que no se realiza constatación de que el objetivo visualizado por las dos cámaras sea la misma persona. Simplemente se realiza una estimación de por dónde irá un individuo. También cabe destacar que si el plano actual está incompleto y no se contiene la información de todas las cámaras del sistema, los resultados que se presentan no serán muy fiables ya que el sistema carece de conocimiento. El uso de esta técnica es uno de los grandes puntos a refinar en el software ya que se requiere de bastantes cámaras que cubran el terreno a

monitorizar y se añadan estrategias más sofisticadas en el salto entre cámaras (actualmente por cuantos de tiempo).

5.3.5- Monitorización guiada por eventos

Como se comentó con anterioridad se trata de la única monitorización de las cámaras que no requiere procesamiento de las imágenes por lo que se ahorra el gasto computacional de esta operación y permite al software manejar mayor número de cámaras. Este tipo de monitorización, mostrada en la Figura 5.21, también se apoya en el uso de las colas de prioridad cuyo funcionamiento es totalmente igual que el comentado en uno de los puntos anteriores.

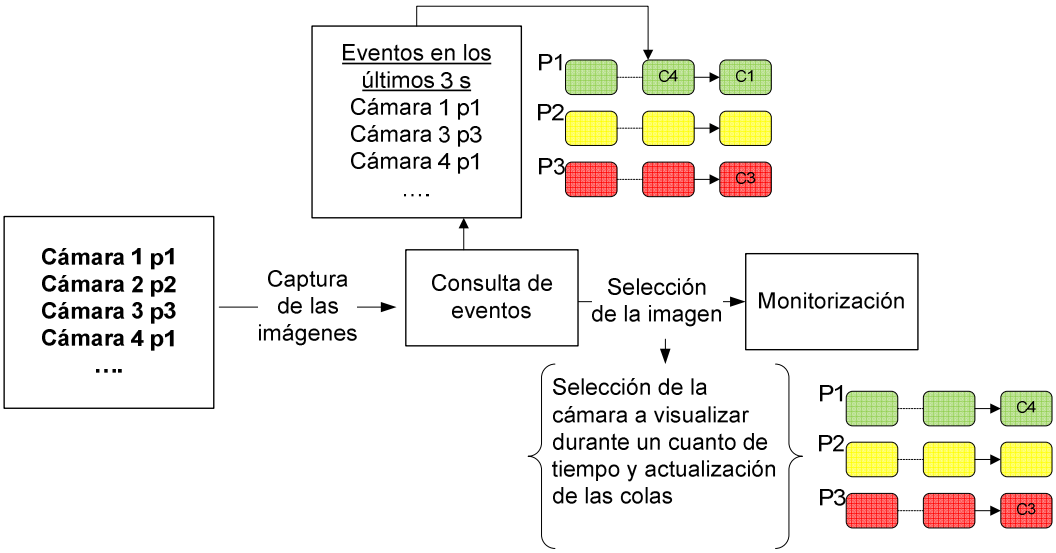

Figura 5. 21 Esquema de la técnica de monitorización guiada por eventos

Este tipo de configuración de monitorización permite que el software funcione de manera distribuida, como se muestra en la Figura 5.22. Donde el S.A.V. con monitorizando por eventos tiene acceso a todas las cámaras mientras que los demás S.A.V. se ocupan de un subconjunto de cámaras del sistema general realizando cualquiera de los demás tipos de monitorizaciones.

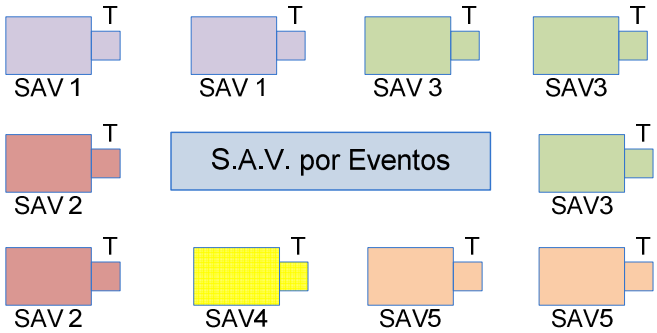

Figura 5. 22 Esquema de funcionamiento distribuido del software

5.4- Registro de eventos y captación de pruebas

En el siguiente punto se detalla el registro de eventos por parte del software y la realización del proceso de recogida de pruebas en éste.

5.4.1- Registro de eventos

El software posee una ventana de registro (*Ventana de Log*) donde quedan registradas de manera interna las acciones y alertas que presenta el software con el objetivo de que el usuario se pueda percatar de operaciones que no se han llegado a producir o simplemente en *echo,* si éstas transcurren correctamente. No obstante existen varias operaciones que se consideran críticas debido a los resultados que pueden generar en el funcionamiento del software. Así, en la base de datos existente, donde se almacena la información de las cámaras y los usuarios, se decidió añadir un componente más que recogiera las alertas de los usuarios en las operaciones críticas (en la administración del sistema) y de las entradas y salidas de objetos en los campos de visualización de las cámaras.

El formato de las alertas recoge los siguientes datos: fecha y hora en la que se produce, el actor que ejecuta una operación o se ve afectado por un evento y una descripción de la alerta. Dentro de las operaciones de administración de S.A.V. las operaciones que se registran son:

- *Añadir* o *Eliminar* una cámara, ya que modifica el sistema.
- *Guardar* o *Borrar* el sistema actual y *Cargar* un sistema, ya que son operaciones que varían la configuración de cámaras contempladas por el software.
- *Gestionar* la BBDD, ya que se incluyen operaciones que pueden afectar a las cámaras existentes y/o los usuarios del software.

5.4.2- Captación de pruebas

A la hora de monitorizar un sistema y dependiendo del tipo de configuración se vaya a ejecutar el usuario puede activar procesos que permiten obtener pruebas más relevantes que un simple registro en la base de datos de intrusismo en una zona vigilada. Dichas pruebas pueden tener dos formatos, en vídeo si lo que se monitoriza es una sola cámara o una tanda de imágenes de la cámara que ha recogido el evento.

La primera técnica es bastante sencilla ya que el único esfuerzo que se realiza es montar un vídeo con las imágenes que se van obteniendo en la monitorización de la cámara. La segunda de las técnicas se realiza mediante una cola circular donde se van almacenando

hasta *n* imágenes de cada cámara durante la monitorización de un sistema y en el transcurso de una alerta de entrada de un individuo en una zona vigilada se genera una serie de hasta *n* imágenes de la cámara donde se ha captado la anomalía.

Para poder acceder a estas pruebas el usuario debe de situarse en la carpeta donde tiene instalado el software e ir a la carpeta de Vid_pruebas o Img_pruebas y dentro de ésta a la carpeta con la fecha de los eventos a visualizar. En el caso de Vid_pruebas encontraremos el vídeo de las imágenes captadas por la cámara seleccionada, mientras que en la carpeta Img_pruebas se recogerán pruebas siempre y cuando se haya producido una detección en el sistema.

5.5- Sistema de gestión del software

Inicialmente se consideró que software fuese gestionado totalmente por fichero pero hubo que reconsiderarlo para facilitar el mantenimiento de los sistemas, por lo que se añadió una base de datos que complementase al software. En dicha base de datos se encuentra centralizada la información de todos los usuarios y las cámaras posibilitándose que el software pueda actuar de forma distribuida y compartir la misma información en cada instante con todos los S.A.V. activos.

La gestión por ficheros del software se ha reservado para los parámetros locales de éste y por lo tanto son optativos. El software está pensado para manejar un fichero de configuración que evite que el usuario este definiendo constantemente los parámetros cada vez que se inicie éste. Además, desde el software se pueden crear y cargar ficheros que definan un sistema de cámaras que se cargará desde la base de datos y un fichero que represente la interconexión de las cámaras (plano). Una vez configurado el software para que se cargue una configuración y un sistema de cámaras, se puede monitorizar dicho sistema desde el inicio del software sin realizar ninguna gestión previa.

6- Resultados.

El objetivo de este proyecto se planteó desde el punto de vista de aportar una solución accesible a todas aquellas personas que necesiten un software sencillo y que además funcionase con una infraestructura extensa, barata y de fácil alcance. Debido a esto hay que tener en cuenta el nivel de exigencia que se desea y cuál es capaz de ofrecer el software.

6.1- Ámbito de utilización del proyecto

El software actualmente funciona al 100% de sus posibilidades bajo el sistema operativo Windows ya que el acceso a las cámaras IP sólo se encuentra implementado bajo esta arquitectura, no obstante el resto de las funcionalidades del software es posible ejecutarlas bajo cualquier otro sistema operativo. El software está centralizado desde una base de datos PostgreSQL donde se almacena la información de los usuarios, cámaras y los log de la aplicación por lo que es necesario que éstas existan y sean accesibles durante el funcionamiento del software. El desarrollo del software se ha realizado casi en su totalidad bajo software abierto y multiplataforma teniendo en cuenta la salvedad de la adquisición de imágenes IP que se ha realizado con el FrameWork de Visual Studio herramienta de acceso gratuito al disponer de una licencia de Windows.

Al realizar las interfaces en todo momento se ha pensado en el usuario final donde los conocimientos para poner en marcha la aplicación, una vez configurada, son nulos ya que basta con apretar un botón. A la hora de realizar la configuración del software ya se requiere un poco más de pericia dependiendo del nivel de privilegios del que disponga el usuario. Comenzando desde el caso más básico al actuar como administrador con privilegios de vigilante, en el que los acceso a los menús son muy intuitivos y guiados, a actuar como administrador del sistema o S.A.V. donde se tiene acceso operaciones tan complejas como la administración de la bases de datos. No obstante, junto al software se incluye un manual de uso con una guía visual del funcionamiento de la aplicación. Además, el software desarrollado es una aplicación libre y abierta en su totalidad para todos aquellos usuarios que deseen mejorarla, por ello también se adjunta la guía de programación en el DVD.

