

PROYECTO FIN DE CARRERA VVD: Vídeo Vigilancia Distribuida



ugr

Universidad
de **Granada**

Rafael Martos Llaveró

Universidad de Granada

Para los estudios de
INGENIERÍA DE TELECOMUNICACIÓN

2011/2012

El tribunal constituido para la evaluación del proyecto PFC titulado:

VVD: Vídeo Vigilancia Distribuida

Realizado por el alumno: **Rafael Martos Llaveró**

Y dirigido por el tutor: **Juan José Ramos Muñoz**

Ha resuelto asignarle la calificación de:

- SOBRESALIENTE (9 - 10 puntos)
- NOTABLE (7 - 8.9 puntos)
- APROBADO (5 - 6.9 puntos)
- SUSPENSO

Con la nota: puntos.

El Presidente:

El Secretario:

El Vocal:

Granada, a de de 2012.



ugr

Universidad
de **Granada**

ESTUDIOS DE INGENIERÍA DE TELECOMUNICACIÓN

VVD: Vídeo Vigilancia Distribuida

REALIZADO POR:

Rafael Martos Llaveró

DIRIGIDO POR:

Juan José Ramos Muñoz

DEPARTAMENTO:

Teoría de la Señal, Telemática y Comunicaciones

Granada, septiembre de 2012

PALABRAS CLAVE

Vídeo vigilancia, servidores distribuidos, *streaming* de vídeo, redes IP, dispositivos *Android*, cifrado de datos, PTZ, control remoto de cámaras motorizadas

RESUMEN

En nuestros días vivimos bajo infinidad de “ojos” que cuidan de nuestros negocios, de la gente que utiliza servicios públicos, de la guardería de nuestros hijos y/o hijas, etc. El desarrollo de redes IP sobre las que se fundamenta la vídeo vigilancia actual permite pensar en multitud de funcionalidades innovadoras para la interconexión de sedes y puntos remotos. Además, la proliferación de necesidades relacionadas con gestión remota de alarmas, unificación de servicios de vídeo, voz y datos, etc. obligan en gran parte al uso de tecnologías IP que den solución a todo tipo de retos.

Bla, bla, bla ... No pasarse de una cara

KEY WORDS

Video surveillance, distributed servers, streaming video, IP networks, Android devices, data encryption, PTZ, camera control

ABSTRACT

Today we live under numerous "eyes" who care for our businesses, for people who use public services, day care of our children and / or daughters, etc. The early development of IP networks that are based on currently existing video surveillance suggests a multitude of innovative features to interconnect offices and remote locations. In addition, the proliferation of remote management needs related to alarm, unified video services, voice and data, etc.. largely require the use of IP technologies that provide solutions to all kinds of challenges.

Blah, blah, blah ...

D. Juan José Ramoz Muñoz, profesor del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, como director del Proyecto Fin de Carrera de D. Rafael Martos Llaveró

Informa:

que el presente trabajo, titulado:

VVD: Vídeo Vigilancia Distribuida

Ha sido realizado y redactado por el mencionado alumno bajo mi dirección, y con esta fecha autorizo a su presentación.

Granada, a de de 2012.

Fdo. Juan José Ramos Muñoz

Los abajo firmantes autorizan a que la presente copia de Proyecto Fin de Carrera se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultada por las personas que lo deseen.

Granada, a de de 2012.

Fdo. Juan José Ramos Muñoz

Fdo. Rafael Martos Llaveró

Agradecimientos

En primer lugar quisiera agradecer a todos aquellos que estáis utilizando la plantilla y que me enviáis comentarios ya sea para sugerir mejoras o para agradecermelo. :)

También al equipo de BorrowBITS por el gran trabajo que están haciendo y por dejarme ser parte de ellos.

Bla, Bla, Bla ...

Índice general

1. Introducción	1
1.1. Contexto	1
1.1.1. La importancia de la vídeo vigilancia	1
1.1.2. Seguridad emocional	1
1.1.3. Vídeo Vigilancia Inalámbrica	2
1.2. Motivación	2
1.3. Objetivos y alcance del proyecto	2
2. Estado del Arte	3
2.1. TeleEye iView	3
2.1.1. Características	3
2.1.2. Ventajas	4
2.1.3. Desventajas	4
2.2. Otro Ejemplo	4
2.2.1. Características	4
2.2.2. Ventajas	4
2.2.3. Desventajas	4
2.3. Así todos los que necesites	4
3. Especificación de requisitos	5
3.1. Requisitos funcionales	5
3.2. Requisitos de los datos	6
3.2.1. <i>Streaming</i> de vídeo	6
3.2.2. Más cosas	6
3.3. Requisitos no funcionales	6
3.4. Requisitos de la interfaz	7
3.4.1. Interfaz gráfica del servidor	7
3.4.2. Interfaz gráfica del cliente	7

