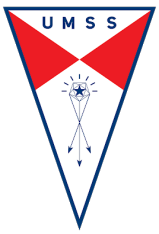

UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIA Y TECNOLOGÍA
INGENIERIA INFORMÁTICA

SISTEMA DE VIDEO VIGILANCIA INTELIGENTE PARA LA ALERTA INMEDIATA ANTE SITUACIONES DE PELIGRO EN EL HOGAR

Proyecto de Grado Presentado para optar el grado en Ingeniera Informática

Presentado por: Sergio Rodrigo Cárdenas Rivera

Tutor: Jorge Orellana Araoz

COCHABAMBA - BOLIVIA
Diciembre - 2020

Dedicatoria

Dedico este proyecto especialmente a mis padres porque sin su apoyo incondicional no lo hubiera logrado. Su compañia y bendición a lo largo de mi vida me reconforta y me lleva por el camino del bien. Por eso les entrego este trabajo en ofrenda por su paciencia y amor infinito. Los quiero mucho.-

Agradecimientos

Agradezco a la prestigiosa Universidad Mayor de San Simón y a su prestigiosa Facultad de Ciencia y Tecnología, por darme una oportunidad más para poner en práctica lo que he aprendido y desarrollar herramientas útiles para la sociedad. A mi familia, por todo el impulso y apoyo incondicional que me dieron para superar todos los obstáculos que se presentaron en mi vida hasta ahora.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El término "Seguridad" es usado para referir a la ausencia de riesgo o a la confianza explicita en algo o alguien; pero este panorama toma diversos sentidos según el campo en el que se enfoca la seguridad. Aunque el objetivo consista en reducir el riesgo a niveles aceptables, el mismo es inherente a cualquier actividad o situación y en ninguna circunstancia puede ser eliminado.

Desde la aparición del hombre sobre la faz de la tierra, siempre prevaleció el instinto de supervivencia, donde surge la necesidad de obtener y/o brindar seguridad ante cualquier peligro que ponga en riesgo la integridad física propia y la de sus seres más cercanos. Cuando las primeras sociedades se formaron, una de las principales tareas del estado fue administrar justicia y brindar seguridad.

En el ámbito de la seguridad, la video-vigilancia se define como el acto de observar una escena o escenas en busca de comportamientos específicos que podrian ser anormales o podrian indicar una posible emergencia o existencia de un comportamiento impropio (Norman, 2017). Los sistemas de video-vigilancia en la actualidad, se han convertido en una herramienta esencial de la seguridad para mantener "observado/vigilado" un espacio muy importante para quien requiere el sistema; donde el mismo esta compuesto por un conjunto de cámaras, monitores y grabadoras los cuales forman parte esencial del sistema. Estos sistemas pueden ser instalados tanto en interiores como en exteriores de una propiedad o establecimiento, especialmente en lugares donde se desea mantener una vigilancia constante.

Gracias a la tecnología actual se ha podido automatizar la mayoría de las tareas que los humanos realizan y el campo de la video-vigilancia no ha sido la excepción. Con los continuos avances tecnológicos cada vez se desarrollan sistemas más completos y avanzados, permitiendo incrementar su eficacia y confiabilidad; por ejemplo la capacidad de poder vigilar en la oscuridad gracias a la tecnología de visión nocturna. Pero el campo más fascinante dentro de estos avances es el de la Inteligencia Artificial y específicamente la rama de la "Visión por Computadora". Gracias a las técnicas utilizadas en este campo de investigación, una computadora con el apoyo de algoritmos específicos y clasificadores, tiene la capacidad de identificar objetos, siluetas y/o elementos dentro de una escena captada por una cámara.

Estas nuevas características pueden ser explotadas en un sin fin de actividades diarias donde es necesaria la supervisión de una persona, permitiendo aún más la automatización de tareas de

vigilancia. El problema a afrontar a partir de este escenario es evaluar si lo que esta siendo identificado en una escena representa un peligro para las personas.

1.1. Antecedentes

En la actualidad es común que empresas e instituciones tengan instalados sistemas de seguridad en sus ambientes como ser: oficinas, sitios de producción, almacenes, entradas, recepción, etc. pero realmente no solo las empresas tienen algún riesgo de situación de peligro o robo, si no también las personas en sus respectivos hogares.

Con el contínuo crecimiento del mercado de la seguridad, el precio de los equipos de video-vigilancia tendieron a decrecer pero aun no son accesibles para todo el mundo. Este hecho asociado con el incremento de la inseguridad independientemente de cada país, promueve los siguientes escenarios: un incremento en el uso de sistemas de video-vigilancia, sistemas con varias cámaras funcionando al mismo tiempo siendo monitoreadas solo por un usuario el cual no esta disponible todo el tiempo y la ausencia de características avanzadas de reconocimiento de escenas en sistemas de video-vigilancia convencionales.

1.2. Descripción del Problema

Cuando el responsable de una casa esta ausente y nadie esta vigilando su hogar, la posibilidad de acontecer una situación anormal siempre esta presente. Si en el peor de los casos llegase a ocurrir algo en su hogar, esta persona solo se llega a enterarse si algún vecino se comunica con él para avisarle lo sucedido o en el peor de los casos, enterarse directamente a su regreso. Un sistema de video-vigilancia con las características de identificar movimiento y situaciones de peligro como ser: presencia de intrusos, fuego y humo, puede reducir el daño causado por los sucesos antes descritos por medio de la acción inmediata por parte del usuario en el momento de ser notificado, apoyado por la visualización en tiempo real de lo que estan captando las cámaras.

1.2.1. Definición del problema

Dificultad para advertir de forma inmediata situaciones de peligro en el hogar.

1.3. Objetivos del Proyecto

A continuación se presentan el objetivo general y los objetivos específicos.

1.3.1. Objetivo General

Facilitar la alerta inmediata ante situaciones de peligro en el hogar por medio de un sistema de video-vigilancia inteligente.

1.3.2. Objetivos Específicos

1. Describir todos los factores que implican el proceso de transmisión de datos por la red.

2. Especificar el proceso de análisis y procesamiento de imágenes con inteligencia artificial.
3. Proveer una red neuronal para el reconocimiento y análisis de video.
4. Identificar las partes que conforman el proceso de transmisión de video.
5. Describir medios para la interacción entre la transmisión y el análisis de imágenes.
6. Proveer el medio de acceso y notificación entre el sistema y el usuario.

1.4. Justificación

El riesgo de que un suceso ponga en peligro la integridad física y material de las personas esta presente cada día y en cualquier lugar. A pesar de que esta posibilidad es imposible de eliminar, se pueden crear mecanismos que contrarresten el impacto que ocasionan dichos sucesos en los sitios que se quieren evitar. Algunas situaciones más comunes que pueden representar un peligro a la integridad física y/o material del hogar son: la presencia de intrusos en ausencia del responsable en el hogar y la presencia de fuego y/o humo en el interior y/o exterior del hogar.

Los sistemas de video-vigilancia permiten la visualización en tiempo real de lo que las cámaras estan captando, pero es necesario una persona que ejecute la acción constante de revisar dicha transmisión para identificar y alertar sobre algunas situaciones que según su criterio pueden llegar a ser peligrosas. Si la cantidad de cámaras es considerable, la eficacia del operador del sistema disminuye al tener que revisar la transmisión de varias cámaras.

Con el aprovechamiento de la tecnología actual se plantea la implementación de un prototipo para un sistema de video-vigilancia inteligente que permita retransmitir de manera remota los fotogramas captados por las cámaras, con la característica de alertar al usuario sobre los sucesos antes descritos una vez que se identifican por medio de técnicas de visión por computadora y redes neuronales, para la acción inmediata del usuario con el fin de disminuir su impacto.

1.5. Alcances y límites

- La transmisión de fotogramas será implementado tanto para su ejecución en un ambiente local, como en línea.
- La notificación del evento identificado se realizará por medio de correo electrónico.
- La visualización en vivo del registro de la cámara se realizará por medio de un reproductor web de video con la capacidad de reproducir video en vivo HLS (HTTP Live Streaming).
- Se implementará la detección de: movimiento, silueta humana, fuego y humo.
- Los fotogramas capturados serán analizados por medio de técnicas de visión artificial y redes neuronales.
- La transmisión de video en tiempo real se realizará por medio de un servidor web que implementa software libre en el streaming de video.

Capítulo 2

Marco Teórico

2.1. Sistema de video vigilancia

El término "video-vigilancia" es usado para hacer referencia al despliegue de cámaras de vídeo que cumplen el rol de videofilmadoras, las cuales guardan el contenido recolectado en un almacén digital y puede ser visualizado en un monitor central (Wikipedia, 2020). Por lo tanto, un sistema de video-vigilancia consiste en una instalación de seguridad cuya finalidad es el control y supervisión visual en tiempo real de instalaciones locales y remotas, mediante el uso de múltiples cámaras de vigilancia, así como de sistemas de visualización, grabación y archivo. Estos sistemas ayudan a proteger a las personas, bienes y recursos, manteniendo una alerta activa con un gran efecto disuasorio.

El sistema llega a capturar imágenes y vídeos, que pueden ser comprimidos, almacenados, o enviados por una red de comunicación y pueden ser instalados en cualquier ambiente. En la figura 2.1 se visualiza el conjunto de elementos que forman un sistema de video-vigilancia. Este sistema compone de un conjunto de cámaras que estan conectadas directamente a un grabador de video en red o N.V.R. (Network Video Recorder), el cual permite la visualización de las imágenes captadas por las cámaras en un monitor local y por medio de una conección a internet, permite su visualización en dispositivos externos a la red local.

Figura 2.1: Componentes de una sistema de video-vigilancia.
Fuente: (Solintel, 2018)

Existe una amplia oportunidad para el mercado de la video-vigilancia en todas las regiones del mundo especialmente en Asia y la región del Pacífico, debido a la apertura de pequeños negocios y la construcción de áreas residenciales como también ciudades "inteligentes" (MarketsAndMarkets, 2020). El mercado creciente de la vigilancia ha permitido que desarrolladores independientes y

fabricantes diseñen nuevas implementaciones de sistemas de video-vigilancia, los cuales aplican nuevas características realizables gracias a la tecnología actual.

En la figura 2.2 se muestra como el mercado global de la video-vigilancia tuvo un valor de 42.9 billones de dólares en el 2019 y esta proyectado alcanzar a los 69.1 billones de dólares hasta el 2026; cuyo incremento registra una taza de crecimiento anual compuesto del 10 % desde el 2020 al 2026. (MarketsAndMarkets, 2020)

Figura 2.2: Proyección del mercado de la video-vigilancia.
Fuente: (MarketsAndMarkets, 2020)

El aspecto más relevante del crecimiento en las oportunidades de este campo, se da en la implementación de nuevas características en este tipo de sistemas; gracias a las técnicas de visión por computadora e inteligencia artificial (I.A.), a la escalabilidad gracias al uso de servicios basados en la nube. Las ramas afines a la inteligencia artificial, como el Machine Learning (Aprendizaje Automático) y el Deep Learning (Aprendizaje profundo) han permitido lograr grandes avances en el campo de la video-vigilancia.

Para la implementación del prototipo propuesto, es necesario describir el marco teórico relevante en la captura de fotogramas de video, transmisión de datos por medio de la red por el protocolo tcp/ip, reconstrucción de información, consolidación y procesamiento de imágenes, como también la transmisión de video en vivo. A continuación se detallan los conceptos teóricos relevantes anteriormente descritos.

2.2. Arquitectura de red

Una arquitectura de red es un esquema completo de comunicación entre computadoras, el cual provee: un esquema de trabajo, un diseño principal, la construcción y manejo de una red. (Technologies Javvin, 2004, 1). La arquitectura de red más importante es la de Interconexión de Sistemas Abiertos (OSI), desarrollada por la Organización Internacional para la Estandarización (ISO).

La arquitectura OSI, es un estándar abierto para la comunicación en red a través de diferentes equipos y aplicaciones. Aunque no está ampliamente implementado, el modelo de 7 capas OSI es considerado el modelo de arquitectura de red principal para la intercomputación y comunicación entre redes. En la figura 2.3 se puede apreciar el modelo OSI de 7 capas, detallando a continuación las siguientes capas:

1. Capa física (Physical)
2. Capa de enlace (Data Link)
3. Capa de red (Network)
4. Capa de transporte (Transport)
5. Capa de sesión (Session)
6. Capa de presentación (Presentation)
7. Capa de aplicación (Aplication)

Este modelo se organiza de la siguiente manera: las capas 7 a 4 se ocupan de las comunicaciones de extremo a extremo entre la fuente de datos y destinos, mientras que las capas 3 a 1 se ocupan de las comunicaciones entre los dispositivos de red. Por otro lado, las siete capas del modelo OSI pueden dividirse en dos grupos: **capas superiores** (capas 7, 6 y 5) y **capas inferiores** (capas 4, 3, 2, 1).

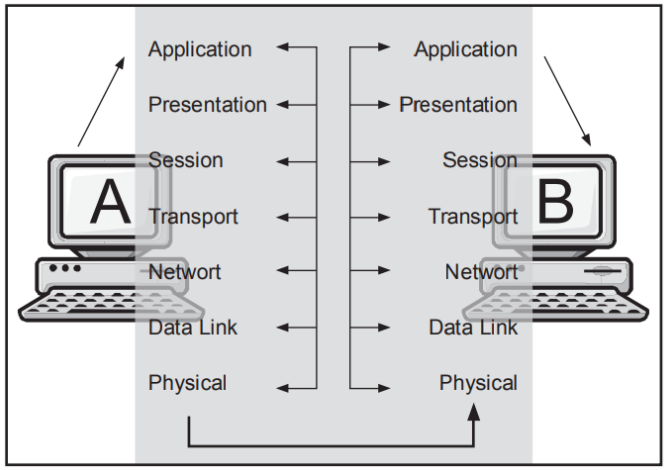

Figura 2.3: Esquema de capas del modelo OSI.
Fuente: (Technologies Javvin, 2004, 3)

Su contraparte en las arquitecturas de red del modelo OSI, es el modelo TCP/IP, que no sigue exactamente el modelo OSI. Desafortunadamente, no existe un acuerdo universal sobre cómo describir TCP/IP con un modelo en capas. Generalmente se acepta que TCP/IP tiene menos niveles (de tres a cinco capas) que las siete capas del modelo OSI. En la figura 2.4 se visualiza las capas que se adoptan en esta arquitectura.

