

**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 compañía 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

<b>Dedicatoria</b>	<b>I</b>
<b>Agradecimientos</b>	<b>III</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	2
1.2. Descripción del Problema . . . . .	2
1.2.1. Definición del problema . . . . .	2
1.3. Objetivos del Proyecto . . . . .	2
1.3.1. Objetivo General . . . . .	2
1.3.2. Objetivos Específicos . . . . .	2
1.4. Justificación . . . . .	3
1.5. Alcances y límites . . . . .	3
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Sistema de video vigilancia . . . . .	5
2.2. Arquitectura de red . . . . .	7
2.2.1. Protocolos . . . . .	8
2.2.2. Modelo cliente-servidor . . . . .	10
2.2.3. HTTP . . . . .	12
2.3. Inteligencia Artificial . . . . .	13
2.3.1. Redes Neuronales . . . . .	14
2.4. Machine Learning (Aprendizaje de Máquina) . . . . .	16
2.4.1. Métodos de Machine Learning . . . . .	17
2.5. Deep Learning (Aprendizaje profundo) . . . . .	18
2.6. Técnicas de visión por computadora . . . . .	18
2.6.1. Aplicaciones . . . . .	18
2.6.2. Librerías . . . . .	19
2.7. Transmisión de video en vivo . . . . .	20
2.7.1. Protocolos . . . . .	20
2.8. Lenguaje de Programación . . . . .	21
2.9. Metodología de desarrollo . . . . .	21
2.9.1. Modelo Cascada (Waterfall) . . . . .	22
<b>3. Seguridad en el hogar</b>	<b>25</b>

3.1.	Introducción . . . . .	25
3.2.	Seguridad . . . . .	25
3.3.	Ausencia en el hogar . . . . .	25
3.3.1.	Ausencias cotidianas . . . . .	26
3.3.2.	Ausencias de termino medio . . . . .	26
3.3.3.	Ausencias prolongadas . . . . .	26
3.4.	Situaciones de riesgo . . . . .	27
3.4.1.	Presencia de intrusos . . . . .	27
3.4.2.	Fuego y humo . . . . .	27
3.5.	Sistemas de seguridad . . . . .	28
3.5.1.	Alarmas . . . . .	28
3.5.2.	Sensores . . . . .	29
3.5.3.	Cámaras . . . . .	29
<b>4.</b>	<b>Inicialización</b>	<b>31</b>
4.1.	Planificación General . . . . .	31
4.2.	Identificación de Requerimientos . . . . .	31
4.2.1.	Requerimientos del sistema . . . . .	31
4.2.2.	Requerimientos del software . . . . .	34
4.3.	Análisis . . . . .	34
4.4.	Diseño de Módulos . . . . .	34
4.5.	Identificación de Subsistemas . . . . .	35
4.6.	Comunicación de Sistemas . . . . .	37
4.6.1.	Sockets . . . . .	37
4.7.	Planificación . . . . .	37
<b>5.</b>	<b>Implementación</b>	<b>39</b>
5.1.	Módulo de Cámara . . . . .	39
5.1.1.	Diseño de clases . . . . .	39
5.1.2.	Diseño de interfaz de simulador . . . . .	39
5.1.3.	Captura de video . . . . .	39
5.1.4.	Conexión a Servidor TCP . . . . .	39
5.1.5.	Comunicacion y envío de fotogramas . . . . .	39
5.2.	Módulo de Servidor TCP . . . . .	39
5.2.1.	Diseño de clases . . . . .	39
5.2.2.	Manejador de conectores de clientes . . . . .	39
5.2.3.	Streaming HLS . . . . .	39
5.2.4.	Detectores . . . . .	39
5.3.	Servidor HTTP . . . . .	39
5.4.	Modulo de envío de correo electrónico . . . . .	39
<b>6.</b>	<b>Pruebas</b>	<b>43</b>
6.1.	Pruebas de integracion . . . . .	43
6.2.	Prueba de transmision . . . . .	43



6.3. Prueba de transmision en vivo . . . . .	43
<b>7. Conclusiones</b>	<b>45</b>
<b>Referencias</b>	<b>47</b>
<b>Anexos</b>	<b>50</b>
<b>Anexo A: Manual de instalacion de la camara</b>	<b>53</b>
<b>Anexo B: Instalación del servidor</b>	<b>55</b>
<b>Anexo C: Instalación de la aplicación</b>	<b>57</b>