ÍNDICE GENERAL

4. Planificación y estimación de costes	9
4.1. Recursos	9
4.1.1. Humanos	9
4.1.2. <i>Hardware</i>	9
4.1.3. <i>Software</i>	9
4.2. Fase de desarrollo	10
4.2.1. WP1: Revisión del estado del arte	10
4.2.2. WP2: Especificación de requisitos	10
4.2.3. WP3: Diseño	10
4.2.4. WP4: Implementación	10
4.2.5. WP5: Evaluación y pruebas	10
4.2.6. WP6: Documentación	11
4.3. Estimación de costes	11
4.3.1. Recursos humanos	11
4.3.2. Herramientas	11
4.4. Presupuesto	11
5. Diseño	13
5.1. Diseño de la arquitectura	13
5.1.1. Servidor	13
5.1.2. Cliente	13
5.2. Diseño de los datos	13
5.2.1. Módulos del servidor	14
5.2.1.1. Clase <code>servidorVVD</code>	14
5.2.1.2. Clase <code>masClases()</code>	15
5.2.2. Módulos del cliente	15
6. Implementación	17
6.1. Servidor	17
6.1.1. Servidor HTTP	17
6.1.1.1. Tarea <code>otroEjemplo</code>	19
7. Evaluación	21
7.1. Topología 1: Descubrimiento de una cámara no operativa	21
7.1.1. Escenario	21
7.1.2. Funcionamiento	21
7.1.3. Estimación del rendimiento	22

7.1.4. Tráfico generado	22
7.2. Topología 2: Otro ejemplo	23
7.2.1. Escenario	23
7.2.2. Funcionamiento	23
7.2.3. Estimación del rendimiento	23
7.2.4. Tráfico generado	23
8. Conclusiones	25
8.1. Resultados	25
8.2. Líneas de trabajo futuras	25
A. Manual de usuario del servidor	27
A.1. Instalación	27
A.2. Documentación	27
A.2.1. Formato del archivo de configuración	27
A.2.2. Formato de los archivos de información de las cámaras	27
A.3. Uso	28
A.3.1. Archivos generados tras la ejecución	29
B. Manual de usuario del cliente para Android	31
B.1. Instalación	31
B.2. Uso	31
Referencias	33

ÍNDICE GENERAL

Índice de figuras

2.1. TeleEye iView para Android.[5]	3
3.1. Diagrama de contexto.	5
4.1. Diagrama de Gantt, con la planificación temporal del proyecto[7].	10
5.1. Relación entre módulos del servidor.	14
6.1. Diagrama de flujo de la actualización automática de la información de las cámaras.	19
7.1. Topología 1.	22
A.1. Interfaz de control del servidor VVD.	28
A.2. Ventana para añadir un nuevo servidor.	29
B.1. Menú Android VVD conectado.	32

ÍNDICE DE FIGURAS

Índice de cuadros

3.1. Parámetros de vídeo aconsejados por Android Developers [8]	6
4.1. Coste temporal del proyecto.	11
4.2. Presupuesto.	12
7.1. Topología 1. Listas de los servidores al inicio.	22

ÍNDICE DE CUADROS

Índice de algoritmos

6.1. Constructor <code>HttpServidor</code> y declaración de los contextos.	18
6.2. Contexto <code>/listaCamaras</code>	18
6.3. <i>Layout</i> de la fila de la lista: <code>filalayout.xml</code>	20

Capítulo 1

Introducción

1.1. Contexto

1.1.1. La importancia de la vídeo vigilancia

La captación y/o el tratamiento de imágenes con fines de vigilancia es una práctica muy extendida en nuestra sociedad. La vídeo vigilancia generalmente persigue garantizar la seguridad de los bienes y las personas Bla, Bla, Bla.

1.1.2. Seguridad emocional

La existencia de alarmas Bla, Bla, Bla.

Así existe lo que se llama “seguridad emocional”[\[3\]](#), (Aquí acabo de poner una Referencia) es decir, aquellas alarmas o alertas que no están relacionadas con la seguridad simplemente sino que contienen elementos de información y sucesos ocurridos.

Podemos dividir en dos categorías el tipo de alarmas de las que estamos hablando:

1. Alarmas de actividad: Aquellas que nos informan de la presencia de personal doméstico, del regreso de un hijo o hija al hogar con la persona que los trae del colegio, la llegada de los demás miembros de la familia, etc.
2. Alarmas de ausencia de actividad: Si uno deja a un niño, niñas o personas mayores en el hogar y durante un período de tiempo prolongado no se produce ningún tipo de aviso, esto puede deberse a que ha ocurrido algo y, por tanto, requiere de nuestra atención.

Bla, Bla, Bla...

1. INTRODUCCIÓN

1.1.3. Vídeo Vigilancia Inalámbrica

El sistema de vídeo vigilancia inalámbrica representa una solución alternativa a la mayoría de los Bla, Bla, Bla. Proporciona características como:

- Alto grado de funcionalidad.
- Totalmente Escalable.
- Despliegue rápido y sencillo.
- Flexibilidad.
- Rentabilidad de la inversión.

Bla, Bla, Bla...

1.2. Motivación

Como ya se ha apuntado en la Sección 1.1, los sistemas de vídeo vigilancia sirven para una gran cantidad de propósitos. En el ámbito de las empresas, bancos, instituciones públicas, aeropuertos, grandes eventos con aglomeraciones de personas, etc. Bla, Bla, Bla.

