Facultad 4

Centro de Informática Industrial

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Tema: Aplicación web para monitoreo y control de una cámara Pan Tilt basada en servomotores.

Autor: Richard Anca Perurena

Tutores: Ing. Julio Alberto Leyva Durán

MSc. Sailyn Salas Hechavarría

Co Tutor: Ing. Yassiel Gómez Díaz

La Habana, 2017

"El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia"

Fidel Castro

Agradecimientos:

Quiero agradecerle ante todo a mis padres Julián y Dailé por su apoyo incondicional, por estar ahí cada vez que los he necesitado, por darlo todo sin esperar nada a cambio, los quiero con la vida.

A mis Abuelos Deysi y Santos que han sido siempre como mis segundos padres.

A mi hermano Henry que, aunque jode bastante siempre puedo contar con él.

A mi novia Yusi, que me ha brindado siempre su apoyo y demostrado que puedo contar con ella para lo q necesite.

A mi tutor Julio que confió en mí para esta tarea, me apoyó siempre que lo necesité.

A mi tutora Sailyn por su apoyo y su ayuda en el transcurso de la carrera.

A Claudia que fue como una tutora más y me guió en el transcurso de este proyecto.

A Delvis que sobre todas las cosas lo considero como mi hermano.

Al Flaco (David) que se sentó a estudiar conmigo y me dedicó bastante tiempo cada vez que lo necesité.

A Miguel, Yosvani, Oscar, Jorgito, Alik, Felix, Liosvel, los Gustavos, Rodolfo y Adrian por estar siempre ahí cuando los he necesitado sin decir nunca que no.

A Dailenis que me ayudo en el trascurso de la carrera y me dedicó todo el tiempo del mundo cada vez que la necesité.

Al piquete del edificio y del aula, a Rogelio, Gabriel, el Lamis, el Rafa y Addiel. A Pavel, el Guille, Yamira y Yaima.

A los profesores de la Facultad 5 y los trabajadores del centro.

A todas las personas que de una forma u otra ayudaron a la realización de este trabajo.

Dedicatoria:

Dedico el presente trabajo a mis padres ***Julián Anca*** *y* ***Dailé Perurena****.*

Declaración de Autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los _____ días del mes de _____ del año _____.

Firma del autor

Resumen

Las Tecnologías de la Información y las Comunicaciones juegan un papel relevante en la economía mundial, siendo estas, un factor influyente en cualquier esfera de la sociedad. Su continuo avance ha influido en el desarrollo de la automatización de las cámaras de vigilancia, específicamente las cámaras Pan Tilt. Las palabras *pan* y *tilt* se traducen como panorámica e inclinación, las cuales son los movimientos que realiza dicha cámara. Estas cámaras nos permiten de manera remota visualizar las áreas en tiempo real y están compuestas por una cámara, dos servomotores y un sistema de control. El presente trabajo persigue como objetivo desarrollar un prototipo de cámara Pan Tilt y una aplicación web, que permita visualizar en tiempo real lo capturado por la cámara y, girar la misma sobre sus ejes horizontal y vertical controlando sus movimientos mediante una Raspberry Pi. Para el desarrollo del sistema se emplea la metodología de desarrollo AUP-UCI, el lenguaje de programación Python desde el lado del servidor y HTML Java Script y CSS del lado del cliente. Además de la biblioteca PWM para el control de ancho de pulso. Como resultado se logra un prototipo de cámara Pan Tilt capaz de recorrer 180 grados de forma horizontal y vertical. Además, una aplicación web capaz de mostrar al usuario el flujo de imágenes capturado por la cámara en tiempo real y permitirle al mismo controlar los movimientos de la plataforma.

Palabras claves: Cámara Pan Tilt, cámara, servomotor, sistema de control, Raspberry Pi.

Abstract

Information and Communication Technologies play an important role in the world economy, and these are an influential factor in any sphere of society. Its continuous advance has influenced the development of the automation of the surveillance cameras, specifically the Pan Tilt cameras. The words pan and tilt are translated as panoramic and vertical, which are the movements that the camera performs. These cameras allow us to remotely visualize the areas in real time and are composed of a camera, two servomotors and a control system. The objective of this work is to develop a Pan Tilt camera prototype and a web application that allows real time visualization of the captured camera and rotates it on its horizontal and vertical axes, controlling its movements using a Raspberry Pi. The development of the system uses the development methodology AUP-UCI, the Python programming language from the server side and HTML Java Script and CSS on client side. In addition to the PWM library for pulse width control. As a result, a prototype Pan Tilt camera able to rotate 180 degrees horizontally and vertically was obtained. In addition, a web application capable of showing the user the flow of images captured by the camera in real time and allow it to control the movements of the platform.

Keywords: Pan Tilt camera, camera, servomotor, control system, Raspberry Pi.

Índice

Índice de Imágenes

Índice de Tablas

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) juegan un papel relevante en la economía mundial, siendo estas, un factor influyente en cualquier esfera de la sociedad (1). A medida que han transcurrido los años, el hombre y las tecnologías han llevado un desarrollo acelerado, siendo, cada vez más necesario el uso de medios tecnológicos en las organizaciones. Estos medios a su vez necesitan ser controlados para preservar su integridad como activos de una organización, siendo así necesario el uso de equipos de vigilancia. Como parte de estos equipos de vigilancia están las cámaras, las cuales con su continuo avance han sido capaz de colocarse a un nivel bien elevado dentro de las instituciones públicas y privadas, así como en muchos de los hogares del mundo (2). Para su funcionamiento, es necesario el desarrollo de un software especializado a cada marca, modelo y tipo de cámara a utilizar, por lo que existen empresas desarrolladoras encaminadas precisamente a este trabajo.

En Cuba se han desarrollado numerosas aplicaciones informáticas para contribuir a desarrollar la economía, así como la informatización de empresas tanto del país como del exterior. Una de las instituciones que ha dado grandes pasos en este sentido es la Universidad de las Ciencias Informáticas (UCI). Este es un centro con características especiales determinada por la combinación Docencia–Producción, tiene entre sus objetivos lograr una competitividad nacional e internacional de la industria cubana del software, así como insertarse en el mercado mediante sus líneas de productos y servicios. La UCI cuenta con varios centros productivos, que se encargan de poner en alto el nombre de la universidad, así como su prestigio a nivel nacional e internacional.

Como parte de los centros mencionados anteriormente, se encuentra el Centro de Informática Industrial (CEDIN), este cuenta con la línea de Sistemas Embebidos la cual se dedica al desarrollo de prototipos de dispositivos y software para los mismos, también se dedica a desarrollar software (aplicaciones y manejadores) para dispositivos desarrollados por terceros, además de software para dispositivos móviles o para tarjetas de hardware abierto como Raspberry Pi o Arduino. En la línea se dispone de componentes que permiten el desarrollo de un prototipo de cámara Pan Tilt y con un firmware desarrollado en la misma el cual permite, desde una aplicación desktop o móvil controlar el desplazamiento horizontal y vertical de la plataforma.

Actualmente en la línea de Sistemas Embebidos se tiene una cámara accesible por interfaz USB, la cual no cuenta con manejadores que permitan la comunicación entre los dispositivos de visualización y la misma. Lo antes plateado trae consigo que sea imposible visualizar el flujo de video capturado por la cámara, dificultando esto el monitoreo de las áreas de forma remota y la visualización en tiempo real de lo que sucede. Además, la plataforma PT solo se puede controlar desde dispositivos que tengan previamente

instalada la aplicación. Dada la diversidad de sistemas operativos que existen en la actualidad, los clientes pueden tener instalado Linux, Windows o Android, trayendo consigo la presencia de más de una aplicación en el proceso de control, por lo que puede existir una desactualización del software o incompatibilidad entre las diferentes aplicaciones. Todos estos elementos impiden cumplir de forma correcta el principal objetivo de tener esta cámara, el cual, no es más que vigilar los activos y medios de la organización.

Ante la problemática planteada anteriormente se define como **problema** a resolver: ¿Cómo acceder a los periféricos de una plataforma de cámara PT para el monitoreo de las áreas de forma remota?

Objeto de Estudio: Monitoreo y control de forma remota de una cámara PT basada en servomotores.

Para dar solución al problema de la investigación científica se define como **objetivo general**: Desarrollar una aplicación web que permita visualizar el *stream* de video en tiempo real y controlar los movimientos horizontal y vertical de la plataforma PT para el monitoreo de las áreas de forma remota.

Campo de Acción: Control de cámaras.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes tareas de investigación:

1. Estudio de los protocolos de comunicación con la plataforma Pan Tilt (acceso a los motores de paso Horizontal y Vertical, y la cámara de video incorporada).
2. Definición de los requisitos funcionales para el desarrollo de la solución propuesta.
3. Selección de estándares, protocolos de comunicación, metodologías, herramientas y tecnologías a usar para el desarrollo del proyecto.
4. Confección de los artefactos relacionados con el análisis y diseño.
5. Diseño e implementación de la solución propuesta.
6. Elaboración de pruebas y corrección de los errores en la solución.

El presente trabajo se desarrolló con el empleo de los **métodos científicos de investigación** que se describen a continuación:

De nivel teórico:

- **Histórico–Lógico:** En la presente investigación se utilizó este método para realizar el estudio del arte de los sistemas o soluciones similares. Además de los lenguajes, herramientas y metodologías para el desarrollo de una aplicación web para el monitoreo y control de una cámara Pan Tilt.
- **Modelación:** Se empleó para realizar los diagramas necesarios para darle cumplimiento a los requisitos funcionales y no funcionales asociados al sistema de monitoreo y control propuesto.

De nivel empírico:

- **Consulta de toda fuente de información:** Se consulta la información referente al funcionamiento de las cámaras Pan Tilt, así como otras fuentes que tributen al adecuado desarrollo de la investigación.
- **Pruebas de validación:** Se realizaron las pruebas pertinentes a la aplicación de monitoreo y control para así tener constancia del cumplimiento de los objetivos.

El documento contará para una mejor comprensión del mismo, con la siguiente estructura capitular:

Capítulo 1. Fundamentación teórica: En este capítulo se presentan los principales conceptos que se deben tener en cuenta para un mejor entendimiento del trabajo, además del estudio y selección de las herramientas, tecnologías y metodología para el desarrollo de la propuesta de solución.

Capítulo 2. Análisis y diseño del sistema: En este capítulo se realiza una descripción del sistema que se pretende desarrollar, se elaboran las Historias de Usuarios para identificar las funcionalidades con las que debe cumplir la aplicación, así como los principales requerimientos no funcionales, además de presentar la arquitectura seleccionada para el desarrollo del sistema.

Capítulo 3: En el capítulo se realiza la validación de la solución propuesta.

Capítulo 1 Fundamentación Teórica

En el presente capítulo se introducen las principales características y conceptos asociados a las cámaras de vigilancia Pan Tilt para lograr una mejor comprensión del tema en cuestión. Además, se hace una descripción de las principales tecnologías que se emplean en el desarrollo del *software*, incluyendo: los lenguajes de programación, *framework*, y las herramientas de desarrollo empleadas.

1.1 Plataforma y cámara Pan Tilt

Pan Tilt (PT), significa Pan(Panorámica), y Tilt(Inclinación) por sus siglas en inglés. Esto se refiere a las capacidades de estas plataformas de moverse sobre sus ejes horizontal y vertical respectivamente. Esencialmente, una plataforma PT se establece en un lugar en particular, pero tiene la habilidad de girarse, usualmente en un círculo completo, inclinarse en ángulos diferentes hacia arriba y hacia abajo, los movimientos se los debe a los motores de paso acoplados en su plataforma (3). Estas plataformas por lo general tienen acoplada una cámara de video, lo que las convierte en cámaras Pan Tilt.

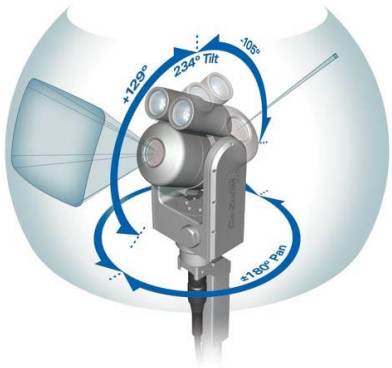


Ilustración 1 Cámara Pan Tilt.

1.1.1 Motores paso a paso

Un motor "**paso a paso**" (o "**PAP**") es un dispositivo electromecánico capaz de interpretar una serie de impulsos eléctricos como desplazamientos angulares discretos. Esto significa que, a diferencia de un motor convencional (que gira de forma continua), es capaz de avanzar una serie de grados (o pasos) a la vez, dependiendo del estado de sus entradas de control. Un motor paso a paso se comporta de la misma manera que un convertidor digital-analógico y puede ser gobernado por impulsos procedentes de sistemas lógicos, tales como microcontroladores u ordenadores (4).

Ilustración 2 Motor de paso.