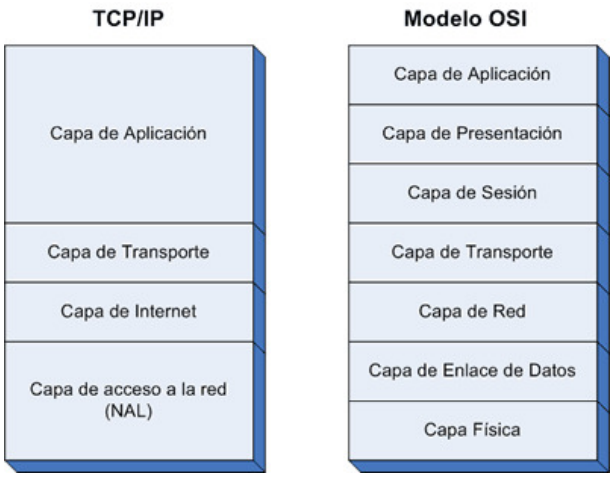

Figura 2.4: Modelo TCP/IP frente al modelo OSI.
Fuente: (Garza, 2013)

La arquitectura TCP/IP omite algunas características que se encuentran en el modelo OSI, combina las características de algunas capas OSI adyacentes y separa otras. La estructura de 4 capas de TCP/IP (capa de aplicación, transporte, internet y acceso a la red) se construye a medida que la información se transmite de la capa de aplicacion a la capa de red física.

Cuando los datos son envíados, cada capa trata la información que recibe de la capa superior como datos y agrega información de control (encabezado) al frente de esos datos y luego los pasa a la capa inferior. Cuando se reciben los datos, se lleva a cabo el procedimiento opuesto ya que cada capa procesa y elimina su encabezado antes de pasar los datos a la capa superior.

2.2.1. Protocolos

El modelo OSI, y cualquier otro modelo de comunicación de red, proporciona solo un esquema conceptual para la comunicación entre computadoras, pero el modelo en sí mismo no proporciona métodos específicos de comunicación (Wikipedia, 1994). La comunicación real está definida por varios protocolos de comunicación.

En el contexto de la comunicación de datos, un protocolo es un conjunto formal de reglas, convenciones y estructuras de datos que determinan cómo las computadoras y otros dispositivos de red intercambian información a través de una red. Este método estándar permite la comunicación entre procesos (que potencialmente se ejecutan en diferentes equipos) y agrega un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red.

Similar a la manera de hablar el mismo lenguaje entre dos personas; un protocolo, simplifica la comunicación entre dispositivos. La arquitectura de red proporciona solo un esquema conceptual para la comunicación. El modelo no proporciona métodos específicos de comunicación, sino mas bién, la comunicación real está definida por varios protocolos de comunicación que son usados en la comunicación analógica y digital, y pueden ser usados en el procesos de transferencia de archivos y acceso a internet.

Los protocolos de comunicación en red más populares, incluyen:

- **Automatización**: Automatizan diferentes procesos tanto en entornos comerciales como personales: en edificios inteligentes, tecnología en la nube o vehículos autónomos.
- **Mensajeria Instantánea**: La comunicación basada en texto, en teléfonos inteligentes y computadoras suceden debido a una serie de diferentes protocolos de mensajeria instantánea.
- **Enrutamiento**: Protocolos de enrutamiento permiten la comunicación entre routers y otros dispositivos de red.
- **Transferencia de archivos**: Envio de archivos por medio de un canal de comunicación.
- **Acceso a Internet**: El protocolo de Internet (IP) permite que los datos sean enviados entre dispositivos por medio de la red de internet.

A continuación se detallan algunos de los protocolos más conocidos:

- **HTTP - Protocolo de transferencia de hipertexto**: Este protocolo de internet define la manera en la que los datos son enviados por internet y determina como los navegadores web y buscadores deben responder a determinados comandos.
- **SSH - Secure Socket Shell**: Este protocolo provee un acceso seguro al dispositivo, incluso si se encuentra en una red no segura. SSH es particularmente usado por administradores de red quienes manejan diferentes sistemas de manera remota.
- **SMS - Servicio de envío de mensajes cortos**: Este protocolo ha sido creado para enviar y recibir mensajes de texto sobre redes de telefonía celular. SMS refiere exclusivamente a mensajes basados en texto. Las imágenes, videos u otro contenido multimedia requiere el protocolo de Servicio de mensajeria multimedia (MMS), que es una extensión del protocolo SMS.
- **ICMP - Protocolo de control de mensajes de Internet**: Trabaja como asistente del protocolo de Internet y se encarga de identificar fallos en la información y enviar mensajes de error hacia el usuario o servidor. Por ejemplo, si una dirección de este no está disponible o si una solicitud presenta fallas.
- **SMTP - Protocolo de transferencia de correo simple**: Este se encarga del intercambio de datos por texto en mensajes de correo electrónico entre ordenadores por medio de la red.

2.2.2. Modelo cliente-servidor

El modelo cliente-servidor es una estructura de aplicación distribuida que divide tareas o cargas de trabajo entre los proveedores de un recurso o servicio, denominados servidores, y los solicitantes del servicio, denominados clientes (Wikipedia, 2022). A menudo, los clientes y los servidores se comunican a través de una red informática en hardware independiente, pero tanto el cliente como el servidor pueden residir en el mismo sistema.

Un servidor ejecuta uno o más programas de servidor, que comparten sus recursos con los clientes. Un cliente normalmente no comparte ninguno de sus recursos, pero solicita contenido o servicio

de un servidor. En la figura 2.5 se aprecia un diagrama que representa el modelo cliente-servidor, donde los clientes acceden al servicio del servidor por medio de la misma red de internet.

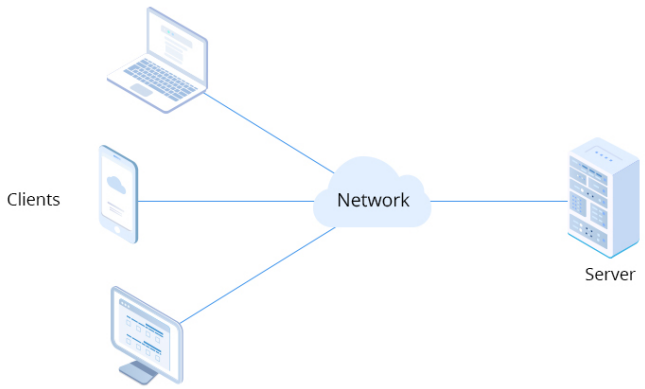

Figura 2.5: Ilustración del Modelo cliente-servidor
Fuente: (Howard, 2022)

Por lo tanto los clientes, inician sesiones de comunicación con los servidores, que esperan las solicitudes entrantes. Para mencionar algunos ejemplos de aplicaciones informáticas que utilizan este modelo cliente-servidor son el correo electrónico, la impresión en red y la World Wide Web (Internet).

La comunicación entre el cliente y el servidor se realiza por medio de una conexión o enchufe "socket" el cual define la dirección y puerto por la cual va a ser enviada y/o recibida la información entre ambos actores. En la figura 2.6 se visualiza la forma en la que se realiza la conexión entre el lado del cliente y el servidor por medio de un conector o "socket". En el lado del cliente: este conoce el nombre de host de la máquina en la que se ejecuta el servidor y el número de puerto en el que escucha el servidor.

Para realizar una solicitud de conexión, el cliente intenta conectarse con el servidor en la máquina y el puerto del servidor, además el cliente también necesita identificarse ante el servidor, para vincularse a un número de puerto local que se utiliza durante esa conexión. Normalmente lo asigna el sistema.

Figura 2.6: Solicitud de conexión por medio de un socket
Fuente: (Oracle, 2015)

Si la solicitud tiene éxito, el servidor acepta la conexión. Después esta aceptación, el servidor obtiene un nuevo "socket" o conector vinculado al mismo puerto local como también tiene su punto final remoto establecido en la dirección y el puerto del cliente. En la figura 2.7 se visualiza

el comportamiento del conector después de la aceptación a la petición de conexión del cliente; una vez establecida la conexión, el servidor necesita un nuevo socket para continuar escuchando las futuras solicitudes de conexión mientras atiende las necesidades del cliente conectado.

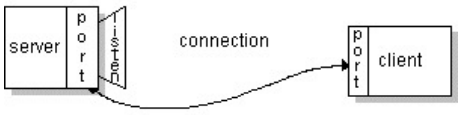

Figura 2.7: Conexión establecida entre sockets
Fuente: (Oracle, 2015)

El flujo general de interacción es como sigue: el servidor TCP establece un conector en una determinada dirección y puerto, y se mantiene escuchando constantemente si algún conector cliente (que conoce la dirección y puerto del servidor) solicita una conexión; cuando el servidor acepta la conexión, se brinda un nuevo puerto diferente de manera automática para comunicarse exclusivamente con la nueva coexión y mantiene el puerto original a la escucha de nuevas conexiones, como se ve en la figura 2.8. Cuando se establece esta conexión, tanto como el cliente y el servidor pueden enviar y recibir datos. Cuando el cliente decide cerrar la conexión, el servidor pone de nuevo el puerto utilizado anteriormente como disponible.

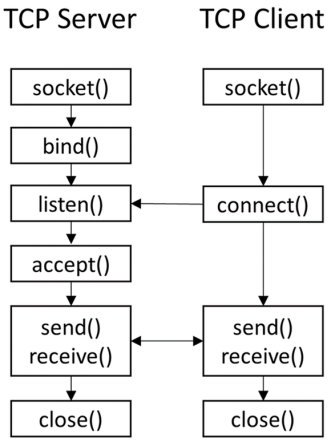

Figura 2.8: Flujo de interacción entre servidor y cliente TCP.
Fuente: (Gama et al., 2021)

A continuación se detalla los métodos que se muestran en la figura 2.8:

- socket(): Crea un punto final para la comunicación en el servidor.
- bind(): Asigna un número único al socket y reservar una combinación única de dirección IP y puerto para el socket creado.
- listen(): Espera a que un cliente se conecte.
- accept(): Recibe una solicitud de conexión de un socket de cliente.

- connect(): El cliente y el servidor están conectados entre sí.
- send()/recieve(): Intercambian datos entre el cliente y el servidor
- close(): El servidor y el cliente cortan la conexión.

Los clientes y servidores intercambian mensajes en un patrón de mensajería de solicitud-respuesta. Para comunicarse, las computadoras deben tener un lenguaje común y deben seguir reglas para que tanto el cliente como el servidor sepan qué esperar. El idioma y las reglas de comunicación se definen en un protocolo de comunicaciones. Todos los protocolos operan en la capa de aplicación.

2.2.3. HTTP

El protocolo de transferencia de hipertexto (HTTP) es el más utilizado en Internet. Se usa en cada transacción de la Web (www) y permite la transferencia de archivos (principalmente, en formato HTML) entre un navegador (cliente) y un servidor web. HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Este protocolo se orienta a las transacciones y sigue el esquema "Petición - Respuesta" entre un cliente y un servidor. En la figura 2.9 se aprecia la interacción entre el cliente y el servidor web.

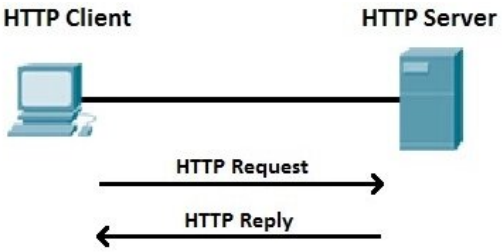

Figura 2.9: Interacción del protocolo HTTP.
Fuente: (CCNA-Certification, 2022)

Este protocolo se compone las siguientes características:

- El cliente que efectúa la petición (un navegador o un spider) es denominado "user agent" (agente del usuario).
- La información transmitida se denomina "recurso" y se identificada mediante una cadena de caracteres definida como dirección URL.
- Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

2.3. Inteligencia Artificial

La Inteligencia Artificial (I.A.), como área de estudio de las ciencias de la computación, en los últimos tiempos dejó de estar reservada para la investigación y pasó a formar parte del desarrollo de la sociedad. El cerebro es el órgano más increible del cuerpo humano; establece la forma en la que percibimos las imágenes, sonido, olores, sabores y el tacto; por lo tanto permite al ser

humano almacenar recuerdos, experimentar emociones e incluso soñar. Sin él, el ser humano sería un organismo primitivo, incapaz de otra cosa que el más simple de los reflejos. Por lo tanto, el cerebro es lo que hace inteligente al ser humano (Lopez, 2017).

Durante décadas se ha investigado para construir máquinas inteligentes con cerebros como el del ser humano; asistentes robotizados para limpiar los hogares, coches que se conducen por solos, microscopios que detecten enfermedades automáticamente. Pero en la construcción de estas máquinas artificialmente inteligentes se presentan problemas computacionales complejos; problemas que el cerebro humano puede resolver en una fracción de segundos. Las formas de analizar y resolver este tipo de problemas, son estudiados por la inteligencia artificial.