1.3. Objetivos y alcance del proyecto

El principal objetivo del presente proyecto es el diseño e implementación de un sistema Bla, Bla, Bla

Capítulo 2

Estado del Arte

En el Capítulo 1, se ha destacado la importancia que tiene la seguridad en nuestra sociedad, y Bla, Bla, Bla

2.1. TeleEye iView

The TeleEye Group ha desarrollado un sistema de vídeo vigilancia para móviles iOS y Android que permite el control y visualización remota de videocámaras instaladas en hogares o en oficinas desde cualquier parte del mundo[4].

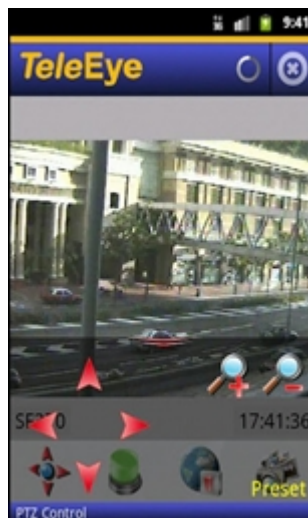


Figura 2.1: TeleEye iView para Android.[5]

2.1.1. Características

TeleEye iView está desarrollado siguiendo Bla, Bla, Bla... y puede realizar PTZ.

2. ESTADO DEL ARTE

2.1.2. Ventajas

- El cliente está disponible para un amplio número de dispositivos diseñados tanto para Android como para iOS.
- Transmisión de vídeo sobre Wi-Fi, HSDPA, 3G, EDGE o GPRS.

Bla, Bla, Bla

2.1.3. Desventajas

- Es necesario un *hardware* específico que actúe como servidor, que será el encargado del control de las cámaras y del envío en *streaming* del vídeo llamado **TeleEye Mobile Monitoring Solution (M-303)**, por lo tanto, no es gratuito.
- En el cliente sólo es posible visualizar una de las cámaras Bla, Bla, Bla

2.2. Otro Ejemplo

Bla, Bla, Bla

2.2.1. Características

Bla, Bla, Bla

2.2.2. Ventajas

Bla, Bla, Bla

2.2.3. Desventajas

Bla, Bla, Bla

2.3. Así todos los que necesites

Capítulo 3

Especificación de requisitos

La especificación de requisitos queda dividida en los siguientes apartados: requisitos funcionales, no funcionales, de datos y de la interfaz.

3.1. Requisitos funcionales

Un requisito funcional define el comportamiento interno del software: Bla, Bla, Bla

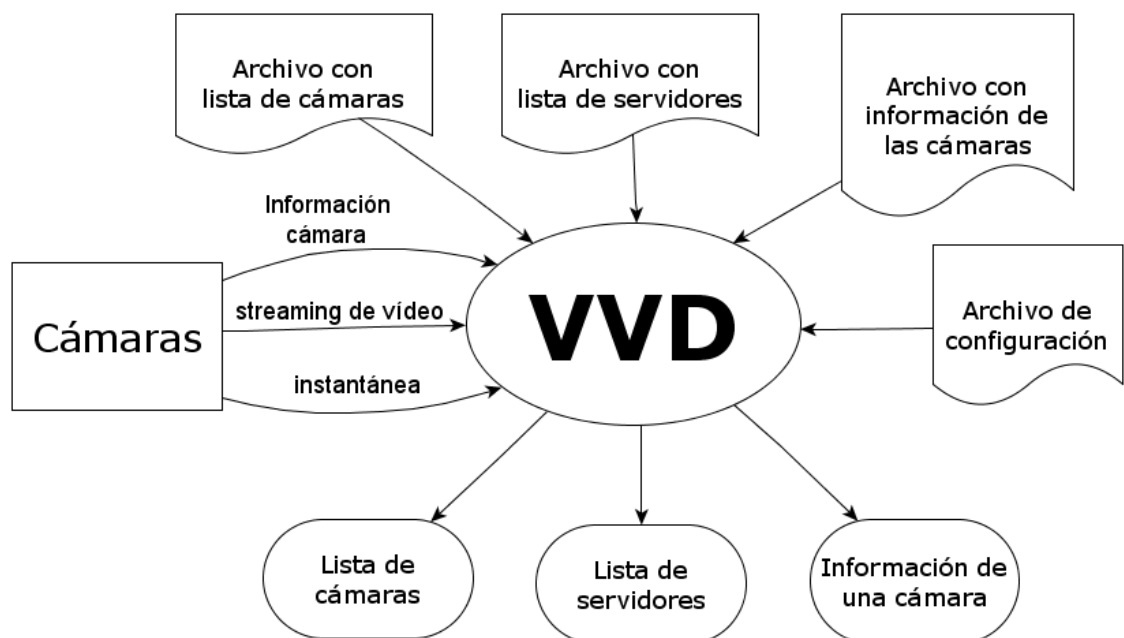


Figura 3.1: Diagrama de contexto.

3. ESPECIFICACIÓN DE REQUISITOS

3.2. Requisitos de los datos

Bla, Bla, Bla.