1.1.2 Servomotor

Los servomotores son dispositivos electromecánicos que consisten en un motor eléctrico, un juego de engranes y una tarjeta de control, todo confinado dentro de una carcasa de plástico. Los servomotores funcionan por medio de modulación de ancho de pulso (PWM por sus siglas en inglés). Las duraciones de cada pulso se interpretan como comandos de posicionamiento del motor, mientras que los espacios entre cada pulso son despreciados. En la mayoría de los servomotores los anchos de pulso son de 1 ms a 2 ms, que cuando son aplicados al servomotor generan un desplazamiento de -90° a +90° por lo que, de una manera más sencilla, el ángulo de giro está determinado por el ancho de pulso; si el ancho de pulso fuera de 1.5 ms, el motor se posicionará en la parte central del rango a 0° (5).

Ilustración 3 Servomotor(5).

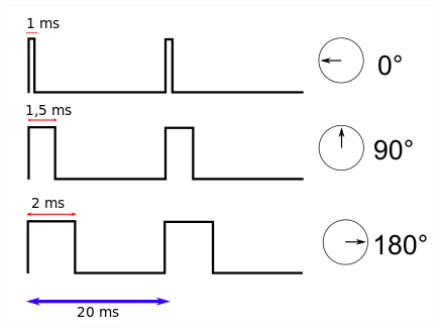


Ilustración 4 Secuencia de Pulsos (5).

1.1.3 Diferencia entre Motor de paso y Servomotor

Un motor paso a paso es esencialmente un servomotor que utiliza un método diferente de motorización. Un motor servo utiliza la rotación continua de un motor Corriente Directa (DC por sus siglas en inglés) controlando el giro con el circuito de control integrado, mientras que los motores paso a paso utilizan múltiples electroimanes dentados dispuestos alrededor de un engranaje central para definir su posición (6).

1.1.4 Cámara de Vigilancia

Una cámara es un aparato que sirve para registrar imágenes estáticas o en movimiento (7). Actualmente las cámaras de vigilancia, representan un papel muy importante para el avance de la tecnología en cuanto a la seguridad tanto en hogares como en sectores comerciales, empresas e instituciones. Existe una gran variedad de cámaras de seguridad en cuanto a modelo y forma, pero básicamente existen dos tipos, las cámaras de seguridad analógicas y las de protocolo de internet (IP, por sus siglas en inglés).
Las cámaras de seguridad analógicas son las que siempre se han usado en CCTV (Circuito Cerrado de Televisión), hoy día se mantiene su uso vigente. La imagen sale en estos equipos de manera analógica, debido a una señal de corriente alterna que varía en el tiempo con diferente amplitud. Poseen una salida de 75 ohm, es por eso que se requiere el uso de cable Coaxial o Par Trenzado (8).

Las Cámaras IP (también conocidas como cámaras Web o de Red) son videocámaras especialmente diseñadas para enviar las señales (video, y en algunos casos audio) a través de Internet desde un explorador (por ejemplo, Mozilla Firefox) o a través de concentrador (un HUB o un SWITCH) en una Red Local (LAN). En las cámaras IP pueden integrarse aplicaciones como detección de presencia (incluso el envío de mail si detectan presencia), grabación de imágenes o secuencias en equipos informáticos (tanto en una red local o en una red externa (WAN), de manera que se pueda comprobar por qué ha saltado la detección de presencia y se graben imágenes de lo sucedido (9).

Hasta este punto se han abordado los conceptos de cámara de seguridad y motores de paso, así como los de servomotores y las cámaras Pan Tilt. A continuación, se hace énfasis en el tema de *live stream* y las herramientas más utilizadas en la actualidad para realizarlo.

1.2 *Live-streaming*

La **transmisión en vivo** o *live-streaming* (por su significado en inglés) es una gran manera de interactuar en tiempo real. Sirve tanto para compartir eventos en directo, realizar entrevistas, mostrar cómo se crea un producto, realizar sesiones de preguntas-respuestas como para monitorear un área de forma remota (10).

1.2.1 Métodos para realizar una transmisión live-streaming

Existen varias formas de realizar una transmisión *streaming*, la solución adecuada para un evento puede no ser la misma para otros. Para realizar el *streaming* en algún lugar se necesita codificar la señal de video en un formato de *streaming*, lo cual debe suceder en el lugar del evento, para esto el hardware debe tener la capacidad de codificar o convertir el video en tiempo real, de otra forma dejaría de ser *live-streaming* (11). Hay gran variedad de aplicaciones que realizan *streaming* de video, algunas de ellas son las siguientes:

VLC

VLC *media player* que puede ser utilizado como servidor y como cliente para transmitir y recibir flujos de red. VLC es capaz de transmitir todo lo que puede leer (12).

VLS

(VideoLAN Server), que puede transmitir archivos MPEG-1, MPEG-2 y MPEG-4, DVD, canales digitales por satélite, canales de televisión terrestres digitales y videos en vivo en la red en unicast o multicast. La mayoría de la funcionalidad VLS ahora se puede encontrar en VLC (12).

WebRTC

WebRTC, también conocido como *Web Real-Time Communications* (Comunicación Web en Tiempo Real por su significado), es un proyecto de código abierto – promovido por Google y Mozilla– que permite comunicaciones en tiempo real sin plugins a través de una API Javascript. Facilita las aplicaciones de llamadas de voz, chat de video y compartimiento de archivos entre navegadores (13).

Motion

Permite la grabación de videos todo el tiempo o solo cuando hay movimiento y puede enmarcar donde ha detectado el movimiento en la imagen grabada. También captura fotogramas igual que un video, pero grabando imágenes sueltas. Permite realizar *streaming* en vivo de las fuentes de origen, aunque no esté grabando nada se puede ver en vivo lo que las cámaras estén captando (14).

1.2.2 Ventajas de utilizar Motion

Luego de realizar un esbozo de las principales aplicaciones que permiten realizar *live-streaming*, se decidió utilizar Motion. Este permite capturar fotograma por fotograma, garantizando así que no se tenga que codificar un video en tiempo real y se pueda utilizar un hardware que no sea tan potente como una PC convencional.

Después de seleccionar Motion como la herramienta adecuada para realizar el envío en vivo del flujo de video, se abarcan conceptos relacionados a la Web y una serie de protocolos y estándares asociados.

1.3 Web

World Wide Web(www), literalmente telaraña de alcance mundial, es un término usado en informática cuya traducción podría ser Red Global Mundial o "Red de Amplitud Mundial", es un sistema de documentos de hipertexto o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario

visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces (15).

1.4 Servicio Web

Programa informático que permite la comunicación y el intercambio de datos entre aplicaciones y sistemas heterogéneos en entornos distribuidos. Los servicios web son por ende un conjunto de funcionalidad expuesta en una intranet o a través de Internet, por y para aplicaciones y computadoras sin la intervención humana. Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet (16).

1.5 Sitio Web

Conjunto organizado y coherente de páginas Web que tiene como función ofrecer, informar, publicitar o vender contenidos, productos y servicios al resto del mundo. Para que un sitio Web pueda ser visitado por otras personas es necesario que se encuentre alojado en un servidor. Se trata de una computadora conectada a la World Wide Web con espacio en disco y conectividad suficiente para albergar sitios y servirlos al resto de la comunidad de usuarios de Internet a través de direcciones IP o nombres de dominio (17).

1.5.1 Conjunto de Protocolos y Estándares

Los servicios Web implementan su lógica mediante la utilización de estándares. Para el transporte suele utilizarse TCP/IP, URI/URN/URL, MIME, HTTP/SMTP o SSL/TLS. Para el contenido suele utilizarse XML y SOAP. La utilización de estándares permite que cualquier tecnología que utilice esos estándares pueda hacer uso de estos servicios web, facilitando así la interoperabilidad de las aplicaciones (30).

Estándares:

- Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML (*Extensible Markup Language*): Es el formato estándar para los datos que se vayan a intercambiar.
- SOAP (*Simple Object Access Protocol*) o XML-RPC (XML *Remote Procedure Call*): Protocolos sobre los que se establece el intercambio.

- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos como: HTTP (*Hypertext Transfer Protocol*), FTP (*File Transfer Protocol*), o SMTP (*Simple Mail Transfer Protocol*).
- WSDL (*Web Services Description Language*): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- UDDI (*Universal Description, Discovery and Integration*): Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.
- WS-Security (*Web Service Security*): Protocolo de seguridad aceptado como estándar por OASIS (*Organization for the Advancement of Structured Information Standards*). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados (30).

A partir de este punto, luego realizar un esbozo sobre las principales particularidades de la web, se abarcan los conceptos asociados a los principales protocolos de comunicación.

1.6 Protocolos de comunicación

Los protocolos de comunicaciones definen las normas que posibilitan que se establezca una comunicación entre varios equipos o dispositivos, ya que estos equipos pueden ser diferentes entre sí. Entre los principales protocolos de comunicación se encuentran los siguientes (18).

1.6.1 Protocolo UDP

UDP por sus siglas en inglés *User Datagram Protocol* es un protocolo no orientado a conexión. Es decir, cuando una maquina A envía paquetes a una maquina B, el flujo es unidireccional. La transferencia de datos es realizada sin haber realizado previamente una conexión con la máquina de destino (maquina B), y el destinatario recibirá los datos sin enviar una confirmación al emisor (la maquina A). Esto es debido a que la encapsulación de datos enviada por el protocolo UDP no permite transmitir la información relacionada al emisor. Por ello el destinatario no conocerá al emisor de los datos excepto su IP (19).

1.6.2 Protocolo TCP

Contrariamente a UDP, el protocolo TCP por sus siglas en inglés *Transmission Control Protocol* está orientado a la conexión. Cuando una máquina A envía datos a una máquina B, la máquina B es informada de la llegada de los datos, y confirma su buena recepción. Aquí interviene el control de verificación por redundancia cíclica (CRC por sus siglas en inglés) de datos que se basa en una ecuación matemática que permite verificar la integridad de los datos transmitidos. De este modo, si los datos recibidos son corruptos,

el protocolo TCP permite que los destinatarios soliciten al emisor que vuelvan a enviar los datos corruptos (36).

1.6.3 Protocolo MMS

Microsoft Media Server (MMS), es un protocolo de transmisión de red propietario de Microsoft, sirve para transferir datos unicast en *Windows Media Services*. MMS se puede transportar a través de UDP o TCP. El puerto por defecto de MMS es UDP / TCP 1755 (20).

1.6.4 Protocolo HTTP

El protocolo de transporte de hipertexto HTTP, protocolo encargado de controlar la transferencia de datos en la World Wide Web, el cual proporciona un vehículo de entrega para las imágenes, gráficos, video, hipertexto u otros datos en la Web. HTTP es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener el estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto permite a las aplicaciones web instituir la noción de sesión, y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado (40).

El protocolo que se escoge es http porque cumple con los requisitos necesarios para enviar y recibir información, además tiene la capacidad de transportar video e imágenes. A partir de este se selecciona la arquitectura del proyecto, la cual queda explicada en el siguiente epígrafe.

1.7 Arquitectura Cliente - Servidor

La Arquitectura Cliente Servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario (21).

1.7.1 Características de la arquitectura Cliente/Servidor

Permite la combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras o módems. Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del

disco y dispositivos entrada salida. Cliente/Servidor establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red (22).

Hasta este punto se han abarcado los conceptos referentes a cámara Pan Tilt, servomotor y *live stream,* además de los conceptos asociados a la web y los protocolos de comunicación más usados en este tipo de proyecto. La cámara Pan Tilt además de los servomotores que facilitan el movimiento y la cámara que captura las imágenes, cuentan con un hardware capaz de entregar la información de la cámara al usuario, y permitirle a este último, interactuar con los motores para lograr los movimientos horizontal y vertical de la plataforma, lo que las convierte en sistemas embebidos. A continuación, se abarca el tema de los sistemas embebidos y las principales plataformas utilizadas en ellos por el CEDIN.

1.8 Sistemas embebidos

Los sistemas embebidos son dispositivos integrados en productos, que controlan una o varias funciones con recursos limitados y en condiciones ambientales hostiles. Por ejemplo, los sistemas de transporte como ascensores, trenes y coches que se utilizan frecuentemente contienen un gran número de sistemas embebidos que se encargan de manera “transparente para el usuario” del control del movimiento, conectividad, autodiagnóstico, gestión energética, confort y seguridad de las personas. Estos sistemas de procesamiento que integran hardware, software y comunicaciones son capaces de dotar a los objetos en los que están integrados de capacidades como confiabilidad (seguridad, disponibilidad, fiabilidad, mantenibilidad), inteligencia, conectividad, gestión energética, interacción con el entorno y productividad (23).

1.8.1 Microcontroladores

Un microcontrolador es un circuito integrado digital que puede ser usado para muy diversos propósitos debido a que es programable. Está compuesto por una unidad central de procesamiento (CPU), memorias (ROM y RAM) y líneas de entrada y salida (periféricos) (24).

Características

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o CPU (Unidad Central de Proceso).
- Memoria RAM para Contener los datos.
- Memoria para el programa tipo ROM/PROM/EPROM.

- Líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos (temporizadores, Puertas Serie y Paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico) (25).

Unos de los microcontroladores más populares son los Arduino y los PIC, a continuación, se muestran algunas características de ambos.

Arduino. Es una plataforma de hardware libre basada en un hardware y software de fácil uso. Es destinada a cualquiera que desee desarrollar proyectos interactivos. Las tarjetas Arduino son capaces de leer un sensor de luz, detectar si un botón es presionado, hacer girar un motor o encender un led (26).

