



**Universitat Autònoma  
de Barcelona**

# **Sistema de videovigilancia a través de una Raspberry Pi**

## **Informe de Progreso II**

**Aitor Domene Sánchez  
1332008**

**Grau en Enginyeria Informàtica**

**Escola d'Enginyeria**

**Curso 2016-17**

## Tabla de contenidos

1. INTRODUCCIÓN.....	4
2. PALABRAS CLAVE .....	4
3. OBJETIVOS DEL PROYECTO .....	5
4. METODOLOGÍA.....	6
5. PLANIFICACIÓN DEL PROYECTO .....	7
6. CONCLUSIONES.....	10
7. BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN.....	10
ANEXO A: WIREFRAMES DE LA PLATAFORMA WEB.....	11
ANEXO B: PLATAFORMA WEB.....	14
ANEXO C: DIAGRAMA DE FLUJO PLATAFORMA WEB.....	16
ANEXO D: TRANSFERENCIA FICHEROS: SISTEMA – WEB.....	17
ANEXO E: ARQUITECTURA SOCKETS SISTEMA – WEB.....	20

## Tabla de figuras

TABLA 1. OBJETIVOS DEL PROYECTO .....	5
TABLA 2. TAREAS DE LOS PRIMEROS SPIRINTS POR % COMPLETADO ..	7
TABLA 3. COSTES DEL PROYECTO ACTUALIZADOS.....	8
FIGURA 1. PLANIFICACIÓN DEL PROYECTO.....	9
FIGURA 2. WIREFRAME DE LA PÁGINA INICIO DE LA WEB.....	11
FIGURA 3. WIREFRAME DE LA PÁGINA HISTORIAL DE LA WEB.....	12
FIGURA 4. WREFRAMDE DE LA PÁGINA STREAMING DE LA WEB.....	13
FIGURA 5. PÁGINA INICIO IMPLEMENTADA .....	14
FIGURA 6. PÁGINA HISTORIAL IMPLEMENTADA .....	15
FIGURA 7. PÁGINA STREAMING IMPLEMENTADA .....	15
FIGURA 8. DIAGRAMA DE FLUJO PLATAFORMA WEB .....	16
FIGURA 9. PROCESO COMPARTICIÓN CLAVE PÚBLICA.....	18
FIGURA 10. ARQUITECTURA COMUNICACIÓN RASPBERRY'S .....	19
FIGURA 11. ARQUITECTURA SISTEMA – WEB CON SOCKETS .....	21

## 1. Introducción

En este documento se encuentran los avances efectuados en el desarrollo del proyecto de fin de grado *Sistema de videovigilancia a través de una Raspberry Pi*. Básicamente se detallará todo el desarrollo de la plataforma web que forma la última parte del proyecto. Además, se presentan los ajustes realizados y metodologías del proyecto, así como una explicación de estas últimas.

También, se hace un resumen a modo de conclusiones del proyecto hasta ahora y se presentan las fuentes bibliográficas y de información que han sido necesarias para elaborar este informe.

Al final del documento se encuentran tanto los anexos sobre el diseño de *wireframes* y el funcionamiento de la plataforma web, como la arquitectura de comunicación entre dos Raspberry's Pi, las cuales una de ellas es la que actúa como sistema de videovigilancia, explicada en el anterior informe de progreso, y la otra, que se añade nueva al proyecto, como servidor web que contendrá la plataforma web.

Hasta ahora se ha hecho uso de una sola Raspberry Pi. El uso de una segunda Raspberry surge por la necesidad de tener toda la información almacenada, es decir, todo lo que capte el sistema de videovigilancia gestionada por una de las Raspberry's se almacenará de manera segura en la otra Raspberry evitando así la pérdida de la información a causa del extravió o pérdida de la Raspberry que gestiona el sistema de videovigilancia.

## 2. Palabras claves

**Sprint:** subconjuntos de requerimientos para ser ejecutados durante un periodo de tiempo.

**Wireframe:** representación estática de una aplicación en baja calidad de un diseño.

**Flask:** framework escrito en Python que permite la creación de aplicaciones web.

**SCP** (Secure Copy Protocol): protocolo de transferencia segura de archivos entre dos hosts.

**SSH** (Secure Shell): protocolo que facilita las comunicaciones seguras entre dos sistemas permitiendo a los usuarios conectarse a un host remotamente.

**RSA** (Rivest, Shamir, Adleman): sistema criptográfico de clave pública.

**Socket:** túnel de comunicación que ayuda a que 2 aplicaciones se comuniquen.

### 3. Objetivos del proyecto

Los objetivos del desarrollo del proyecto que se describieron en el informe inicial, son los siguientes:

1. Integrar una comunicación entre el sistema y el usuario.
2. Facilitar al usuario el uso del sistema, así como poder estar siempre informados de lo que suceda en todo momento.
3. Gestionar los diferentes componentes conectados a la Raspberry Pi para proporcionar un sistema de videovigilancia con capacidades avanzadas.
4. Integrar una plataforma web con un historial de imágenes/videos proporcionadas por el sistema de videovigilancia.

El objetivo que pertenece a este informe es el último de los descritos, *Integrar una plataforma web con un historial de imágenes/videos proporcionadas por el sistema de videovigilancia*.

Para conseguir que el objetivo detallado sean finalizado de forma satisfactoria se introdujeron tareas que se han ido llevando a cabo durante la última parte del proyecto.

En la Tabla 1 se pueden ver los objetivos establecidos y añadidos, y su grado de prioridad:

Nº	Objetivo	Prioridad
1	Diseño de la plataforma web	PRIORITARIO
2	Comunicar plataforma web con Raspberry Pi	CRITICO
3	Test de las funcionalidades del sistema	PRIORITARIO
4	Implementar 'video streaming' en plataforma web	CRITICO

Tabla 1. Objetivos del proyecto

Como se puede ver en la Tabla 1, los objetivos del proyecto respecto a la parte *plataforma web* se mantienen igual que en el *Informe de Progreso I* salvo un nuevo objetivo introducido con prioridad crítica (n4 Implementar 'video streaming' en plataforma web). Este último objetivo pretendía la posibilidad de poder acceder a un video streaming del sistema de videovigilancia desde la plataforma web. Esto permite al usuario ser capaz de poder ver lo que ocurre tanto desde su dispositivo móvil (explicado en el *Informe Progreso I*) como desde cualquier ordenador a través de la web.