3.2.1. *Streaming* de vídeo

El cliente, una vez que haya seleccionado una cámara Bla, Bla, Bla. Los parámetros de vídeo aconsejados por Android Developers son:

	SD (Baja Calidad)	SD (Alta calidad)	HD
Códec de vídeo	H.264	H.264	H.264
Resolución de vídeo	176 x 144 <i>píxeles</i>	480 x 360 <i>píxeles</i>	1280 x 720 <i>píxeles</i>
Frame rate	12 fps	30 fps	30 fps
Bitrate	56 Kbps	500 Kbps	2 Mbps

Cuadro 3.1: Parámetros de vídeo aconsejados por Android Developers [8].

3.2.2. Más cosas

Bla, bla, bla

- Elemento número uno.
- Elemento número dos.

3.3. Requisitos no funcionales

Un requisito no funcional o atributo de calidad es un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar.

Software y compatibilidad

En la implementación del servidor se usará el bla,bla,bla.

Dependencia de otras partes

bla,bla,bla.

Gestión de fallos

bla,bla,bla.

Extensibilidad

bla,bla,bla.

Seguridad

bla,bla,bla.

Portabilidad

bla,bla,bla.

Concurrencia

bla,bla,bla.

3.4. Requisitos de la interfaz

Tanto para el servidor como para el cliente, se proporcionará una interfaz gráfica para bla,bla,bla.

3.4.1. Interfaz gráfica del servidor

La interfaz gráfica contará con bla,bla,bla.

3.4.2. Interfaz gráfica del cliente

La interfaz gráfica contará con un conjunto de bla,bla,bla.

3. ESPECIFICACIÓN DE REQUISITOS

Capítulo 4

Planificación y estimación de costes

4.1. Recursos

4.1.1. Humanos

- D. Juan José Ramos Muñoz, profesor del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, en calidad de tutor del proyecto.
- Rafael Martos Llaveró, alumno de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada, autor del proyecto.

4.1.2. *Hardware*

- Ordenador personal.
- *Tablet* con sistema operativo *Android 2.3 Gingerbread*.

bla,bla,bla.

4.1.3. *Software*

- Sistema operativo Microsoft® Windows 7.
- Ofimática: LyX 2.0.4 y MiKTeX 2.9 [6].
- SDK de Android (*Software Development Kit*) [9].

bla,bla,bla.

4. PLANIFICACIÓN Y ESTIMACIÓN DE COSTES

4.2. Fase de desarrollo

La distribución temporal de los seis paquetes de trabajo puede verse en el diagrama de Gantt de la Figura 4.1. A continuación se procederá a enumerar los paquetes junto con las tareas que los componen:

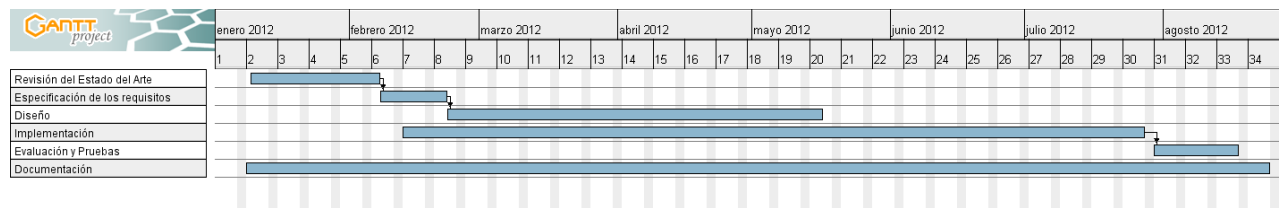


Figura 4.1: Diagrama de Gantt, con la planificación temporal del proyecto[7].

4.2.1. WP1: Revisión del estado del arte

Se hace un repaso de las soluciones existentes hasta el momento, totales o parciales, sobre el mismo ámbito de aplicación o relacionado, del problema a bla,bla,bla.

4.2.2. WP2: Especificación de requisitos

Se analiza pormenorizadamente el problema en pos de una correcta especificación de requisitos, funcionalidades y restricciones que delimitan el posterior diseño de una solución. bla,bla,bla.

4.2.3. WP3: Diseño

Se adopta justificadamente una solución previamente especificada y se procede a su diseño.

bla,bla,bla.

4.2.4. WP4: Implementación

Se implementa el diseño anterior en el lenguaje de programación escogido. Bla,bla,bla.

4.2.5. WP5: Evaluación y pruebas

Tras la comprobación y puesta a punto, se procede a la evaluación de la solución, contrastando su funcionamiento y resultados, si procede, con otras soluciones vistas durante la revisión del estado del arte.

Fase	Duración (días)
Revisión estado del arte	X
Especificación de los requisitos	X
Diseño	X
Diseño + Implementación	X
Implementación	X
Evaluación y Pruebas	X
TOTAL	X

Cuadro 4.1: Coste temporal del proyecto.

4.2.6. WP6: Documentación

La redacción de la documentación es una tarea que se lleva a cabo de forma paralela al resto.

4.3. Estimación de costes

4.3.1. Recursos humanos

En el Cuadro 4.1 puede verse el coste temporal de los diferentes paquetes de trabajo. En total se computa en días (jornadas de 8 horas a 20 €/hora), sin contar sábados, domingos y festivos. La documentación no aparece por ser Bla,bla,bla.