PIC. Es una familia de microcontroladores fabricados por *Microchip Technology Inc*. El nombre completo es PICmicro, aunque generalmente se utiliza como *Peripheral Interface Controller* o Controlador de Interfaz periférico. Microchip ofrece soluciones para toda la gama de rendimiento de los microcontroladores de 8 bits, 16 bits y 32 bits, con una arquitectura potente, tecnologías de memoria flexibles, herramientas de desarrollo completas fáciles de usar, documentación técnica completa y soporte de post-diseño a través de una red global de ventas y distribución (27).

1.8.2 Single Board Computer SBC

Las siglas SBC significan, *Single Board Computer* o computador de placa única por su significado en español. Esto quiere decir que, a diferencia de los ordenadores tradicionales, los SBC son placas que contienen todos, o la mayor parte de los componentes de un ordenador. La principal característica de los SBC u ordenadores con placas SBC son sus reducidas dimensiones. Los ordenadores SBC están montados sobre placas con medidas pequeñas, desde tamaños relativos a un USB hasta medidas similares a una tarjeta de débito como la Raspberry Pi que mide 8,5 x 5,3 cm. Las placas SBC por lo general son económicas, normalmente sus precios no suelen sobrepasar los 100 USD. Estas placas ofrecen poca potencia de cálculo en comparación con un ordenador con CPU core i3 o core i7. A continuación algunos ejemplos de placas SBC (28).

1.8.3 Placas SBC

Raspberry Pi. La placa SBC más popular se llama Raspberry Pi. Es una pequeña placa que cuenta con varias versiones y que tiene una amplia comunidad. El proyecto nació para buscar hardware económico y libre para enseñar computación en los colegios de primaria. Actualmente y gracias a su comunidad se puede hacer casi cualquier cosa con esta placa, desde un servidor hasta un clúster pasando por ser el hardware de una pesada Tablet (28).

BeagleBone Black. Es la alternativa estadounidense a Raspberry Pi. Por lo general no suele existir mucha diferencia entre la potencia de esta placa con el resto, BeagleBone Black puede soportar Ubuntu o funcionar como un accesorio más del pc tradicional (28).

PcDuino. Es una placa SBC libre. PcDuino está basado en los esquemas de Arduino e incorpora lo necesario para ser una placa SBC, es decir: procesador y memoria RAM. A diferencia del resto, PcDuino es bastante grande, alcanza los 12 cm de largo por 6 cm de ancho. El último modelo de esta placa admite y soporta Ubuntu y Android (28).

	Raspberry Pi	**BeagleBone Black**	**PcDuino.**
Precio USD	37	55	60
USB	4	1	1 + 1 mini USB
CPU	1.2 GHz 64-bit quad-core	1-GHz	1-GHz dual-core
Almacenamiento	microSD	2 GB + microSD	4 GB + microSD
RAM	1 GB	512 MB	1 GB
GPIO	40 GPIO pines	5 puertos serial + 8 PWM + 7 convertidores analógico/digital	14 GPIO + 2 PWM

Tabla 1 Prestaciones SBC

1.8.4 Plataformas electrónicas empleadas en el dominio de sistemas embebidos del CEDIN

Entre las principales plataformas empleadas en el centro CEDIN para el desarrollo de aplicaciones embebidas se identifican Raspberry Pi y la placa Arduino. Aunque ambas están destinadas a resolver problemas similares y puedan lucir parecidas, son muy diferentes. Raspberry Pi es identificada como una microcomputadora completamente funcional, mientras que Arduino, es identificado como un microcontrolador. La principal diferencia entre estas pequeñas placas se encuentra en la velocidad de cálculo y la capacidad de memoria RAM, siendo Raspberry Pi mejor en este sentido. En la tabla 2 se pueden apreciar las principales características de estas pequeñas placas.

1.8.5 Comparación entre Arduino y Raspberry Pi

	 Ilustración 5 Arduino.	 Ilustración 6 Raspberry Pi.
Precio en Dólares	$30	$37
Tamaño	7.6 x 1.9 x 6.4 cm	8.6 x 5.4 x 1.7 cm
Memoria	0.002MB	1GB
Velocidad del Reloj	16MHz	1.2 GHz
Network	Ninguna	10/100 wired Ethernet RJ45, Wi-Fi 802.11n, Bluetooth 4.1
Multitarea	No	Si
Voltaje de entrada	7 a 12 V	5 V
Memoria Flash	32 KB	Tarjeta SD (2 a 16GB)
Puertos USB	Uno	Cuatro
Sistema Operativo	Ninguno	Distribuciones de Linux y Windows
Entorno de desarrollo integrado(IDE)	Arduino	Scratch, IDLE, cualquiera con soporte Linux.

Tabla 2 Comparación entre Arduino(5) y Raspberry Pi(7).

Teniendo en cuenta la comparación antes realizada se decidió utilizar Raspberry Pi 3 debido a que ésta plataforma está disponible para las condiciones de desarrollo actual en el centro CEDIN de la UCI. Raspberry Pi posee Bluetooth, Wifi, es de hardware y código abierto y además cumple con las políticas de software libre de la soberanía tecnológica del país.

1.8.6 Raspberry Pi 3

Raspberry Pi 3 no es más que una diminuta placa base de 85 x 54 milímetros en la que se aloja un chip Broadcom BCM2837 con procesador ARMv8 hasta a 1.2 GHz de velocidad, GPU VideoCore IV y hasta 1 Gbytes de memoria RAM. En cuanto a su precio, suele estar por debajo de los 40 dólares. Para que funcione, basta con que se añada un medio de almacenamiento (como por ejemplo una tarjeta de memoria microSD), enchufarlo a la corriente gracias a cualquier cargador de tipo micro USB (el mismo que sirve para recargar la mayoría de los teléfonos móviles). La fundación de Raspberry Pi pone a disposición desde su página web a Raspbian, una distribución de Linux basada en Debian. Raspberry Pi cuenta con un puerto de salida de video HDMI y otro de tipo mini Jack compuesto que exporta audio y video, además con cuatro puertos USB 2.0 a los que se puede conectar teclado, ratón y otros periféricos. Dispone de Ethernet RJ45, Wifi 802.11n y Bluetooth 4.1. Raspberry Pi es una microcomputadora completamente funcional, (29) pero aun así no logra tener un rendimiento como el de una PC de escritorio, ya que sus capacidades son inferiores.

1.8.7 GPIO Raspberry Pi

GPIO (*General Purpose Input/Output*) es, como su propio nombre indica, un sistema de E/S (Entrada/Salida) de propósito general, es decir, una serie de conexiones que se pueden usar como entradas o salidas para usos múltiples. Estos pines están incluidos en todos los modelos de Raspberry Pi, para que se puedan realizar proyectos interesantes como se hacen con Arduino. Los GPIO son pines que representan la interfaz entre la Raspberry Pi y el mundo exterior. Y con ellos se puede hacer multitud de proyectos, desde hacer parpadear un LED o hacer girar un motor hasta otros mucho más sofisticados (30).

1.8.8 Sistemas operativos

Conjunto de programas destinados a abstraer el hardware de un dispositivo al usuario, gestionando sus recursos de forma eficiente. Todas las aplicaciones del usuario se ejecutarán haciendo uso de las primitivas (funciones) que aporta el sistema operativo, evitando la comunicación directa de éstos con el hardware (31).

¿Por qué y para qué el uso de un SO?

- Ofrecen una visión menos compleja del hardware.
- Facilitan el manejo del hardware aportando funciones al usuario.
- Portabilidad de las librerías y programas.
- Hace parecer al usuario que se ejecutan todas las aplicaciones al mismo tiempo.
- Aportan seguridad en el funcionamiento del sistema y/o uso malintencionado de él mismo.

- Soporte ante posibles fallos de las aplicaciones que se ejecutan (31).

1.8.9 Sistemas operativos soportados por Raspberry Pi

Existe varios sistemas operativos soportados por Raspberry Pi, en su sitio oficial se pueden descargar distribuciones tanto de Linux como de Windows. Dentro de las distribuciones disponibles actualmente se encuentran Noobs, Raspbian, Ubuntu Mate, Snappy, Ubuntu Core, Windows 10 Iot Core, OSMC y Android. **Raspbian** es un sistema operativo libre basado en Debian optimizado para el hardware de Raspberry Pi. Raspbian proporciona más que un sistema operativo puro, viene con más de 35.000 paquetes, software recompilado incluido en un formato agradable para una fácil instalación en Raspberry Pi (32). Raspbian es el sistema operativo con soporte oficial de la fundación.

1.8.10 Raspbian OS

Raspbian OS es la distribución por excelencia para la Raspberry Pi. Es la más completa y optimizada de las existentes, por eso cuenta con apoyo oficial. Raspbian OS se basa en la potente distribución Debian Wheezy (Debian 7.0), optimizando el código de ésta para la Raspberry Pi. La distribución es ligera para moverse ágilmente en el hardware de la Raspberry Pi, comenzó con un entorno de escritorio LXDE y Midori como navegador web predeterminado, pero la fundación Raspberry Pi ha creado un entorno de escritorio especial llamado PIXEL (*Pi Improved Xwindows Environment Lightweight*). Además, incluye herramientas de desarrollo como IDLE para Python (33).

A partir de este punto se definen las tecnologías y herramientas a utilizar para realizar una aplicación web para el monitoreo y control de las áreas de forma remota, utilizándose para ello servomotores para garantizar el movimiento de la plataforma Pan Tilt, una cámara USB para capturar la imagen y una Raspberry Pi como hardware.

1.9 Tecnologías y herramientas usadas para el desarrollo de la propuesta de solución

En este epígrafe se hace referencia a la tecnología de comunicación, lenguaje y herramienta de modelado. Además, a los lenguajes de programación, entorno de trabajo y metodología de desarrollo a utilizar en el desarrollo de la aplicación web para el monitoreo y control de las áreas de forma a remota.

1.9.1 Tecnología de comunicación

Hoy día, la conexión entre dispositivos para el intercambio de datos se puede establecer mediante cables o de manera inalámbrica. En este caso la comunicación al servidor web se realizará mediante Ethernet. A continuación, se describe en que consiste esta tecnología.

Ethernet es la tecnología de red de área local más extendida en la actualidad, la norma IEEE 802.3 define las reglas para configurar una red Ethernet. Es una red CSMA/CD de banda base a 10 Mbps, que funciona con cableado coaxial fino y grueso, par trenzado y fibra óptica. Como tecnología de red maneja dos aspectos, la capa física y la lógica. La capa física y la capa de enlace de datos. La capa física describe las características físicas de la red y el hardware usado. Esta capa incluye: topología, hardware de transmisión y equipo usado (34).

Ethernet es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a Ethernet la tecnología ideal para la red de la mayoría los usuarios de la informática actual. En términos generales, Ethernet es un sistema para el trasporte digital de datos a través de sistemas de cómputo local (34).

1.9.2 Lenguaje de modelado

Lenguaje Unificado de Modelado (UML 8.0)

Lenguaje Unificado de Modelado (*Unified Modeling Lenguaje*, UML por sus siglas en inglés), es un lenguaje basado en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un sistema programado. Por otra parte, facilita a los integrantes de un equipo multidisciplinario participar e intercomunicarse fácilmente. UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones, ha sido ampliamente aceptado debido al prestigio de sus creadores. Hay que tener en cuenta que el estándar UML no es un proceso, no es una metodología de desarrollo, sino una notación, un lenguaje (35).

1.9.3 Herramientas de Modelado

Visual Paradigm for UML 8.0

Es una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador por su significado en español) profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Dentro de sus características se aprecia que soporta BPMN (Notación de Modelado de Procesos de Negocio) y UML versión 2.1, además permite:

- Construir Diagramas de Procesos de Negocio.
- Generar informes para generación de documentación.

- Distribución automática de diagramas, reorganización de las figuras y conectores de los diagramas UML.
- Dibujo de diagramas UML con plantillas (*stencils*) de MS Visio.
- Realizar el Modelado colaborativo con CVS y subversión.
- Editor de figuras (36).

1.9.4 Lenguajes de programación del lado del cliente

Los lenguajes del lado cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre tratamiento, es totalmente independiente del servidor.

Lenguaje de Marcas de Hipertexto (HTML)

El Lenguaje de Marcas de Hipertexto (*Hyper Text Markup Language*, HTML por sus siglas en inglés), no es más que un conjunto de etiquetas o comandos, complementados en la mayoría de los casos por extensiones que permiten dar formato a un archivo, con el objetivo básico de crear un documento que pueda ser visualizado en forma de Página Web y que además, pueda por medio de dichas etiquetas, tener la estructura o forma deseada por quien la diseñó. La principal ventaja de los lenguajes de etiquetas es que son muy sencillos de leer y escribir por parte de las personas y de los sistemas electrónicos. La principal desventaja es que pueden aumentar mucho el tamaño del documento, por lo que en general se utilizan etiquetas con nombres muy cortos (37).

JavaScript

Es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web, puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos (38).