6.2- Análisis de recursos al ejecutar el software

Para determinar los recursos del sistema (memoria y CPU) que son requeridos por el software se sometió a éste a distintas situaciones de estrés. Recordar que las pruebas se han

realizado en un PC Intel(R) Core(TM)2 CPU T7200 @ 2.00GHz con 2.00 GB RAM y los resultados obtenidos son los siguientes:

- **Prueba 1:** Realizar operaciones de administración.
    - el uso de CPU ronda el 1-3 %.
    - el consumo de memoria está entre los 15-20.000KB.
- **Prueba 2:** Monitorización manual de una cámara
    - el uso de CPU ronda el 1-3 %.
    - el consumo de memoria está entre los 18.000KB.
- **Prueba 3:** Monitorizar dos cámaras IP de manera automática con el algoritmo GMM+HMI.
    - el uso de CPU ronda el 5%.
    - el consumo de memoria está entre los 62.000KB.
- **Prueba 4:** Monitorizar dos cámaras IP de manera automática con el algoritmo Mov. Promedio.
    - el uso de CPU ronda el 3%.
    - el consumo de memoria está entre los 38.000KB.
- **Prueba 5:** Monitorizar dos cámaras IP de manera automática con el algoritmo GMM+Contorno
    - el uso de CPU ronda el 3%
    - el consumo de memoria está entre los 60.000KB.
- **Prueba 6:** Simular una monitorización de un sistema de 3 vídeos con el algoritmo GMM+HMI.
    - el uso de CPU ronda el 50 %.
    - el consumo de memoria está entre los 93.000KB de media.
- **Prueba 7:** Simular una monitorización de un sistema de 3 vídeos con el algoritmo Mov. Promedio.
    - el uso de CPU ronda el 32 %.
    - el consumo de memoria está sobre los 42.000KB de media.

- **Prueba 8:** Simular una monitorización de un sistema de 3 vídeos con el algoritmo GMM+Contorno.
    - el uso de CPU ronda el 50 %.
    - el consumo de memoria está sobre los 88.000KB de media.
- **Prueba 9:** Visualizar simultáneamente todas las cámaras (2 IP).
    - el uso de CPU ronda el 1-3 %.
    - el consumo de memoria está entre los 30.000KB.

Las siguientes Figuras 6.1 y 6.2 muestran diagramas de barras en los que se representan los resultados de los picos de consumo de CPU y memoria respectivamente.

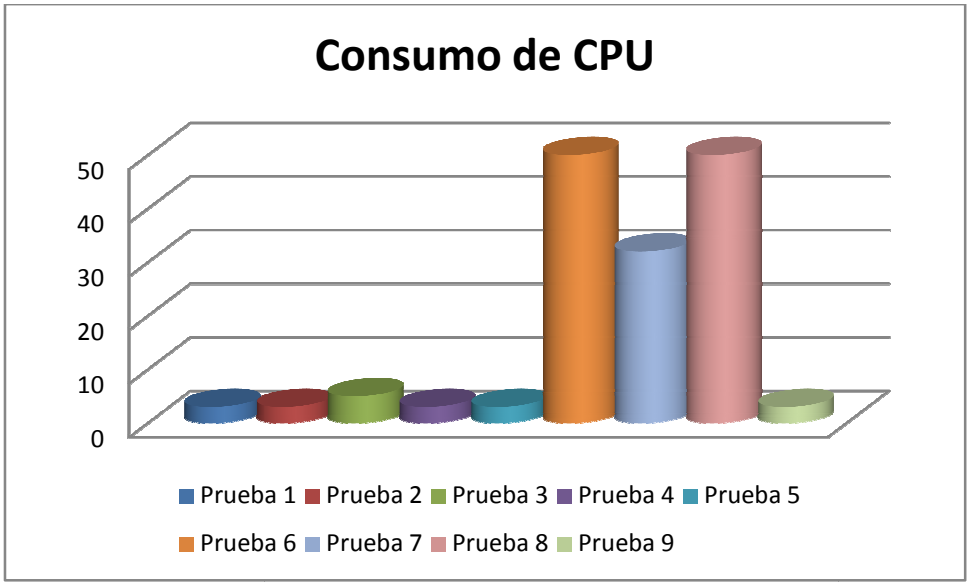

Figura 6. 1 Diagrama del consumo de CPU

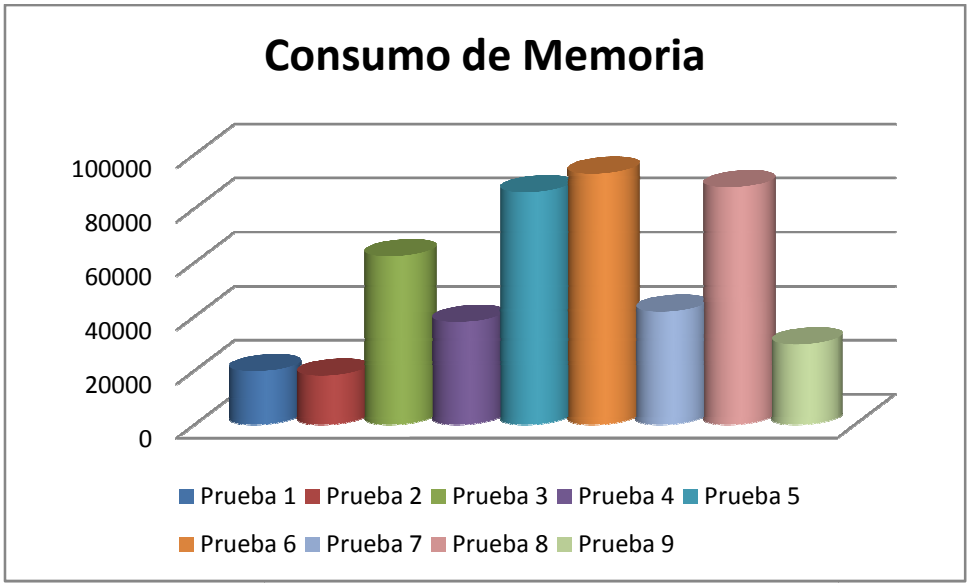

Figura 6. 2 Diagrama del consumo de memoria

6.3- Validación de los resultados

En este punto se llevarán a cabo estudios para determinar la eficiencia del software y del comportamiento de los componentes de éste.

6.3.1- Estudio de la capacidad de procesamiento de imágenes

Este estudio tiene por objetivo concluir el número de imágenes que el software puede procesar de forma simultánea en tiempo real, entendido como tiempo real el análisis por parte del software de todos los cuadros que componen una fracción de vídeo y que éste se realice en como máximo en dicha fracción de tiempo. Es decir, que si se analiza una fracción de un segundo de un vídeo compuesto por diez imágenes y el software lleva a cabo el análisis de las diez imágenes en un segundo o menos se considera que el software está trabajando en tiempo real.

Para poder evaluar el rendimiento del software se calculó el tiempo de procesamiento de un segundo de vídeo con uno de los algoritmos más exigentes (*GMM+HMI*). En el estudio se utilizó vídeos compuestos por imágenes de 480x272 píxeles, tamaño medio de las imágenes ofrecidas por las cámaras usadas en el proyecto, para posteriormente analizar la frecuencia de cuadros que deben presentar los componentes de adquisición (cámaras, vídeos, etc.). En la Tabla 6.1 se muestra la primera comparativa que sirvió para determinar el número de imágenes que se podían analizar, según las características de los vídeos ya comentada. Para ello se evaluó al software con un mismo vídeo a diferentes cuadros por segundos, a 30cps, 25cps y 10cps.

Configuración	Tiempo de procesamiento
Vídeo a 30cps	1.4 s
Vídeo a 24cps	1.2 s
Vídeo a 10cps	0.5 s

Tabla 6. 1 Tiempos de procesamiento del análisis de un segundo de vídeo según los cuadros/segundos

De este estudio se determinó que el software era capaz de analizar unas 20 imágenes por segundo (50 ms/imagen) de un solo dispositivo en tiempo real y del análisis del mercado se concluyó que de diez a quince cuadros por segundos es la velocidad media para la adquisición de imágenes. Por lo que se decidió optar como frecuencia de referencia para los dispositivos de adquisición 10 cps.

Una vez conocido el tiempo de procesamiento para un dispositivo medio, se realizó el siguiente estudio para determinar el número de dispositivos que se podían atender de forma simultánea en el ordenador de referencia. Para ello se analizó durante un segundo distintas configuraciones de sistemas, sometiéndose al software al análisis de sistemas

compuesto por el mismo vídeo en 1, 2, 3 y 4 veces, obteniéndose los resultados que se muestran en la Tabla 6.2.

Configuración del sistema	Tiempo de procesamiento
1 Vídeo a 10cps	0.5 s
2 Vídeos a 10cps	0.8 s
3 Vídeos a 10cps	1.1 s
4 Vídeos a 10cps	1.4 s

Tabla 6. 2 Tiempos de procesamiento del análisis de un segundo de vídeo variando el número de vídeos

De los datos obtenidos en este estudio destacamos que el punto de saturación, para que el comportamiento del software en el análisis de los distintos sistemas se considere en tiempo real en la máquina de referencia, aparece al utilizar más de tres dispositivos bajo las condiciones usadas en el estudio: vídeos con frecuencia de 10cps y 480x272píxeles, y analizados con el algoritmo *GMM+HMI*. Cualquier variación de algunos de los parámetros ocasionará distintos resultados por lo que las conclusiones de este estudio sólo pretenden ser orientativas.