# Índice de figuras

2.1. Componentes de una sistema de video-vigilancia. . . . .	5
2.2. Proyección del mercado de la video-vigilancia. . . . .	6
2.3. Esquema de capas del modelo OSI. . . . .	7
2.4. Modelo TCP/IP frente al modelo OSI. . . . .	8
2.5. Ilustración del Modelo cliente-servidor . . . . .	10
2.6. Solicitud de conexión por medio de un socket . . . . .	11
2.7. Conexión establecida entre sockets . . . . .	11
2.8. Flujo de interacción entre servidor y cliente TCP. . . . .	11
2.9. Interacción del protocolo HTTP. . . . .	12
2.10. Diagrama de la Inteligencia Artificial. . . . .	13
2.11. Ilustración de una Neurona artificial . . . . .	14
2.12. Modelo de capas de una red neuronal. . . . .	14
2.13. Ilustración de una Red Neuronal Convolutacional. . . . .	15
2.14. Movimiento del Kernel. . . . .	16
2.15. Diferencias entre programación clásica y M.L. . . . .	17
2.16. Logotipo de la librería Fuente: Web . . . . .	19
2.17. HLS. . . . .	20
2.18. Dash. . . . .	21
2.19. Modelo Cascada . . . . .	22
2.20. Modelo Cascada: Relación iterativa entre las fases sucesivas. . . . .	23
3.1. Ilustración ejemplo de un intruso. . . . .	26
3.2. Ilustración ejemplo de un ladrón . . . . .	27
3.3. Ilustración ejemplo de un fuego en interiores. . . . .	27
3.4. Ilustración de un ladrón . . . . .	28
3.5. Ilustración de un ladrón . . . . .	29
3.6. Ilustración de un ladrón . . . . .	29
3.7. Variedad de cámaras de vigilancia. . . . .	29
3.8. Cámara de vigilancia de interiores. . . . .	30
4.1. Diagrama de Gannt. . . . .	31
4.2. Diagrama de Gannt. . . . .	32
4.3. Diagrama de Gannt. . . . .	32
4.4. Diagrama de Gannt. . . . .	33
4.5. Ejecución del servidor TCP. . . . .	33

4.6. Ilustración de un ladrón . . . . .	36
4.7. Ilustración de un ladrón . . . . .	36

# Índice de tablas

4.1. Tabla de planificación de las diferentes fases del modelo Cascada . . . . .	31
4.2. Detalle de las pruebas realizadas . . . . .	34
4.3. Detalle de las pruebas realizadas . . . . .	34
4.4. Detalle de las pruebas realizadas . . . . .	35
4.5. Detalle de las pruebas realizadas . . . . .	35
5.1. Título de tabla multipágina . . . . .	40



# Capítulo 1

## Introducción

Seguridad, es un término usado para referir a la ausencia de riesgo o a la confianza explícita 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 llega a ser el acto de observar una escena o escenas en busca de comportamientos específicos que podrían ser anormales o podrían indicar una posible emergencia o existencia de un comportamiento impropio (Norman, 2017). Los sistemas de video-vigilancia de la actualidad se han convertido en una herramienta esencial de la seguridad para mantener “observado” un espacio muy importante para el que requiere el sistema; donde el mismo está compuesto por un conjunto de cámaras, monitores y grabadoras donde estos elementos 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 cada vez más robustos 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 redes neuronales tiene la capacidad de identificar objetos, siluetas y/o elementos dentro de una escena captada por una cámara.

Estas nuevas capacidades pueden ser explotadas en un sin fin de actividades diarias donde es necesaria la supervisión de una persona, permitiendo aún más una automatización inminente. 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 continuo 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 elementos 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 estará 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 comunicarle 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 no es imposible de eliminar, es posible crear mecanismos que contrarresten el impacto que ocasionan dichos sucesos en los sitios que se quiere 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 captan, pero es necesaria 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 adaptativo 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 audio y 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). Entonces 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, mantienen una alerta activa y poseen 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 conexión 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 nego-

cios, como la construcción de áreas residenciales y 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 alcanzadas por 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 tasa 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)

Lo más relevante del crecimiento de estas oportunidades, es la implementación de nuevas características en este tipo de sistemas; gracias a la implementación de técnicas de visión por computadora e inteligencia artificial (I.A.), como a la escalabilidad lograda por el uso de servicios basados en la nube. Las ramas de la inteligencia artificial como el Machine Learning (Aprendizaje Automático) y el Deep Learning (Aprendizaje profundo) permiten lograr estas características.

Para el desarrollo del prototipo propuesto se implementan los siguientes componentes involucrados en el sistema:

- Cámaras (A ser denominados "nodos")
- Servidor TCP (Servicio que emplea el protocolo TCP/IP)
- Servidor HTTP (Servicio que emplea el protocolo de la Web)
- Módulo SMTP (Módulo de envío de correo electrónico)

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 y 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).