#### 4. Metodología a utilizar

El desarrollo del proyecto ha ido avanzando satisfactoriamente conforme a lo que se especificó en la planificación que se hizo presente en el Informe Inicial. Las tareas se han ido realizando secuencialmente para poder conseguir una visión global y más precisa de lo que se va implementando.

Una vez finalizada la primera parte del proyecto que consistía en el desarrollo de la lógica y gestión del sistema de videovigilancia en la Raspberry Pi, los últimos *sprints* a desarrollar son respecto a la plataforma web. En esta plataforma web el usuario dispondrá de una sección en la que contendrá todo lo que el sistema de videovigilancia haya captado, es decir, un historial. Esto resulta muy útil ya que siempre tendremos una copia de seguridad de las imágenes y videos. A parte, dispondrá de una sección para poder visualizar un video streaming.

El primer sprint de la parte del desarrollo de la web, era el sprint 6 que consistía en el diseño de la plataforma web. La idea era desarrollar una plataforma web muy intuitiva y sencilla para el usuario. En el ANEXO A se visualiza el *wireframe* de la web.

Una vez diseñada la plataforma web a partir de los wireframes, daba paso al sprint 7. En este sprint se implementó la *página Inicio* y la *página Historial*. La *página Inicio* consistía en tener un sistema de acceso mediante un *usuario* y *password* para poder visualizar el contenido del historial. Respecto a la *página Historial*, consistía en mostrar todo el contenido que el sistema de videovigilancia captase, identificadas por la fecha de captura. En el ANEXO B se encuentra la plataforma web desarrollada.

El siguiente sprint, era el sprint 8. Consistía en comunicar el sistema de videovigilancia con la plataforma web para poder transferir de una Raspberry a otra toda la información recogida (imágenes y videos). Para ello, como se ha mencionado anteriormente, es necesario el uso de una segunda Raspberry Pi que se utilizará como servidor web en el que se dispondrá de dicha plataforma web. El uso de una segunda Raspberry Pi como servidor web nos da cierta ventaja ya que en el caso de que se extraviara el sistema de videovigilancia no se perdería la información recogida ya que se almacenaría en otra Raspberry Pi y podríamos acceder a la información en cualquier momento.

La idea era la siguiente: en el momento en el que el sistema de videovigilancia detectase alguna presencia (en este momento el sistema de videovigilancia habrá capturado una imagen y un video de periodo corto), tanto la imagen como el video capturado por el sistema de videovigilancia, se enviaría de forma automática al servidor web. En el ANEXO D se detalla la arquitectura de comunicación para la transferencia de las imágenes y videos entre las dos el sistema de videovigilancia y el servidor web.

Tal y como se ha comentado en el apartado *Objetivos del proyecto*, se añadió un nuevo objetivo, el cual daba a surgir el ultimo sprint del proyecto. Este objetivo se trataba de implementar un video streaming (realizado por el sistema

de videovigilancia) en el que poder visualizarse en la plataforma web. Esta idea tenía un gran problema ya que no se podía realizar la misma comunicación entre las dos Raspberry's detallado en el ANEXO D para este caso. Era necesaria la ayuda de los *sockets*.

Un socket es un método para la comunicación entre un programa del cliente (sistema de videovigilancia) y un programa del servidor (servidor web). A través de los *sockets* era posible que las dos Raspberry's Pi, el sistema de videovigilancia y el servidor web, pudiesen intercambiar cualquier flujo de datos de manera fiable y ordenada, en este caso, los fotogramas de la PiCamera. Esto quiere decir, que si se envía por el socket de flujo 30 fotogramas por segundo desde la Raspberry Pi que gestiona el sistema de videovigilancia a la Raspberry Pi que actúa como servidor web, llegará al destino esos 30 fotogramas en el mismo orden dando como resultado un *video streaming*. En el ANEXO E se pretende comprender mejor la arquitectura de comunicación.

## 5. Planificación del proyecto

A continuación se presentan las tareas asociadas a los 4 últimos sprints del proyecto – Sprint 6, Sprint 7, Sprint 8, Sprint 9:

Nº	Nombre tarea	% Completado
1	Diseño de la plataforma web	100%
2	Implementación Pagina Inicial	100%
3	Implementación Login	100%
4	Implementación Pagina Historial	100%
5	Test del diseño	100%
6	Implementar comunicación Raspberry – Servidor Web	100%
7	Test de comunicación	100%
8	Visualización streaming en plataforma web	100%

Tabla 2. Tareas de los primeros sprints por % Completado

Como se ha podido comprobar, las tareas asociadas a los 3 últimos sprints han sido llevadas a cabo de forma satisfactoria y sin cambios en la planificación.

Una vez finalizados los últimos sprints del proyecto, tal y como se ha mencionado anteriormente, se añadió un último sprint con la intención de enriquecer el proyecto (n8. Visualización streaming en plataforma web).

Por lo que respecta al último sprint, *visualización streaming en plataforma web*, provoca el reajuste para calcular el coste del proyecto, tal y como se muestra en la Tabla 3.

Componente	Horas	Cantidad	Coste unitario	Coste total
Ingeniero de Requisitos	5 h	1	25€	125€
Diseñador	12 h	1	15€	180€
Programador	146 h	1	20€	2920€
Tester	42 h	1	20€	840 €
Raspberry Pi	0 h	2	40 €	80 €
Sensor HC-SR501	0 h	1	5 €	5 €
Buzzer	0 h	1	5 €	5 €
Pi Camera	0 h	1	21€	21€
Cables	0 h	1	7€	7€
Servo Motor	0 h	1	8€	8€
IR Remote	0 h	1	12€	12€
Sublime Text	0 h	1	0 €	0 €
phpMyAdmin	0 h	1	0 €	0 €
Xampp	0 h	1	0 €	0 €
Dropbox	0 h	1	0 €	0 €
Microsoft Office	0 h	1	0 €	0 €
			<b>Sub Total</b>	4.203,00€
			<b>15% Imprevistos</b>	630,00€
			<b>25 % Ganancias</b>	1.050,00€
			<b>Total</b>	<b>5.884,00€</b>

Tabla 3. Costes del proyecto actualizados

Todos los datos expuestos en este apartado pueden analizarse mejor gracias a la Figura 1 que se encuentra en la siguiente página.