6.3.2- Estudio de la eficiencia de los algoritmos de detección

En este estudio someteremos a los diferentes algoritmos de detección que se incluyen en el software a una fase de análisis exhaustivo sobre cinco escenarios con diferentes configuraciones. Se expondrán los resultados obtenidos para cada escenario y algoritmo con los que se determinará de forma empírica cuál es el más conveniente en cada caso. Antes de adentrarnos en el análisis de los resultados hay que comentar algunos de los aspectos a tener en cuenta en los resultados del procesamiento de los vídeos. Otros puntos a considerar son las situaciones donde se produce algún solapamiento u ocultaciones de individuos en la escena, ya que el software no es capaz de diferenciar entre ambos individuos dando como resultado la detección de un único individuo. También se debe considerar que en la implementación del algoritmo de Viola-Jones sólo se encuentra la detección del patrón cabeza-hombro en su representación frontal por lo que su uso no es recomendable en aquellas cámaras cuya orientación no sea frontal.

Además, hay que tener en cuenta las limitaciones del campo de acción de los algoritmos a la hora de detectar objetos en una escena, ya que a diferencia del ojo humano los algoritmos están pensados para detectar objetos de un tamaño mínimo, aquellos que estén en un campo de acción de alrededor de 5-7m., en los escenarios que se analizarán a continuación este límite estará marcado en la zona de los 7m. aproximadamente, por lo que todos aquellos individuos monitorizados que superen este perímetro empezarán a no ser detectados por el software. Por último comentar que la comparativa de los resultados de la

detección de individuos de los algoritmos frente a los resultados obtenidos por un observador humano se ha realizado de manera completa y exhaustiva al analizarse en ambos casos cuadro por cuadro.

- **Vídeo 1**

El primero de los escenarios que se analiza se puede encontrar en la ruta *Códigos\\Escenarios\\Vídeos\\Vacio\\cam3.mp4* del DVD. En él se monitoriza la escena mostrada en la Figura 6.3, a lo largo de aproximadamente un minuto.

Figura 6. 3 Escenario 1

Durante todo este tiempo no se producirá actividad en el escenario es decir, ninguna persona entraré en la escena, lo que nos servirá para evaluar el comportamiento de los algoritmos frente a este tipo de situación. Además, cabe destacar que se trata de un entorno no controlado en cuanto a iluminación ya que combina iluminación solar y artificial, originándose reflejos en paredes y suelo dificultándose la labor de detección.

Datos	V	A1	A2	A3	A4	A5	A6
Detecciones positivas	0	0	0	0	0	0	0
Detecciones negativas	613	613	613	613	613	613	613
Falsas detecciones positivas	-	0	0	0	0	0	0
Falsas detecciones negativas	-	0	0	0	0	0	0
Tiempo de procesado medio (ms)	-	55	63	7	12	25	32

Tabla 6. 3 Datos del escenario 1

En la Tabla 6.3 se muestran los resultados obtenidos por el vigilante y los algoritmos tras el análisis de la secuencia de vídeo donde se analizaron 613 imágenes. Donde ***V*** hace referencia a los datos del vigilante, ***A1*** a los resultados del algoritmo de GMM+HMI, ***A2*** a la versión de éste algoritmo con Viola-Jones , ***A3*** y ***A4*** son los datos del algoritmo Mov. Promedio sin y con Viola-Jones respectivamente y, ***A5*** y ***A6*** son los datos obtenidos y tras aplicar el algoritmo de GMM+Contorno en sus dos modalidades (sin y con Viola-Jones).

En esta tabla se contabilizará el número de detecciones positivas y negativas, las falsas detecciones positivas y negativas, y el tiempo de procesamiento medio del análisis de las imágenes de la secuencia en milisegundos.

De este escenario hay poco que vislumbrar ya que como se esperaba ante un escenario donde no se presenta actividad, la respuesta de los algoritmos es nula. Con ello probamos que la configuración genérica de los algoritmos es correcta, al menos para las condiciones de este escenario, puesto que no provoca falsos positivos.

- **Vídeo 2**

Este segundo escenario lo podemos encontrar en la siguiente ruta dentro del DVD *Códigos\\Escenarios\\Vídeos\\D\\cam1.mp4*. En el vídeo se monitoriza la escena de la Figura 6.4 a lo largo de aproximadamente un minuto, en la que observaremos la entrada de un usuario en la escena sobre el segundo trece dirigiéndose a la primera puerta que se localiza en la pared blanca, alcanzando ésta en cuatro segundos y ocultándose durante dieciséis segundos tras ella para volver a aparecen en el escenario monitorizado durante otros cinco segundos.

Figura 6. 4 Escenario 2

Datos	V	A1	A2	A3	A4	A5	A6
Detecciones positivas	91	91	91	53	53	86	86
Detecciones negativas	512	512	512	550	550	517	517
Falsas detecciones positivas	-	0	0	2	0	1	0
Falsas detecciones negativas	-	0	0	38	38	6	6
Tiempo de procesado medio (ms)	-	58	68	6	12	27	33

Tabla 6. 4 Datos del escenario 2

En la Tabla 6.5 se muestran los resultados obtenidos por el vigilante y los algoritmos tras el análisis de la secuencia de vídeo donde se analizaron 603 imágenes. Hay que recordar que cada secuencia de 1 segundo de vídeo está compuesta por 10 imágenes, es decir se analizan imágenes cada 100 ms. A partir de estos datos se han realizado una

serie de gráficas que representan las secuencias de detecciones de los tres mejores algoritmos frente a la obtenida por un vigilante humano con una fiabilidad del 100%. Los algoritmos se han clasificado atendiendo a la cantidad de detecciones positivas obtenidas, menor número de falsas detecciones y como último discriminante se ha tomado el tiempo de procesamiento promedio.

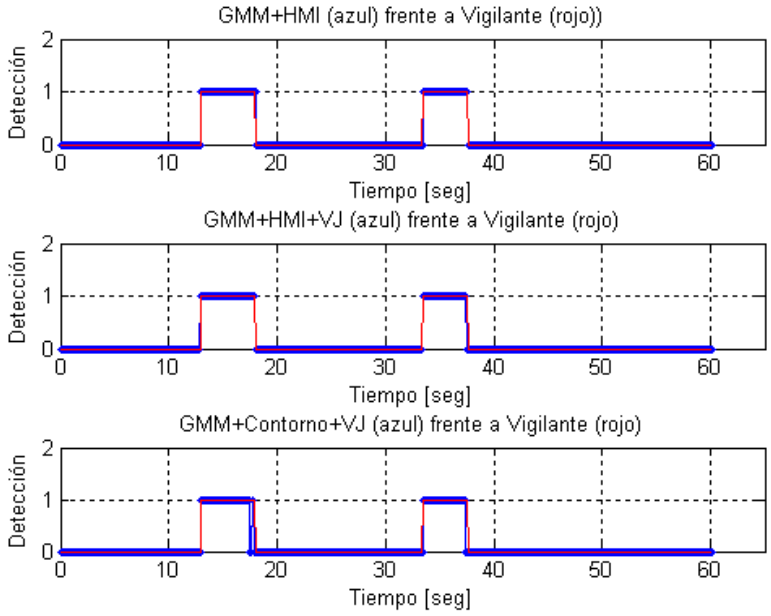

Figura 6. 5 Los 3 mejores algoritmos para el escenario 2

La figura 6.5, como ya se ha comentado, presenta la comparativa de los tres mejores algoritmos que para este escenario son GMM+HMI (A1), GMM+HMI+VJ (A2) y GMM+Contorno+VJ (A6). En el caso de los dos primeros algoritmos las detecciones conseguidas por ambos son exactamente iguales a la obtenida por el vigilante humano. Este resultado es difícil de repetir aun siendo un escenario propicio, puesto que se deben de dar una serie de condiciones, como que el modelado del fondo sea lo suficientemente estable en los cambios de luz y que las detecciones parciales de los sujetos sean suficientes para considerar una detección de una persona, aspectos que se han dado en esta ocasión. Mientras que el tercer algoritmo, si lo estudiamos a lo largo del tiempo, presenta una detección casi constante sufriendo los mayores altibajos cuando el individuo se oculta por primera vez (tras una puerta) provocando un cambio brusco del fondo. No obstante numéricamente la tasa de acierto que presenta es superior al 94% de acierto.

El algoritmo que no ha conseguido colocarse entre los tres primeros es el Mov. Promedio y su variante, se trata del algoritmo más ligero y para este escenario presenta

una tasa de acierto del 60%, no obstante las detecciones a lo largo del tiempo no son malas y permiten detectar al individuo durante la secuencia, los fallos son aislados y en la mayoría de las ocasiones no superan el medio segundo. Otro de los aspectos a destacar en el análisis de este escenario es la aportación que realiza Viola-Jones a la hora de reducir el número de falsos positivos que sufren los algoritmos (en un 100% para GMM+Contorno y en un 50% para Mov. Promedio) a costa de aumentar ligeramente el tiempo de computación de éstos (~8 ms en el ordenador de referencia).