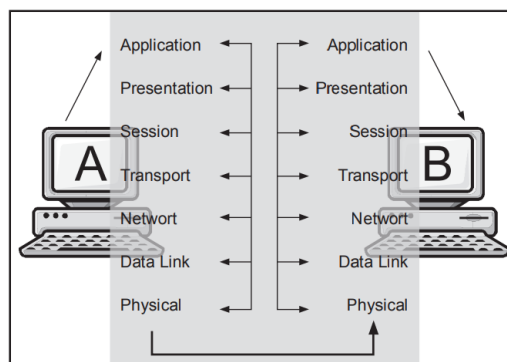


Figura 2.3: Esquema de capas del modelo OSI.

Fuente: (Technologies Javvin, 2004, 3)

Su contraparte en las arquitecturas de red del modelo OSI, es 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.

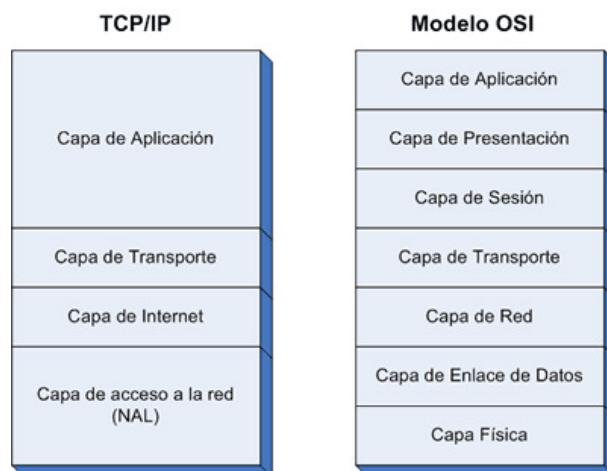


Figura 2.4: Modelo TCP/IP frente al modelo OSI.

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 capas. 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 aplicación a la capa de red física.

Cuando son enviados los datos, cada capa trata toda la información que recibe de la capa superior como datos, 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 **j**cita. 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. La arquitectura de red proporciona solo un esquema conceptual para la comunicación. El modelo no proporciona métodos específicos de comunicación, sino mas bien, 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:** Estos protocolos se utilizan para automatizar diferentes procesos tanto en entornos comerciales como personales, como en edificios inteligentes, tecnología en la nube o vehículos autónomos.
- **Mensajería Instantánea:** La comunicación basada en texto, en teléfonos inteligentes y computadoras suceden debido a una serie de diferentes protocolos de mensajería instantánea.
- **Enrutamiento:** Protocolos de enrutamiento permiten la comunicación entre routers y otros dispositivos de red.
- **Transferencia de archivos:** Envío de archivos por medio de un canal de comunicacion.
- **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 detalla 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 mensajería 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 [cita 1]. 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 una red.

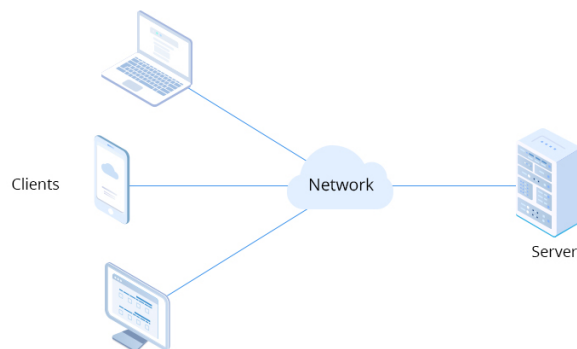


Figura 2.5: Ilustración del Modelo cliente-servidor

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.



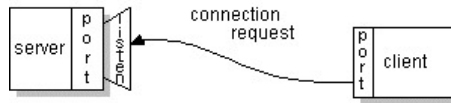


Figura 2.6: Solicitud de conexión por medio de un socket  
Fuente: (Oracle, 2015)

Si todo va bien, el servidor acepta la conexión. Tras la aceptación, el servidor obtiene un nuevo socket 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 o socket después de la aceptación a la petición de conexión del cliente; ya que después establecida la conexión, el servidor necesita un nuevo socket para continuar escuchando las nuevas solicitudes de conexión mientras atiende las necesidades del cliente conectado.



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

El flujo de interacción es como sigue: el servidor TCP establece un conector o socket en una determinada dirección y puerto, y se mantiene escuchando constantemente si un conector cliente (que conoce la dirección y puerto del servidor) solicita una conexión, una vez que el servidor acepta la conexión, se brinda un nuevo puerto diferente de manera automática para comunicarse exclusivamente con ese cliente y mantiene el puerto original a la escucha de nuevos clientes, como se ve en la figura 2.8. Una vez que esta conexión es establecida, tanto como el cliente y el servidor pueden enviar y recibir datagramas. Cuando el cliente decide cerrar la conexión, el servidor pone de nuevo el puerto utilizado anteriormente como disponible.