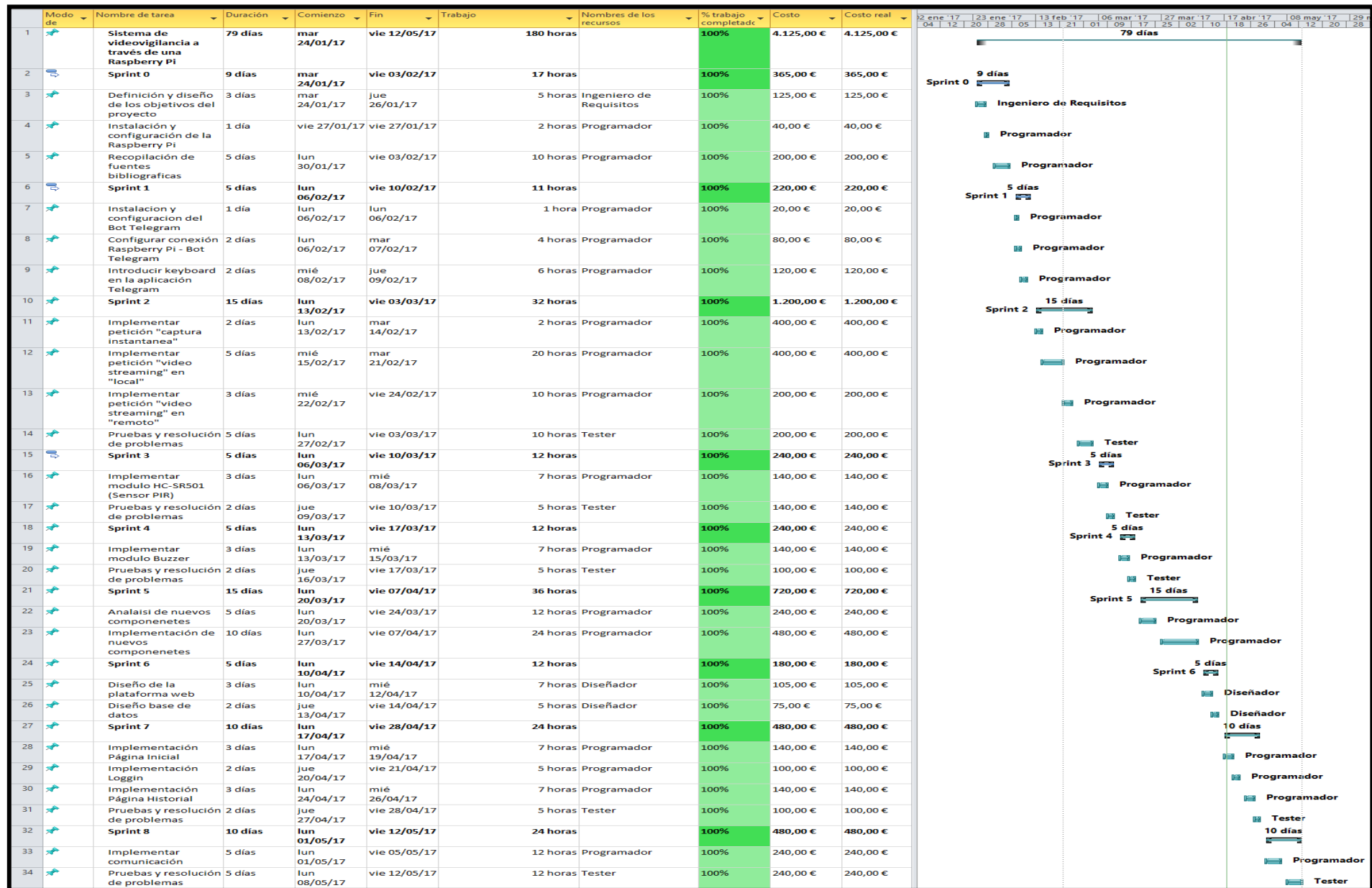


Figura 1. Planificación del proyecto (Microsoft Project)

## 6. Conclusiones

Las tareas asociadas a los últimos *sprints* han sido correctamente diseñadas e implementadas, siguiendo los plazos establecidos que se establecieron en la planificación inicial. El desarrollo del proyecto hasta hoy ha sido satisfactorio ya que el trabajo planificado se ha ajustado al cien por cien de lo que se planificó en un principio.

Una posible extensión en el proyecto desarrollado a día de hoy, cambiaría significativamente el título del trabajo de final de grado: "Sistema de videovigilancia con reconocimiento de personas a través de una Raspberry Pi". Esta *extensión* básicamente consiste en proporcionar una opción avanzada al sistema: la posibilidad de reconocer a las personas. Esta idea permitiría evitar que el sistema de videovigilancia captase movimientos no pertenecientes a personas como podría ser los animales, entre otros.

## 7. Bibliografía y fuentes de información

Anaya Multimedia. (2007). *Gestión de proyectos con MS Project*

Gestió de Projecte, UAB (2016). *Plantilla informe del Projecte*

Gestió de Projecte, UAB (2016). *Planificació del Projecte*

Diagrama Gliffy. Accedido en abril de 2017: Disponible: <https://www.gliffy.com>

Sockets en Python. Accedido en mayo de 2017: Disponible: <http://developeando.net/sockets-python/>

[1] "Ninja Mock". Accedido en abril de 2017. Disponible: <https://ninjamock.com/Designer/Workplace>

[2] "SSH Keys". Accedido en abril de 2017. Disponible: [https://wiki.archlinux.org/index.php/SSH\\_keys](https://wiki.archlinux.org/index.php/SSH_keys)

## ANEXO A

### Wireframes de la plataforma web

En esta sección se presenta el desarrollado el wireframe para el diseño de la plataforma web.

El motivo del desarrollo de un wireframe para la plataforma web es para tener una mejor visión del diseño y funcionalidades que tendrá. Para realizar el wireframe se ha utilizado la herramienta *NinjaMock* [1]. Esta herramienta se centra en ser una herramienta de nivel intermedio para la creación de wireframes tanto para móviles como para diseño web

La primera parte implementada pertenece a la página principal de la plataforma web, la *página Inicio*. En esta página, tal y como se ha comentado anteriormente, dispone de un sistema de acceso para poder acceder al contenido. Con este sistema de autenticación de usuarios se limitara el acceso a usuarios no autorizados.

La idea era desarrollar una plataforma web sencilla e intuitiva. Para ello, en la página de Inicio se implementó la parte del sistema de autenticación acompañada del logo y título del sistema.