En términos generales si observamos las detecciones proporcionadas por los algoritmos podremos comprobar que se ajusta a la escena descrita inicialmente y en un alto porcentaje a la obtenida por un vigilante ideal. Si clasificamos todos los algoritmos por su rendimiento, es decir atendiendo al número de detecciones positivas, menor número de falsos negativos, menor número de falsos positivos y al tiempo de procesamiento de la secuencia, la lista quedaría de la siguiente manera:

$$A1 > A2 > A6 > A5 > A3 > A2$$

- **Vídeo 3**

Este tercer escenario se trata de la continuación de la secuencia anterior y lo podremos encontrar en la ruta *Códigos\\Escenarios\\Vídeos\\D\\cam2.mp4*. En el vídeo se monitoriza la siguiente escena durante aproximadamente un minuto, en ella observaremos la entrada de un usuario en la escena en el trigésimo noveno segundo dirigiéndose a las escaleras y bajando éstas en apenas cinco segundos para posteriormente desaparecer del plano de monitorización de la cámara. En este escenario se producirán ocultamientos parciales del sujeto que los algoritmos no son capaces de identificar.

Figura 6. 6 Escenario 3

En la Tabla 6.6 se muestran los resultados obtenidos tras el análisis de 606 imágenes correspondientes al escenario de la Figura 6.6. A partir de los resultados obtenidos se

han representado y comparado las secuencias de las detecciones de los tres mejores algoritmos frente a las de un vigilante ideal, Figura 6.7.

Datos	V	A1	A2	A3	A4	A5	A6
Detecciones positivas	62	57	58	48	48	58	58
Detecciones negativas	544	549	548	558	558	548	548
Falsas detecciones positivas	-	1	2	0	0	5	0
Falsas detecciones negativas	-	5	4	14	14	4	4
Tiempo de procesado medio (ms)	-	61	66	9	15	29	35

Tabla 6. 5 Datos del escenario 3

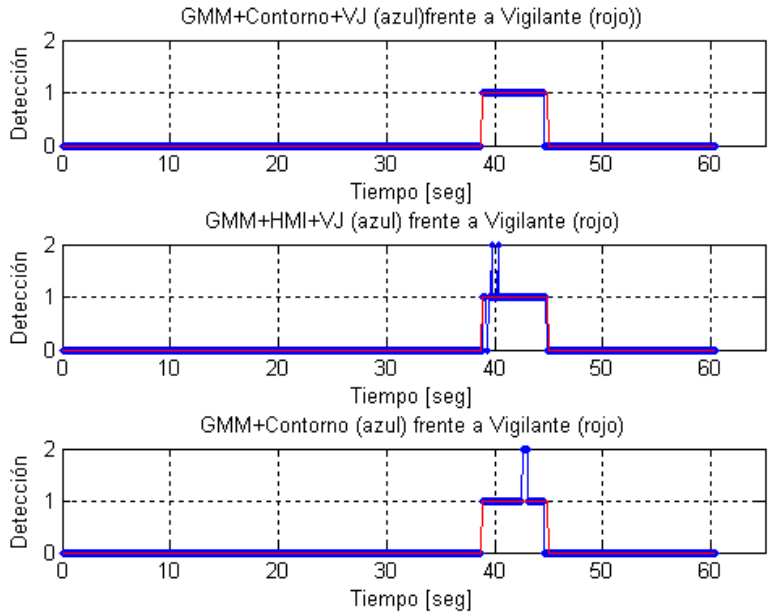

Figura 6. 7 Los 3 mejores algoritmos para el escenario 3

La tres mejores algoritmos para este escenario son, GMM+Contorno+VJ (A6), GMM+HMI+VJ (A2) y GMM+Contorno (A5). Como se ha comentado en la introducción del escenario, el aspecto más interesante en el estudio de éste es el comportamiento de los algoritmos frente a las ocultaciones parciales de los individuos. En este aspecto los algoritmos más afectados por esta problemática son los que mejores resultados presentan, ya que sus procesados y modelados del fondo son más precisos y laboriosos. Como en el anterior escenario la inclusión de Viola-Jones al procesamiento de los algoritmos elimina los falsos positivos salvo para el algoritmo GMM+HMI donde no los disminuye sino que los aumenta en uno.

La tasa de acierto de los tres mejores algoritmos es superior al 93% por lo que dicho escenario es propicio para ellos y una vez más el algoritmo más ligero es el que mayor tasa de fallos presenta (77%) no obstante la mayoría de ellos son esporádicos y nunca superan el medio segundo. En general el comportamiento de los algoritmos se ajusta en gran

medida a la situación descrita en la introducción del escenario y si listamos todos los algoritmos por su rendimiento quedarían de la siguiente manera:

$$A6 > A1 > A2 > A5 > A3 > A4$$

Esta clasificación ha variado en comparación con el anterior escenario, aunque con resultados muy ajustados, por lo que se concluye que para condiciones distintas los algoritmos presentan diferentes comportamientos y eficiencias.

- **Vídeo 4**

El cuarto escenario del análisis al vídeo de la ruta *Códigos\\Escenarios\\Vídeos\\2P\\cam4.mp4*, en el que se monitoriza la escena de la Figura 6.8 durante aproximadamente un minuto y en la que se observa la entrada de dos individuos de manera progresiva. El primero de ellos aparece en escena en el segundo treinta y ocho y el segundo en el cuarenta y ocho siguiendo la ruta mostrada en la Figura 6.8. Con este escenario podemos evaluar el rango de acción de los algoritmos a la hora de detectar a los intrusos y ver cómo afectan a los resultados las pequeñas perturbaciones iniciales (movimientos de la cámara en los primeros segundos). Para la evaluación de este experimento se han tenido en cuenta todos resultados y la marca del límite de detección es sólo orientativa (situada a ~7 m lineales de la cámara).

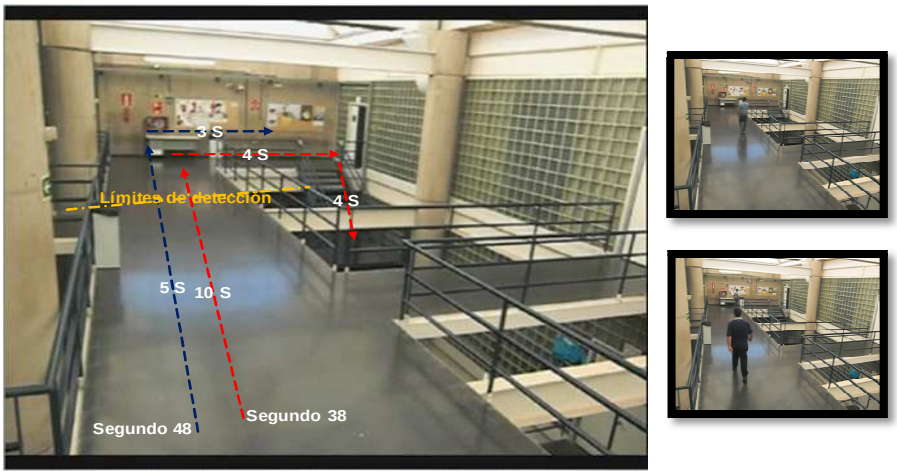

Figura 6. 8 Escenario 4

En la Tabla 6.7 se muestran los resultados obtenidos por el vigilante y los algoritmos tras el análisis de las 615 imágenes que componen la secuencia.

Datos	V	A1	A2	A3	A4	A5	A6
Detecciones positivas	235	72	73	119	120	71	72
Detecciones negativas	380	543	542	496	495	544	543
Falsas detecciones positivas	-	4	4	9	6	5	5
Falsas detecciones negativas	-	163	162	116	115	164	163
Tiempo de procesado medio (ms)	-	68	75	7	15	31	41

Tabla 6. 6 Datos del escenario 4

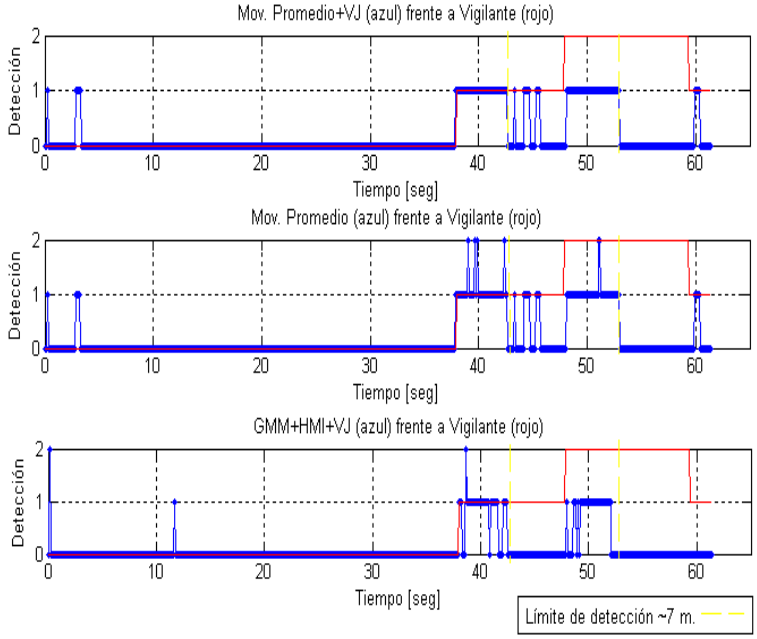

Figura 6. 9 Los 3 mejores algoritmos para el escenario 4

En la Figura 6.9 se muestran los resultados de la comparativa de los tres mejores algoritmos con respecto a un vigilante ideal, en esta ocasión los tres mejores algoritmos son: Mov. Promedio+VJ (A4), Mov. Promedio (A3) y GMM+HMI+VJ (A2). El escenario monitorizado en este estudio es sin duda el más complejo, junto a las perturbaciones iniciales y a la monitorización fuera de rango ya comentadas, la composición de luces del escenario es un factor a considerar, ya que se mezclan gran cantidad de luz natural y artificial provocando sombras en el escenario que dificultan aun más el reconocimiento de individuos.