4.3.2. Herramientas

Se han empleado para el diseño del proyecto herramientas gratuitas siempre que ha sido posible para minimizar el coste del proyecto. Bla, bla,bla.

4.4. Presupuesto

En el Cuadro 4.2 se registra el presupuesto del proyecto.

4. PLANIFICACIÓN Y ESTIMACIÓN DE COSTES

CONCEPTO	CANTIDAD
<i>Recursos humanos</i>	
1 desarrollador x X días x 8 h/día x 20 €/h	X €
<i>Herramientas</i>	
1 ordenador personal x X € unidad x X meses / 36 meses	X €
Tablet de 7 pulgadas	X €
Sistema operativo	X €
Entorno de desarrollo	0 €
Software complementario y librerías	0 €
Paquetes de ofimática	0 €
TOTAL	X €

Cuadro 4.2: Presupuesto.

Capítulo 5

Diseño

A continuación, se establecerá el diseño de la aplicación en dos pasos: en primer lugar y con un mayor nivel de abstracción, se procederá al diseño de la arquitectura de forma que quede definido el comportamiento global del sistema; en segundo lugar, se profundizará en su estructura a través de los diagramas de clases en el apartado de diseño de los datos.

5.1. Diseño de la arquitectura

Tal y como se ha especificado en el Capítulo 3, la arquitectura escogida es de tipo cliente-servidor. La herramienta constará de Bla,bla,bla.

5.1.1. Servidor

Como se puede verse en Bla,bla,bla.

5.1.2. Cliente

El cliente, por su parte, consta de Bla,bla,bla.

5.2. Diseño de los datos

Se dividirá entre la aplicación del servidor y la del cliente, definiendo los módulos (clases de Java o *Activities*[\[1\]](#) de Android[\[2\]](#)) que intervienen y la relación existente entre ellos, tal y como se muestra en la Figura 5.1 y en Bla,bla,bla.

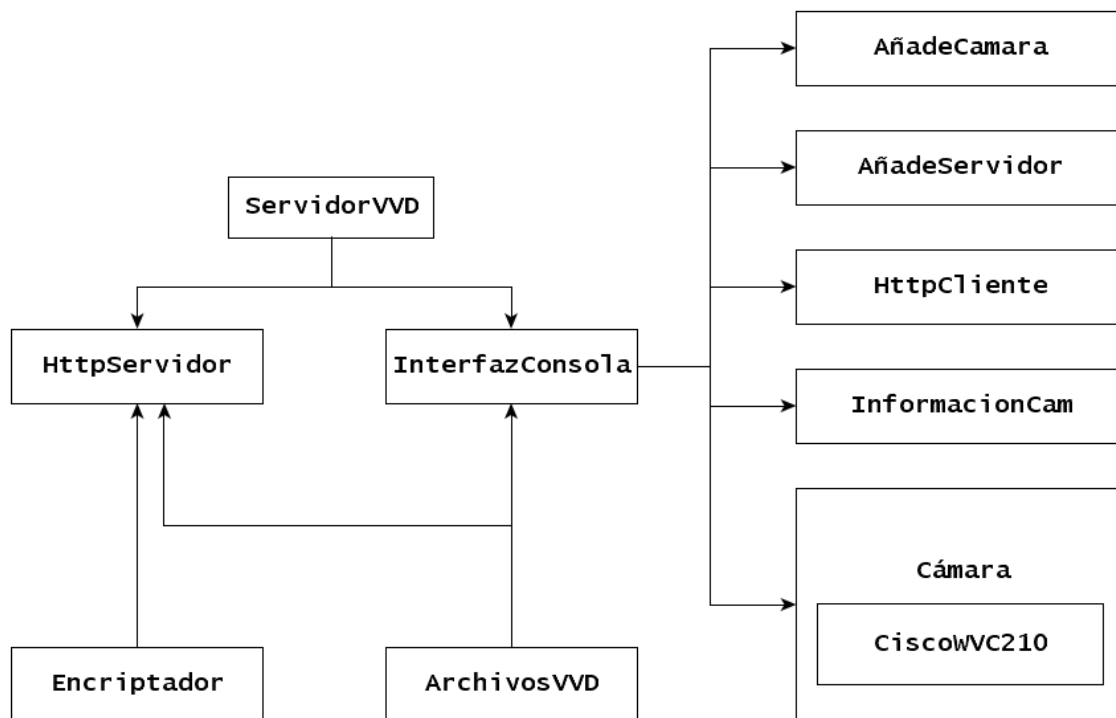


Figura 5.1: Relación entre módulos del servidor.

5.2.1. Módulos del servidor

Se definen los siguientes módulos (clases de Java) para el servidor, cuyas relaciones vienen definidas en la Figura 5.1:

ServidorVVD Núcleo del programa. Bla,bla,bla.

otroMas Clase que actúa como un servidor HTTP lanzado en modo *standalone*, para Bla,bla,bla.

muchosMas Bla,bla,bla.

5.2.1.1. Clase servidorVVD

Arranca el servidor de HTTP propio en modo *standalone*, escuchando las Bla,bla,bla.

Constructores

`constructor()` Bla,bla,bla.