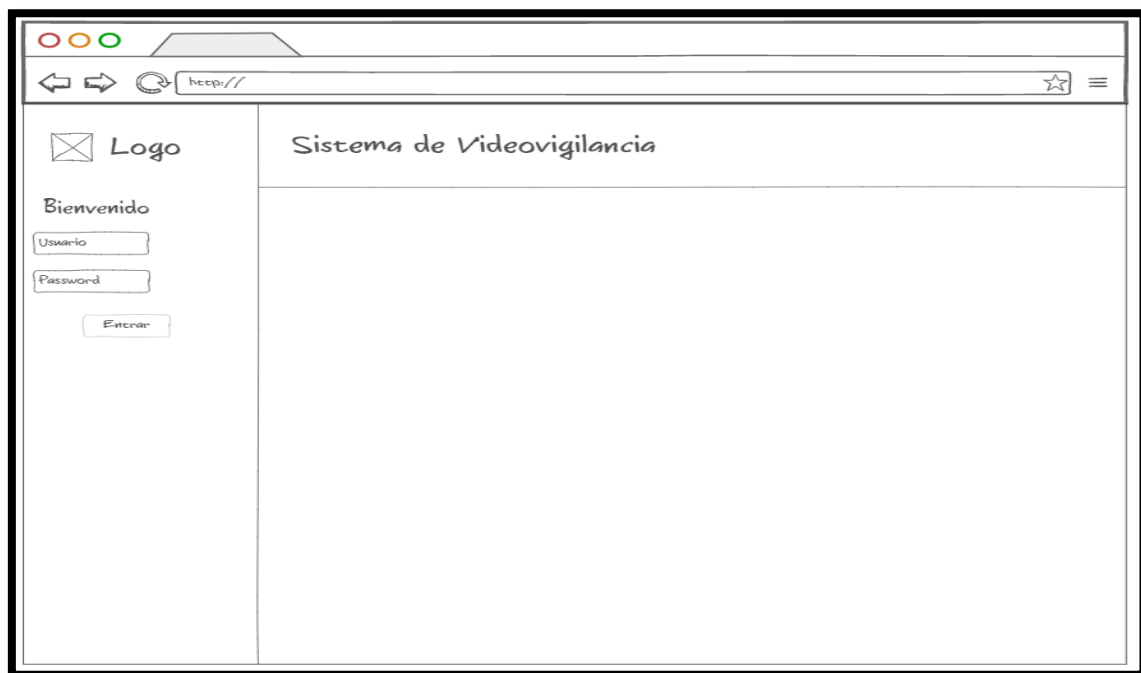


Figura 2. Wireframe de la página Inicio de la plataforma web

La segunda parte implementada pertenece a la *página Historial*. En esta página utilizamos el mismo diseño que la *página Inicio*. A diferencia de la página anterior, en la sección del sistema de autenticación pasa a ser un pequeño menú en el que le permite al usuario cerrar sesión. La sección más importante es la que se posiciona en medio de la página web. En esta sección se visualizara el contenido de imágenes y videos captados por el sistema mediante la identificación de la fecha de captura.

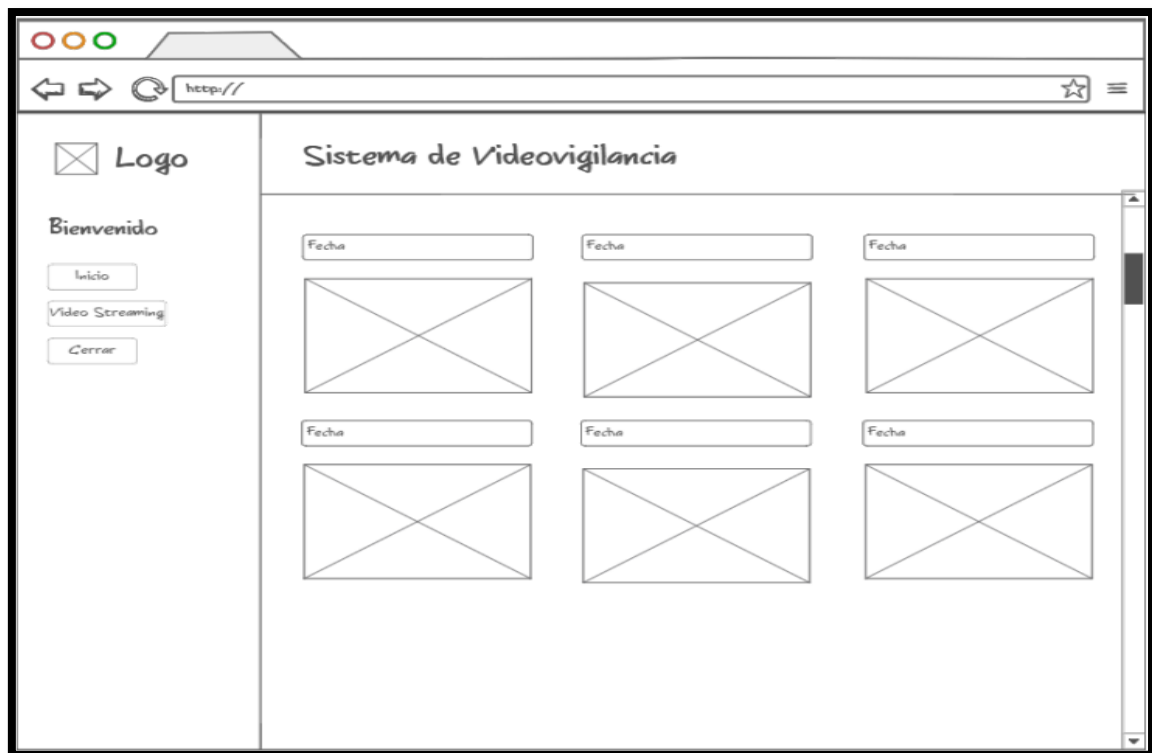


Figura 3. Wireframe de la página Historial de la plataforma web

En la figura 4 que aparece a continuación hace referencia al wireframe de la *página Streaming*. En él se puede visualizar el mismo diseño que las anteriores páginas pero con la diferencia de que en el cuerpo de la página se mostrará el video streaming del sistema de videovigilancia.