En cuanto a los resultados que se muestran en la Figura 6.9 cabe destacar que sea el algoritmo más simple y ligero el que mejor resultado ofrezca, presentando rendimientos tan rotundos como una tasa de acierto un 60% mejor que el resto de los algoritmos, aunque en su contra hay que destacar que es el algoritmo que más falsos positivos presenta. Su rendimiento es debido en parte a su mayor rango de detección y a que el análisis que realiza este algoritmo es más simple y por lo tanto menos sensible a las condiciones del escenario.

No obstante el análisis de esta secuencia se debe de realizar en dos pasos, estudiándose los resultados de los algoritmos antes de superar el límite de detección y tras superar éste. Antes de superar el límite de detección se puede observar como en los primeros segundos todos los algoritmos presentan falsos positivos debido a los movimientos que sufrió la cámara a la hora de grabar la secuencia y en cuanto a las detecciones de los individuos

dentro de la zona de monitorización aceptada, los algoritmos presentan una tasa de acierto que ronda el 80% con pequeñas pérdidas del individuo inferiores al medio segundo.

Después del límite de 7 metros el estudio de las detecciones para todos los algoritmos es nulo salvo para el Mov. Promedio que presenta pequeñas detecciones esporádicas, que como se puede observar en la figura no ocurren para los dos individuos por igual, hecho que se puede achacar al contraste más favorable que ofrece el primer individuo con respecto al fondo.

Una vez más añadir Viola-Jones al procesamiento reduce los falsos positivos correspondientes a las detecciones múltiples de un mismo individuo, siendo inevitable los falsos positivos obtenidos al inicio. En esta ocasión el algoritmo que no se encuentra dentro de los tres mejores es el algoritmo GMM+Contorno debido al método de valoración de los algoritmos ya que si observamos sus resultados en la Tabla 6.7 son muy similares a los aportado por el método GMM+HMI.

Concluyéndose que este escenario deja a la vista la principal debilidad de los algoritmos de detección, el rango de acción. Por lo tanto la efectividad de los algoritmos para la monitorización de zonas amplias baja considerablemente con respecto a los otros escenarios. Quedando la clasificación general de la siguiente manera:

A4 > A3 >A2> A1 > A5 > A6

- **Vídeo 5**

En el último de los escenarios, cuya ruta en el DVD es *Códigos\\Escenarios\\Vídeos\\S\\S1.mp4*, se monitoriza la escena de la Figura 6.10 alrededor de un minuto. En la escena monitoriza aparecen dos individuos en direcciones opuestas de manera no coincidentes, el primero de ellos lo hace sobre el segundo diecinueve siguiendo una trayectoria rectilínea desde la puerta de cristal. El segundo individuo aparece alrededor del segundo cuarenta realizando el camino inverso al primer sujeto, es decir saliendo desde el pasillo en dirección a la puerta de cristal y una vez alcanza ésta retorna al plano monitorizado por la cámara. En el transcurso de su trayectoria este sujeto se dedica a realizar movimientos más complejos que los caminares monitorizados en los anteriores escenarios (hace giros a la vez que camina, se coloca de perfil y realiza una pequeña coreografía al final de la secuencia).

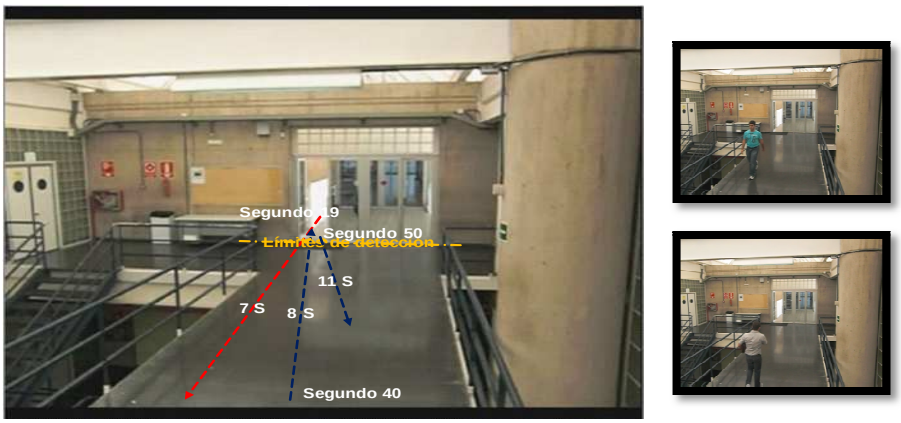

Figura 6. 10 Escenario 5

Del análisis de las 612 imágenes que componen la secuencia de vídeo se han obtenido los resultados que se muestran en la Tabla 6.8.

Datos	V	A1	A2	A3	A4	A5	A6
Detecciones positivas	303	232	232	213	213	240	240
Detecciones negativas	309	380	388	399	399	372	372
Falsas detecciones positivas	-	0	0	0	0	2	0
Falsas detecciones negativas	-	71	71	90	90	63	63
Tiempo de procesado medio (ms)	-	57	65	7	14	26	35

Tabla 6. 7 Datos del escenario 5

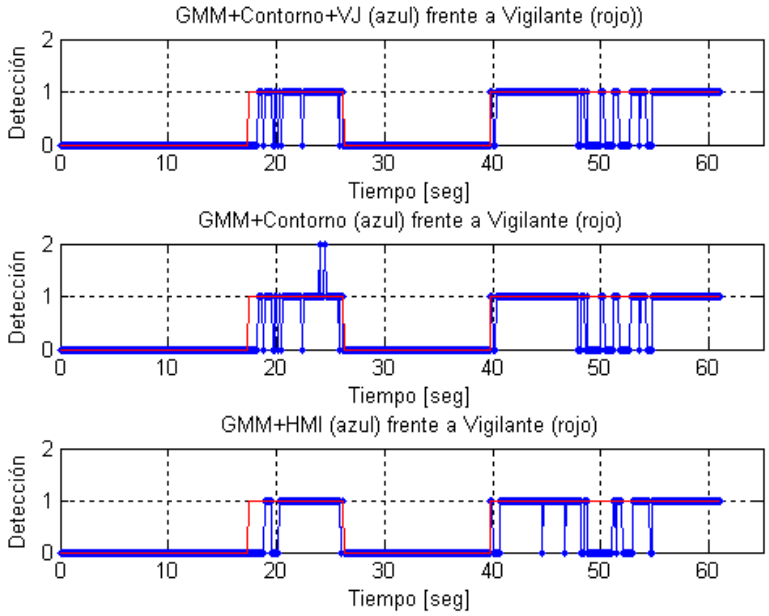

Figura 6. 11 Los 3 mejores algoritmos para el escenario 5

Como se ha hecho en los demás escenarios, en la Figura 6.11 se representan los resultados obtenidos por los tres mejores algoritmos, que para este escenario son GMM+Contotno+VJ (A6), GMM+Contotno (A5) y GMM+HMI (A1), frente a los

resultados presentados por un vigilante ideal. Se trata de un escenario bastante luminoso en el que se presentan dos situaciones difíciles de controlar por los algoritmos, la primera de ellas son las oclusiones de los individuos tras las puertas de cristal, donde un vigilante es capaz de discernir a una persona a través de ellas pero no los algoritmos. Esta situación se da dos veces a lo largo de la secuencia, pudiéndose observar en todos los algoritmos fallos en las detecciones sobre los segundos 18 y 49. La segunda de las situaciones son los movimientos no lineales por parte del segundo sujeto frente a una zona con luz frontal, como se ha comentado el sujeto realiza giros, se coloca de perfil a la cámara y por último realiza una pequeña coreografía en una situación estática, situaciones que producen pequeñas pérdidas en las detecciones del sujeto de alrededor de dos tercios de segundo en todos los algoritmos. No obstante para los algoritmos que están entre los tres mejores la tasa de acierto ronda entre el 80-75%, descendiendo hasta el 70% para los restantes. El algoritmo Mov. Promedio presenta unos resultados bastante intermitente para esta escena, sobre todo con el segundo de los individuos donde constantemente se pierde su detección por cuantos de medio segundo llegando a casi dos al final de la secuencia.

En general la aportación de Viola-Jones para este escenario es nula salvo para el algoritmo GMM+Contorno donde se corrigen los dos falsos positivos. Con este vídeo se ha sometido a los algoritmos a secuencias exigentes, detectándose sujetos en movimientos inusuales, aun así la capacidad de detección se mantiene por encima del 70% y nunca se pierde a los individuos por cuantos superiores a los dos segundos. Quedando para este escenario la clasificación de los algoritmos según su rendimiento de la siguiente manera:

A6 > A4 > A1 > A2 > A3> A4

Conclusiones Generales

Este estudio se ha realizado mediante una selección aleatoria de cinco escenarios que forman parte del conjunto de escenarios desarrollados para la evaluación del software. Las conclusiones que se obtienen de este estudio no deben de ser generalizadas ya que los factores que pueden influir en un entorno monitorizado son infinitos y es imposible realizar pruebas en todos ellos. No obstante para entornos que presenten una configuración similar a la usada por el estudio podremos concluir lo siguiente.

La gráfica de la Figura 6.37 enfrenta los tiempos medios de procesado de los vídeos frente a todos los algoritmos para cada escenario. De donde cabe destacar que los tiempos de respuesta de los algoritmos dependen en un factor de los escenarios que se monitoricen:

- GMM+HMI (A1): entre 55 y 75 ms.

- Mov. Promedio (A3): entre 5 y 15 ms.
- GMM+Controno (A5): entre 25 y 40 ms.