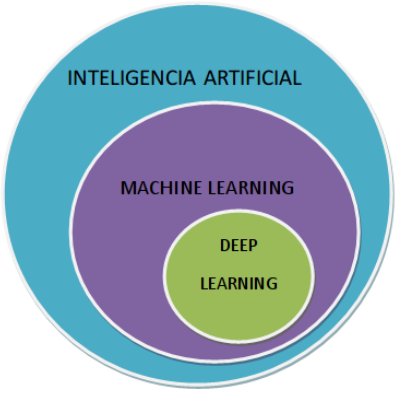

Figura 2.10: La Inteligencia Artificial y sus sub-areas
Fuente: (Tejada, 2019)

A menudo los conceptos de Inteligencia Artificial, Aprendizaje Automático (Machine Learning) y Aprendiazaje Profundo (Deep Learning) son usados de manera indistinta. Por los años '80 la Inteligencia Artificial era una característica que se alcanzaba al definir un conjunto de reglas que decían que hacer en un determinado momento, de esta manera un sistema 'inteligente' solo obedecía reglas de acción programadas (Banda, 2017). En la figura 2.10 se ilustra como la Inteligencia Artificial engloba los subcampos de estudio como son el Machine Learning y el Deep Learning.

2.3.1. Redes Neuronales

Las redes neuronales artificiales, son un modelo inspirado en el funcionamiento del cerebro humano. Está formado por un conjunto de nodos conocidos como neuronas artificiales que están conectadas y transmiten señales entre sí. Estas señales se transmiten desde la entrada hasta generar una salida. En la figura 2.15 se aprecia la estructura propia de una neurona artificial.

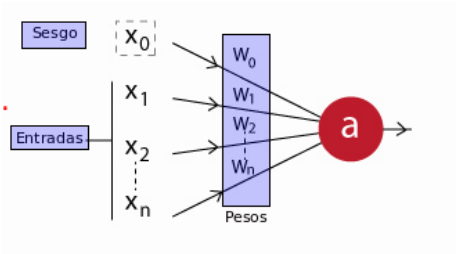

Figura 2.11: Ilustración de una Neurona artificial
Fuente: (Lopez, 2016)

El objetivo principal de un modelo neuronal, es aprender por medio de la modificación automática de si misma, llegando a realizar tareas complejas que no podrían ser realizadas mediante programación clásica basada en reglas. De esta forma se pueden automatizar funciones que al principio solo podrían ser realizadas por personas. Semejante al del cerebro humano, las redes neuronales reciben una serie de valores de entrada, donde cada una de estas llegan a un nodo llamado neurona.

Las neuronas de una red están a su vez agrupadas en capas formando una red neuronal. Cada una de las neuronas de la red poseen un peso, que es un valor numérico con el que modifica la entrada recibida. Los nuevos valores obtenidos por las neuronas continúan su camino por la red. En la figura 2.12 se aprecia el funcionamiento anteriormente descrito.

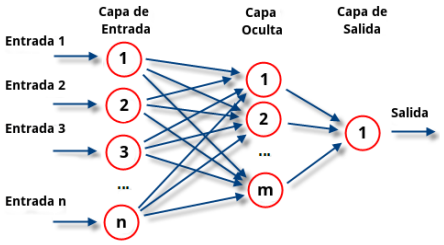

Figura 2.12: Modelo de capas de una red neuronal.
Fuente: (Wikipedia, 2012)

Cuando este proceso llega al final de la red, se obtiene una salida que será la predicción calculada por la red. Cuantas más capas posea la red y aumente la complejidad de sus conexiones; mas complejas serán las funciones que pueda realizar. Para que una red neuronal realice las funciones deseadas, es necesario la ejecución del entrenamiento correspondiente.

El entrenamiento de la red neuronal se realiza por medio de la modificación de los pesos de sus neuronas para que consiga extraer los resultados deseados. Para ello, se introduce datos de entrenamiento en la red, según el resultado obtenido y el error se modifican los pesos de las neuronas en función de cuanto haya contribuido cada neurona a dicho resultado. Este método es conocido como "backpropagation" o propagación hacia atrás. Con este método se consigue que la red "aprenda", consiguiendo un modelo capaz de obtener resultados muy acertados incluso con datos muy diferentes a los de su entrenamiento (Innovation, 2019).

Estas redes neuronales son utilizadas para tareas de predicción y clasificación. Esta técnica se ha convertido en una pieza clave para el desarrollo de la Inteligencia Artificial. Como se describió previamente, es uno de los principales campos de investigación y el que más ha evolucionado con el tiempo, ofreciendo cada vez soluciones más complejas y eficientes.

Redes neuronales convolucionales

Dentro del conjunto de tipos de redes neuronales están las convolucionales, que se usan en el campo de la visión artificial. Las redes neuronales convolucionales representan un algoritmo de Aprendizaje Profundo (Deep Learning) que está diseñado para trabajar con imágenes. Toma las imágenes como entrada, les asigna un valor (peso) a ciertos elementos en la imagen para su clasificación.

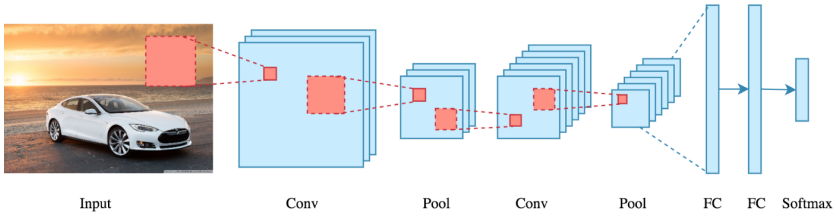

Figura 2.13: Ilustración de una Red Neuronal Convolucional.
Fuente: (Bootcamp-ia, 2019b)

Estas redes han contribuido en el desarrollo y perfeccionamiento del campo de la visión por computadora. Las redes convolucionales contienen varias capas ocultas como se ilustra en la figura 2.12, donde las primeras puedan detectan líneas, curvas y así se van especializando hasta poder reconocer formas complejas como un rostro, siluetas, etc. (Saha, 2018).

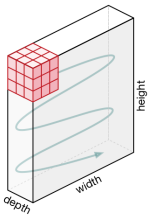

Figura 2.14: Movimiento del Kernel.
Fuente: (Bootcamp-ia, 2019a)

Estas redes funiconan en base de convoluciones. Este proceso consiste en tomar un grupo de píxeles de la imagen de entrada e ir realizando un producto escalar con un kernel. El kernel recorrerá todas las neuronas de entrada y obtendremos una nueva matriz, la cual será una de las capas ocultas. En el caso de que la imagen sea de color se tendrán 3 kernels del mismo tamaño que se sumarán para obtener una imagen de salida. El "kernel" dentro las redes convolucionales es considerado como un filtro que se aplica para extraer ciertas características importantes o patrones. Se usa para detectar bordes, enfoque, desenfoques, y características de la imagen por la convolución entre la imagen y el kernel. Este proceso se desarrolla entre la imagen y el kernel,

con la finalidad de que el kernel recorra toda la imagen (pixel por pixel). Por lo general, el kernel es de menor tamaño que la imagen permitiendo multiplicar el kernel con la porción de imagen escogida, realiza un producto escalar a medida que el kernel se va desplazando; por esta razón es un proceso iterativo. Las tareas comunes de este tipo de redes son: detección o categorización de objetos, clasificación de escenas y clasificación de imágenes en general.

2.4. Machine Learning (Aprendizaje de Máquina)

Es un subcampo de la Inteligencia Artificial cuyo objetivo es entender la estructura de la información y ajustar estos datos en modelos que puedan ser entendidos y utilizados por las personas. (Tagliaferri, 2017). A diferencia de la computación tradicional, donde los algoritmos resuelven problemas específicos, los algoritmos de Machine Learning entrenan a las computadoras con datos de entrada y emplean análisis estadístico para generar valores de salida que se clasifican según a un rango específico. Por eso el Machine Learning facilita a las computadoras construir modelos a partir de datos ejemplo para automatizar el proceso de toma de decisiones basados en estos datos.

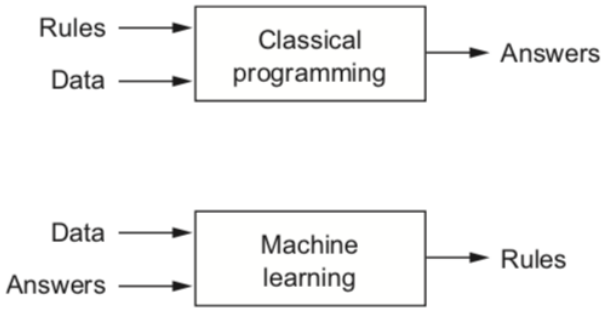

Figura 2.15: Diferencias entre programación clásica y M.L.
Fuente: (Chollet y Allaire, 2018)

En la figura 2.15 se visualiza la diferencia y similitud entre la programación clásica de la inteligencia artificial de los años '80 y las novedosas técnicas del aprendizaje automático. La programación clásica necesita de reglas y datos de entrada para que esta funcione como un sistema inteligente y pueda dar una respuesta, mientras que el Machine Learning necesita de datos y sus respectivas respuestas esperadas, para identificar patrones que los relacionen y asi desarrollar reglas que generan respuestas para nuevos datos.

2.4.1. Métodos de Machine Learning

En el Machine Learning, las tareas se clasifican en grandes categorias, las cuales estan basadas en el modo en el que el "aprendizaje" es ejecutado. Los métodos más adoptados en el Machine Learning son: el aprendizaje supervisado y el no supervisado.

Aprendizaje Supervisado

La computadora es provista de entradas ejemplo las cuales se categorizan con sus respectivas salidas esperadas. El propósito de este metodo consiste en que el algoritmo pueda "aprender" comparando la actual salida con las salidas esperadas para encontrar errores y en consecuencia modificar el modelo. El aprendizaje supervisado por lo tanto usa patrones para predecir valores categorizados en datos no categorizados.

Aprendizaje No Supervisado

La información provista a la computadora no está categorizada, por lo que los algoritmos de aprendizaje buscan similitudes entre los datos de entrada. Como los datos no etiquetados son más abundantes que los datos etiquetados, los métodos de aprendizaje automático que facilitan el "aprendizaje" pasan a ser más importantes.

2.5. Deep Learning (Aprendizaje profundo)

En la figura 2.10, se visualiza que el aprendizaje profundo es un subcampo dentro del Machine Learning, el cual hace uso de distintas redes neuronales para lograr el "aprendizaje" de sucesivas capas de representación que son relevantes para los datos. El término Deep "profundo", hace referencia a la cantidad de capas de representación que se utilizan en un modelo; en general es posible utilizar decenas incluso cientos capas de representación, los cuales aprenden de forma automática a medida que el modelo es entrenado con los datos (Briega, 2015).

2.6. Técnicas de visión por computadora

La visión por computadora es una técnica de recolección de información que surge por la inspiración en el sistema visual humano, el cual es la principal fuente de información para el cerebro. Su meta es de modelar y automatizar el proceso de reconocimiento visual de objetos en la vida real.

De los cinco sentidos que poseen las personas, la vista es la más importante. Por lo tanto la visión, es una tarea de procesamiento de información; pero tiene un grado de complejidad elevado, saber que es lo qué hay en el mundo, el cerebo humano deben se capaz de representar esta información sobre el color, la forma, movimiento, detalle y belleza de los objetos. (Briega, 2015)

La visión por computadora o visión artificial compone de un conjunto de herramientas y métodos que permiten obtener, procesar y analizar imágenes del mundo real por medio de una computadora. Estos métodos van a permitir automatizar un amplio conjunto de tareas al aportar a las computadoras información que es necesaria para la toma de desiciones en sus tareas asignadas. Esta técnica trata de imitar a la visión humana, usando geometría y un enfoque estadístico para solucionar el problema.

2.6.1. Aplicaciones

Esta rama de la Inteligencia Artificial aún sigue en investigación y mejoras donde sus aplicaciones más comunes son:

- **Reconocimiento óptico de caracteres:** Detección automática de símbolos que pertenecen a un alfabeto.
- **Inspección robotizada:** Revisión rápida de piezas para garantizar la calidad de componentes fabricados.
- **Modelado 3D:** Construcción de modelos 3D a partir de fotografías.
- **Imágenes médicas:** Análisis de radiografías.
- **Conducción segura:** Detección de obstáculos por medio de un sistema de conducción asistida por cámaras.
- **Vigilancia:** Monitoreo de intrusos, análisis del tráfico vial, monitoreo de piscinas, etc.
- **Detección de rostros:** Mediante algoritmos de reconocimiento facial se reconocen rostros usados en métodos de biometría.

2.6.2. Librerias

Una de las librerias mas utilizadas para las técnicas de vision por computadora es OpenCV. Es una biblioteca de uso libre para el desarrollo de aplicaciones usando visión artificial desarrollada por Intel. Esta libreria reune diversas caracteristicas que la hacen popular, por ejemplo:

- Permite su uso para fines comerciales y de investigación.
- Se encuentra disponible par varias plataformas como ser GNU/Linux, Mac OS, Windows y Android.
- Documentación completa y explicada, con una comunidad de desarrolladores activa.
- El procesamiento de imágenes en su escalado, eliminación de ruido y formateo de imagen y video.
- El uso y modificación de sus 2500 modelos pre-optimizados que son incluidos en la libreria, acorde a las necesidades del usuario.
- El uso del estado del arte de modelos de visión por computadora como también de aprendizaje de máquina (Machine Learning).
- El desarrollo de modelos en varias categorías de investigación como ser: reconocimiento facial, detección y seguimiento de objetos, extracción de modelos 3D, etc.

Figura 2.16: Logotipo de la libreria
Fuente: Web

Existen otras librerias que no son tan populares y representan un pago adicional.

2.7. Streamming

Este término también se denomina como transmisión, lectura en continuo, difusión en flujo, lectura en tránsito, difusión en continuo, descarga continua. Es la distribución digital de multimedia a través de una red de computadoras, de manera que el usuario consume el producto (generalmente archivo de vídeo o audio) en paralelo mientras se descarga (Austerberry, 2005).