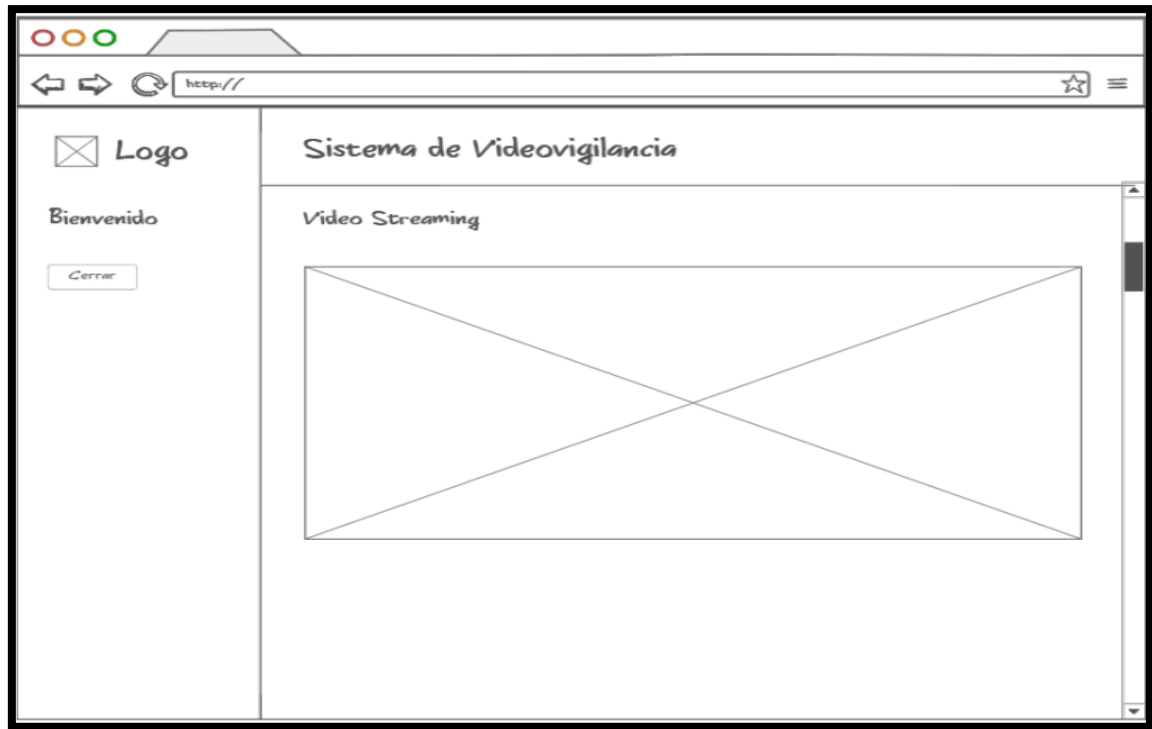


Figura 4. Wireframe de la página Streaming de la plataforma web

Una vez desarrollado el wireframe de la plataforma web se da paso a su implementación.

## ANEXO B

### Plataforma Web

En esta sección se presenta el desarrollo de la plataforma web que pertenece al sprint 7 a partir del wireframe realizado en el anterior sprint.

Para el desarrollo de la plataforma web se ha utilizado el framework *Flask*. Este framework permite configurar un servidor web en la Raspberry Pi. Como se ha mencionado anteriormente, se necesita una segunda Raspberry Pi para que haga de servidor web y proporcionar las imágenes y videos captados por el sistema.

El desarrollo de la plataforma web, además del framework *Flask*, se ha hecho uso de otros lenguajes de programación: HTML5 y CSS.

A continuación se visualiza la *página Inicio* en la Figura 4:



Figura 5. Página Inicio implementada

Tal y como se diseñó en el wireframe, la *página Inicio* contiene un sistema de autenticación en el lateral izquierda, junto al logo y título del sistema.

Para desarrollar la *página historial* y comprobar que, se visualizaban correctamente las capturas de imágenes y videos, se transfirió algunas representaciones graficas al servidor web captadas por el sistema de videovigilancia de manera manual.

Se puede comprobar en Figura 6 que cada representación gráfica es identificada por la fecha en la cual se hizo la captura:

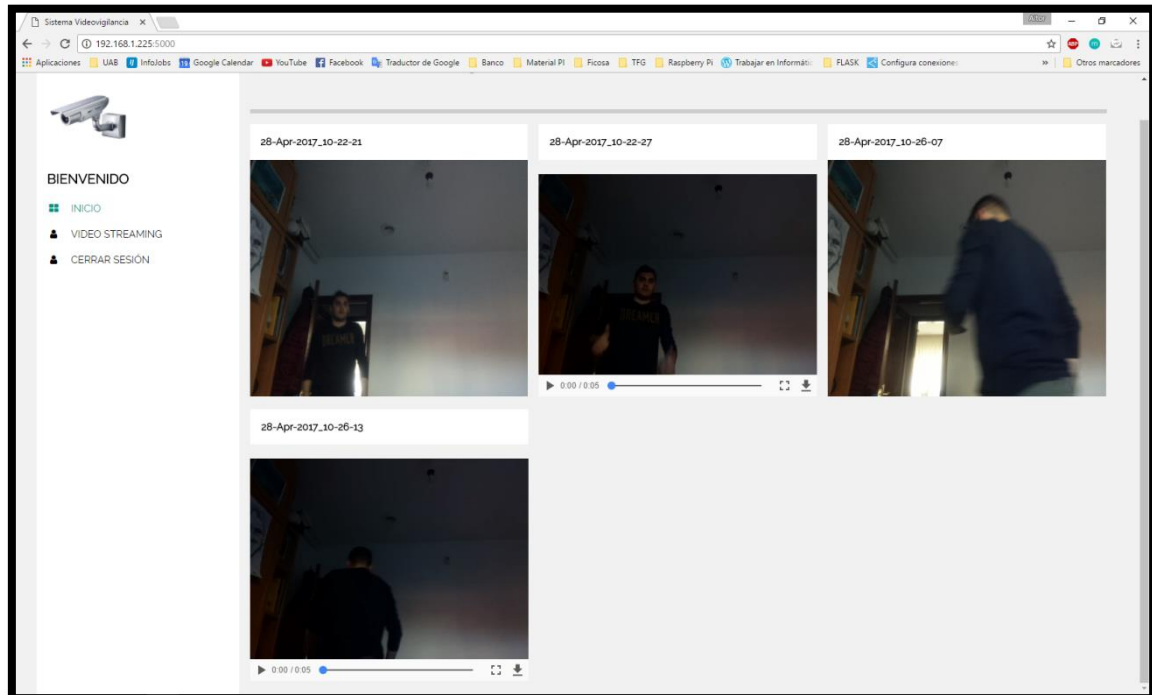


Figura 6. Página Historial implementada

Para poder cerrar sesión, tal y como se diseñó en el wireframe, se dispone de un pequeño menú con dichas opción en el lateral izquierdo.