Hojas de estilo en cascada (CSS 3.0)

Las hojas de estilo en cascada (*Cascading Style Sheets*), es la tecnología desarrollada por el *World Wide Web Consortium* (W3C) con el fin de separar la estructura de la presentación. CSS permite crear páginas web de una manera más exacta, gracias a ellas el desarrollador es mucho más dueño de los resultados finales de la página, logrando resultados imposibles de hacer utilizando solamente HTML, como fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo y posicionamiento avanzado. CSS3 está dividida en varios documentos separados llamados "módulos". Cada módulo añade nuevas

funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad. Las siguientes cuatro funcionalidades han sido publicadas como recomendaciones formales:

- Media Queries (publicado en 2012)
- Namespaces (publicado en 2011)
- Selectors Level 3 (publicado en 2011)
- Color (publicado en 2011)

Las novedades de CSS3 permite ahorrar tiempo y trabajo al poder seguir varias técnicas (bordes redondeados, sombra en el texto, sombra en las cajas) sin necesidad de usar un editor gráfico (37).

1.9.5 Lenguaje de programación del lado del servidor

Se clasifica así al lenguaje de programación en la tecnología cliente servidor que se ejecuta del lado del servidor y del cual los usuarios solo obtienen el beneficio del procesamiento de la información.

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, las cuales ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2 y Mac.
- Además, Python es gratuito, incluso para propósitos empresariales

Dentro de la gran gama de librerías que contiene Python se encuentra la librería RPIO, la cual cuenta con el módulo PWM que permite obtener una salida analógica artificial en un pin digital, facilitando así controlar las posiciones movimientos de los servomotores (39).

1.9.6 Django

Django es un *framework* de alto nivel de Python Web que fomenta un desarrollo rápido y un diseño limpio y pragmático. Construido por desarrolladores experimentados, se encarga de gran parte de la molestia de

desarrollo web, por lo que permite centrarse en escribir la aplicación sin necesidad de reinventar la rueda. Es de código abierto y gratuito (40).

1.9.7 Metodología de desarrollo

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce. Entre algunas de las metodologías más usadas en la actividad productiva de la Universidad de Ciencias Informáticas, se encuentran: *eXtreme Programming* (XP), Proceso Unificado de Rational (RUP), Proceso Unificado Abierto (OPEN UP), *Agile Unified Process* (AUP), y *Agile Unified Process* ajustada a la UCI (AUP-UCI), cuyas características veremos a continuación.

Agile Unified Process ajustada a la Universidad de las Ciencias Informáticas (AUP-UCI): Es una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Persigue como objetivo aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas, para ello nos apoyaremos en el modelo CMMI-DEV v1.3, el cual constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad.

La metodología a usar como guía para alcanzar el cumplimento del objetivo de la investigación es AUP-UCI en su escenario número 4, por las siguientes razones:

- Recomendable para proyectos de mediano alcance, en los que el negocio a informatizar esté bien definido y el equipo de desarrollo esté siempre acompañado por el cliente.
- Simplicidad: Todo se describe concisamente utilizando poca documentación.

Conclusiones parciales

Durante el desarrollo del capítulo se realizó un análisis de los principales conceptos referentes a las cámaras de vigilancia PT para una mejor comprensión del marco teórico de la investigación. Además, se definieron los servomotores para realizar los movimientos de la plataforma, ya que se puede situar la cámara en el ángulo deseado. Se selecciona Motion como herramienta para controlar el flujo de video gracias a su bajo consumo de recursos y facilidad de mostrar la información al usuario. Se definió la arquitectura cliente-servidor y como modelo arquitectónico Modelo Vista *Template*. Además, se decidió usar Raspberry Pi 3 como hardware para controlar los movimientos y gestionar el flujo de video. La metodología que guiará la evolución del desarrollo será AUP UCI por la flexibilidad y documentación de la misma.

Capítulo 2 Análisis y diseño de la solución propuesta

En el presente capítulo se presenta la propuesta del sistema y el prototipo de cámara Pan Tilt propuesto para dar solución a la situación problemática antes plateada. Además, se hace énfasis en los requerimientos del sistema, tanto en los funcionales como en los no funcionales. También se realizan las Historias de Usuario UH y sus estimaciones de esfuerzos e iteraciones correspondientes, se describe la solución y los patrones utilizados.

2.1 Propuesta del sistema

Para darle respuesta la situación problemática antes descrita se propone desarrollar un prototipo de cámara Pan Tilt con dos servomotores, una cámara web y una Raspberry Pi, además implementar una aplicación web para el control de la plataforma y el monitoreo de las áreas de forma remota. Este sistema estará diseñado para controlar los movimientos de la plataforma y visualizar en tiempo real el video capturado por la cámara. La aplicación permitirá al usuario escoger el tipo de movimiento de la plataforma, los diferentes tipos son:

- Manual: Permite el movimiento paso a paso de la cámara, donde un paso equivale a 9° hasta llegar a 180° tanto en el eje horizontal como en el vertical.
- Automático: Permite el movimiento automático de la plataforma sobre el eje horizontal permitiéndole al usuario realizar movimientos verticales de forma manual mientras ésta realiza su recorrido.

2.2 Prototipo de plataforma Pan Tilt

Para lograr el prototipo, se acoplan dos servomotores, el primero con su eje apuntando hacia la parte de bajo, acoplado a una base y el segundo de costado sobre el primero, quedando su eje de forma horizontal. De esta forma el primer servomotor garantizaría los movimientos panorámicos y el segundo los movimientos de inclinación. Como se muestra en la Ilustración número 7:

Ilustración 7 Servomotores acoplados.

La cámara web se acopla al segundo servomotor, quedando como en la figura número 8:

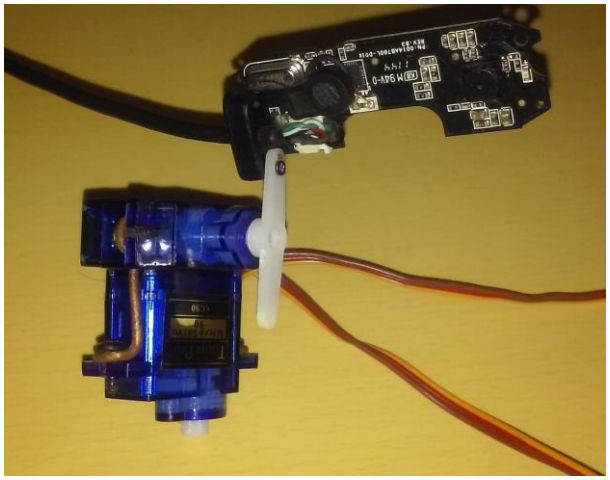


Ilustración 8 Cámara web acoplada a servomotores.

Cada servomotor tiene tres cables, carmelita, rojo y naranja los cuales corresponden a voltaje tierra y pulso respectivamente. Estos cables se conectan a la Raspberry Pi 3 conectándose los cables rojos de ambos servomotores en los pines 2 y 4, los cuales corresponden a 5V. Los cables carmelitas en los pines 6 y 9 los cuales corresponden a tierra. Luego se conecta el cable naranja del servomotor que controla el movimiento vertical al pin 3 el cual corresponde al GPIO 2 y en cable del motor horizontal al pin 5 correspondiente al GPIO 3. En las siguientes figuras se muestra a la izquierda en la ilustración 9 la distribución de pines que se utilizan de la Raspberry Pi 3 y a la derecha el prototipo de cámara Pan Tilt en la ilustración 10.

Ilustración 9 Pines usados Raspberry PI.

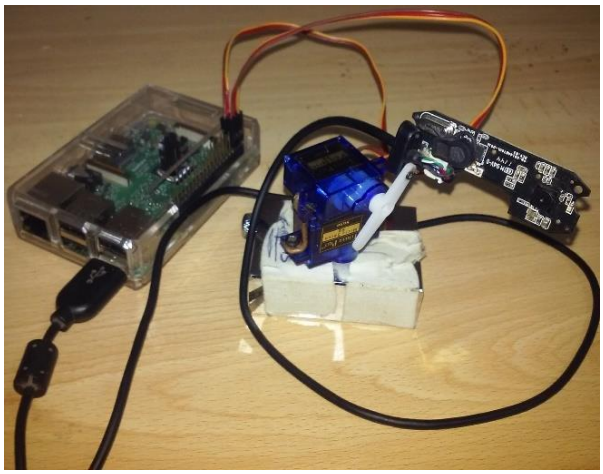


Ilustración 10 Prototipo de cámara Pan Tilt.

2.3 Requerimientos del sistema

2.3.1 Requisitos funcionales del sistema

Los requisitos funcionales representan el comportamiento que tendrá el sistema. Describen las funcionalidades que debe cumplir o que se espera que este provea. Deben ser lo más completo, claro y conciso posible (41). Además, pueden definirse a partir de reglas del negocio o la propia interacción de los usuarios. A continuación, se especifican los requisitos funcionales que responden a la propuesta solución:

1. Direccionar de forma manual la cámara hacia la derecha
2. Direccionar de forma manual la cámara hacia la izquierda.
3. Direccionar de forma manual la cámara hacia arriba.
4. Direccionar de forma manual la cámara hacia abajo.
5. Direccionar la cámara hacia su posición central.
6. Desplazar de forma automática la cámara en el eje horizontal comenzando hacia el lado izquierdo.
7. Desplazar de forma automática la cámara en el eje horizontal comenzando hacia el lado derecho.
8. Detener movimiento automático.
9. Visualizar el *stream* de video capturado por la cámara en tiempo real.

2.3.2 Requisitos no funcionales del sistema

Atributo de Calidad	Software
Sub-atributos/Sub-características	
Objetivo	La aplicación debe ser compatible con Firefox 40 o superior
Origen	Externo
Artefacto	
Entorno	Ejecución de la aplicación de control.
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
1.a Interacción con el sistema	
Medida de Respuesta	

Tabla 3. Requisito no funcional #1.

Atributo de Calidad	Seguridad
Sub-atributos/Sub-características	
Objetivo	Se debe garantizar la conexión vía Wifi por medio de una contraseña encriptada.
Origen	Externo
Artefacto	
Entorno	Ejecución de la aplicación de control.
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
1.a Interacción con el sistema	
Medida de Respuesta	

Tabla 4. Requisito no funcional #2.

2.4 Historias de Usuario

Las HU son la técnica que utilizan las metodologías ágiles (SCRUM, XP, AUP) para especificar los requisitos de software, estas deben ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a las tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra HU. Las estimaciones de esfuerzo, asociado a la implementación de las historias, la establecen los programadores, utilizando como medida el punto. Un punto, equivale a una semana ideal de programación (5 días laborables). Las historias, generalmente, valen de 1 a 3 puntos (42). Cada HU recoge al menos los siguientes aspectos (43):

- Número: Posee el número asignado a la HU.
- Nombre del requisito: Atributo que contiene el nombre del requisito.
- Programador: El programador del sistema.
- Iteración Asignada: Número de la iteración en la cual se desarrollará la HU.
- Prioridad: Evidencia el nivel de prioridad de la HU en el negocio.
- Puntos estimados: Este atributo es una estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo.
- Referencia: Es el conjunto de identificadores de las HU de las cuales depende historia actual que está en desarrollo.
- Descripción: Posee una breve descripción de lo que realizará la HU.
- Observaciones: Algunas aclaraciones que es importante señalar acerca de la HU.
- Prototipo de interfaz: Ilustración de la HU.

A continuación, se describen las HU definidas para llevar a cabo el desarrollo de los módulos. En total se realizaron 9 historias de usuario (todas con prioridad alta). Para su confección se tuvieron en cuenta opiniones de diferentes especialistas, estas se especifican a continuación.