Se aplica habitualmente a la difusión de audio o vídeo. El streaming requiere una conexión por lo menos de igual ancho de banda que la tasa de transmisión del servicio. El streaming de vídeo se popularizó a fines de la década de 2000, cuando el ancho de banda se hizo lo suficientemente barato para gran parte de la población. Sin embargo, con la tecnología del streaming, un archivo puede descargarse y reproducirse al mismo tiempo, con lo que el tiempo de espera es mínimo (Austerberry, 2005). El streamming permite la reproducción directa del contenido multimedia, sin la necesidad de almacenar en memoria la información para después reproducirla, como se puede ver en la figura 2.17.

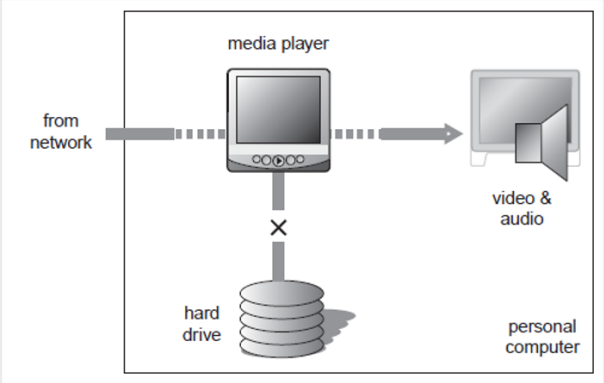

Figura 2.17: Streaming en un servidor web.
Fuente: (Austerberry, 2005)

El servidor más utilizado para la transmisión de contenidos multimedia es el servidor web, tipificado por Apache. Los servidores Web utilizan HTTP a través de TCP/IP para entregar sus páginas HTML y archivos de imagen asociados. El protocolo TCP/IP se utiliza como la capa de transporte a través de Internet. Los archivos se descargan a la memoria caché del navegador web más rápido de lo que el sistema permite. TCP incorpora control de flujo para gestionar la velocidad de descarga.

2.7.1. Servidor web para el Streaming de contenido multimedia

El servidor web reproduce contenido multimedia de la siguiente forma: el servidor mediante una aplicación proporciona una dirección que permite acceder a un archivo multimedia por medio de un reproductor. Luego que se establece una conexión al servidor web mediante el mismo enlace, se manda una señal para que el servidor transmita el contenido. El servidor web redirige al reproductor a un servidor streaming para luego recién realizar la reproducción del contenido digital, como se observa en la figura 2.18.

Un servidor web no realiza este proceso sin un servidor Streaming de por medio; permite la descarga del archivo en su totalidad para que después se realice la reproducción por parte del reproductor. El servidor streaming decodifica el bitrate en el que se codificó. Los oyentes deben tener instalado un reproductor multimedia en su ordenador, pero también se puede escuchar la emisión sin necesidad de éste último. Se podría utilizar: Windows Media Player, VLC, o algún software compatible.

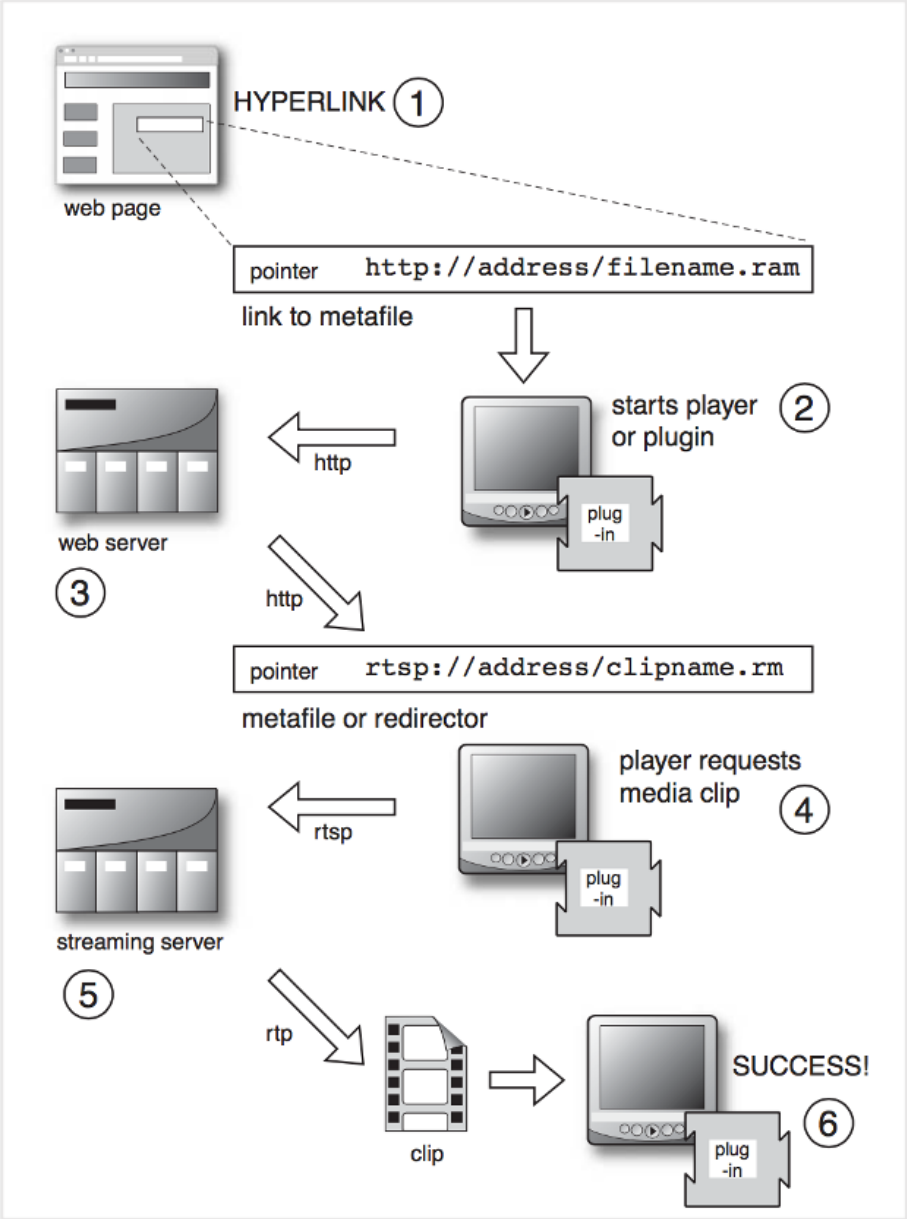

Figura 2.18: Streaming en un servidor web.
Fuente: (Austerberry, 2005)

El termino bitrate se refiere a la calidad de la señal, usualmente se utilizan 24, 32, 48, 64 kbps pero si se requiere se puede emitir a un bitrate superior, por ejemplo 96 kbps o 128 kbps. Se debe tener en cuenta que a mayor bitrate es necesario mayor ancho de banda, así como mayor será el costo. Sin la tecnología de Streaming, para reproducir un archivo multimedia, seria necesario

descargar en memoria del computador para después recién reproducirlo.

Http Live Streaming (HLS)

HLS es un protocolo utilizado para enviar audio y video sobre HTTP, soporta bitrate adaptivo y es compatible con la mayoría de los dispositivos. Este protocolo fue desarrollado por Apple y puede ser utilizado tanto para video en vivo como video en demanda. Entre sus características importantes se tiene:

- Se adapta a las condiciones de la red. Según el ancho de banda de internet del espectador, entrega una calidad o bitrate adecuado.
- Puede ser entregado desde un servidor tradicional de HTTP.
- Es protegido con encriptado o autenticación con el fin de proteger derechos de autor.

El flujo del protocolo es como sigue:

1. El archivo de video se codifica con compresión de video H.264 y compresión de Audio AAC.
2. Selecciona MPEG2-TS como contenedor de video, esto lo hace normalmente un encoder de video.
3. Cuando es codificado el video, se realiza un proceso de segmentación. Este proceso realiza cortes del video cada ciertos segundos, usualmente cada 10 segundos. A estos fragmentos se les conoce como segmentos o "chunks" de video.
4. Se genera un archivo índice. Este archivo contiene la información de los segmentos y en donde se encuentran para que el player los descargue. Este mismo proceso se realiza tanto para video en vivo o video en demanda (video almacenado).
5. Cuando se decide encriptar la información, se ejecuta un algoritmo para proteger cada uno de los segmentos. Donde el player que tenga la llave para desencriptarlos puede ver el contenido.
6. Estas llaves pueden ser estáticas o dinámicas. Puede generarse una llave para todos los videos o una llave por video.

La URL de reproducción tiene la siguiente forma:

```
http://nombre-del-dominio/directorio-del-video/mi-video.m3u8
```

Usando el mismo protocolo que impulsa la web, HLS comparte contenido usando servidores web y redes de entrega de contenido ordinarios. HLS está diseñado para brindar confiabilidad y se adapta dinámicamente a las condiciones de la red al optimizar la reproducción para la velocidad disponible de las conexiones por cable e inalámbricas.

En la figura, 2.19 se visualiza un esquema que representa el proceso de que sigue HLS. El archivo índice ".m3u8" permite a los reproductores de video descargar partes del contenido a medida que se reproduce, según el ancho de banda disponible en ese momento. Si el ancho de banda es bajo

se realizan peticiónes del video de una calidad más baja para no interrumpir la reproducción, caso contrario se realizan peticiones a la calidad normal o más alta.

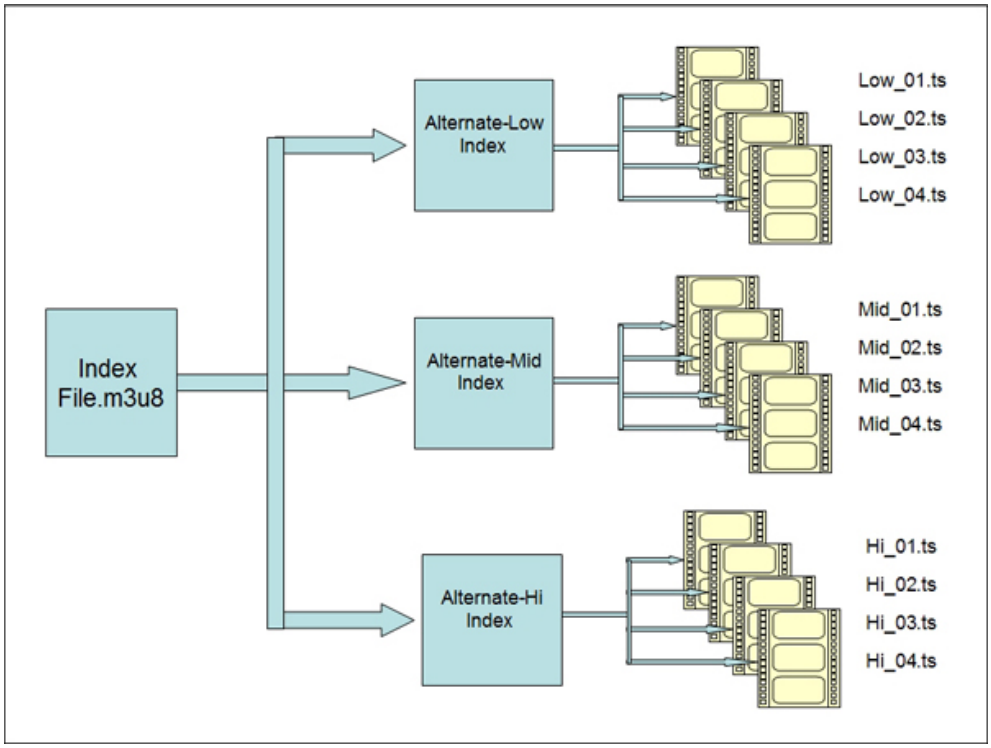

Figura 2.19: Http Live Streaming.
Fuente: (Ozer, 2017)

DASH

MPEG DASH (Dynamic Adaptive Streaming over HTTP) es un estándar para la transmisión adaptativa a través de HTTP que tiene el potencial de reemplazar tecnologías patentadas existentes como Microsoft Smooth Streaming, Adobe Dynamic Streaming y Apple HTTP Live Streaming (HLS) (Ozer, 2018). Todas las tecnologías de transmisión adaptativa basadas en HTTP utilizan una combinación de archivos multimedia codificados y archivos de manifiesto que identifican transmisiones alternativas y sus respectivas URL.

Los reproductores respectivos monitorean el estado del búfer (HLS) y la utilización de la CPU (Smooth Streaming y HTTP Dynamic Streaming) cambiando las transmisiones según sea necesario, ubicando la transmisión alternativa de las URL especificadas en el archivo de manifiesto. HLS usa segmentos MPEG-2 Transport Stream (M2TS), almacenados como miles de pequeños archivos M2TS, mientras que Smooth Streaming y HLS usan código de tiempo para encontrar el fragmento necesario de los flujos elementales MP4 apropiados.

En la figura 2.20 se visualiza el funcionamiento del estándar DASH. El reproductor realiza las peticiones necesarias a un enlace el cual esta disponible y gestiona las peticiones de los clientes como del servidor en el momente de guardar los videos codificados.