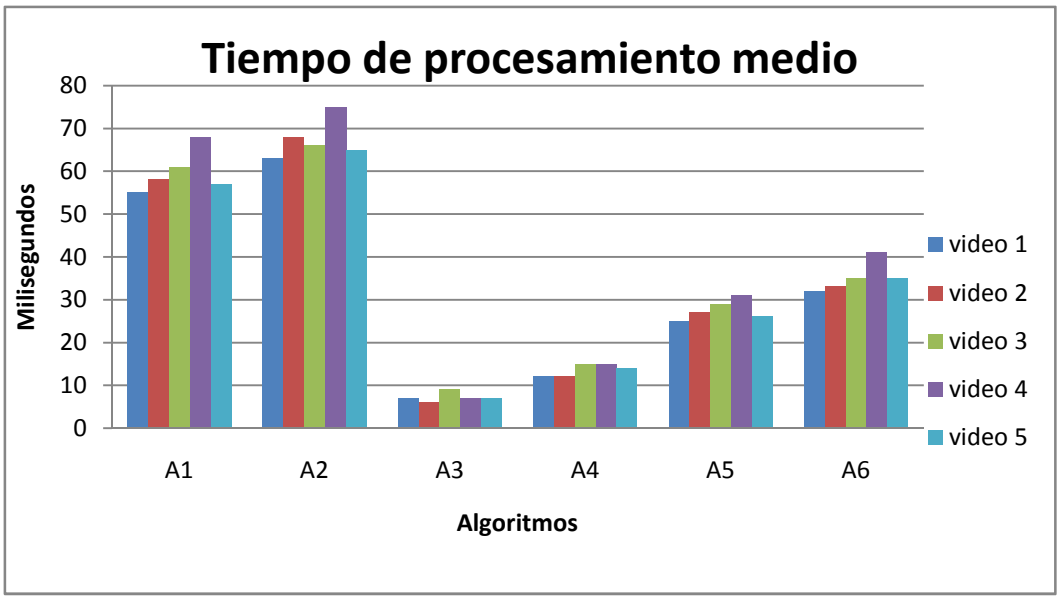

Figura 6. 12 Gráfica del tiempo de procesamiento de los algoritmos

Además de determinar que el uso del método de Viola-Jones añade un coste adicional de entre 7 y 10 ms. Por lo que a la hora de monitorizar un escenario hay que tener en cuenta la composición de éste a la hora de elegir un algoritmo.

A partir de los datos de tiempo de procesamiento y el cálculo de la capacidad de detección de los algoritmos se puede determinar cuál es el mejor algoritmo para cada tipo de escenarios. En la Figura 6.13 se representa el porcentaje de acierto de los algoritmos en cada escenario. Como se puede determinar de los resultados que presenta la gráfica, no existe el algoritmo perfecto para todos los escenarios. Pero en términos generales los algoritmos presentan una tasa de entre un 70 y 80% de aciertos de detecciones. Este hecho puede resultar un poco pobre si se analiza en frío y se comparan con los resultados que tendría un vigilante humano, pero hay que pensar que se ha llegado a los límites en la monitorización de los escenarios, pasando de entornos ideales a otros totalmente extremos, como en el caso del cuarto escenario donde los algoritmos no tienen la suficiente potencialidad para controlar toda la escena. Por lo que los datos que se presentan no son engañosos ni idealistas.

Aun así, hay que considerar estos datos desde otro punto de vista, en este estudio solo se ha analizado las capacidades de análisis sobre un vídeo sin embargo el software es capaz de analizar varios vídeos simultáneamente con idénticos resultados. Además, se elimina la variable del tiempo en el rendimiento del software, ya que siempre será el mismo al

contrario que en los vigilantes humanos donde va decreciendo. Cumpliéndose el principal objetivo del software de vigilancia que es captar las actividades dentro del escenario monitorizado y añadiéndose las actividades de detección y seguimiento propias de los sistemas autónomos.

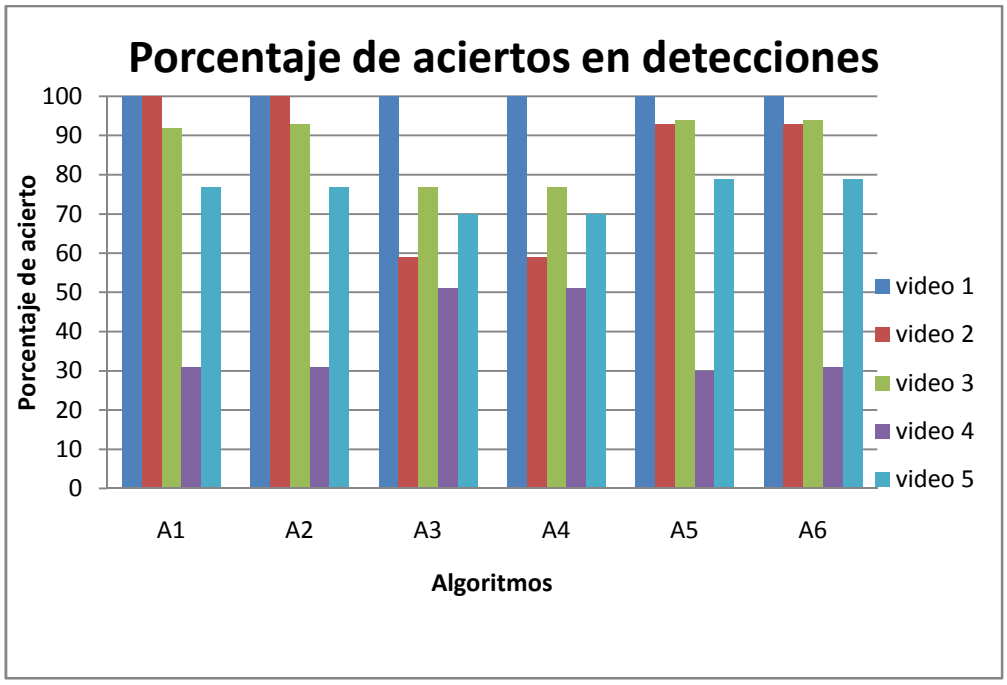

Figura 6. 13 Gráfica con el porcentaje de acierto de los algoritmos

6.4.3- Estudio del comportamiento de los algoritmos de selección

En este estudio se analizarán los comportamientos presentados por los principales algoritmos de selección de cámara al someterlos ante simulaciones de escenarios. Entre ellos se estudiaron el algoritmo rotatorio, por prioridad, no incluyéndose en el estudio la monitorización manual, ya que no aporta nada nuevo y la monitorización clásica puesto que en ella no se realiza procesamiento y tan sólo se limita a visualizar todas las cámaras.

- **Algoritmo de monitorización rotatorio**

El objetivo del algoritmo rotativo es visualizar todo el sistema de cámaras una a una sin necesidad de que el usuario interactúe con el sistema. Al presentar una imagen única permite al usuario centrar toda su atención en un único escenario y al estar constantemente rotando entre los escenarios se consigue romper la monotonía, lográndose mayor atención del usuario. Como contra se puede tener en cuenta lo que se ha considerado una ventaja, que el usuario no ve todo el sistema en todo momento. Pero por ello el software incorpora una opción en la que se pueden visualizar todas las cámaras simultáneamente, además hay que destacar que el algoritmo únicamente está activo cuando no se detecta actividad en ninguna de las cámaras del sistema.

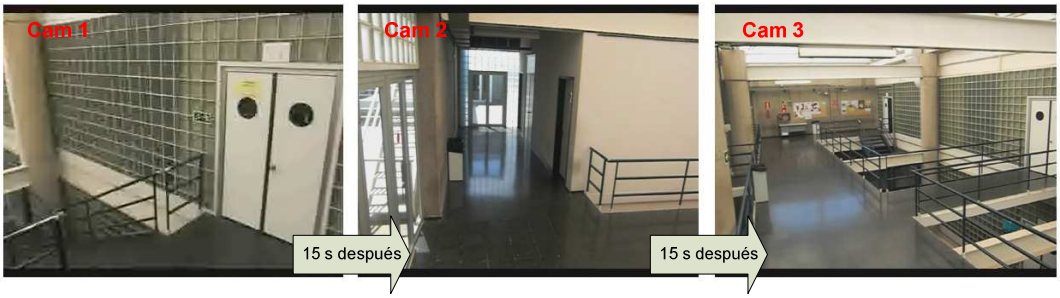

Figura 6. 14 Simulación del algoritmo de monitorización rotatoria

En la Figura 6.14 se muestra un ejemplo real del funcionamiento del algoritmo en el que se monitorizan 3 cámaras sin que se presente actividad alguna en ellas, como se indica en la imagen el cuanto de tiempo para las rotaciones de las cámaras a visualizar se ha fijado en torno a los 15 segundos.

- **Algoritmo de monitorización por prioridad**

El algoritmo de selección por prioridad viene a tratar de poner un poco de control en la monitorización de las cámaras al establecer prioridades entre ellas. Aunque esto puede llegar a ser un tanto anárquico si el administrador no acierta en la configuración de sus prioridades, ya que cuando hay detecciones simultáneas el software monitorizará las cámaras dando saltos entre éstas si ambas poseen la misma prioridad. Si lo miramos por el lado negativo no permite seguir la secuencia por más de un cuanto de tiempo pero, por otro lado se logra llamar la atención del vigilante y que éste no se centre en una sola cámara al mostrarle actividad en varias, dándole la responsabilidad al vigilante de seguir el protocolo de seguridad que mejor se adapte a cada ocasión.

Figura 6. 15 Secuencia de conflicto para el algoritmo de monitorización por prioridad

En la figura 6.15 se muestra una secuencia de conflicto para el algoritmo de monitorización por prioridad, en el estudio que se llevo a cabo se supusieron y probaron los siguientes casos:

- Cam 1 con igual prioridad que Cam2.
- Cam 1 con mayor prioridad y Cam 2 con mayor prioridad.