`constructor(muchosArgumentos)` Bla,bla,bla.

Métodos

`otroMetodo()` Método con el que se inicia la aplicación.

5.2.1.2. Clase `masClases()`

Bla,bla,bla.

Constructores

Bla,bla,bla.

Métodos

Bla,bla,bla.

5.2.2. Módulos del cliente

Bla,bla,bla.

5. DISEÑO

Capítulo 6

Implementación

El presente capítulo tiene por objetivo detallar las soluciones adoptadas a la hora de acometer la implementación del sistema planteado en el capítulo anterior. Los puntos principales a tratar son dos: la implementación del servidor y del cliente, detallando aquellos aspectos fundamentales.

6.1. Servidor

Se procederá a la descripción de las funcionalidades principales del servidor, especificando la Bla,bla,bla.

6.1.1. Servidor HTTP

Tal y como se ha descrito en la fase de diseño, el servidor debe soportar Bla,bla,bla.

6. IMPLEMENTACIÓN

Algoritmo 6.1 Constructor HttpServidor y declaración de los contextos.

```
1 public HttpServidor(int puerto){
2     this.puerto = puerto;
3     try {
4         server = HttpServer.create(new InetSocketAddress(this.puerto), 0);
5     } catch (BindException be){
6         JOptionPane.showMessageDialog(null, "El puerto: "+puerto+" está ocupado por
          otra aplicación.\r\n Cambie su valor en el archivo de configuración \"
          config.txt\" en la siguiente línea: \r\n #Puerto (Esta línea es un
          comentario opcional) \n puerto=Valor_Del_Puerto\n", "Error al abrir el
          puerto "+puerto, 0);
7     } catch (IOException e) {
8         e.printStackTrace();
9     }
10    try {
11        InetAddress address = InetAddress.getLocalHost();
12        ...
13    } catch (UnknownHostException e) {
14        e.printStackTrace();
15    }
16    server.start();
17 }
```

Algoritmo 6.2 Contexto /listaCamaras

```
1 static class ListaCamarasHandler implements HttpHandler {
2     public void handle(HttpExchange t) throws IOException {
3         InputStream is;
4         OutputStream os;
5         String response = "";
6         String camaras[] = new String[100];
7         is=t.getRequestBody();
8         ...
9         int i=0;
10        if(camaras[i]!=null){
11            response = camaras[i];
12            i++;
13        }else{
14            response = "null";
15        }
16        while(camaras[i]!=null){
17            response = response + ";" + camaras[i];
18            i++;
19        }
20        ...
21    }
22 }
```

Para servir la lista de servidores a los clientes que la soliciten, se realizará siguiendo el formato siguiente:

```
200 OK
nombre_servidor1 , IP:puerto; nombre_servidor2 , IP:puerto;
    nombre_servidor3 , IP:puerto; ...
```

6.1.1.1. Tarea otroEjemplo

Una vez que han sido actualizadas las listas de servidores y de cámaras, se procede a la actualización de los archivos de información de cada cámara. El proceso de actualización de la información de las cámaras se muestra en la Figura 6.1 y se explica a continuación:

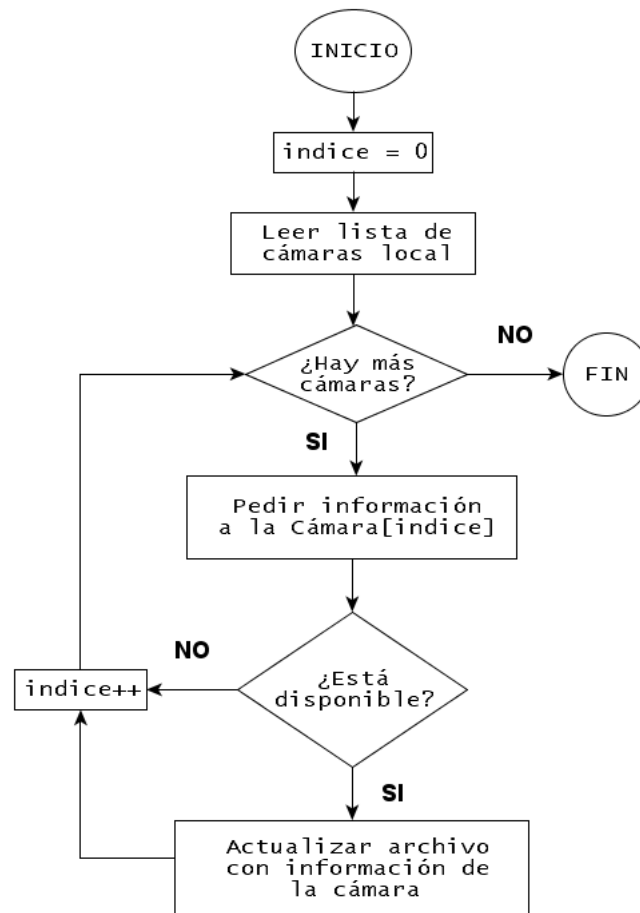


Figura 6.1: Diagrama de flujo de la actualización automática de la información de las cámaras.