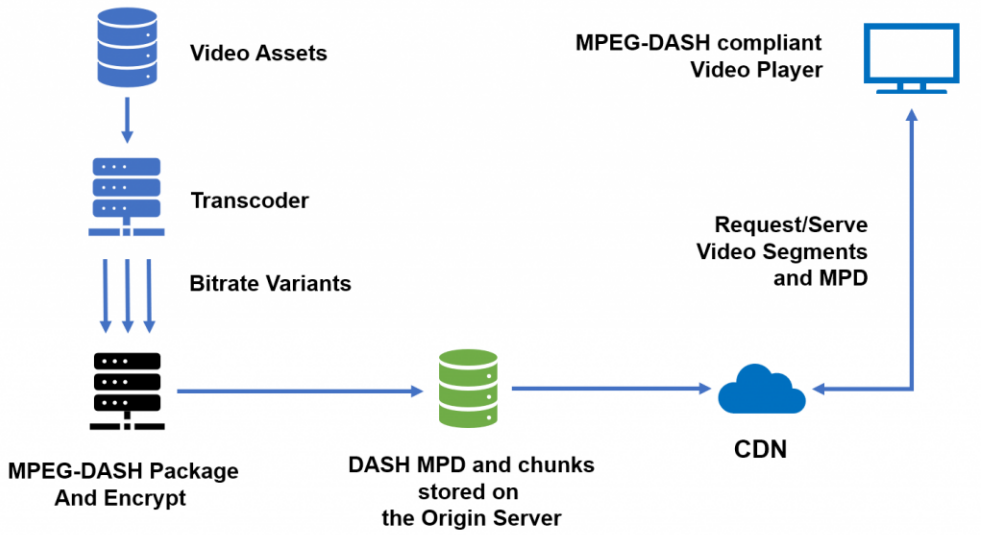

Figura 2.20: Dinamyc Adaptative Streamming Over Http.
Fuente: (Vijayanagar, 2021)

2.8. Lenguaje de Programacion

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Figura 2.21: Logo del lenguaje de programación Python.
Fuente: Web

Python usa tipado dinámico y conteo de referencias para la administración de memoria. Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos). También es usado en el campo del Machine Learning y el Deep Learning. Actualmente es el lenguaje más popular y de ámplio alcance en la inteligencia artificial (Henry, 2022).

2.9. Metodología de desarrollo

El concepto de ingenieria de software es propuesto por primera vez en el conjunto de conferencias históricas organizadas por el comité de ciencia de la OTAN.[1] en los años 60. Para ese tiempo la ingenieria de software tampoco era conocida ni aceptada. En ese entonces los proyectos de

[1]Organización del Tratado del Atlántico Norte.

software complejos eran problemáticos y costosos de completar, entonces se supuso que sería beneficioso considerar el desarrollo de software como Ingeniería (Ganis, 2010).

Los encargados de la codificación del software llamados programadores; en un principio eran ingenieros, pero como el costo computacional era alto, se utilizó un concepto de hardware denominado "mide dos veces, corta una vez" (Ganis, 2010). La naturaleza cautelosa de esta costumbre provocó el desarrollo de metodologías que permitieron a los equipos de proyectos creen planes lentos y metódicos para la creación de sistemas de software.

2.9.1. Modelo Cascada o Waterfall

En sus inicios, este concepto fue abordado por el Dr. Winston Royce, por medio de un artículo escrito sobre la gestión y dirección de proyectos grandes y complejos de software (Royce, 1970). En ese artículo en base en experiencias de desarrollo de Software para la planificación de misiones aereas; el autor describe los fundamentos del desarrollo de Software. Gran parte de esos fundamentos aún son aplicables en la actualidad. En el planteamiento de estos fundamentos, Royce presenta un conjunto de fases los cuales forman parte de una secuencia de desarrollo de software ilustrada en la figura 2.22.

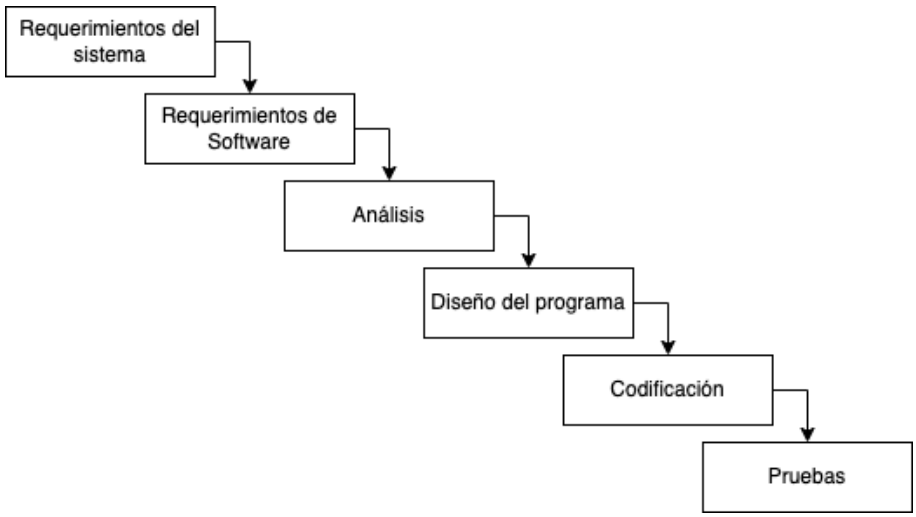

Figura 2.22: Modelo Cascada
Fuente: Elaboracion Propia

Una vez definida esta secuencia, se crea el concepto de "Cascada", como un modelo de desarrollo con actividades bien definidas y organizadas con un objetivo independiente dando origen al diseño del primer S.D.M.[2] (Bell y Thayer, 1976). En la figura 2.22 la fase de análisis y la de codificación entregan la mayor parte del producto esperado, mientras que las otras fases estan puestas para ser organizadas y planificadas de manera independiente para un mejor manejo de los recursos del proyecto.

De acuerdo al modelo Cascada, se enfatiza en la dependecia secuencial del producto entregado

[2]Metodología de Desarrollo de Software

en el paso previo. Es decir existe una dependencia que mantiene en espera el diseño del sistema mientras que el análisis del modelo no sea aprobado o concluido y consecuentemente la fase de codificación se verá en espera también hasta que el diseño se concluya.

Cada una de las fases guarda una relación iterativa con el siguiente paso definido en la metodología que asegura la completitud del producto entregado en la fase antetior. Esta relación esta ilustrada en la figura 2.23. Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida dentro de un proyecto de desarrollo de software.

En la práctica, se aplican diversas versiones del modelo. Los más habituales son los modelos que dividen los procesos de desarrollo en cinco fases. En ocasiones, las fases 1, 2 y 3 definidas por Royce se integran en una sola fase de proyecto a modo de análisis de los requisitos.

1. Análisis: planificación, análisis y especificación de los requisitos.
2. Diseño: diseño y especificación del sistema.
3. Implementación: programación y pruebas unitarias.
4. Verificación: integración de sistemas, pruebas de sistema y de integración.
5. Mantenimiento: entrega, mantenimiento y mejora.

La figura 2.23 se muestra la ampliación de esta metodología donde se añaden funciones iterativas al modelo básico como, por ejemplo, los saltos hacia atrás, que permiten comparar los resultados de cada una de las fases con las hipótesis obtenidas en la fase anterior, de modo que se puedan verificar.

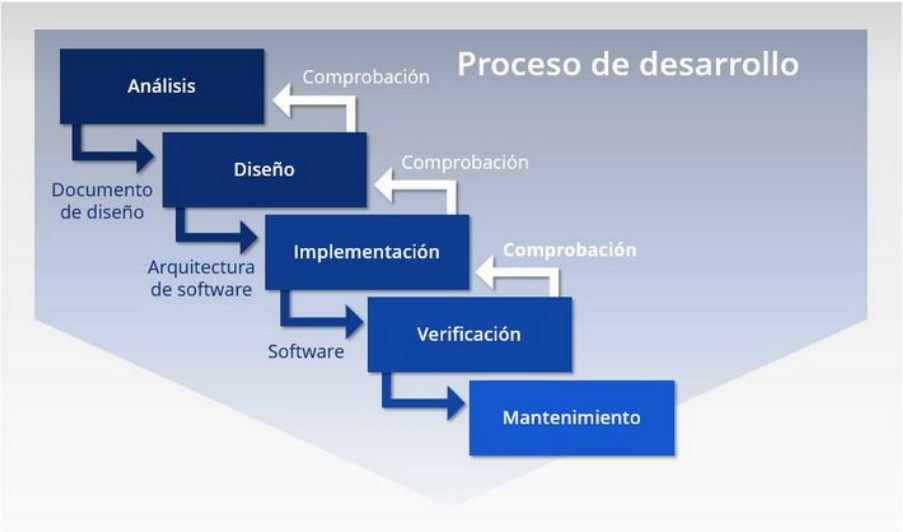

Figura 2.23: Modelo Cascada: Relación iterativa entre las fases sucesivas.
Fuente:

En este modelo, las diferentes fases de un proceso de desarrollo se suceden una detrás de otra como en una cascada. Cada una de las fases concluye con un resultado provisional o "hito". A continuacion se detalla cada una de las fases y las actividades que implica cada fase:

Análisis

Todo proyecto de software comienza con una fase de análisis que incluye un estudio de viabilidad y una definición de los requisitos. En el estudio de viabilidad se evalúan los costes, la rentabilidad y la factibilidad del proyecto de software. El estudio de viabilidad da como resultado un pliego de condiciones (una descripción general de los requisitos), un plan y una estimación financiera del proyecto, así como una propuesta para el cliente, si fuera necesario.

Este es el punto de partida en donde se plasma cada detalle de la idea y tomar el impulso necesario para su desarrollo. Esta fase consiste en determinar cuáles son las necesidades y los objetivos, para luego reunir todos los requisitos que se deben cumplir en el desarrollo del software para llevar a cabo todo el proceso.

Se realiza una definición detallada de los requisitos, incluyendo un análisis de la situación de salida y un concepto. Mientras que los análisis de salida se encargan de describir la problemática en sí, el concepto ha de definir qué funciones y características debe ofrecer el producto de software para cumplir con las correspondientes exigencias. La definición de los requisitos da como resultado un pliego de condiciones, una descripción detallada de cómo se han de cumplir los requisitos del proyecto, así como un plan para la prueba de aceptación, entre otros.

Por último, la primera fase del modelo cascada incluye un análisis de la definición de los requisitos en el que los problemas complejos se dividen en pequeñas tareas secundarias y se elaboran las correspondientes estrategias de resolución.

Diseño

La fase de diseño sirve para formular una solución específica en base a las exigencias, tareas y estrategias definidas en la fase anterior. En esta fase, los desarrolladores se encargan de diseñar la arquitectura de software, así como un plan de diseño detallado del mismo, centrándose en componentes concretos, como interfaces, entornos de trabajo o bibliotecas. La fase de diseño da como resultado un borrador preliminar con el plan de diseño del software, así como planes de prueba para los diferentes componentes.

Implementación

La arquitectura de software concebida en la fase de diseño se ejecuta en la fase de implementación, en la que se incluye la programación del software, la búsqueda de errores y las pruebas unitarias. En la fase de implementación, el proyecto de software se traduce al correspondiente lenguaje de programación. Los diversos componentes se desarrollan por separado, se comprueban a través de las pruebas unitarias y se integran poco a poco en el producto final. La fase de implementación da como resultado un producto de software que se comprueba por primera vez como producto final en la siguiente fase (prueba alfa).

Verificación

La fase de verificación o pruebas incluye la integración del software en el entorno seleccionado. Por norma general, los productos de software se envían en primer lugar a los usuarios finales seleccionados en versión beta (pruebas beta). Las pruebas de aceptación desarrolladas en la fase de análisis permiten determinar si el software cumple con las exigencias definidas con anterioridad. Aquellos productos de software que superan con éxito las pruebas beta están listos para su lanzamiento.

Mantenimiento

Una vez que la fase de prueba ha concluido con éxito, se autoriza la aplicación productiva del software. La última fase del modelo en cascada incluye la entrega, el mantenimiento y la mejora del software. En esta fase se analizan los resultados del hito anterior para realizar los cambios pertinentes para dar por concluido el proyecto.

2.9.2. Ventajas y desventajas del modelo Cascada

Esta metodología permite estructurar la organización de forma clara en aquellos proyectos de desarrollo en los que las diversas fases de proyecto se diferencian claramente entre sí. Como cada una de las fases concluye con un hito, el proceso de desarrollo es muy fácil de comprender. El punto clave del modelo reside en la documentación de todos y cada uno de los pasos de proceso. Los conocimientos adquiridos se registran en pliegos de requisitos o borradores preliminares.

En teoría, el desarrollo en cascada pretende crear los requisitos previos para una ejecución rápida y rentable de los proyectos a través de una cuidada planificación previa. Sin embargo, la utilización del modelo en la práctica es controvertida. Por una parte, en el desarrollo de software las fases de proyecto no suelen estar claramente diferenciadas entre sí.

En sentido estricto, el modelo en cascada no prevé la realización de ajustes a lo largo del proyecto. Sin embargo, un proyecto de software en el que todos los detalles del desarrollo se definieran al comienzo, solo podría concluir con éxito si desde el principio se invirtiera una gran cantidad de tiempo y dinero en análisis y diseño.

En resúmen se tienen las siguientes ventajas:

- Una estructura sencilla gracias a unas fases de proyecto claramente diferenciadas.
- Buena documentación del proceso de desarrollo a través de unos hitos bien definidos.
- Los costes y la carga de trabajo se pueden estimar al comenzar el proyecto.
- Aquellos proyectos que se estructuran en base al modelo en cascada se pueden representar cronológicamente de forma sencilla.

Como también las siguientes desventajas:

- Por norma general, los proyectos más complejos o de varios niveles no permiten su división en fases de proyecto claramente diferenciadas.

- Poco margen para realizar ajustes a lo largo del proyecto debido a un cambio en las exigencias.
- El usuario final no se integra en el proceso de producción hasta que no termina la programación.
- En ocasiones, los fallos solo se detectan una vez finalizado el proceso de desarrollo.

Se elige el modelo cascada dado que todas las ventajas se ajustan perfectamente al desarrollo del prototipo de sistemas de video-vigilancia. Por lo tanto las fases que tiene el modelo de desarrollo de software elegido para la realización del proyecto son:

1. **Análisis**: Esta fase se lleva a cabo a través de la definición de requisitos, la planificación y los casos de uso.
2. **Diseño**: Esta fase se lleva a cabo mediante un diagrama de secuencia y un diseño de la interfaz.
3. **Implementación**: Esta fase se lleva a cabo mediante la implementación del Sistema de video-vigilancia inteligente.
4. **Pruebas**: Esta fase se lleva a cabo mediante la realización de pruebas sobre el sistema de video-vigilancia.