Número:1	Nombre del requisito: Direccionar de forma manual la cámara hacia la derecha.
Programador: Richard Anca Perurena	Iteración Asignada:1
Prioridad: Alta	Punto estimado: 0.6
Riesgo de desarrollo: Alta	Referencia: -
Descripción:	

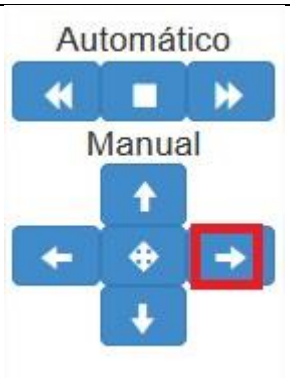
La aplicación permitirá al usuario mover la cámara hacia la derecha tantos pasos como este desee hasta llegar a su límite.
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control. -El movimiento será de 180°.
Prototipo de Interfaz:


Tabla 5. Historia de usuario #1

Número:2	Nombre del requisito: Direccionar de forma manual la cámara hacia la izquierda.
Programador: Richard Anca Perurena	Iteración Asignada:1
Prioridad: Alta	Punto estimado: 0.6
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario mover la cámara hacia la izquierda tantos pasos como este desee hasta llegar a su límite.	
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control. -El movimiento será de 180°.	
Prototipo de Interfaz:	

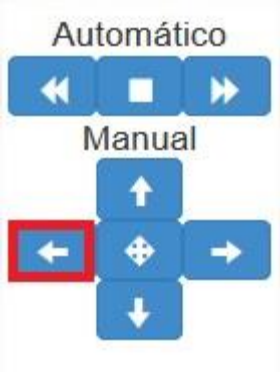


Tabla 6. Historia de usuario #2

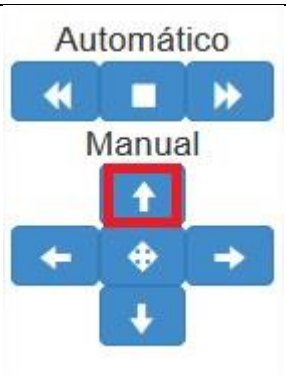
Número:3	Nombre del requisito: Direccionar de forma manual la cámara hacia arriba.
Programador: Richard Anca Perurena	Iteración Asignada:1
Prioridad: Alta	Punto estimado: 0.4
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario mover la cámara hacia arriba tantos pasos como este desee hasta llegar a su límite.	
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control. -El movimiento será de 180°.	
Prototipo de Interfaz:	
	

Tabla 7. Historia de usuario #3

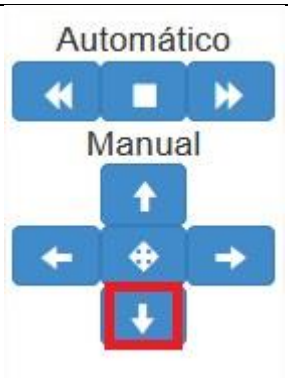
Número:4	Nombre del requisito: Direccionar de forma manual la cámara hacia abajo.
Programador: Richard Anca Perurena	Iteración Asignada:1
Prioridad: Alta	Punto estimado: 0.4
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario mover la cámara hacia abajo tantos pasos como este desee hasta llegar a su límite.	
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control. -El movimiento será de 180°.	
Prototipo de Interfaz:	
	

Tabla 8. Historia de usuario #4

Número:5	Nombre del requisito: Direccionar la cámara hacia su posición central.
Programador: Richard Anca Perurena	Iteración Asignada:1
Prioridad: Alta	Punto estimado: 0.2
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario mover la cámara hacia su centro.	
Observaciones:	

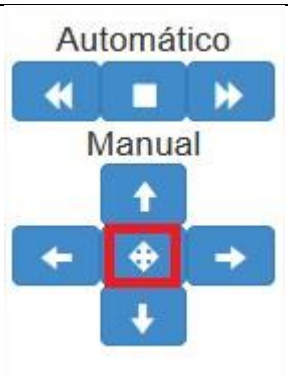
-El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control. -El movimiento será de 180°.
Prototipo de Interfaz:


Tabla 9. Historia de usuario #5

Número:6	Nombre del requisito: Desplazar de forma automática la cámara en el eje horizontal comenzando hacia el lado izquierdo
Programador: Richard Anca Perurena	Iteración Asignada:2
Prioridad: Alta	Punto estimado: 1
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario mover la cámara automáticamente sobre su eje horizontal comenzando por el lado izquierdo.	
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control. -El movimiento será de 180°.	
Prototipo de Interfaz:	

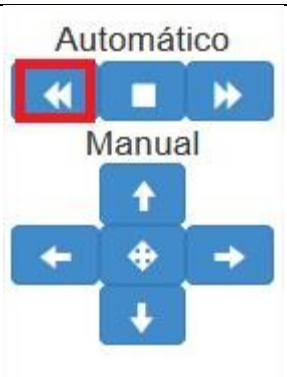


Tabla 10. Historia de usuario #6

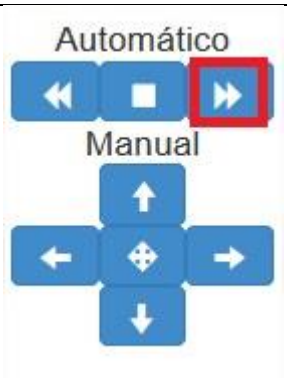
Número:7	Nombre del requisito: Desplazar de forma automática la cámara en el eje horizontal comenzando hacia el lado derecho.
Programador: Richard Anca Perurena	Iteración Asignada:2
Prioridad: Alta	Punto estimado: 0.8
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario mover la cámara automáticamente sobre su eje horizontal comenzando por el lado derecho.	
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control. -El movimiento será de 180°.	
Prototipo de Interfaz:	
	

Tabla 11. Historia de usuario #7

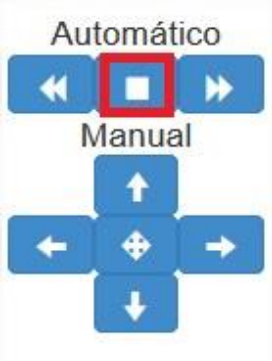
Número:8	Nombre del requisito: Detener movimiento automático
Programador: Richard Anca Perurena	Iteración Asignada:2
Prioridad: Alta	Punto estimado: 0.2
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario detener el movimiento automático.	
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi. -El cliente debe estar situado en el sitio de control.	
Prototipo de Interfaz:	
	

Tabla 12. Historia de usuario #8

Número:9	Nombre del requisito: Visualizar el *stream* de video capturado por la cámara en tiempo real.
Programador: Richard Anca Perurena	Iteración Asignada:3
Prioridad: Alta	Punto estimado: 1.2
Riesgo de desarrollo: Alta	Referencia: -
Descripción: La aplicación permitirá al usuario visualizar el *stream* de video capturado por la cámara en tiempo real.	
Observaciones: -El cliente debe estar correctamente conectado a la red Wifi emitida por la Raspberry Pi.	

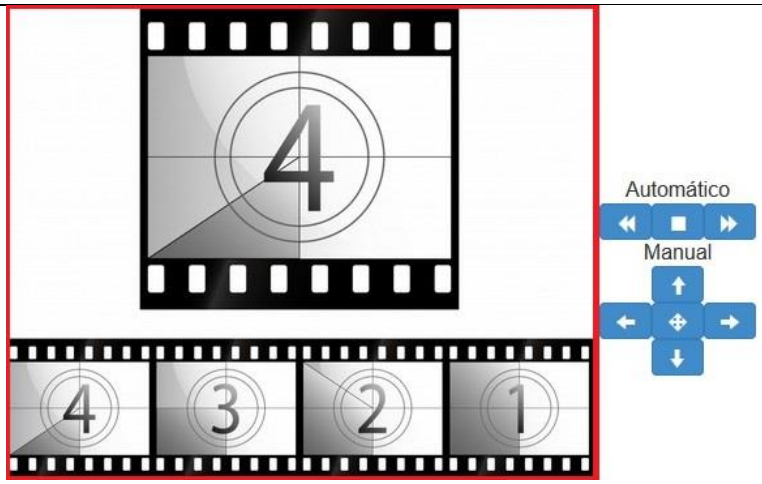
-El cliente debe estar situado en el sitio de control. -Se visualizará una secuencia de imágenes a 20 cuadros por segundo.
Prototipo de Interfaz:


Tabla 13. Historia de usuario #9

2.5 Estimación de esfuerzos e iteraciones por Historia de Usuario

Historias de Usuario	**Puntos de estimación**	**Iteración**	**Duración de las iteraciones (Semanas)**
Direccionar de forma manual la cámara hacia la derecha.	0.6	1	3
Direccionar de forma manual la cámara hacia la izquierda.	0.6		
Direccionar de forma manual la cámara hacia arriba	0.4		
Direccionar de forma manual la cámara hacia abajo	0.4		

Direccionar la cámara hacia su posición central.	0.2		
Desplazar de forma automática la cámara en el eje horizontal comenzando hacia el lado izquierdo.	1	2	3
Desplazar de forma automática la cámara en el eje horizontal comenzando hacia el lado derecho.	0.8		
Detener movimiento automático.	0.2		
Visualizar el *stream* de video capturado por la cámara en tiempo real.	1.2	3	3

Tabla 14. Estimación de esfuerzos e iteraciones por Historia de Usuario

Luego de definir las historias de usuario y estimar el esfuerzo propuesto para su realización, se realizó el sistema en 3 iteraciones las cuales se describen a continuación:

Iteración 1: Esta iteración tiene como objetivo realizar las HU 1, 2, 3, 4 y 5 las cuales se encargan de los movimientos manuales y centrar la plataforma respectivamente.

Iteración 2: Esta iteración tiene como objetivo realizar las HU 6, 7 y 8 las cuales se encargan del movimiento automático de la plataforma y detener dicho movimiento.

Iteración 3: Esta iteración tiene como objetivo realizar la HU 9 la cual se encarga de visualizar el video capturado por la cámara.

2.6 Descripción de la arquitectura del sistema

El Modelo Vista Controlador (MVC) es un patrón arquitectónico para el desarrollo del software que se basa en separar los datos, la interfaz y la lógica por lados diferentes (44). En el framework Django el patrón MVC también es llamado modelo vista *template* (MVT), en este caso la capa de modelo se mantiene, pero la vista no es una vista, sino un controlador que se llama vista y el *template* son las vistas del MVC.(39)

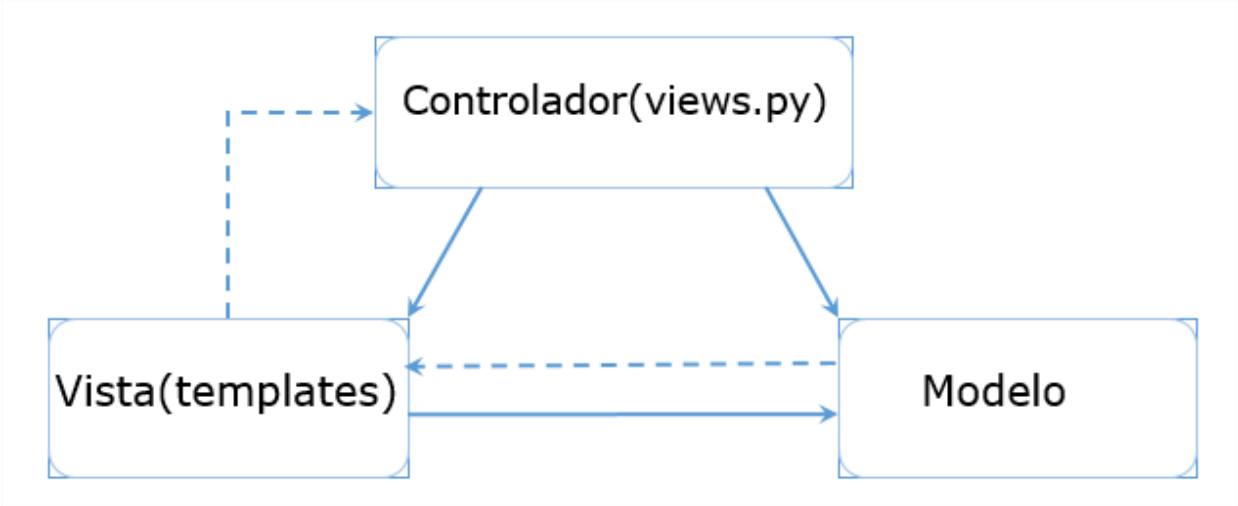


Ilustración 11 Patrón de diseño MVC.

En la solución propuesta, la vista es la página HTML que es mostrada al usuario (templates/index.html) y el controlador es donde están implementadas las funcionalidades de la aplicación (views.py).

2.7 Descripción de la solución

Se pretende lograr una aplicación web en Python, la cual, al recibir una primera conexión, al ser Python un lenguaje interpretado se carguen las bibliotecas necesarias y se inicie Motion logrando visualizar en tiempo real lo capturado por la cámara y permitiéndole al usuario controlar los movimientos de la plataforma. En la siguiente ilustración se muestra cómo se realiza el flujo de datos entre los principales componentes y el cliente.

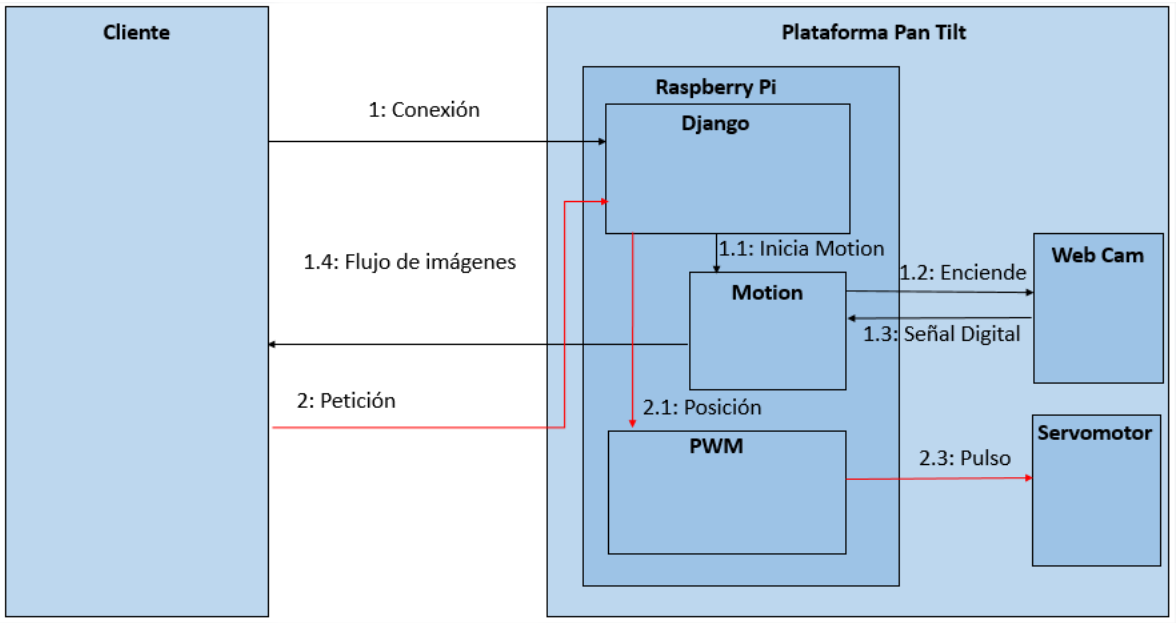


Ilustración 12 Flujo de datos.