1. Se lee la primera cámara almacenada en el servidor en el archivo de **listaCamaras.txt**.
2. Se conecta a la cámara solicitándole su información de configuración.
3. Si la cámara se encuentra disponible, se dibuja en la interfaz gráfica de la lista de cámaras una luz verde, y se actualiza el contenido del archivo **IPcamara.txt**.

6. IMPLEMENTACIÓN

4. En caso de que la cámara no esté operativa, se dibuja un icono rojo y no se altera el contenido del archivo **IPcamara.txt**.
5. Se conecta a la siguiente cámara y se repite el proceso hasta llegar al final del archivo **listaCamaras.txt**.

Algoritmo 6.3 *Layout* de la fila de la lista: `filaLayout.xml`.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/filaLayout"
5     android:layout_width="wrap_content"
6     android:layout_height="wrap_content"
7     android:orientation="horizontal"
8 >
9     <ImageView
10         android:id="@+id/icono"
11         android:layout_width="32dp"
12         android:layout_height="32dp"
13         android:layout_marginLeft="4dp"
14         android:layout_marginRight="10dp"
15         android:layout_marginTop="4dp"
16         android:scaleType="fitCenter"
17         android:src="@drawable/cisco">
18     </ImageView>
19
20     <TextView
21         android:id="@+id/label"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:text="@+id/label"
25         android:textScaleX="1.2"
26         android:gravity="center_vertical"
27         android:layout_gravity="center_vertical">
28     </TextView>
29
30 </LinearLayout>
```

Capítulo 7

Evaluación

En este capítulo se procede a la evaluación del proyecto llevado a cabo. Se realizará el estudio sobre tres escenarios en los que se simulan los casos más comunes. En primer lugar se definen los escenarios empleados para la realización de las pruebas, los pasos que realizan los servidores durante su funcionamiento y cómo quedan sus datos tras la ejecución. Se analizará en cada apartado los Bla,bla,bla.

7.1. Topología 1: Descubrimiento de una cámara no operativa

7.1.1. Escenario

Se establece una red en el rango 192.168.1.0/24. Existen dos servidores: Servidor A con la IP 192.168.1.35:8004 y Servidor B con la IP 192.168.1.100:8006; los puertos de escucha han sido configurados en los respectivos archivos de configuración de cada servidor. Hay un dispositivo Android que actuará como cliente con la IP 192.168.1.10, y dos cámaras Cisco WVC210 con las IP: 192.168.1.34 y 192.168.1.33 (véase Figura 7.1).

7.1.2. Funcionamiento

El tiempo entre actualizaciones se ha establecido en 30 segundos en ambos servidores.

Las listas de los servidores al iniciarse se encuentran con los datos reflejados en la tabla 7.1.

7. EVALUACIÓN

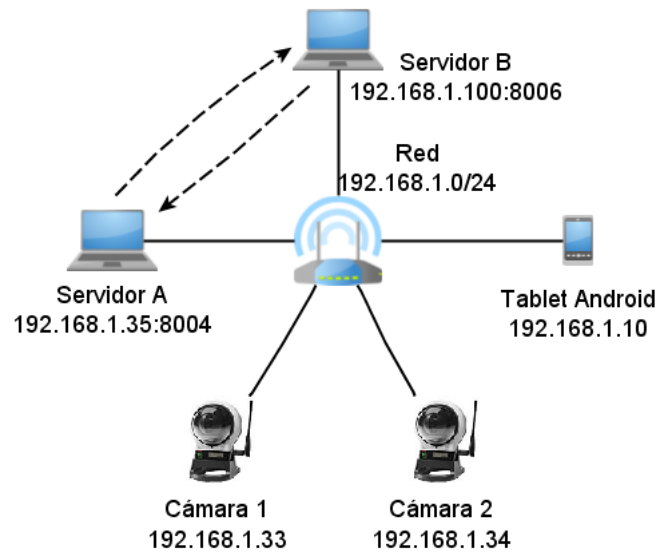


Figura 7.1: Topología 1.

Servidor A		Servidor B	
Cámaras	Servidores	Cámaras	Servidores
-	Servidor B	Cámara 1	-
		Cámara 2	

Cuadro 7.1: Topología 1. Listas de los servidores al inicio.

El Servidor B y las cámaras se encuentra en funcionamiento. Al iniciar el Servidor A se generan los siguientes mensajes de *log*:

```
[11/09/2012 11:22:01.493] SERVER:# INFO: Servidor VVD iniciado.  
[11/09/2012 11:22:01.524] SERVER:# INFO: Temporizador Actualizaciones Automáticas = 0.5 minutos.
```

Anuncia que se ha iniciado el servidor e informa del tiempo entre actualizaciones.