Capítulo 3

Seguridad en el hogar

3.1. Introducción

La seguridad en el hogar es un tema relevante y delicado de manejar, principalmente cuando se trata del espacio vital más importante de todos, donde convive el ser humano. Sea del tipo que sea, en el hogar se establecen los vínculos más íntimos y personales; entonces la presencia de una persona en el hogar es un factor de seguridad de gran importancia. La mayoría de percances como intrusión de un extraño, allanamientos y robos se producen durante su ausencia. Los motivos para dejar un hogar vacío son varios: desde un viaje prolongado a salidas más o menos puntuales, regulares y/o diarias. Una vivienda vacía es más vulnerable que otra ocupada y este aspecto se toma en cuenta en el diseño de un sistema de seguridad para el hogar, el cual debe ofrecer características que ayuden a minimizar el impacto de las situaciones de peligro.

3.2. Ausencia en el hogar

En materia de seguridad del hogar, cualquier precaución, resulta de utilidad para prevenir, que ocurran incidentes y mantener protegidos a los seres más cercanos. De hecho el objetivo de la seguridad implica la toma de precauciones necesarias para que un lugar sea seguro para las personas.

Incluso las personas más retraídas y amantes de la soledad y el aislamiento deben, en un momento u otro, salir de su residencia habitual, algo que se convierte en largas horas de ausencia en la mayoría de los casos (evidentemente por trabajo, obligaciones académicas y otros menesteres cotidianos), y ocasionalmente, con mayor o menor asiduidad, por otras razones menos frecuentes (viajes, vacaciones, etc).

Todas las posibles situaciones a presentarse se definen en función del tiempo de ausencia; de modo que se establecen distintos casos con aspectos peculiares y específicos referentes a la seguridad y los riesgos; por ejemplo exponer las medidas de protección más básicas y elementales que siempre deben tomarse en cuenta, como el contrato de un seguro para la vivienda (con elementos importantes que figuran en toda póliza para su elección y contratación).

Un seguro contra robos puede proteger de alguna manera los daños materiales que pueden ocurrir, pero incluso para probar la veracidad del suceso es necesario presentar pruebas visuales que sirvan de referencia en la denuncia. Una imágen como en la figura 3.1 puede ayudar incluso en una

investigación.

Figura 3.1: Ilustración ejemplo de un intruso.
Fuente: Web

A contunuación se detellan las situaciones más comunes que pueden presentarse con sus respectivas acciones sugeridas para disminuir el peligro.

3.2.1. Ausencias cotidianas

Las ausencias diarias de horas o minutos son las más comunes y brindan oportunidades a asaltantes atentos. Para este caso se puede tener en cuenta medidas de protección sencillas y sin complicaciones que cualquiera puede llevar a cabo apenas sin inversión alguna. Asegurar los cierres de los accesos a la vivienda, disimular las ausencias o evitar proporcionar información sobre nuestros hábitos son algunas de las medidas que se exponen para evitar intrusiones no deseadas en el hogar.

3.2.2. Ausencias de termino medio

Cuando uno sale de casa previamente sabe si uno va ha volver al cabo de pocas horas, de unos días o de semanas; en cada caso se pueden presentar algunas peculiaridades y riesgos específicos que se deben afrontar de distintos modos. En este supuesto, tras las ausencias cotidianas, se detallan los casos de ausencias de pocos días, especialmente en fines de semana, puentes festivos y vacaciones cortas. En estas situaciones convergen la necesidad de contar con alarmas y avisadores técnicos, con la de disponer de sistemas de alarma y dispositivos antiintrusión los cuales, como veremos, pueden ser de muy diversa índole.

3.2.3. Ausencias prolongadas

Las vacaciones y las estancias de cierta duración en lugares alejados de nuestras residencias habituales ofrecen oportunidades únicas a posibles asaltantes. No ofrecer información sobre nuestro paradero, tratar de evitar el efecto de vivienda vacía, contar con la supervisión regular de alguien de confianza en nuestra ausencia y mantener a buen recaudo bienes u objetos de valor serán, en estos casos, las principales prioridades (sobre todo en el caso de las segundas residencias, una cuestión que también consideraremos detalladamente como caso diferenciado).

3.3. Situaciones de riesgo

3.3.1. Presencia de instrusos

Un intruso o persona ajena siempre representa un peligro en el interior de nuestro hogar y aún más cuando se desconoce el motivo de su presencia. La posibilidad de robos en cualquier cuidad del mundo esta presente y aun mas cuando este entra al interior de un hogar forzando cerraduras, encapuchado especialmente cuando los habitantes de la casa no estan. En la figura 3.2, se muestra una ilustracion de ejemplo de un intruso forzando la puerta de una casa.

Figura 3.2: Ilustración ejemplo de un ladrón
Fuente: Web

3.3.2. Fuego y humo

El fuego es una reacción quimica, donde un conjunto de partículas o moléculas incandecentes en materia combustible es capaz de emitir calor y luz. Con el calor se pueden llegar a desintegrar muchos objetos y estos mismos servir de combustion para que el fuego se expanda. Este fenómeno es uno de los principales causantes de tragedias en la actualidad, tanto como incendios forestales y/o colectivos, incendios en interiores, como ser casas, departamentos o sitios cerrados. En la figura 3.3 se visualiza la facilidad con la que el fuego puede expandirse en interiores.

Figura 3.3: Ilustración ejemplo de fuego en interiores.
Fuente: Web

El humo acompañado del fuego son elementos muy perjudiciales tanto como a las personas como al medio ambiente en general. El humo es uno de los factores principales que afectan a la salud respiratoria de las personas y animales en general. Identificar a tiempo la presencia de humo puede incluso prevenir y/o predecir la organización de fuego evitar tragedias.
En la figura 3.4, se visualiza como la presencia de humo puede ayduar a alertar de que hay fuego en el interior de una casa o habitación.

Figura 3.4: Ilustración de la presencia de fuego y humo en una habitación cerrada.
Fuente: Web.

3.4. Sistemas de seguridad

En el mercado, existe una gran variedad de artefactos, que están al alcance de todos para proteger los hogares frente a cualquier tipo de amenaza, tanto interna como externa. Los más eficaces y eficientes, son los sistemas electrónicos de seguridad. No obstante, sea cual sea la opción elegida se debe tener en cuenta los siguientes riesgos y amenazas:

- **Allanamientos, intrusiones y vandalismo:** riesgos procedentes del exterior, que se pueden mitigar fácilmente instalando cierres de alta seguridad en los accesos a la vivienda, alarmas antiintrusión u otros mecanismos disuasorios.
- **Accidentes domésticos:** riesgos procedentes del interior de hogar que pueden poner en riesgo la integridad física y/o moral de sus habitantes, tanto personas como mascotas, así como los bienes que contienen e incluso la misma infraestructura.

Las alarmas técnicas (alertas de fugas y escapes) y de emergencia son los sistemas más adecuados para proteger una vivienda. También es preciso tomar las medidas oportunas para proteger los componentes más sensibles del hogar (instalaciones de suministros y otros elementos de riesgo) de manipulaciones indebidas, golpes y otro tipo de percances que pueden ocasionar accidentes o situaciones indeseables.

3.4.1. Alarmas

Las alarmas son artefactos sonoros que emiten un sonido que provoca la alerta en las personas. Existen de diferentes tipos, medidas y campo de uso. El volúmen y el sonido es claramente diferenciado de cualquier objeto que emita un sonido cualquiera. Este objeto es utilizado generalmente para poner en alerta a todas las personas que lleguen a escucharlo y/o comunicar peligro. En la figura 3.5 se puede apreciar un modelo particular de alarmas sonoras.

Figura 3.5: Ilustración de alarmas con sonido.
Fuente: Web.

3.4.2. Sensores

Los sensores son dispositivos que captan magnitudes físicas (variaciones de luz, temperatura, sonido, etc.) u otras alteraciones de su entorno. Los detectores de humo son dispositivos desarrollados para detectar la presencia de un incendio en el interior de un edificio. En la figura 3.6 se aprecia un modelo en particular de sensor de detección de humo.

Figura 3.6: Ilustración de detector de humo.
Fuente: Web.

3.4.3. Cámaras

Las cámaras son dispositivos que permiten registrar imágenes estáticas y en movimiento. Específicamente las cámaras de vigilancia son las que se encargan de grabar todo lo que puede ocurrir en una casa o negocio. Contar con este tipo de cámara puede proporcionar sensación de seguridad y protección. Disponer de este tipo de sistemas puede resultar ser una solución para mantenerse protegido. El desarrollo de la tecnología ha logrado que el sector de la seguridad disponga de equipos eficientes y con diversas funcionalidades. En la figura 3.7 se visualizan diferentes modelos de cámaras de seguridad que se encuentran en el mercado.

Figura 3.7: Ilustración de diversas cámaras de seguridad.
Fuente: Web.

El tipo más común en el mercado son las cámaras de interiores ya que son las más sencillas y económicas del mercado ya que no necesitan mucho mecanismo ni protección. En la figura 3.8 se visualiza un ejemplo de cámara de interiores.

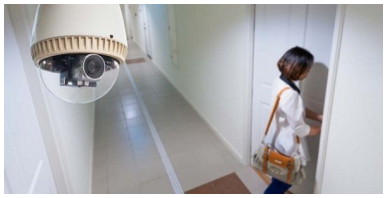

Figura 3.8: Cámara de vigilancia de interiores.
Fuente: Web.

Seguridad y vigilancia son aspectos que se requieren en todo el mundo; gobiernos, empresas, instituciones financieras, organizaciones de salud, necesitan cierto grado de medidas de seguridad y como resultado se generó un dramático incremento en la demanda de aplicaciones de seguridad como por ejemplo video vigilancia, monitoreo y grabación de: fronteras, puertos, transporte, hogares, corporaciones, instituciones educativas, lugares públicos, edificios, etc.

Capítulo 4

Análisis y Diseño

Las etapas iniciales de la metodología "Cascada" consisten en el análisis y diseño que son previos al inicio de la implementación del producto final. Es importante que estas fases sean completadas en su totalidad para que el desarrollo sea éxitoso. Este proyecto plantea un diseñado y desarrollo de 50 dias hábiles calendario con la distribución del tiempo definido según la complejidad de las tareas que implican cada una de las fases.

4.1. Análisis

Para el diseño de un sistema de video-vigilancia inteligente es necesario tener en cuenta los elementos principales que lo componen. Un sistema de video-vigilancia esta compuesto de cámaras individuales y un puesto central o servidor donde todas las conexiones convergen y se centralizan para su control. En el dispositivo central (servidor) se procesan las imágenes que las cámaras capturan y se convierten en video para ser visualizado en un monitor.

El sistema propuesto permite visualizar el video en vivo desde cualquier dispositivo con acceso a la red de internet además de que envía notificaciones automáticas en el instante en que se detecta: movimiento, fuego, humo o silueta de un intruso. El usuario recibe la notificación por correo electrónico el cual adjunta capturas y un enlace web para visualizar en vivo lo que esta captando la cámara.

En la figura 4.1 se visualiza el esquema general del sistema propuesto. Las cámaras de video-vigilancia se encargan de capturar los fotogramas de video y se enlazan por medio de un socket o conector (uno por cada cámara) al servidor TCP. Cuando es registrada una nueva conexión, el sistema notifica al usuario por medio de un correo electrónico, compartiendo información relevante sobre la conexión de una nueva cámara. El servidor TCP que maneja todas las conexiones, realiza el análisis de los fotogramas de manera individual por cada cámara conectada por medio de una libreria de visión por computadora. Paralelamente se construye el video que se va a transmitir a partir de los fotogramas y es decodificado por medio de un conjunto de paquetes de software libre, los cuales arman las partes del archivo para el streaming de video por la red. Cuando un evento (fuego, humo, movimiento) es identificado, automáticamente se envia un correo electronico al usuario notificando una posible identificación de un evento.

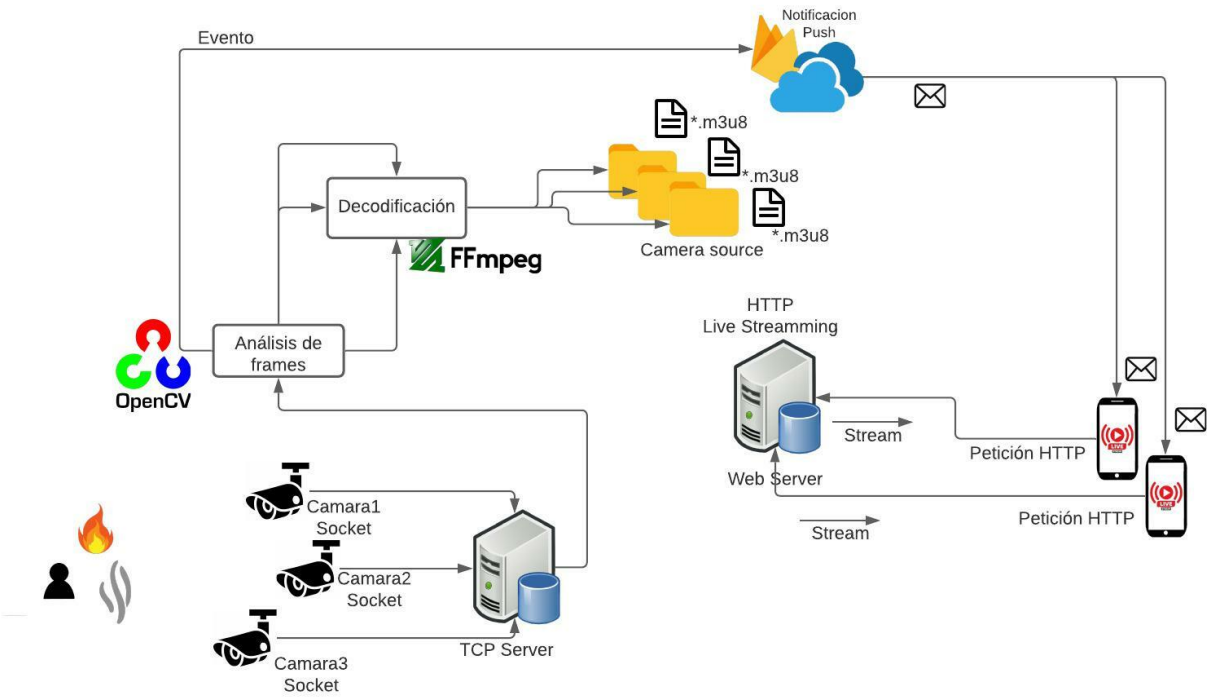

Figura 4.1: Diseño de interacción de los módulos del sistema de video-vigilancia
Fuente : Elaboración propia

4.1.1. Definición de Requerimientos

La definición de requerimientos es una de las actividades más importantes del desarrollo de software; de ello depende el resto de actividades del proyecto. Anteriormente se describe completamente el comportamiento del sistema que se desarrolla para la identificación de los requerimientos. Inicialmente se plantean los criterios de partida a tomar en cuenta en el diseño del sistema de video-vigilancia inteligente:

- Costos altos en la infraestructura de transmisión de video en vivo.
- Las características de identificación automática, estan disponibles para sistemas de video-vigilancia de alto nivel.
- Una alerta inmediata puede minimizar el impacto de alguna situación que ponga en peligro la integridad de bienes materiales y humanos.
- Los usuarios finales son personas que a menudo dejan su hogar para salir a trabajar, y usan constamente su correo electrónico.