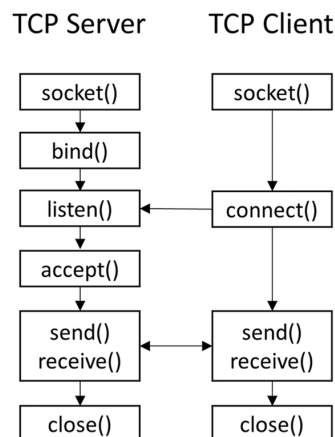


Figura 2.8: Flujo de interacción entre servidor y cliente TCP.  
Fuente: (Gama et al., 2021)

A continuación se detallan los métodos que se muestran en el gráfico anterior:

- `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()/receive()`: 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. El cliente envía una solicitud y el servidor devuelve una respuesta. Este intercambio de mensajes es un ejemplo de comunicación entre procesos. 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 protocolo más utilizado en Internet. Es usado en cada transacción de la Web (www) y permite la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el 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. Es un protocolo orientado a transacciones y sigue el esquema “petición - respuesta” entre un cliente y un servidor. En la figura 2.9 se aprecia la interacción entre un cliente y un servidor web.

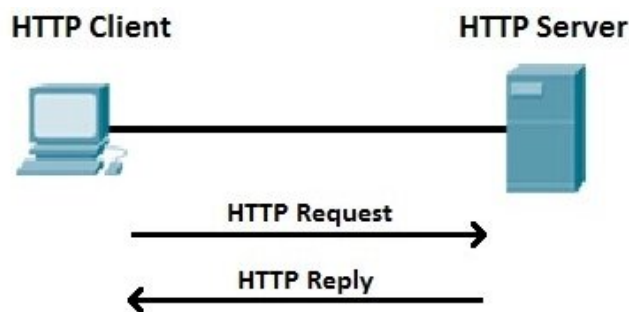


Figura 2.9: Interacción del protocolo HTTP.

Fuente: (CCNA-Certification, 2022)

El protocolo tiene las siguientes partes:

- El cliente que efectúa la petición (un navegador o un spider) es denominado “user agent” (agente del usuario).

- La información transmitida es llamada “recurso” y es identificada mediante una cadena de caracteres denominada 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 las ciencias de la computación, en los últimos tiempos dejó de estar reservada para la investigación y ha formado parte del desarrollo de la sociedad.

El cerebro es el órgano más increíble 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 a este ser, un ser inteligente ().

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.

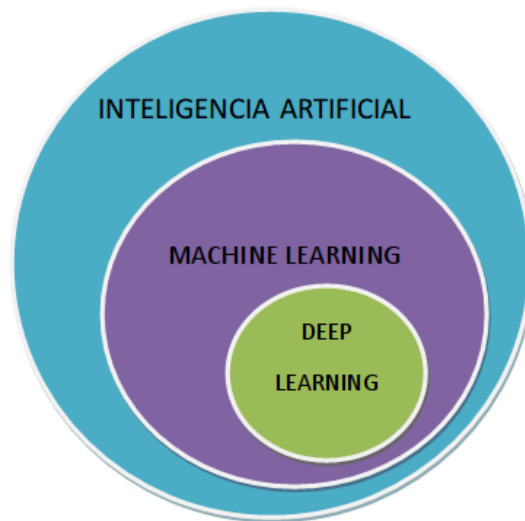


Figura 2.10: Diagrama de la Inteligencia Artificial.

Fuente: (Tejada, 2019)

A menudo los términos Inteligencia Artificial, Aprendizaje Automático (Machine Learning) y Aprendizaje Profundo (Deep Learning) son usados de manera indistinta, pero se debe tener en cuenta su significado diferente. 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 a sus subcampos de estudio como ser 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.

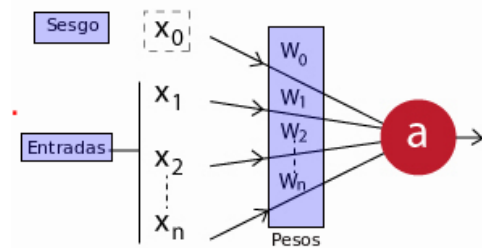


Figura 2.11: Ilustración de una Neurona artificial  
Fuente: (Lopez, 2016)

El objetivo principal de un modelo neuronal, es aprender modificándose automáticamente a si mismo, llegando a realizar tareas complejas que no podrían ser realizadas mediante la clásica programación basada en reglas. De esta forma se pueden automatizar funciones que al principio solo podrían ser realizadas por personas. Con su semejanza al del cerebro humano, las redes reciben una serie de valores de entrada y cada una de estas entradas llega a un nodo llamado neurona.