A continuación en la Figura 7, se visualiza la *página streaming*. En el cuerpo de la página se observa la sección que comprende el video streaming.

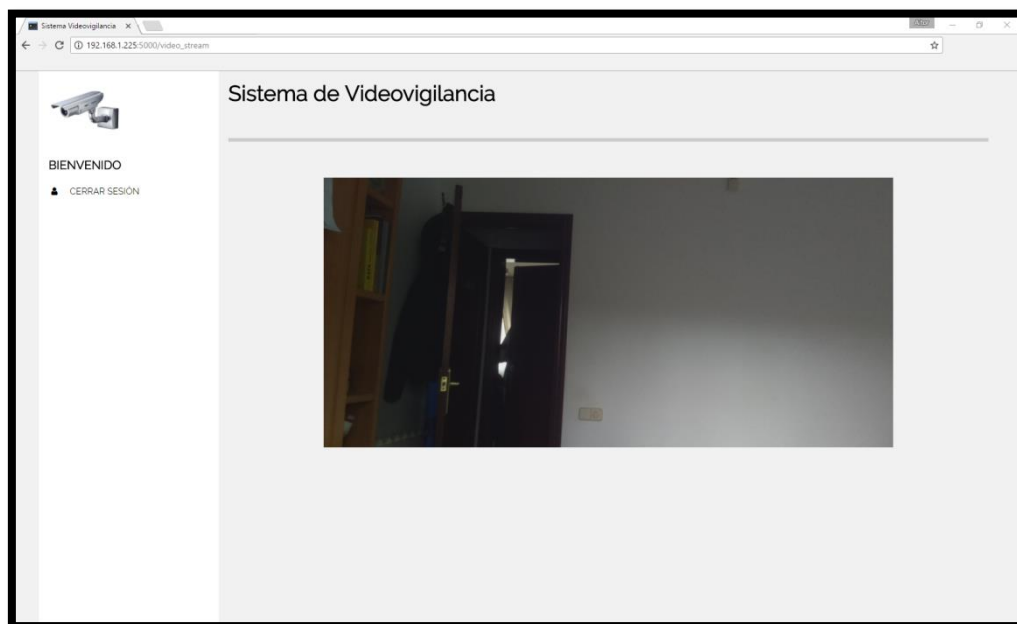


Figura 7. Página Streaming implementada

## ANEXO C

### DIAGRAMA DE FLUJO PLATAFORMA WEB

En esta sección se pretende explicar el funcionamiento de la plataforma web. A continuación, en la Figura 8 se detalla el diagrama de flujo para comprender mejor las acciones que se pueden realizar en la plataforma web.

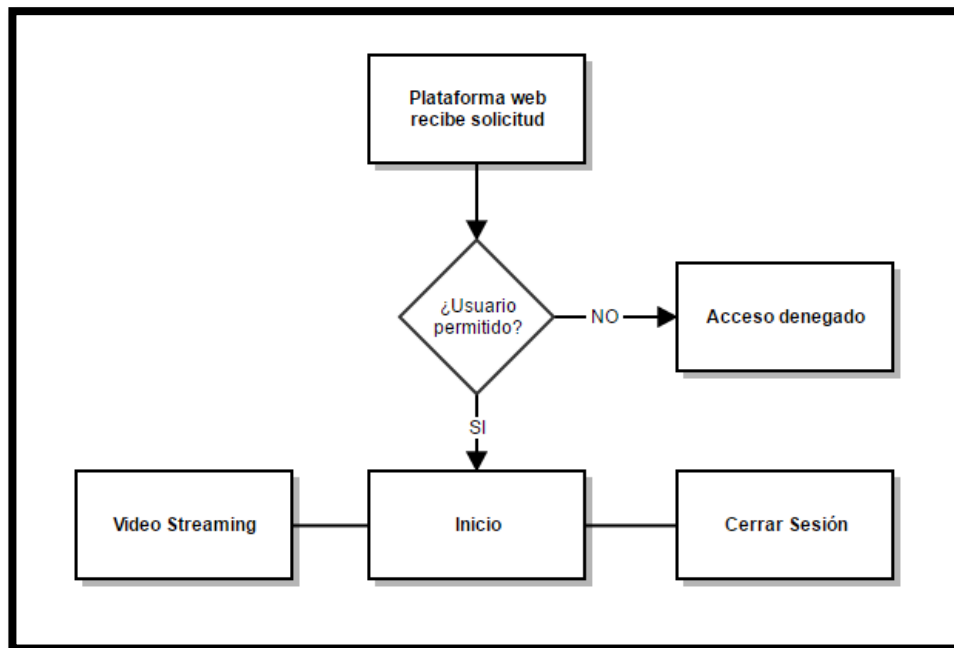


Figura 8. Diagrama de flujo de la plataforma web

Como en la mayoría de plataformas web con información sensible, el acceso a ellas se realiza mediante una autenticación. Para que el usuario pueda acceder a la plataforma web del sistema de videovigilancia deberá de autenticarse. Si esta autenticación no es válida se le denegará el acceso. Una vez el usuario se haya autenticado correctamente dispondrá de 3 opciones para interactuar con la plataforma.

En el momento en el que se autentifica el usuario estará en la página Inicio. En ella, tal y como se ha definido anteriormente, podrá observar todas las imágenes y videos que el sistema de videovigilancia haya captado. Para poder realizar un video streaming, el usuario deberá de seleccionar la pestaña *video streaming*. Para poder terminar con la sesión, solo ha de seleccionar la pestaña *cerrar sesión*. Tal y como se indica en el diagrama de flujo, el usuario podrá acceder a cualquier página este donde este.



## ANEXO D

### Transferencia de ficheros entre el sistema de videovigilancia y la plataforma web

En esta sección se pretende explicar la comunicación entre las dos Raspberry's Pi, el cual se ha desarrollado en el sprint 8. Por un lado está la *Raspberry Pi #A* que actúa de sistema de videovigilancia, tal y como se ha ido explicado en los informes anteriores. Por otro lado, la Raspberry Pi #B como servidor web para proporcionar al usuario una web en el que contenga un historial con todo lo que el sistema de videovigilancia haya captado.

De aquí al final del ANEXO la Raspberry Pi que actúa como sistema de videovigilancia será llamado *Raspberry Pi #A* y la Raspberry que actúa como servidor web, *Raspberry Pi #B*.