Requerimientos Funcionales

En esta fase es necesario delimitar el alcance y las capacidades del sistema planteado para la realización de la planificación inicial, estimación de tiempos, diseño y desarrollo. Para ello se define la lista de requerimientos funcionales del sistema de video-vigilancia inteligente.

Tabla 4.1: Lista de requerimientos funcionales

1.	Capturar fotogramas por medio de una o varias cámaras portatiles.
2.	Visualizar fotogramas capturados en tiempo real (captura de video).
3.	Notificar al usuario cuando una nueva cámara se conecta.
4.	Enviar fotogramas capturados por medio de la red al servicio encargado de su análisis.
5.	Análizar y procesar fotogramas capturados individualmente por cada cámara.
6.	Permitir la recepción de fotogramas de varias fuentes hacia el servicio.
7.	Transmitir los fotogramas convertidos en video en vivo desde fuentes diferentes.
8.	Detectar movimiento a partir de los fotogramas recibidos desde distintas fuentes.
9.	Detectar silueta humana a partir de los fotogramas recibidos desde distintas fuentes.
10.	Detectar fuego y/o humo a partir de los fotogramas recibidos desde distintas fuentes.
11.	Notificar al usuario por medio de un correo electrónico cuando se de una detección.

Fuente: Elaboración propia.

Requerimientos No Funcionales

Los requisitos no funcionales son aquellos relacionados con la calidad y el proceso de desarrollo del sistema. Los tipos de requisitos no funcionales son: rendimiento, disponibilidad, acecesibilidad, usabilidad, estabilidad, portabilidad, costo, operatividad, interoperabilidad, escalabilidad, concurrencia, mantenibilidad, interfaz, plazo de entrega y herramientas. Los requisitos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto y las herramientas utilizadas.

1. **Requisitos de interfaz**
    - Las cámaras tendran una interfaz que permita visualizar el trabajo que estan ejecutando.
    - El servidor mostrará información necesaria sobre las tareas qeu realiza.
    - El sistema deberá ser de fácil configuración.
    - Las notificaciones por correo electrónico deberán ser visualmente agradables y estilizadas.
2. **Requisitos de portabilidad**
    - Los módulos de cámara podran ser conectarse de forma cableada o inalámbrica.
3. **Requisitos de disponibilidad**
    - La transmisión en vivo estará disponible cuando el usuario quiera visualizar lo que captan las cámaras.

4.2. Planificación

Todas las fases del modelo cascada son planificadas según la complejidad y tareas que presenta cada fase. Dado que cada fase debe culminarse por completo para pasar a la siguiente o en su

defecto requerir mínimas modificaciones para regresar a la fase anterior, se plantea una planificación de 60 días hábiles que se visualiza en la tabla 4.2.

Tabla 4.2: Tabla de planificación según fases del modelo Cascada.

Num.	Fase del modelo Cascada	Fecha inicial	Fecha final	Duración (días)
1.	Fase de análilis	6-jun	17-jun	10
2.	Fase de diseño del sistema	20-jun	8-jul	15
3.	Fase de implementación	11-jul	19-ago	30
4.	Fase de pruebas	22-ago	2-sep	10
5.	Fase de mantenimiento	5-sep	9-sep	5

Fuente: Elaboración propia.

En la figura 4.2 se visualiza e diagrama de Gannt en base a la planificación expresada en la anterior tabla.

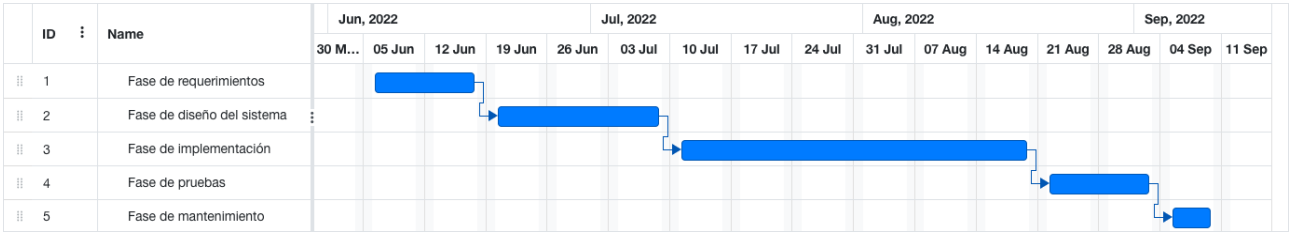

Figura 4.2: Diagrama de Gannt.
Fuente : Elaboración propia

4.3. Diseño

El diseño del sistema de video-vigilancia se divide y se desarrolla según los módulos que componen el sistema completo:

- Módulo de cámaras
- Módulo de servidor

4.3.1. Módulo de cámaras

Este módulo es el encargado de controlar y conectar la cámara al servidor TCP. Esta planificado para funcionar en un dispositivo portatil como ser una raspberry. Gracias a la

Diseño de interfaz

Diagrama de interacción

En la figura 4.3.

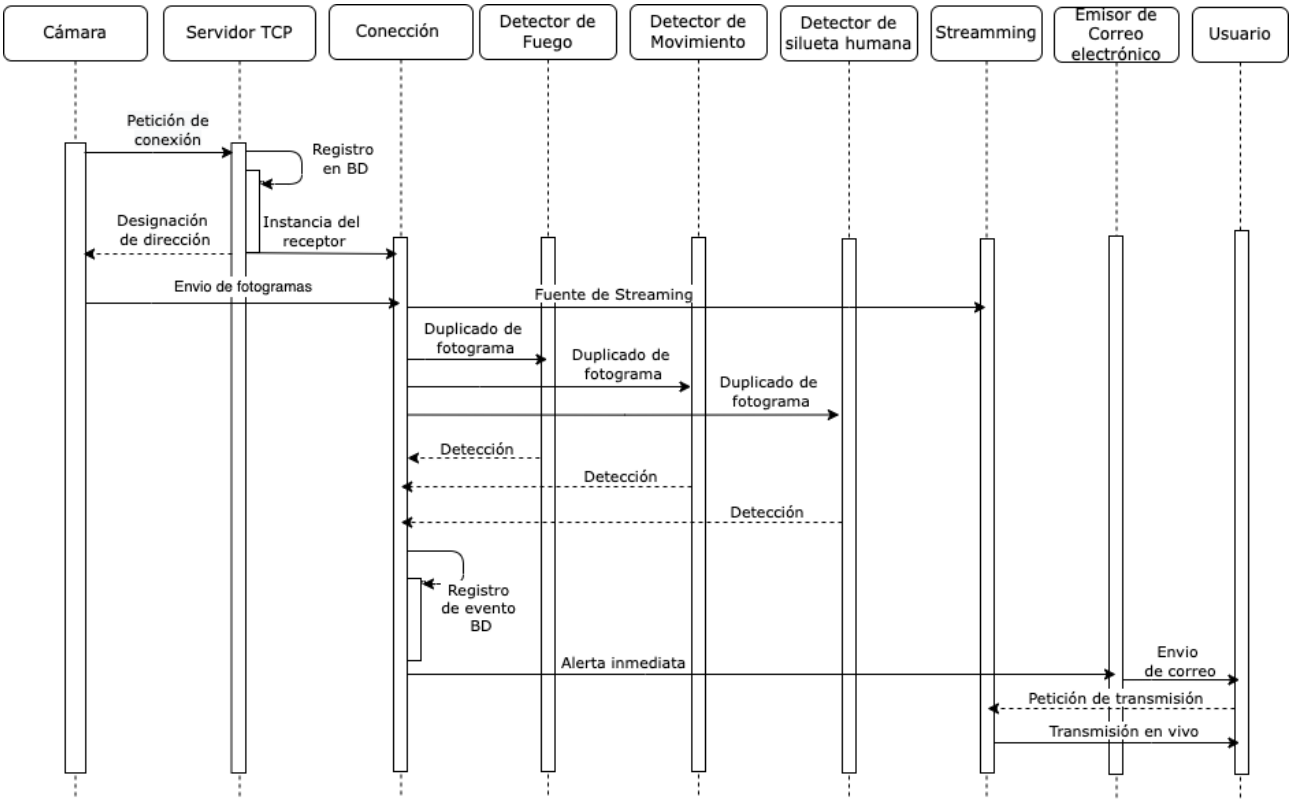

Figura 4.3: Diagrama de interacción entre los módulos del sistema.
Fuente: Elaboración propia.

Diagrama de estado

Diagrama de clases

En la figura 4.4

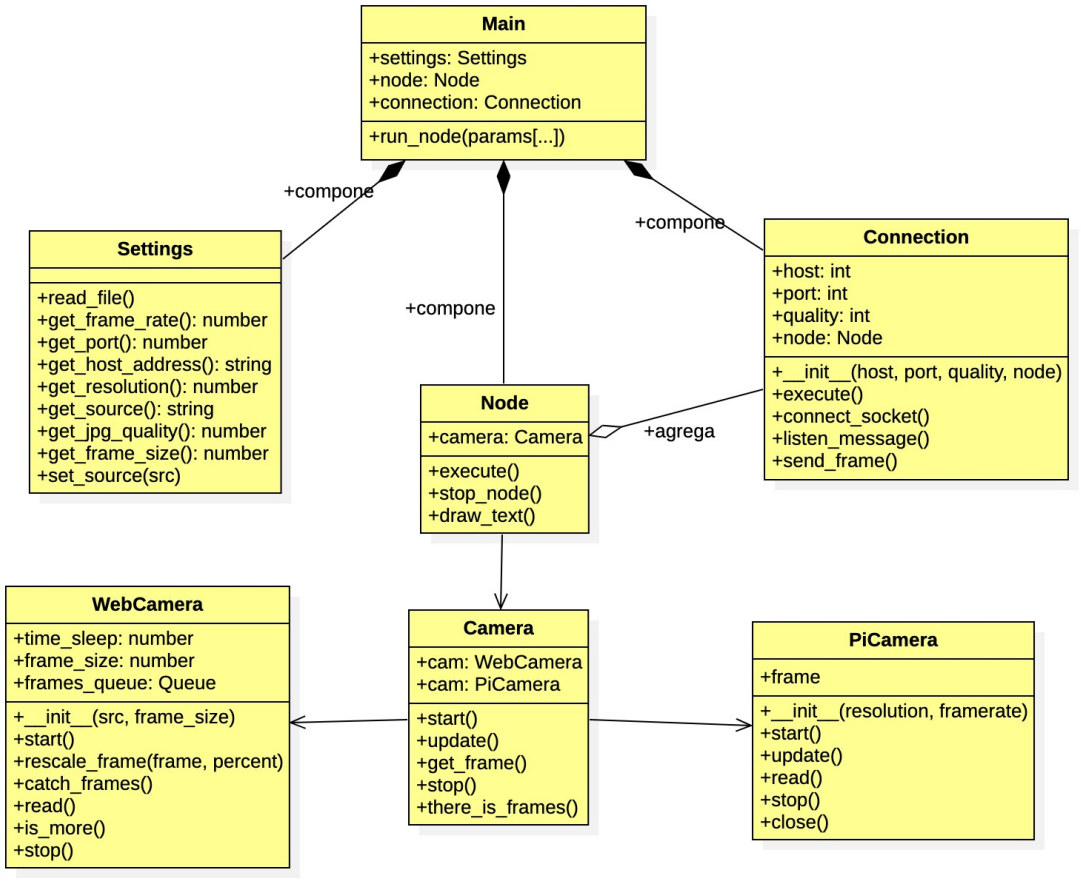

Figura 4.4: Diagrama de clases del módulo de cámara
Fuente : Elaboración propia

4.3.2. Módulo de servidor

Diagrama de interacción

Diagrama de estado

Diagrama de clases

Diagrama de la base de datos

En la figura 4.5.