```
[11/09/2012 11:22:02.553] SERVER:# INICIO: Actualización lista de servidores.  
[11/09/2012 11:22:03.240] SERVER:# FIN: Actualizacion lista de servidores.
```

Bla,bla,bla.

7.1.3. Estimación del rendimiento

Bla,bla,bla.

7.1.4. Tráfico generado

Bla,bla,bla.

7.2. Topología 2: Otro ejemplo

7.2.1. Escenario

Bla,bla,bla.

7.2.2. Funcionamiento

Bla,bla,bla.

7.2.3. Estimación del rendimiento

Bla,bla,bla.

7.2.4. Tráfico generado

Bla,bla,bla.

7. EVALUACIÓN

Capítulo 8

Conclusiones

8.1. Resultados

En este proyecto se ha diseñado e implementado una aplicación que Bla,bla,bla.

8.2. Líneas de trabajo futuras

Quedan abiertas las siguientes líneas de trabajo:

- Bla,bla,bla.
- Bla,bla,bla.

8. CONCLUSIONES

Apéndice A

Manual de usuario del servidor

A.1. Instalación

Bla,bla,bla.

A.2. Documentación

Bla,bla,bla.

A.2.1. Formato del archivo de configuración

El contenido del archivo se compone por pares “llave=valor”. Si una línea comienza por una almohadilla (#), será considerada un comentario.

El archivo de configuración tiene la siguiente estructura:

```
#Puerto a la escucha del Servidor VVD
puerto=8004
...
```

A.2.2. Formato de los archivos de información de las cámaras

La información de las cámaras se ha de almacenar siguiendo un determinado
Bla,bla,bla.

```
hostname=camaraVVD
description=WVC210 WirelessPTZ Video Camera
defname=camaraVVD
...
```

A. MANUAL DE USUARIO DEL SERVIDOR

El significado de cada parámetro es el siguiente:

hostname Nombre del *host*.

description Descripción.

defname Nombre de la cámara definido por el usuario.

etc Bla,bla,bla.

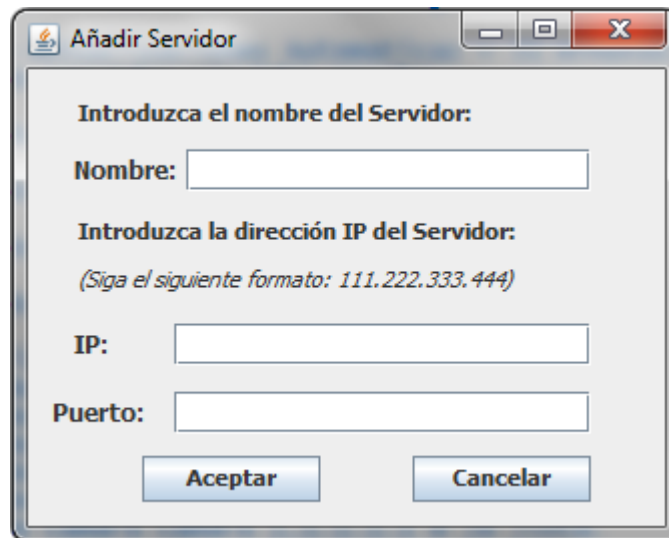
A.3. Uso

El servidor ha de ejecutarse en Bla,bla,bla.



Figura A.1: Interfaz de control del servidor VVD.

Botones para la lista de los servidores: Dentro del grupo de botones de control de la lista de servidores, se encuentra *añadir servidor*, que abre una ventana para introducir el nombre, IP y puerto del nuevo servidor de la lista, tal y como puede verse en la Figura A.2.



Añadir Servidor

Introduzca el nombre del Servidor:

Nombre:

Introduzca la dirección IP del Servidor:

(Siga el siguiente formato: 111.222.333.444)

IP:

Puerto:

Aceptar **Cancelar**

Figura A.2: Ventana para añadir un nuevo servidor.

A.3.1. Archivos generados tras la ejecución

Bla,bla,bla.

A. MANUAL DE USUARIO DEL SERVIDOR

Apéndice B

Manual de usuario del cliente para Android

B.1. Instalación

El cliente de VVD: Vídeo Vigilancia Distribuida es una aplicación diseñada para Android. Tiene las siguientes Bla,bla,bla.

B.2. Uso

Para ejecutar el cliente hay que pulsar sobre el icono Android VVD creado en Bla,bla,bla. La interfaz puede verse en la Figura B.1. Los botones bla,bla,bla.

Otro Apartado Más

Bla,bla,bla



Figura B.1: Menú Android VVD conectado.

Referencias

- [1] ANDROID ACTIVITY. Ventanas de una aplicación android.
<http://developer.android.com/reference/android/app/Activity.html>.
- [2] ANDROID. Sistema operativo. *<http://www.android.com/about/>.*
- [3] SEGURIDAD EMOCIONAL. Definición y ejemplos.
<http://ebookbrowse.com/ticmania-ficha-21-videovigilancia-pdf-d252554299>.
- [4] THE TELEEYE GROUP. Teleeye iview hd. *<http://www.teleeye.com>.*
- [5] IMAGEN TELEEYE IVIEW. *<http://dd6xle2n4vut.cloudfront.net/com.teleeye.gViewFull3.png>.*
- [6] LYX. The document processor. *<http://www.lyx.org/>.*
- [7] GANTT PROJECT. Herramienta para generar diagramas de gantt.
<http://www.ganttproject.biz/>.
- [8] VIDEO ENCODING RECOMMENDATIONS. Android developers.
<http://developer.android.com/guide/appendix/media-formats.html>.
- [9] ANDROID SDK. Software developement kit.
<http://developer.android.com/sdk/index.html>.