En una misma Raspberry Pi sería capaz de actuar como sistema de videovigilancia y de servidor web. El problema es que en el caso de que dicha Raspberry Pi se extraviara perderíamos toda la información recogida por el sistema. Es por ello, que se hace uso de una segunda Raspberry Pi el cual permite asegurar dicha información.

Tal y como se ha comentado anteriormente, en el momento en que el sistema de videovigilancia detecte alguna presencia, tanto la captura de imagen como la captura de video, no solo se enviará al usuario mediante la aplicación Telegram, sino que se enviará automáticamente al servidor web. Es por ello, que para dicha comunicación se establece una capa de seguridad entre la conexión de la Raspberry Pi #A y la Raspberry Pi #B.

Para la transmisión de los archivos que envía la Raspberry Pi #A a la Raspberry Pi #B se hace a través del protocolo *SCP*. El protocolo *SCP* permite la transferencia segura de archivos entre dos host. Para que esta transferencia sea segura y autenticada se hace uso del protocolo *SSH*. El protocolo *SSH* sirve para acceder a maquinas remotas a través de la red, en este caso, a la Raspberry Pi #B que actúa de servidor web. Todo lo que se transmita por *SSH* va encriptado.

Como se ha mencionado anteriormente, el protocolo *SSH* usa autenticación, esto quiere decir, que la Raspberry Pi #B le pedirá en cada conexión que solicite a la Raspberry Pi #A una autenticación. Para evitar que la Raspberry Pi #A se conecte la Raspberry Pi #B con autenticación se hará uso de una clave. Esto quiere decir que se realizará una identificación *SSH* mediante clave [2]. Se trata de una clave *RSA*.

Para ello, primeramente se ha de generar un conjunto de claves de cifrado asimétrico empleando el algoritmo *RSA* en la Raspberry Pi #A. En este momento, hay generados dos claves: clave privada y clave pública. La clave privada solo será conocida por la Raspberry Pi #A, en cambio, la clave pública la ha de compartir con la Raspberry Pi #B.

En la figura 6 se visualiza gráficamente el proceso anterior:

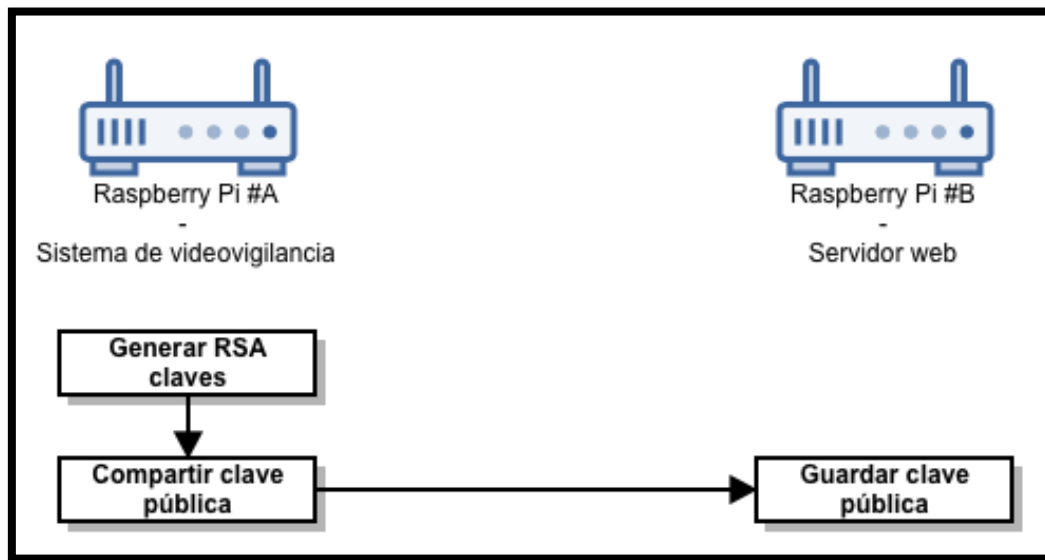


Figura 9. Proceso de compartir clave pública

Una vez que se ha compartido la clave pública con la Raspberry Pi #B (servidor web), el proceso para que la Raspberry Pi #A acceda a la Raspberry Pi #B mediante SSH es la siguiente:

La Raspberry Pi #A hace uso de RSA para acceder a la Raspberry Pi #B. Para ello, envía su clave pública. Cuando la Raspberry Pi #B recibe dicha clave pública realiza una búsqueda en el fichero que contiene todas las claves públicas. Si encuentra la clave pública recibida cifra un número aleatorio a partir de esa clave pública y se la enviará a la Raspberry Pi #A. En el caso de que no encuentre la clave pública en su fichero de claves denegará el acceso. Una vez enviado el número aleatorio cifrado, la Raspberry Pi #A recibirá el mensaje (el cual contiene el número cifrado) y a partir de su clave pública dará paso al descifrado. Una vez haya descifrado el mensaje (dando lugar a un número) lo enviará a la Raspberry Pi #B. La Raspberry Pi #B comprueba que ese número devuelto por parte de la Raspberry Pi #A coincide con el número aleatorio que mandó cifrado. En el caso de que los dos números coincidan, permitirá el acceso a la Raspberry Pi #A y así podrá hacer la transferencia de la imagen y video que haya captado el sistema. En caso de no coincidir, se deniega el acceso.

En la Figura 7 se resume de manera gráfica esta comunicación para el acceso entre las dos Raspberry's Pi:

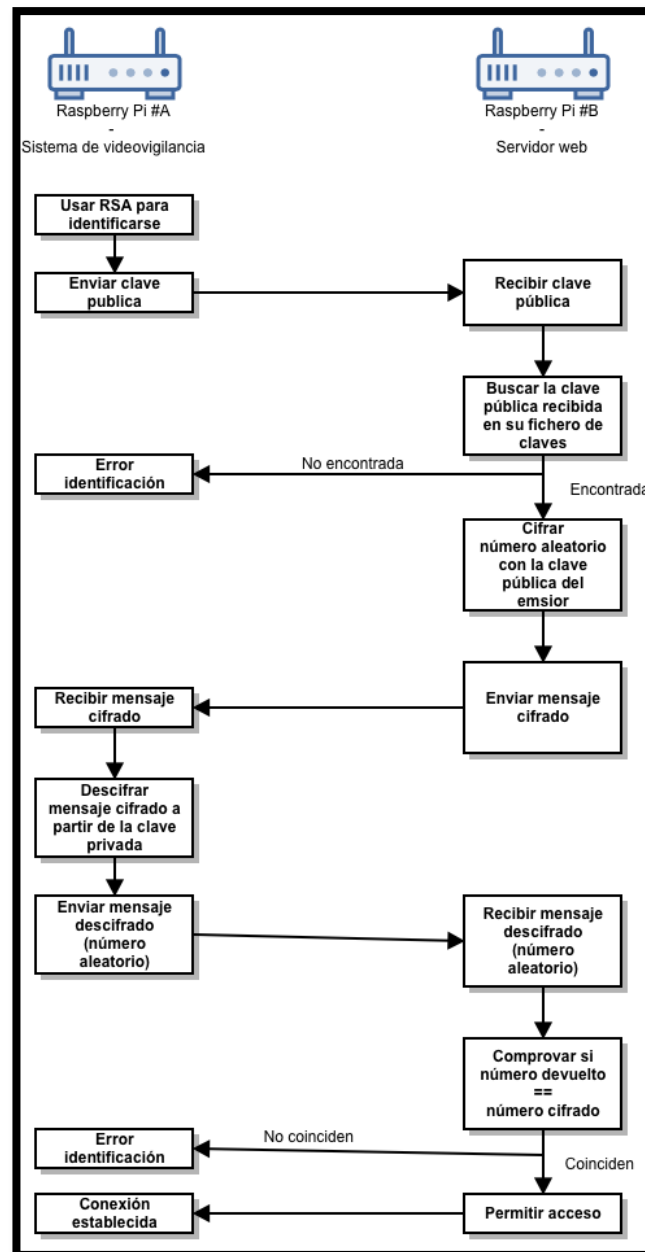


Figura 10. Arquitectura comunicación Raspberry's Pi

Este sistema le da una capa de seguridad a los datos transferidos de una Raspberry a otra.

## ANEXO E

### Arquitectura sistema de videovigilancia y servidor web con sockets

En esta sección se detalla la arquitectura de comunicación entre el sistema de videovigilancia y el servidor web para hacer posible la implementación del *video streaming* por parte del sistema de videovigilancia en el servidor web.

Ambas partes trabaja unidas, es por ello que es necesario que haya una comunicación entre las dos. Por un lado tenemos el sistema de videovigilancia que será el encargado de realizar el *video streaming*. Por otro, el servidor web que contiene la plataforma web en la que deberá visualizar el video streaming. Al tratarse de dos programas completamente diferentes surge la necesidad de hacer uso de los sockets para realizar una conexión entre ellas dos.

Tal y como se ha descrito anteriormente, un socket es un túnel de comunicación que ayuda a que dos programas se comuniquen. En este caso, se trata de una arquitectura cliente-servidor. El sistema de videovigilancia (cliente) enviara el "video streaming al servidor web (servidor) para que se pueda visualizar en la plataforma web.

Para su implementación, cada parte tiene una serie de características. El sistema de videovigilancia (cliente) ha de definir una conexión utilizando el socket creado a partir de una dirección IP y puerto hacia donde se iniciara la conexión, es decir, al servidor web. Después, debe solicitar el inicio de una conexión con el servidor el cual le responderá diciéndole que está conectado correctamente. Una vez establecida la conexión, es posible el intercambio de comunicación. Respecto al servidor web (servidor), se crea una conexión utilizando el socket, indicando una dirección IP y puerto por el que se iniciará la conexión del socket. A partir de aquí, se iniciaría un bucle infinito en el que estará todo el tiempo en modo escucha del cliente.

Una vez que tenemos las dos partes configuradas a través de los sockets, es posible él envió de información. Para ello, en la parte del sistema de videovigilancia se ha de transformar los fotogramas capturados por el sistema de videovigilancia en binario. Esto es necesario ya que la información baja en forma de datos binarios por lo que se ha de encargar el sistema de videovigilancia en transformar la información a binario. Una vez hecho eso, se envía el paquete y el servidor web es quien lo recibe. Una vez el servidor web haya recibido el paquete, se dispone abrirlo y a transformar de nuevo a su tipo original.

Esto hace posible que mientras el sistema de videovigilancia esté realizando un video streaming, por la parte del servidor web se podrá visualizar en la plataforma web dicho video streaming.

A continuación en la Figura 11 se detalla la arquitectura cliente-servidor escrita en el ANEXO para comprender mejor su funcionamiento.

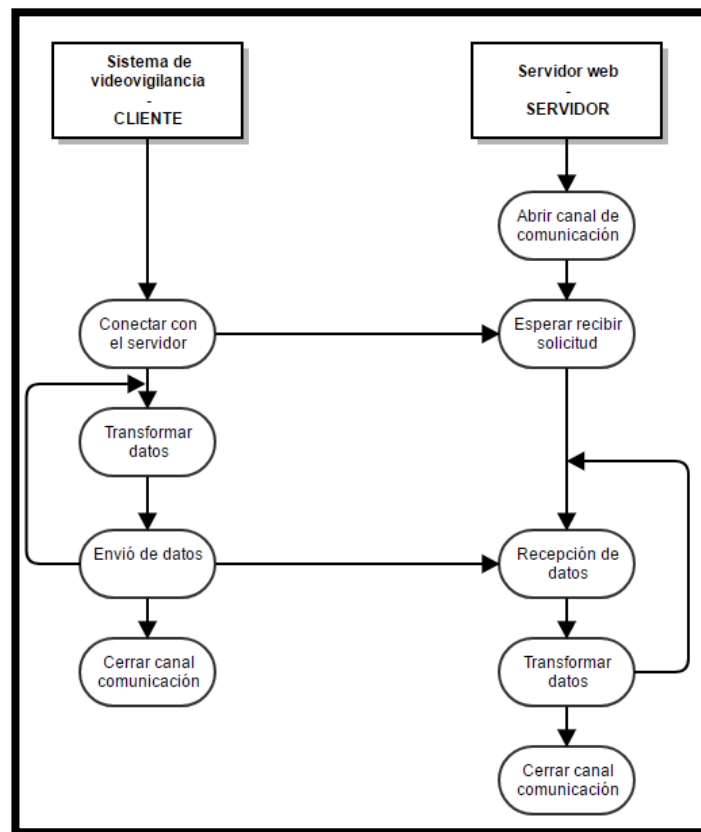


Figura 11. Arquitectura sistema videovigilancia – servidor web con sockets