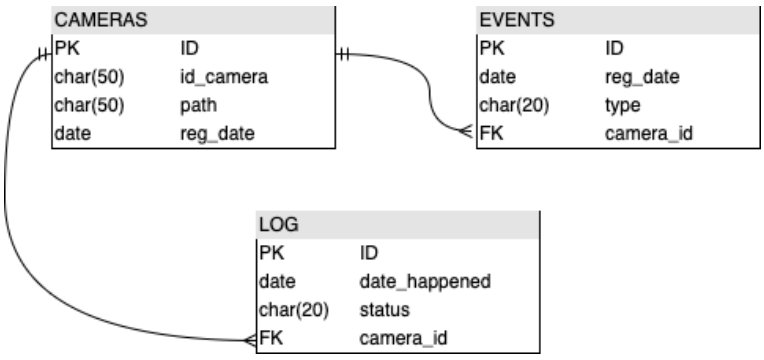

Figura 4.5: Diagrama entidad-relación.
Fuente : Elaboración propia

Diseño de notificación por correo electrónico

col1	col2	col3
Multiple row	cell2 cell5 cell8	cell3 cell6 cell9

Capítulo 5

Implementación

En este capitulo se describe a detalle el proceso de implementación de los diferentes módulos que forman parte del sistema de video-vigilancia inteligente.

5.1. Módulo de Cámara

5.1.1. Fuente de fotogramas

Webcam

Raspberry-picam

5.1.2. Configuración

5.1.3. Interfaz gráfica de usuario

Referenciando a la figura 5.1.

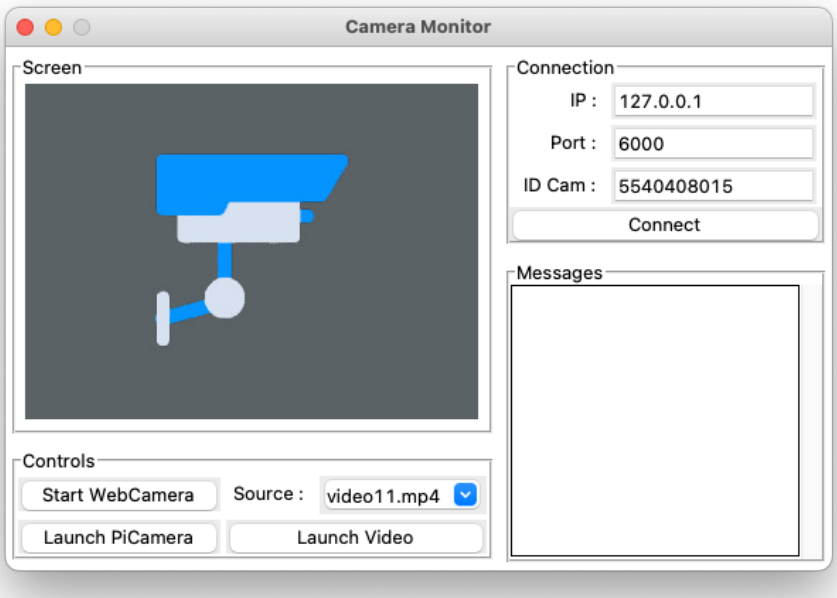

Figura 5.1: Interfaz de la camara desarrollada.
Fuente : Elaboración Propia.

Referenciando a la figura 5.1.

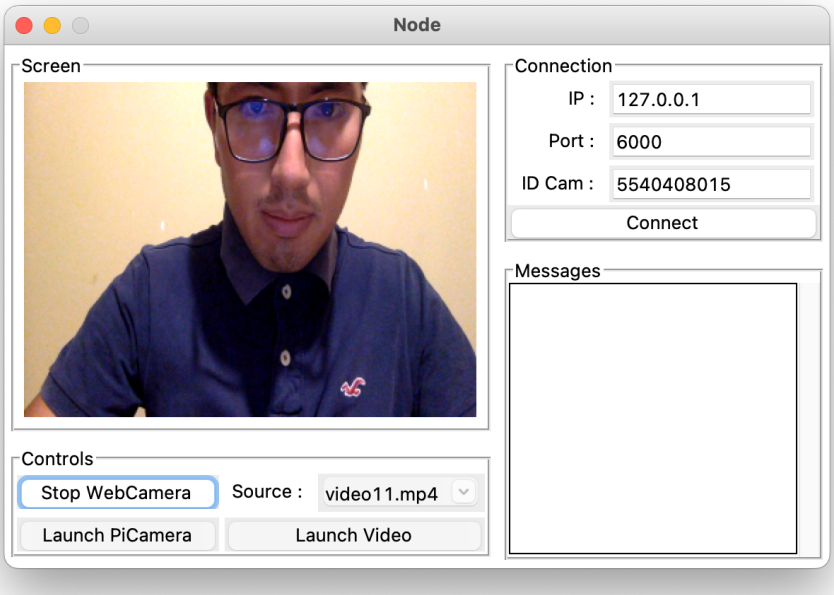

Figura 5.2: Diagrama de Gannt.
Fuente : Elaboración propia

Referenciando a la figura 5.3.

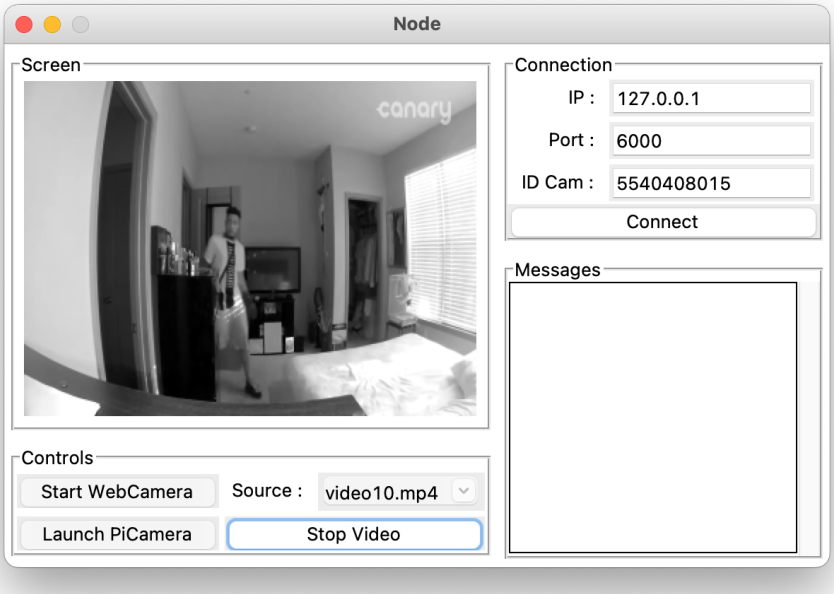

Figura 5.3: Diagrama de Gannt.
Fuente : Elaboración propia

5.1.4. Conexión al servidor

5.2. Módulo de Servidor

5.2.1. Interfaz de usuario

Referenciando a la figura 5.4.

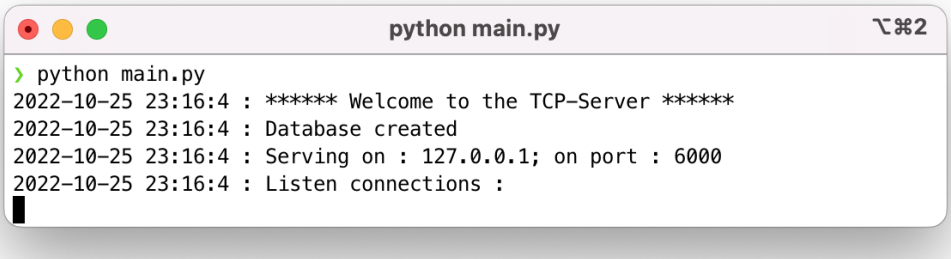

Figura 5.4: Ejecución del servidor TCP.
Fuente : Elaboración propia

5.2.2. Manejador de conexiones

5.2.3. Análisis de fotogramas

Fuego y humo

Silueta humana

Movimiento

5.2.4. Streamming Http

5.3. Notificación por correo electrónico

Capítulo 6

Pruebas

6.1. Pruebas de conexión al servidor

6.2. Pruebas de notificación

6.3. Prueba de transmisión en vivo

Capítulo 7

Conclusiones

Concluimos que...

Referencias

Austerberry, D. (2005). *The technology of video and audio streaming* (Vol. 1). ELSEVIER publisher.

Banda, H. (2017, Abril). Inteligencia artificial: Principios y aplicaciones. *Research Gate*, *50*. Recuperado de https://www.researchgate.net/publication/262487459_Inteligencia_Artificial_Principios_y_Aplicaciones

Bell, T. E., y Thayer, T. A. (1976). Software requirements: Are they really a problem? *IEEE Computer Society Press*, 61-68. Recuperado de https://static.aminer.org/pdf/PDF/000/361/405/software_requirements_are_they_really_a_problem.pdf

Bootcamp-ia. (2019a, Noviembre). *Comprensión de las redes neuronales convolucionales 1d y 3d.* Recuperado de https://programmerclick.com/article/89511747730/

Bootcamp-ia. (2019b, Noviembre). *Introducción a las redes neuronales convolucionales.* Recuperado de https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8

Briega, R. E. L. (2015, Septiembre). *Libro online iaar.* Recuperado de https://iaarbook.github.io/

CCNA-Certification. (2022, Enero). *Http and https explained.* Recuperado de https://study-ccna.com/http-https/

Chollet, F., y Allaire, J. J. (2018). *Deep learning with r.* Javvin Editiorals. Recuperado de https://bookdown.org/paul/computational_social_science/machine-learning-as-programming-paradigm.html#ref-Chollet2018-vf

Gama, B., Sant'Ana, W., Lambert-Torres, G., Salomon, C., Bonaldi, E., da Silva, L. E., ... Steiner, F. (2021, 02). Fpga prototyping using the stemlab board with application on frequency response analysis of electric machinery. *IEEE Access*, *PP*, 1-1. doi: 10.1109/ACCESS.2021.3058059

Ganis, M. (2010). Agile methods: Fact or fiction. *Research Gate*. Recuperado de https://tcf.pages.tcnj.edu/files/2013/12/ganis-tcf2010.pdf

Garza, J. (2013, Septiembre). *Breve historia tcp/osi.* Recuperado de https://www

.javiergarzas.com/2013/09/tcpip-se-impuso-a-osi-2.html/

Henry. (2022, Septiembre). *Python, ¿será el lenguaje más popular en 2023?* Recuperado de https://blog.soyhenry.com/tag/mundo-henry/

Howard. (2022, Julio). *Redes cliente-servidor vs. redes peer-to-peer.* Recuperado de https://community.fs.com/es/blog/client-server-vs-peer-to-peer-networks.html

Innovation, A. (2019, Octubre). *Qué son las redes neuronales y sus funciones.* Recuperado de https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/

Lopez, R. (2016, Junio). *Tensorflow y redes neuronales.* Recuperado de https://relopezbriega.github.io/blog/2016/06/05/tensorflow-y-redes-neuronales/

Lopez, R. (2017, Junio). *Introducción a la inteligencia artificial.* Recuperado de https://relopezbriega.github.io/tag/inteligencia-artificial.html

MarketsAndMarkets. (2020, Noviembre). *Video surveillance market with covid-19 impact analysis.* Recuperado de https://www.marketsandmarkets.com/Market-Reports/video-surveillance-market-645.html

Norman, T. L. (2017). Chapter 6 - electronics elements: A detailed discussion originally from integrated security systems design. thomas norman: Butterworth-heinemann, 2015. updated by the editor, elsevier, 2016. , 95-137. Recuperado de https://www.sciencedirect.com/science/article/pii/B9780128044629000063 doi: https://doi.org/10.1016/B978-0-12-804462-9.00006-3

Oracle. (2015, Septimbre). *The java™ tutorials.* Recuperado de https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html#:~:text=Definition%3A,address%20and%20a%20port%20number.

Ozer, J. (2017, Noviembre). *What is mpeg dash?* Recuperado de https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HLS-(HTTP-Live-Streaming)-78221.aspx?utm_source=related_articles&utm_medium=gutenberg&utm_campaign=editors_selection

Ozer, J. (2018, Diciembre). *What is hls (http live streaming)?* Recuperado de https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=79041

Royce, W. W. (1970). Managing the development of large software systems. *IEEE WESCON*, 328-338. Recuperado de https://www.praxisframework.org/files/royce1970.pdf

Saha, S. (2018, Diciembre). *A comprehensive guide to convolutional neural networks.* Recuperado de https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Solintel. (2018, Diciembre). *Sistemas de seguridad con cámaras ip.* Recuperado de http://www.solintelsa.net/sistemas-de-videovigilancia/

Tagliaferri, L. (2017, Septiembre). An introduction to machine learning. *Digital Ocean*. Recuperado de https://www.digitalocean.com/community/tutorials/an-introduction-to-machine-learning

Technologies Javvin, I. (2004). *Network protocols handbook*. Javvin Editiorals. Recuperado de https://bkarak.wizhut.com/www/lectures/networks-07/NetworkProtocolsHandbook.pdf

Tejada, A. G. (2019). Aplicaciones de "deep learning" en entorno ros. *Research Gate*. Recuperado de https://idus.us.es/bitstream/handle/11441/102311/TFG-2951-GOMEZ%20TEJADA.pdf?sequence=1&isAllowed=y

Vijayanagar, K. R. (2021, Abril). *What is mpeg-dash video streaming protocol? how does mpeg-dash work?* Recuperado de https://ottverse.com/mpeg-dash-video-streaming-the-complete-guide/

Wikipedia. (1994, Diciembre). *Modelo osi.* Recuperado de https://es.wikipedia.org/wiki/Modelo_OSI

Wikipedia. (2012, Julio). *Perceptor multicapa.* Recuperado de https://es.wikipedia.org/wiki/Perceptr%C3%B3n_multicapa

Wikipedia. (2020, Diciembre). *Videovigilancia ip.* Recuperado de https://es.wikipedia.org/wiki/Videovigilancia_IP

Wikipedia. (2022, Octubre). *Cliente-servidor.* Recuperado de https://es.wikipedia.org/wiki/Videovigilancia_IP

Anexos

Anexo A: Manual de instalacion de la camara

Contenido de Anexo A

Anexo B: Instalación del servidor

Contenido de Anexo B

Anexo C: Instalación de la aplicación

Contenido de Anexo C