El usuario para lograr la conexión, se debe conectar a la dirección web http://192.168.1.1:8000/main, al establecerse la conexión desde Django se cargan las bibliotecas PWM, time y os. PWM se utiliza para simular señales analógicas y lograr el movimiento de los servomotores, pasándole por parámetros el pin GPIO y la posición donde situarse entre 600 y 1500 esta biblioteca envía a los servomotores cada 20 mili segundos un ancho de pulso en un rango de 500 micro segundos a 2500 micro segundos dependiendo de la posición que se desee. La biblioteca time permite hacer pausas de un tiempo pasado por parámetros. La biblioteca os permite iniciar servicios del sistema, en este caso se utiliza para iniciar Motion. Motion es una aplicación código abierto, la cual permite capturar imágenes desde una cámara web y servirlas al usuario en una url. Se decidió utilizar Motion debido a que Raspberry Pi no consta de recursos para codificar un video en tiempo real como hacen otros servidores de *stream* como vlc. Motion es configurable, en este caso está configurada para no guardar datos y enviar un flujo de 20 imágenes por segundo en una resolución de 800 por 600. Al iniciarse Motion se enciende la cámara web y este convierte la señal digital recibida en imágenes y se las muestra al usuario.

2.8 Diagrama de Componentes

El diagrama de componentes está estructurado por componentes lógicos y físicos, como se describen a continuación.

- La vista es un componente lógico en el cual se muestra al usuario lo que está capturando la cámara en tiempo real y una serie de botones con los que el usuario puede interactuar para controlar los movimientos de la plataforma.
- El controlador es un componente lógico en el cual se importan las bibliotecas PWM para controlar los movimientos de los servomotores, *time* para los tiempos de espera entre movimientos continuos y os para iniciar procesos del sistema. Además, se inicia Motion y se dan respuesta a las peticiones de movimientos realizadas por el usuario.
- La plataforma es un componente físico compuesto por dos servos motores y una cámara USB.

En la ilustración siguiente se hace referencia a los componentes descritos.

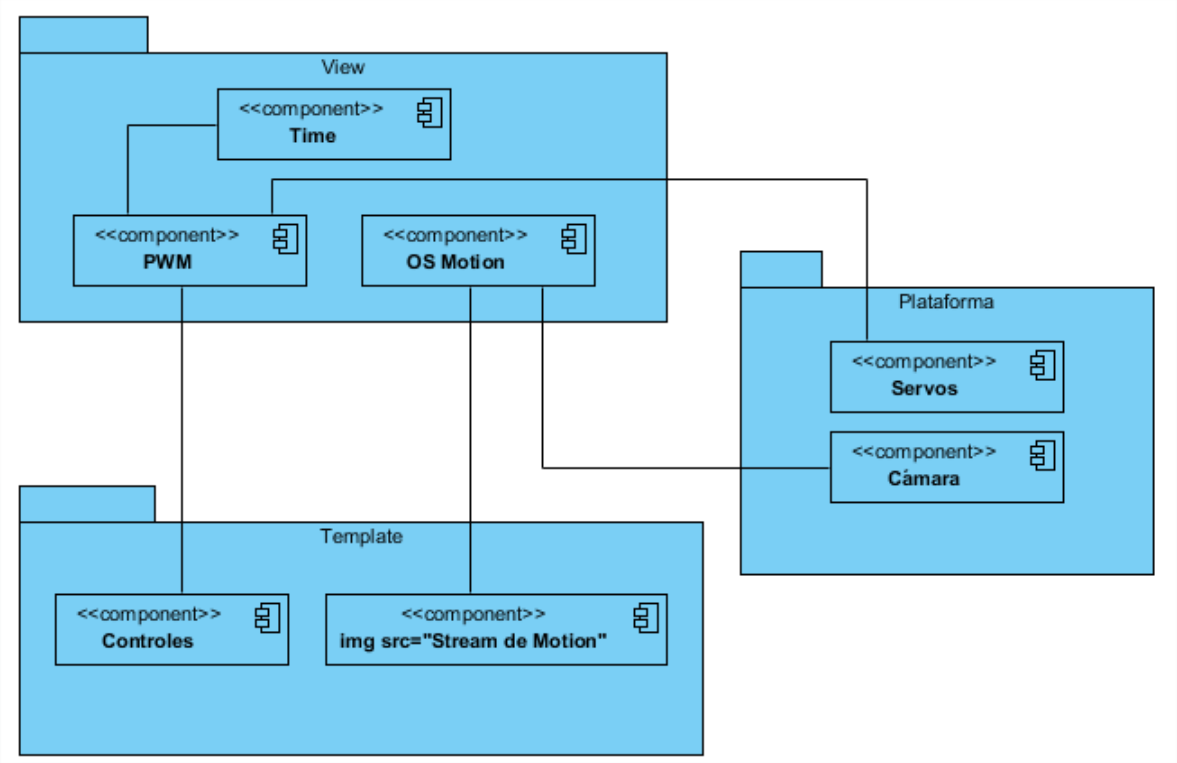


Ilustración 13 Diagrama de Componentes.

En el controlador para interpretar las peticiones del usuario para el movimiento de la cámara se define un simple protocolo, el cual consiste en enviar en la url el identificador asignado a la funcionalidad deseada, donde el identificador 1 y 2 son los movimientos verticales arriba y abajo, 3 y 4 los movimientos horizontales manuales, el número 5 lleva los servomotores a su posición central, centrando la cámara, 6 y 8 están asignados a los movimientos horizontales automáticos y por último el 7 detiene el movimiento automático como se muestra en la siguiente tabla.

ID	Respuesta
1	Inclinación hacia arriba.
2	Inclinación hacia abajo.
3	Movimiento panorámico a la derecha.
4	Movimiento panorámico a la izquierda.
5	Lleva ambos servomotores a su posición central.
6	Movimiento automático comenzando por el lado derecho.
7	Detiene el movimiento automático si se está realizando.
8	Movimiento automático comenzando por el lado izquierdo.

Tabla 15 Protocolo de atención a peticiones de movimiento.

Para lograr los movimientos verticales hacia arriba y hacia abajo se crea un archivo con el nombre motor1 y extensión pos, en el cual se guarda la última posición del servomotor. Al recibir una solicitud de movimiento se lee el archivo de posición, se le suma o resta 100 al valor actual en dependencia de si el movimiento es hacia arriba o hacia abajo, se le pasa a la biblioteca PWM el pin GPIO 2 el cual corresponde al servomotor vertical y el valor de la posición deseada. Por último, se guarda la posición actual en el archivo motor1.pos.

Los movimientos horizontales se logran de forma similar a los verticales, en este caso el pin GPIO utilizado es el 3, el archivo de posición es motor2.pos y se utiliza una bandera para detectar si se está realizando un movimiento automático. Si se está realizando un movimiento automático se escribe ¨false¨ en la bandera y se hace una pequeña pausa con la biblioteca timer para garantizar que se detiene el movimiento automático y luego se atiende la petición de movimiento del usuario.

La función centrar cámara se realiza pasando a PWM los pines GPIO 2 y 3 donde están conectado ambos servomotores y el valor 1500, el cual se traduce como su posición central. Luego el valor 1500 se guarda en los archivos de posición correspondientes a cada servomotor.

Cuando se recibe una petición de movimiento automático, primeramente, se consulta el archivo auto.flag para identificar si la plataforma está actualmente en movimiento. Si la plataforma no está en movimiento se pone en ¨true¨ el archivo y se realiza un ciclo aumentando o disminuyendo la posición en dependencia del sentido deseado cada 0.1 segundo, y cada una actualización de movimiento se revisa si se actualizó la bandera a ¨false¨, si se actualizó se detiene el ciclo, se guarda la posición actual y se atiende la siguiente

petición del usuario, en caso de permanecer la bandera en ¨true¨ cuando siclo alcanza la posición final del servomotor, termina y comienza un siclo en sentido contrario, así hasta que el usuario intervenga. En caso de realizar la petición de movimiento automático y la plataforma este en movimiento, se pone la bandera ¨false¨ se espera 0.11 segundos y comienza el movimiento en la nueva dirección.

La función detener simplemente escribe en el archivo bandera ¨false¨.

2.9 Patrones de Diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular (45).

2.9.1 Patrones Grasp

Patrones generales de softare para asignación de responsabilidades, por sus siglas en ingles GRASP.

Bajo acoplamiento

Se asignan las responsabilidades de manera que el acoplamiento sea bajo. Un componente con bajo acoplamiento no mantiene fuertes dependencias con otros componentes. Cuando este es alto se pueden presentar estos problemas:

- Los cambios locales son difíciles de realizar.
- Son difíciles de analizar de forma independiente.
- Se vuelven difíciles de reutilizar.

Este patrón se evidencia en las funcionalidades de movimiento, ya que estas no presentan ninguna dependencia entre ellas se pueden realizar modificaciones sin alterar su comportamiento.

Alta Cohesión

La cohesión es una medida de la fuerza con la que se relacionan las clases y el grado de focalización de las responsabilidades de un elemento. Cada elemento de nuestro diseño debe realizar una labor única dentro del sistema, no desempeñada por el resto de los elementos y auto-identificable, una clase con baja cohesión hace muchas cosas no relacionadas o hace demasiado trabajo.

Este patron se evidencia en el sistema de manera tal que las funcionalidades contienen la informacion necesaria para trabajar sin involucrar a otras funcionalidades.

2.9.2 Patrones Gof

Front controller

Front controller es el primer punto de entrada para la solicitud. Maneja el envío general al controlador y se ocupa de las necesidades comunes tales como, autenticación, gestión de sesiones, seguridad, redirección. Contiene la lógica que todas las solicitudes deben considerar, reduciendo la duplicación de código. También agrega datos al contexto de ejecución (46).

Observer

Define una dependencia de uno a muchos entre objetos de forma que, cuando un objeto cambia de estado, se notifica a los objetos dependientes para que se actualicen automáticamente. Este patrón se aplica en la funcionalidad de centrar la cámara, cuando el usuario realiza la petición centrar, las posiciones de ambos motores se sitúan en 1500, es decir, en su posición central horizontal y vertical respectivamente.

Conclusiones parciales

Durante el desarrollo del capítulo quedó definida la propuesta de solución a desarrollar y sus características, obteniéndose una lista de las funcionalidades que debe tener el sistema. Las mismas fueron descritas mediante las Historias de Usuario. Se estableció como patrón arquitectónico patrón MVT para facilitar al usuario realizar las peticiones de movimiento y la visualización del flujo de video. Se definieron como patrones de diseño Grasp, Bajo acoplamiento y Alta cohesión para independizar las funcionalidades y como patrones de diseño Gof, Front controller y Observer para responder las solicitudes del usuario y actualizar la posición de los motores respectivamente.

Capítulo 3 Implementación y prueba

El objetivo de este capítulo luego de haber realizado el diseño es desarrollar la solución y luego someter la herramienta a pruebas internas pues podría presentar fallas, mediante casos de pruebas donde se validarán las soluciones y finalmente poder realizar el despliegue en los sistemas de producción.

3.1 Implementación del Sistema

La implementación constituye una de las fases más importantes del desarrollo de software. En ella se toman como punto de partida los resultados obtenidos en el diseño, implementándose el sistema en términos de componentes como ficheros de código binario, código fuente, scripts y ejecutables. Su importancia reside en que se obtiene como consecuencia un sistema ejecutable, siendo esto uno de los principales objetivos en el desarrollo de software.

3.1.1 Diagrama de Despliegue

Un diagrama de despliegue, es un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Además, muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución. A continuación, se muestra el diagrama de despliegue del sistema.

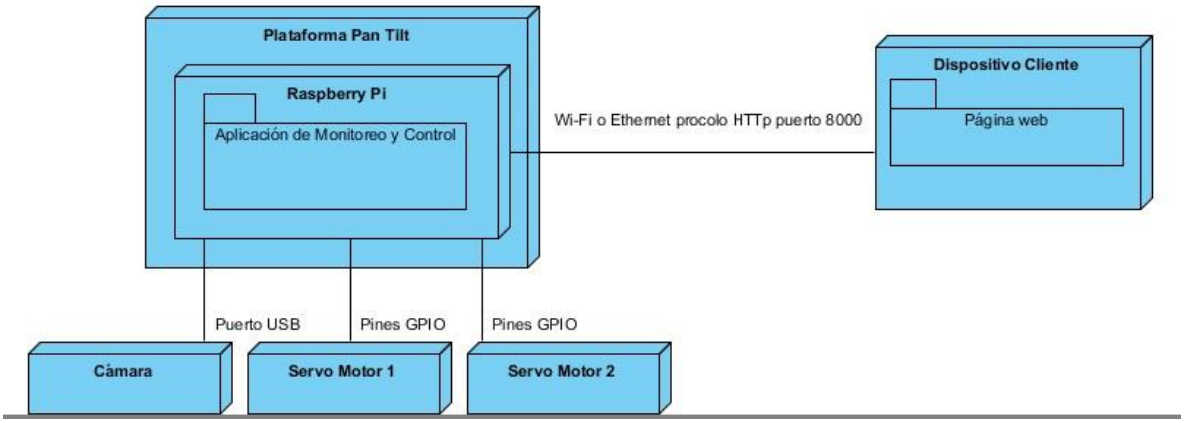


Ilustración 15 Diagrama de despliegue.