Para el primer caso donde ambas cámaras poseían la misma prioridad, el algoritmo de selección daba botes entre las cámaras cada 5 segundos siempre y cuando no se produzcan pérdidas del individuo superiores a 1 segundo en alguna de las escenas, ya que en este caso saltaría inmediatamente a la otra cámara repitiéndose este comportamiento mientras las cámaras presenten alertas. Esta situación se dio durante el estudio de este algoritmo, ya que en algunas secuencias la detección de los individuos es intermitente, provocando que la persistencia de la cámara se agotara y se forzara el salto hacia la otra. En el segundo de los casos ante la activación de una cámara con mayor prioridad el algoritmo tiende a centrarse en esta siempre y cuando no se produzcan pérdidas del individuo. Siguiéndose la política de monitorizar la cámara con mayor prioridad y alertar mediante el log del sistema la detección en las otras cámaras.

- **Algoritmo de monitorización por plano**

El algoritmo de selección guiado por plano pretende aportar mayor inteligencia al proceso de monitorización de objetivos en escenarios contiguos. Aunque siempre atendiendo a las necesidades del algoritmo, información de todas las cámaras, una configuración inicial aceptable y sobre todo el ajuste de los tiempos de espera entre las transiciones de las cámaras, ya que se tiene que llegar a un convenio de si ha sido una elección correcta o incorrecta. Para testear este algoritmo se utilizaron los 3 vídeos del escenario de la carpeta *vídeos/D* donde un individuo presenta la siguiente secuencia de detección C1-C1-C2 usándose las configuraciones mostradas en la Figura 6.16 y 6.17.

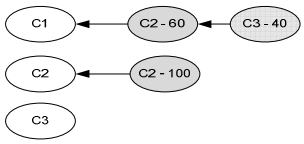

Figura 6. 16 Configuración A

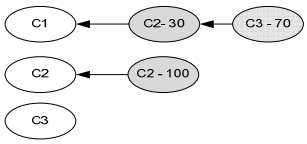

Figura 6. 17 Configuración B

Según la experiencia con este escenario se consideró como cuanto óptimo de espera entre transición de las cámaras cuya separación no supera los 7 metros 3 segundos. Dicho cuanto se respetará siempre y cuando no se produzca ninguna detección en las cámaras vecinas, ya que en caso de que se produzca alguna alerta se monitorizará la cámara responsable. Como se explicó en la presentación del algoritmo, cuando la estimación de la vecina a monitorizar es correcta el porcentaje de ésta se incrementara en una cantidad de 10 reajustándose todos los pesos de los nodos. Esta situación se da en el caso de la

configuración A, mientras que con la configuración B lo que se observa es un salto desde la cámara C1 a la C3 y de la C3 a la C2 sin que llegue a transcurrir los tres segundos.

- **Algoritmo de monitorización por eventos**

La monitorización por evento, es una monitorización sin procesamiento de imagen que se apoya de la información recogida de otros S.A.V. que realizan monitorizaciones simultáneamente con alguno de los métodos comentados con anterioridad de las cámaras que componen el sistema. Sin esta última premisa no tiene ningún sentido usar esta monitorización ya que el resultado que obtendríamos sería una monitorización rotatoria sin análisis de las imágenes, o lo que es lo mismo un sistema de vigilancia actual con la limitación de sólo observar una cámara sino se activa la opción de monitorizar múltiple. Para probar este método lo que se hizo fue simular actividad en la base de datos de algunas de las cámaras que componían el sistema y ver como éstas se activaban. Al realizar una monitorización sin coste de análisis, ni información de las cámaras se presupone que todas las cámaras poseen la misma prioridad y a la hora de atender varias alertas la prioridad será la que primero se haya recogido en la base de datos, cediéndole a dicha cámara un cuanto de monitorización de 3 segundos. Otro de los inconvenientes de esta monitorización es que no se lleva un chequeo de las actividades de las cámaras por lo que es probable que se llegue a monitorizar una cámara que haya presentado una detección previa a su monitorización y a la hora de monitorizarla no haya ningún sujeto en la escena.
Tras comentarse las experiencias con cada algoritmo donde se estudiaron sus fuertes y debilidades se puede concluir que el análisis del comportamiento presentado en este punto puede considerarse simple y artificial, al llevarse a cabo mediante las experiencias obtenidas a partir de simulaciones en vídeos o incluso la edición manual del registro de eventos. Por lo que para poderse presentar unos resultados más serios, sobre todo para los dos últimos algoritmos, monitorización guiada por planos y la monitorización por eventos, haría falta una infraestructura mucho más completa de la que actualmente se dispone y realizar un estudio que abarque un periodo más grande, es por ello que realizar un análisis más profundo es una de las tareas propuesta como posible trabajo futuro.

7- Conclusiones y trabajo futuro.

En el transcurso del proyecto se han estudiado herramientas y técnicas de tratamiento de imagen, de bases de datos, de desarrollo de interfaces, de herramientas de concurrencia, ampliando los conocimientos de programación de objeto en C++ y empleado técnicas de análisis y diseño de ingeniería del software con el propósito de realizar un software para la gestión y monitorización autónoma de sistemas de cámaras en el ámbito de la vídeo vigilancia. Aunque las vías comerciales y de investigación que ofrece este campo son enormes, a día de hoy están por explotar, debido a que la existencia de los sistemas de vídeo vigilancia es de apenas de diez años, y que el hardware para este propósito estaba constantemente evolucionando y eran pocos los que se lo podían permitir. A día de hoy, estas barreras se han empezado a superar gracias en parte al abaratamiento de la tecnología y a la liberación del código propietario, pero sobre todo al aumento de seguridad que se ha demandado en los últimos años, lo que ha posibilitando la inversión de capital para la investigación en este terreno y la aparición de herramientas que permiten el desarrollo de proyectos como éste.

7.1- Conclusiones finales

Como se ha intentado dar a entender S.A.V. está pensado como un software para la monitorización y administración de sistemas de cámaras estacionarias que cubran áreas de tamaño medio (pequeños comercios, almacenes, etc.), atendiendo a un número de cámaras variable según el tipo de monitorización que se realice y al ordenador donde se ejecute la aplicación. Como impacto social cabe destacar que el objetivo de las aplicaciones de monitorización automática no es sustituir a los vigilantes sino facilitarles sus tareas y aumentar su eficiencia y rapidez a la hora de aplicar los protocolos de seguridad.

Tras todos los estudios realizados y hacer una breve reflexión S.A.V. no es un sistema de seguridad pensado para ningún banco o departamento de seguridad nacional, pero desde el inicio de la implementación del software quedó claro que eso no era el propósito de éste.

A título personal estoy bastante satisfecho de los resultados que presenta el software y los conocimientos y retos que he superado en la realización de este proyecto. Y como posible vía de explotación de la aplicación aún queda mucho mercado en el que el software cubre las exigencias de los escenarios, como por ejemplo la tan de moda vídeo vigilancia de hogares, almacenes y pequeños comercios que no se pueden permitir instalar un sistema de vídeo vigilancia de altas prestaciones, donde ya se van requiriendo más que simples sistemas de monitorización. Con este proyecto tan sólo se han establecido las bases para un

software de vídeo vigilancia y a partir de él las posibilidades que admite abarcar son muchas, algunas de ellas se comentan en el siguiente punto.

Para terminar con estas conclusiones quiero instar a todas aquellas personas que le interesen realizar proyectos en el campo de la visión por computación que se tiren a la piscina. He de reconocer que es un campo bastante costoso, cuando como yo no se tenía conocimiento alguno, pero sin duda las posibilidades que abarca este campo bien merecen el sacrificio. Además para todas aquellas personas que no le gustan enfrascarse en el estudio de algo concreto y estancado este es su mundo.

7.2- Trabajo futuro

Las vías de continuidad que abarca este proyecto responden a distintas necesidades. Comenzando por la mejora del software cabe citar los siguientes aspectos:

- Paralelizar más el código.
- Implementar la adquisición de imágenes bajo todas las plataformas.
- Añadir métodos de detección y seguimientos de personas.

Actualmente el software corre con dos hilos (interfaz y procesamiento de imágenes) más los que se puedan lanzar en la visualización sin procesamiento de las cámaras, por lo que sería interesante que el proceso de adquisición de imagen y el sistema de obtención de prueba se realizaran en hilos independientes aumentando el rendimiento de éste. El segundo de los ítems así como el tercero sólo pretenden aumentar el nivel de prestaciones y funcionamiento del software.

En segundo lugar, se plantea añadir nuevas funcionalidades al software que lo complementen, algunas opciones serían:

- Realizar de detección de movimiento con cámaras no estacionarias.
- Añadir al software un planificador de colocación de las cámaras.

La primera opción sería interesante de implementar puesto que requiere realizar un estudio del movimiento de las cámaras que añade bastante complejidad en el software aunque no obstante se puede reducir al caso ideal y representar una cámara móvil como si se tratase de varias (según el nivel de puntos que se visualicen). La segunda idea que se propone presenta una gran complejidad a la hora de realizar la tarea de satisfacción de restricciones puesto que cada escenario es diferente, no obstante siempre hay puntos en común si nos centramos en escenarios interiores.

Por último, en relación con el ámbito investigador se proponen las siguientes sugerencias:

- Realizar estudios más exhaustivos en entornos reales y con red de datos dedicadas exclusivamente para las cámaras IP.

- Realizar técnicas de análisis de comportamiento de los usuarios en los escenarios.