Plataforma Pan Tilt: Plataforma donde están acoplados los servomotores y la cámara USB. Además, contiene en su interior la Raspberry Pi.

Raspberry Pi: Es donde está ejecutándose la Aplicación de Monitoreo y Control.

Aplicación de Monitoreo y Control: Es la encargada de interpretar las peticiones del usuario y mostrarle al mismo lo capturado por la cámara USB en tiempo real.

Cámara USB: Dispositivo para capturar las imágenes en tiempo real.

Servo Motores: Dispositivos electromecánicos que permiten a la plataforma realizar los movimientos.

Dispositivo Cliente: Dispositivo desde el cual el cliente puede controlar los movimientos de la plataforma y visualizar en tiempo real lo que la cámara captura. Puede ser una PC o un teléfono inteligente.

3.2 Pruebas

Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación. El objetivo fundamental de las pruebas es descubrir diferentes clases de errores con la menor cantidad de tiempo y de esfuerzo. Aunque las pruebas no pueden asegurar la ausencia de defectos; sí pueden demostrar que existen defectos en el software (47).

Estas actividades se planean con anticipación y se realizan de manera sistemática. Cuando se aplican pruebas a un software es necesario tener en cuenta el objetivo que se persigue, debido a las pruebas son agrupadas por niveles que se encuentran en distintas etapas del proceso de desarrollo.

3.3.1 Pruebas de aceptación

El objetivo de las pruebas de aceptación no es validar el comportamiento lógico específico de un componente sino un escenario, caso de uso o caso de negocio de la aplicación. Su característica principal es que se realiza con el cliente para validar que la aplicación permite a los clientes completar las tareas para la que fue diseñada.

Es la prueba planificada y organizada formalmente para determinar si se cumplen los requisitos de aceptación marcados por el cliente. Sus características principales son las siguientes (47):

- Participación del usuario.
- Está enfocada hacia la prueba de los requisitos de usuarios especificados.
- Está considerada como la fase final del proceso para crear una confianza en que el producto es el apropiado para su uso en explotación.

Para representar las pruebas de aceptación se definieron los siguientes elementos:

- Código: Representa al caso de prueba, incluye el número de HU, de la prueba y si posee diferentes escenarios.
- Historia de Usuario: Número de la HU a la cual pertenece.
- Nombre: Junto al código conforma el identificador del caso de prueba.
- Descripción: Acción que debe realizar el sistema.
- Condiciones de ejecución: Describe las características y elementos que debe contener el sistema para realizar el caso de prueba.
- Entrada/Pasos de Ejecución: Incluye las entradas necesarias para realizar el sistema, además de los pasos para realizar el caso de prueba.
- Resultados Esperados: Descripción de la respuesta del sistema ante el caso de prueba.
- Evaluación de la prueba: Clasificación de la prueba en satisfactoria o insatisfactoria.

Caso de Prueba Aceptación	
Código: 1	**Historias de usuario: 1,2 3 y 4**
Nombre: Movimiento manual.	
Descripción: Gira o inclina la plataforma tantos pasos como ordene el usuario en el sentido que este desee.	
Condiciones de Ejecución: El usuario debe estar conectado en línea con la aplicación.	
Entradas/ Pasos de Ejecución: - Conectarse a la aplicación. - Clic izquierdo con el ratón en la dirección en la cual se desea mover la plataforma.	
Resultado esperado: La plataforma ha girado o inclinado en la dirección deseada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 16 Caso de prueba aceptación 1.

Caso de Prueba Aceptación

Código: 2	**Historia de usuario: 5**
Nombre: Centrar la cámara.	
Descripción: Centra la cámara vertical y horizontalmente llevándola al medio de ambos ejes.	
Condiciones de Ejecución: El usuario debe estar conectado en línea con la aplicación.	
Entradas/ Pasos de Ejecución: - Conectarse a la aplicación. - Clic izquierdo con el ratón en el botón centrar.	
Resultado esperado: La plataforma se ha centrado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 17 Caso de prueba aceptación 2.

Caso de Prueba Aceptación	
Código: 3	**Historias de usuario: 6 y 7**
Nombre: Movimiento automático.	
Descripción: Hace girar la plataforma de forma continua de un lado a otro sobre la totalidad de su eje horizontal comenzando por el lado deseado.	
Condiciones de Ejecución: El usuario debe estar conectado en línea con la aplicación.	
Entradas/ Pasos de Ejecución: - Conectarse a la aplicación. - Clic izquierdo con el ratón en el botón asociado a la dirección automática deseada.	
Resultado esperado: La plataforma se mueve de manera continua sobre su eje horizontal comenzando por la dirección deseada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 18 Caso de prueba aceptación 3.

Caso de Prueba Aceptación

Código: 4	**Historia de usuario: 8**
Nombre: Detener.	
Descripción: Detiene el movimiento automático de la plataforma en la posición actual.	
Condiciones de Ejecución: El usuario debe estar conectado en línea con la aplicación y la plataforma moviéndose automáticamente.	
Entradas/ Pasos de Ejecución: - Conectarse a la aplicación. - Clic izquierdo con el ratón en el botón detener.	
Resultado esperado: La plataforma detuvo su movimiento.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 19 Caso de prueba aceptación 4.

Caso de Prueba Aceptación	
Código: 5	**Historia de usuario: 9**
Nombre: Flujo de video.	
Descripción: Muestra en tiempo real el flujo de video capturado por la cámara.	
Condiciones de Ejecución: El usuario debe estar conectado en línea con la aplicación.	
Entradas/ Pasos de Ejecución: - Conectarse a la aplicación.	
Resultado esperado: Se muestra lo que está capturando la plataforma en tiempo real de manera fluida.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 20 Caso de prueba aceptación 5.

3.3.2 Diseños de casos de pruebas

Pruebas del sistema	HU	Iteración	NC	Cerrada	No procede
		1ra	5	5	0

	9	2da	4	4	0
		3ra	1	1	0

Tabla 21 Diseño de casos de prueba.

Una vez realizados los casos de Pruebas para un total de 9 historias de usuarios, con tres iteraciones se eliminaron las 10 No Conformidades.

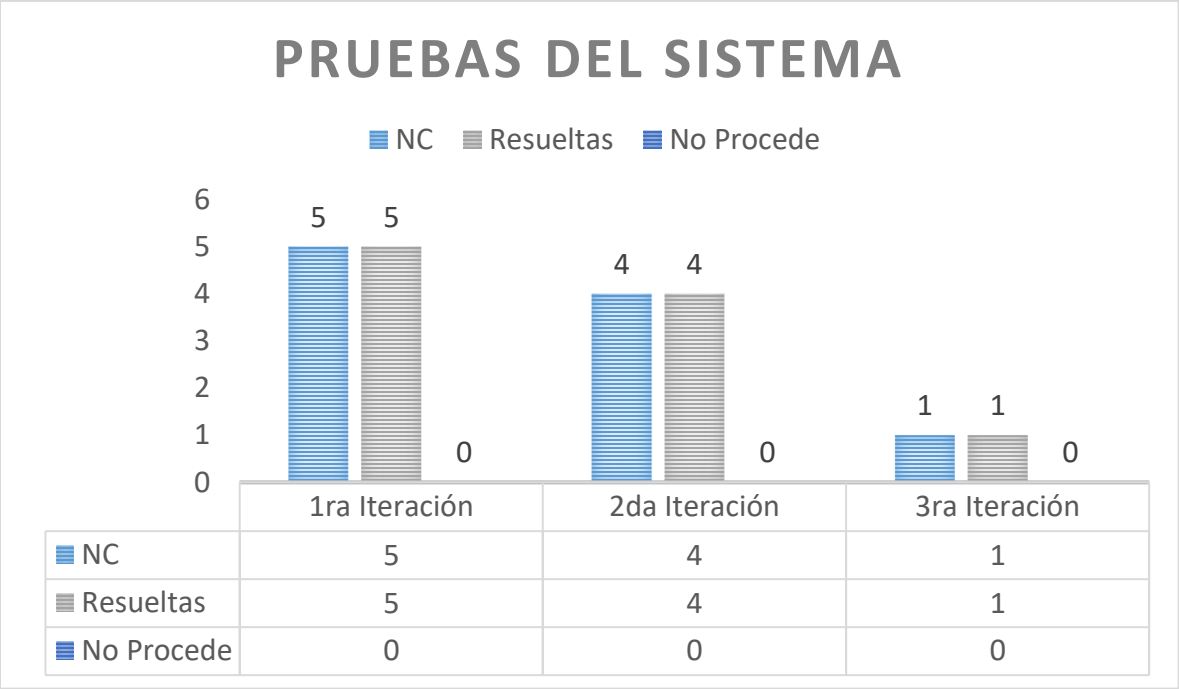


Ilustración 16 Pruebas del sistema.

En la primera iteración de pruebas se encontraron un total de 5 no conformidades (errores) distribuidos de la siguiente manera: cuatro pertenecientes a los requerimientos del sistema y uno en cuanto al diseño de interfaz. Según propone la metodología AUP-UCI para pasar a la siguiente iteración de pruebas es necesario que todas las no conformidades detectadas en iteraciones anteriores sean corregidas satisfactoriamente. En la segunda iteración se encontraron un total de 4 no conformidades todas significativas ya que se trataban de requerimientos que el sistema debía cumplir, las cuales fueron corregidas. En la tercera iteración se encontró una no conformidad significativa ya que también se trataba de un requisito que el sistema debía cumplir la cual fue corregida satisfactoriamente.

A modo de conclusión, los resultados obtenidos luego de la realización de las pruebas fueron satisfactorios, debido a que las no conformidades encontradas durante su ejecución fueron deficiencias de tipo

significativas y no significativas para el correcto funcionamiento de la aplicación, las cuales fueron corregidas.

3.3 Relación de Costo

En la actualidad el precio de las cámaras Pan Tilt oscila por los 100 dólares, y el de las cámaras Pan Tilt profesionales por los 1000, también existen cámaras de inferior calidad y precio, así como de muy altos precios. En la tabla 21 se pueden apreciar varios modelos, su calidad de imagen y precio. El costo del prototipo propuesto se aproxima a los 60 USD como se puede apreciar en la tabla 22. Con la solución propuesta se evidencia un gran ahorro para el país, un paso más hacia a la soberanía tecnológica y no sería necesaria la importación de las cámaras Pan Tilt.

Marca	**Modelo**	**Calidad de imagen**	**Lente (Megapíxel)**	**Precio (USD)**
FDT	FD7903	960p	1.3	159.99
Amcrest	IP3M-941B	UltraHD 2K	3	119.99
Zmodo	-	1280 x 720	1	49.99
Sony	GW Security	1080P	3	429.00
Sony	BRC-H800	1080p	20.4	8,999.00
SPI	M7 Long Range	HD 4K	-	40,239.00

Tabla 22 Cámaras Pan Tilt.

Servomotor	**Cámara USB (Baja Calidad)**	**Raspberry Pi**	**Total**
2x2 USD	15 USD	37 USD	56 USD

Tabla 23 Coste del prototipo.

Conclusiones parciales

En este capítulo se ejecutaron las pruebas de aceptación para darle validez a los requisitos descritos en las historias de usuarios. Se presentó la distribución física del sistema permitiendo un mejor entendimiento de la distribución física y lógica del sistema. A partir del resultado arrojado por las pruebas realizadas se llegó a la conclusión de que la aplicación se encuentra lista para su puesta en funcionamiento.

Conclusiones generales

Con el desarrollo de la presente investigación se arriba a las siguientes conclusiones:

- El estudio de las tecnologías actuales facilitó la construcción de un prototipo de cámara Pan Tilt y la puesta en marcha de su software correspondiente
- La implementación del sistema de control propuesto permite un correcto control del movimiento de una cámara Pan Tilt y una visualización fluida del video capturado por la misma, proporcionando un mejor monitoreo de las áreas a visualizar de forma remota.

Recomendaciones

Se recomienda en la investigación:

- Incorporar la funcionalidad de almacenar la información obtenida por el sistema en un medio externo para su posterior acceso.
- Incorporar un control de acceso a la aplicación para garantizar una mayor seguridad.
- Incorporar un componente gráfico que muestre la posición actual de los motores.

Bibliografía

1. **Tagua de Pepa, Marcela, García Garino, Carlos y Marianetti, Osvaldo.** http://www.unidiversidad.com.ar. [En línea] 15 de julio de 2016. [Citado el: 21 de mayo de 2017.] http://www.unidiversidad.com.ar/las-tic-su-importancia-en-la-actualidad-y-el-mercado-laboral. 1.

2. **DO.** http://siesa.com.ar. [En línea] 6 de julio de 2015. [Citado el: 11 de junio de 2017.] http://siesa.com.ar/el-importante-rol-del-sistema-de-cctv-en-la-seguridad-privada. 2.

3. **IEEE.** http://ieeexplore.ieee.org. [En línea] 23 de febrero de 2012. [Citado el: 23 de junio de 2017.] http://ieeexplore.ieee.org/document/6156786/.

4. **Palazzesi , Ariel.** http://www.neoteo.com. [En línea] 7 de marzo de 2008. [Citado el: 7 de diciembre de 2016.] http://www.neoteo.com/motores-paso-a-paso. 6.

5. **Mendez, Uriel.** https://www.330ohms.com. [En línea] 14 de marzo de 2016. [Citado el: 15 de marzo de 2017.] https://www.330ohms.com/blogs/blog/112837444-que-son-los-servomotores.

6. **IEEE.** http://ieeexplore.ieee.org/document/6371791/. [En línea] 25 de 5 de 2010. [Citado el: 17 de 2 de 2017.] http://ieeexplore.ieee.org/document/6371791/.

7. **Oxford.** https://es.oxforddictionaries.com. [En línea] 7 de 12 de 2016. [Citado el: 7 de 12 de 2016.] https://es.oxforddictionaries.com/definition/camara.

8. **Gómez Diaz, Yassiel.** *Control de cámara Pan Tilt.* La Habana : s.n., 2016.

9. **Valeriano A.J, J.** http://valetron.eresmas.net. [En línea] 2016. [Citado el: 7 de diciembre de 2016.] http://valetron.eresmas.net/CamarasIP.htm.

10. **Zenith.** http://blogginzenith.zenithmedia.es. [En línea] 5 de junio de 2015. [Citado el: 6 de marzo de 2017.] http://blogginzenith.zenithmedia.es/que-es-y-como-funciona-el-live-streaming-diczionario/.

11. **Orlin, Jon.** http://www.streamingcolombia.co. [En línea] 19 de julio de 2014. [Citado el: 6 de marzo de 2017.] http://www.streamingcolombia.co/2014/07/19/10-consejos-para-un-streaming-exitoso/.

12. **VideoLan.** http://www.videolan.org. [En línea] 2017. [Citado el: 6 de marzo de 2017.] http://www.videolan.org/vlc/streaming.html.

13. **3CX.** https://www.3cx.es. [En línea] 2017. [Citado el: 6 de marzo de 2017.] https://www.3cx.es/webrtc/que-es-webrtc/.

14. **Mundo Bip.** http://mundobip.com. [En línea] 8 de enero de 2011. [Citado el: 6 de marzo de 2017.] http://mundobip.com/Introducci%C3%B3n-Gu%C3%ADa-de-instalaci%C3%B3n-de-Servidor-con-Linux-Debian-Lenny-al-m%C3%ADnimo-%28Parte-VIII%3A-Servidor-de-seguridad-Webcam%29_197.htm.

15. **Ecured.** https://www.ecured.cu. [En línea] 2017. [Citado el: 5 de marzo de 2017.] https://www.ecured.cu/Web.

16. —. https://www.ecured.cu. [En línea] 2017. [Citado el: 24 de enero de 2017.] https://www.ecured.cu/Servicio_Web.

17. —. https://www.ecured.cu. [En línea] 2017. [Citado el: 5 de marzo de 2017.] https://www.ecured.cu/Sitio_Web.

18. **Reno.** http://html.rincondelvago.com. [En línea] 2017. [Citado el: 24 de enero de 2017.] http://html.rincondelvago.com/protocolos-de-comunicacion_1.html.

19. **CCM.** http://es.ccm.net. [En línea] 11 de agosto de 2016. [Citado el: 7 de diciembre de 2016.] http://es.ccm.net/faq/1559-diferencias-entre-los-protocolos-tcp-y-udp.

20. **VideoLan.** https://wiki.videolan.org. [En línea] 23 de noviembre de 2010. [Citado el: 5 de marzo de 2017.] https://wiki.videolan.org/MMSH/.

21. **Ecured.** https://www.ecured.cu. [En línea] 2015. [Citado el: 7 de diciembre de 2016.] https://www.ecured.cu/Cliente-Servidor.

22. **Marquez, Lis.** *Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español .* Mexico : s.n., 2004.

23. **Perez, Jon.** http://www.ikerlan.es. [En línea] 2015. [Citado el: 24 de enero de 2017.] http://www.ikerlan.es/es/que-investigamos/sistemas-embebidos.

24. **Sherlin.** http://sherlin.xbot.es. [En línea] 2016. [Citado el: 24 de enero de 2017.] http://sherlin.xbot.es/microcontroladores/introduccion-a-los-microcontroladores/que-es-un-microcontrolador.

25. **Merlynck, David.** http://www.monografias.com. [En línea] 25 de enero de 2007. [Citado el: 24 de enero de 2017.] http://www.monografias.com/trabajos12/microco/microco.shtml.

26. **Arduino.** https://www.arduino.cc/. [En línea] Arduino, 23 de junio de 2017. [Citado el: 23 de junio de 2017.] https://www.arduino.cc/. 3.

27. **Microchip.** http://www.microchip.com. [En línea] 2017. [Citado el: 23 de junio de 2017.] http://www.microchip.com/design-centers/microcontrollers.

28. **Garcia Cobo, Joaquin.** https://www.hwlibre.com. [En línea] 3 de noviembre de 2014. [Citado el: 23 de junio de 2017.] https://www.hwlibre.com/que-es-una-placa-sbc/.

29. **Castro, Alberto.** http://computerhoy.com. [En línea] 23 de 1 de 2014. [Citado el: 23 de 1 de 2017.] http://computerhoy.com/noticias/hardware/que-es-raspberry-pi-donde-comprarla-como-usarla-8614.

30. **PE, Isaac.** http://comohacer.eu. [En línea] 26 de 5 de 2015. [Citado el: 14 de 1 de 2017.] http://comohacer.eu/gpio-raspberry-pi/.

31. **Gonzalez Martinez, Oscar.** Sistemas operativos en microcontroladores. [En línea] 23 de 3 de 2006. [Citado el: 24 de 1 de 2017.] http://www.alcabot.com/alcabot/seminario2006/SEM06_SOenMicrocontroladores.pdf.

32. **Raspbian.** https://www.raspbian.org. [En línea] 2017. [Citado el: 24 de enero de 2017.] https://www.raspbian.org/.

33. **ADNPI.** https://espberry.wordpress.com. [En línea] 7 de marzo de 2016. [Citado el: 12 de junio de 2017.] https://espberry.wordpress.com/2016/03/07/instalar-mis-aplicaciones-en-raspbian/.

34. **Galeon.** http://www.galeon.com. [En línea] 2012. [Citado el: 23 de enero de 2017.] http://www.galeon.com/redesljd/atm.htm.

35. **Hernández Orallo, Enrique.** El Lenguaje de Modelado Unificado de (UML). [En línea] [Citado el: 2 de mayo de 2017.] http://www.disca.upv.es/enheror/pdf/ActaUML.PDF.

36. **Visual Paradigm.** https://www.visual-paradigm.com/. [En línea] 2017. [Citado el: 23 de enero de 2017.] https://www.visual-paradigm.com/.

37. **HTML.** http://html.net/. [En línea] 2017. [Citado el: 23 de enero de 2017.] http://html.net/.

38. **w3schools.** http://www.w3schools.com. [En línea] 2017. [Citado el: 23 de enero de 2017.] http://www.w3schools.com/js/js_intro.asp.

39. **P. George, Damien y Sokolovsky, Paul .** https://docs.micropython.org. [En línea] 6 de marzo de 2017. [Citado el: 7 de marzo de 2017.] https://docs.micropython.org/en/latest/esp8266/esp8266/tutorial/pwm.html.

40. **Django.** https://www.djangoproject.com/. [En línea] 2017. [Citado el: 6 de marzo de 2017.] https://www.djangoproject.com/.

41. **Fernández, Thaimara.** https://fundamentodeingenieria.wordpress.com/. [En línea] 1 de junio de 2010. [Citado el: 10 de marzo de 2017.] https://fundamentodeingenieria.wordpress.com/.

42. **Tarantino, Quentin.** *Django Unchained.* 2016.

43. **Ecured.** https://www.ecured.cu. [En línea] 2017. [Citado el: 5 de marzo de 2017.] https://www.ecured.cu/Protocolo_de_Transferencia_de_Hipertexto.

44. **Axterciyo.** https://debuenamano.wordpress.com. [En línea] 3 de mayo de 2012. [Citado el: 10 de marzo de 2017.] https://debuenamano.wordpress.com/2012/03/05/modelo-vista-controlador/.

45. **Alexander.** http://siul02.si.ehu.es. [En línea] 2017. [Citado el: 10 de marzo de 2017.] http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf.

46. **LibrosWeb.** http://librosweb.es. [En línea] 2017. [Citado el: 11 de marzo de 2017.] http://librosweb.es/libro/symfony_1_2_en/chapter_6/the_front_controller.html.

47. **Wordpress.** *Estrategias de pruebas de software.* 2009.

48. **Hacedores.** http://hacedores.com. [En línea] 12 de mayo de 2014. [Citado el: 24 de enero de 2017.] http://hacedores.com/arduino-o-raspberry-pi-cual-es-la-mejor-herramienta-para-ti/.

49. **Marta.** http://fresqui.com. [En línea] 22 de enero de 2017. [Citado el: 5 de marzo de 2017.] http://fresqui.com/news/caracteristicas-y-ventajas-del-raspberry-pi-3.

50. **HTML.** http://es.html.net. [En línea] 2017. [Citado el: 23 de enero de 2017.] http://es.html.net/tutorials/css/lesson1.php.

Anexos

Anexo 1: Prototipo de interfaz de usuario

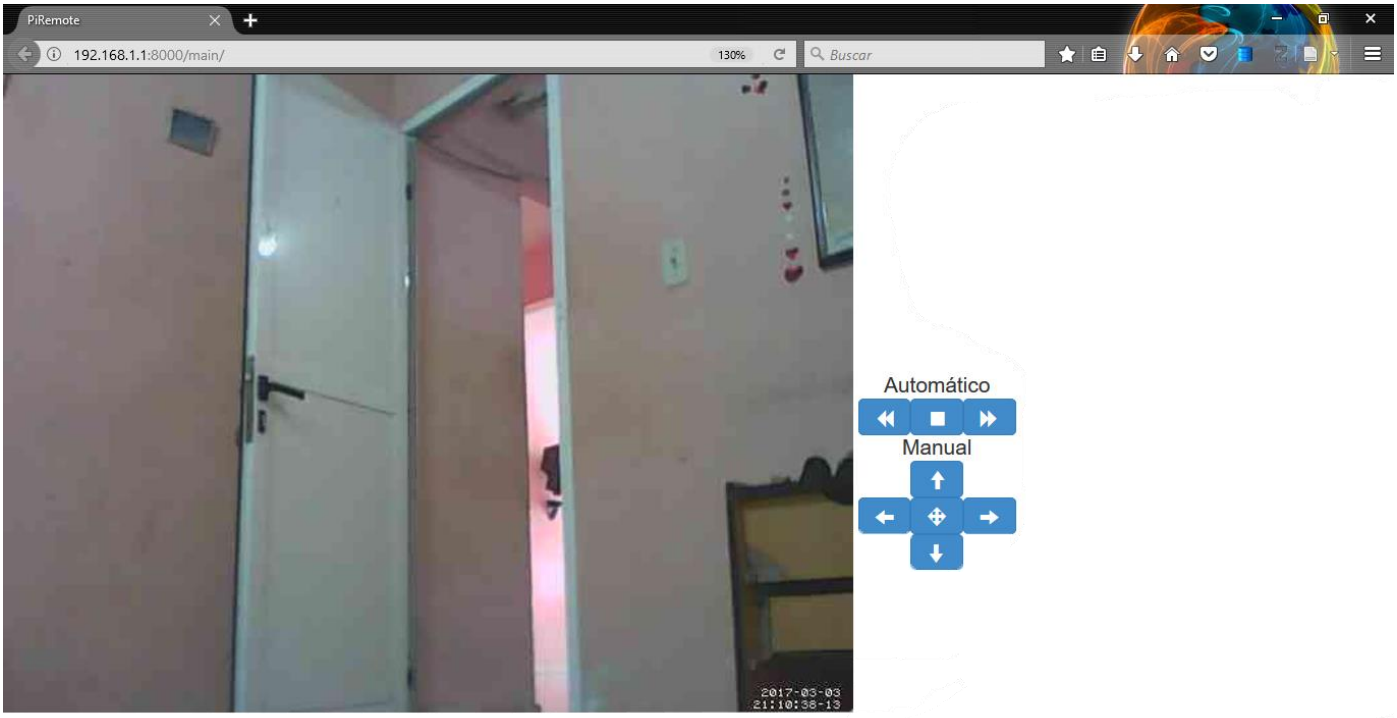


Ilustración 17 Prototipo de interfaz de usuario.