La primera sugerencia de este tercer grupo corresponde a una dimensión del proyecto que no se pudo cubrir en la realización de mi proyecto ya que por falta de infraestructura no se ha podido realizar un estudio a fondo. La segunda idea hace alusión a una de las principales vías de investigación actuales con más salida en el mundo laboral y comercial.

8- Anexo.

a) Detalles sobre la implementación del proyecto

Se recuerda que los detalles de la implementación de todas las clases las puedes encontrar en el manual de programación Doxygen que se incluye en el DVD del proyecto. Y en este apartado se hará hincapié únicamente en los componentes estructurales del proyecto.

La clase que se encarga definir una cámara, no presenta ninguna estructura a destacar, siendo lo único a tener en cuenta la cantidad de atributos necesarios para definir los parámetros de la cámara. Cabe destacara que muchos parámetros pese a ser numéricos están declarados como *string* ya que al adquirir estos desde la base de datos se obtienen en este formato y es a la hora de operar con ellos cuando se hacen las apropiadas transformaciones.

En las clases gestoras se ha hecho uso de la estructura de la *STD deque* que es un template que representa una cola doblemente encadenada. Esta estructura incluye operaciones realizadas de manera eficiente y su uso es recomendado en estructuras donde su tamaño varíe de forma significativa. Por lo que se ha usado para la creación de todas aquellas estructuras de datos en las que se necesitaba almacenar secuencias de estructuras/datos con necesidades de accesos no secuenciales.

Para la estructura de comunicación con la base de datos se hizo uso de la librería de programación con C++ de PostgreSQL "*libpq-fe.h*", en la que se incluyen las dos clases *PGconn* y *PGresult* que permiten acceder y manejar toda la información de estas.

Otro aspecto que quiero destacar es la implementación del mapa topológico de las cámaras se llevo a cabo mediante la implementación de un grafo mediante lista de adyacencia. Esta representación es bastante sencilla pero suficiente para tener una representación simplista del mundo que me permitiera moverme por él.

En cuanto a la estructura para manejar las imágenes se hizo uso de la librería OpenCV que incluye la clase IplImage diseñada para esta tarea.

Por último comentar que la herramienta wxWidgets usada para realizar la interfaz me ha proporcionado todas las funciones y estructuras necesarias:

- Clases y métodos para la creación y manejo de ventanas y diálogos.
- Una clase template con la que implementar mis hilos.
- Así como métodos para dibujar.

Aunque en la mayoría de los casos algo arcaicas y su usos requieren de conversiones entre los datos usados de C++ y las librerías y wxWidgets.

b) Contenidos del DVD.

Dentro del DVD que se adjunta con el PFC podemos encontrar los siguientes contenidos:

- Memoria del proyecto (Formato PDF).
- Manual de usuario (Formato PDF).
- Manual de programación (Formato Doxygen).
- Manual de instalación de las herramientas (Formato PDF).
- Legislación de 8 de noviembre, de la Agencia Española de Protección de Datos (Formato PDF).
- Libros en PDF sobre las herramientas usadas.
- Los programas y librerías usados en el PFC.
- Un proyecto Visual Studio en el que se incluye el código fuente del software.
- El instalador del software.

Glosario.

- **Agente Autónomo.** Un agente autónomo es un sistema anidado y parte integrante de un ambiente (environment) y que detecta o percibe (percepts) datos ambientales, momento a momento, y actúa sobre él con la intención de usar (actions) esos datos para su propia tarea (task) o agenda, afectando así lo que va a detectar en el futuro, sin intervención de terceras partes. En computación, aquellos agentes que actúan de forma automática, siendo capaces de tomar decisiones independientes y realizar acciones para satisfacer objetivos internos basados en el entorno que perciben. Todos los agentes de software en aplicaciones importantes son supervisados bien de cerca por personas que los inician, monitorean y modifican su comportamiento, y los apagan cuando sea necesario.
- **CPS o FPS.** Cuadros/Frames por segundos es la medida que la que indicamos el nivel de composición que posee un vídeo o la capacidad de imágenes que es capaz de transmitir una cámara.
- **Code Book**. Es una técnica de visión por computación basado en operaciones de segmentación de imagen usadas para detectar objetivos y usado en las técnicas de sustracción de fondo.
- **Echo.** Es un comando para la impresión de un texto en pantalla.
- **GMM.** Acrónimo de Gaussian Mixture Model técnica usada para el modelado de fondos.
- **HMI.** Acrónimo de Historial Motion Image técnica para la detección de movimiento de objetos en las escenas.
- **Sistema multiagente o de inteligencia artificial distribuida.** Es una ciencia y una técnica que trata con los sistemas de inteligencia artificial en red. En cierto modo, un sistema multiagente es un sistema distribuido en el cual los nodos o elementos son sistemas de inteligencia artificial, o bien un sistema distribuido donde la conducta combinada de dichos elementos produce un resultado en conjunto inteligente. Hay que notar que los agentes no son necesariamente inteligentes. Existen como en todo el resto del dominio de la inteligencia artificial, dos enfoques para construir sistemas multiagentes: El enfoque formal y el enfoque constructivista.

Bibliografía.

Referencias bibliográficas de libros de texto:

[CL, 2006]

Craig Larman (2006). "UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado".
Editorial Pearson Prentice Hall.

[DD, 2003]

Korry Douglas y Susan Douglas (2003). "PostgreSQL".
Editorial Developer's Library.

[Hartley and Zisserman, 2000]

Hartley, R. and Zisserman, A. (2000). Multiple View Geometry in Computer Vision. Cambridge University Press.

[Javed-Shah, 2008]

Omar Javer y Mubarak Shah. (2008). "Automated Multi-Camera Surveillance".
Editorial Springer.

[Pressman, 1997]

Pressman, R.S. (1997). "Ingeniería del software. Un enfoque práctico".
Editorial McGraw-Hill.

[SCDS, 2000]

S. Sánchez, C. Canal, F. Durán y M. Sicilia. (2000). "El Proceso Unificado de Desarrollo de Software".
Editorial Addison Wesley.

[SHB, 2008]

Milan Sonka,Vaclav Hlavac y Roger Boyle (2008). "Image Processing, Analysis and Machine Vision".
Editorial Thomson.

[Zhang, 1998]

Zhang, Z. (1998). A flexible new technique for camera calibration. Technical Report MSR-TR-98-71, Microsoft Research, Microsoft Corporation, Redmond, WA 98052.

Referencias bibliográficas de pdf(Adjuntos en el DVD):

[PDF, CP]

Cross-Platform GUI Programming with wxWidgets. (Libro en formato digital)

[PDF, Ocv]

OReilly-LearningOpenCV. (Libro en formato digital)

[PDF, OT]

Object Tracking: A Survey. (Articulo de 2006 escrito por: Alper Yilmaz, Omar Javed y Mubarak Shah)

Referencias bibliográficas de páginas web:

[PW, Agpd]

https://www.agpd.es

Página de asociación española de protección de datos.

Consultada en mayo de 2009

[PW, Blob]

wibirama.com/ngaji/data/OpenCV/docs/.../Blob_Tracking_Tests.doc

Diagrama de de los módulos *Blob Tracking*

.Consultada en octubre de 2009

[PW, Cod]

http://www.codeproject.com

Consulta en referencia a la comunicación con cámaras IP y vídeo vigilancia

Consultada entre octubre-noviembre 2008

[PW, Com]

http://es.wikipedia.org/wiki/Command_%28patr%C3%B3n_de_dise%C3%B1o%29

Imagen del patrón Comando.

Consultada entre octubre-noviembre 2008

[PW, Fatih]

http://www.merl.com/papers/docs/TR2003-36.pdf

Human Body Tracking by Adaptive Background Models and Mean-Shift Analysis

Consultada en Abril del 2009.

[PW, IJCV]

http://www.cmucam.org/attachment/wiki/viola-jones/viola-ijcv04.pdf

Robust Real-Time Face Detection

Consultada en Abril del 2009.

[PW, InsPos]

http://www.arpug.com.ar/trac/wiki/PgInstaller

Instalación del PostgreSql

Consultada en Marzo del 2009.

[PW, Kim]

http://www.umiacs.umd.edu/~knkim/paper/Kim-RTI2005-

FinalPublished.pdfImagen del patrón Comando.

Real-time foreground–background segmentation using codebook model

Consultada en febrero 2009

[PW, Msdn]

http://msdn.microsoft.com/es-es/default.aspx

Consulta sobre las claves de Visual Studio 2005

Consultada durante el periodo de desarrollo del proyecto.

[PW, MVC]

http://www.proactiva-calidad.com/java/patrones/mvc.html

Imagen del patrón MVC.

Consultada entre octubre-noviembre 2008

[PW,TutPos]

http://etutorials.org/SQL/Postgresql/Part+II+Programming+with+Postgre

SQL/Chapter+8.+The+PostgreSQL+C+API+-+libpq/Client+3+-

+Simple+Processing+-+PQexec+and+PQprint/

Tutorial de PostgreSql

Consultada en Marzo-Abril del 2009.

[PW, UsoPos]

http://www.postgresql.org/docs/8.3/static/libpq.html

Manual de la librería de comunicación con PostgreSql desde C/C++

Consultada en Marzo del 2009.

[PW, wxWidgets_conf]

http://www.codeproject.com/KB/library/wxwidgets.aspx

Configuración de proyectos con wxWidgets usando Visual Studio 2005

Consultada en diciembre de 2008.

[PW, wxWidgets_install]

http://wiki.wxwidgets.org/Microsoft_Visual_C%2B%2B_Guide

Instalación de wxWidgets usando Visual Studio 2005

Consultada en diciembre de 2008.