Las neuronas de la red están a su vez agrupadas en capas que forman la red neuronal. Cada una de las neuronas de la red posee a su vez un peso, un valor numérico, con el que modifica la entrada recibida. Los nuevos valores obtenidos salen de las neuronas y continúan su camino por la red. Este funcionamiento puede observarse de forma esquemática en la figura 2.12.

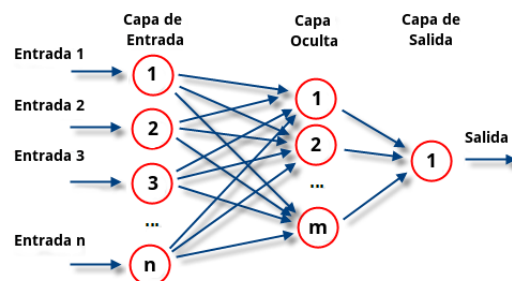


Figura 2.12: Modelo de capas de una red neuronal.  
Fuente: (Wikipedia, 2012)

Una vez que se alcanza el final de la red se obtiene una salida que será la predicción calculada

por la red. Cuantas más capas posea la red y más compleja sea, también serán mas complejas las funciones que pueda realizar. Para conseguir que una red neuronal realice las funciones deseadas, es necesario **entrenarla**.

El entrenamiento de una red neuronal se realiza modificando los pesos de sus neuronas para que consiga extraer los resultados deseados. Para ello lo que se hace es introducir datos de entrenamiento en la red, en función del resultado que se obtenga, se modifican los pesos de las neuronas según el error obtenido y 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 que han sido utilizados durante 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 esta evolucionando con el tiempo, ofreciendo cada vez soluciones más complejas y eficientes.

### Redes neuronales convolucionales

Dentro del conjunto de tipos de redes neuronales tenemos las convolucionales, que específicamente servirán en el campo de la visión artificial. Las redes neuronales convolucionales son un algoritmo de Aprendizaje Profundo (Deep Learning) que está diseñado para trabajar con imágenes, tomando estas como entrada, asignándoles importancias (pesos) a ciertos elementos en la imagen para así poder diferenciar unos de otros.

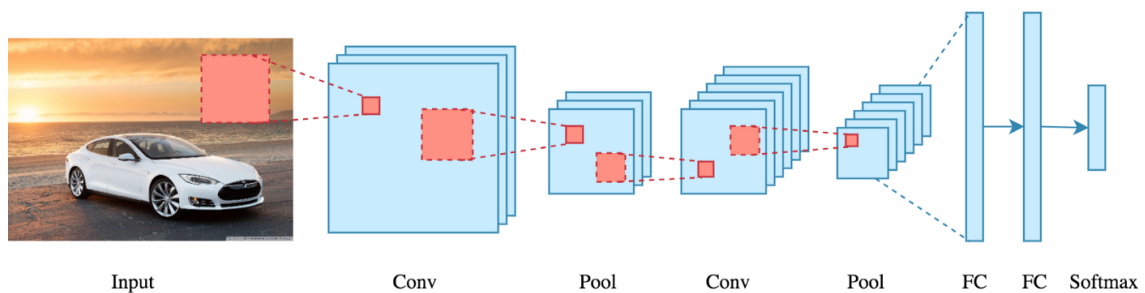


Figura 2.13: Ilustración de una Red Neuronal Convolucional.

Fuente: (Bootcamp-ia, 2019b)

Este es uno de los principales algoritmos que ha contribuido en el desarrollo y perfeccionamiento del campo de visión por computadora. Las redes convolucionales contienen varias capas ocultas como se ilustra en la figura 2.12, donde las primeras pueden detectar líneas, curvas y así se van especializando hasta poder reconocer formas complejas como un rostro, siluetas, etc. (Saha, 2018).

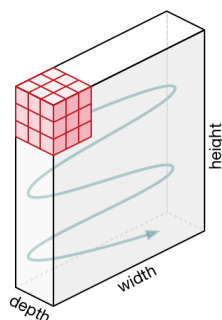


Figura 2.14: Movimiento del Kernel.

Fuente: (Bootcamp-ia, 2019a)

El proceso que se distingue de estas redes son las convoluciones. El cual 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 en las redes convolucionales se considera como un filtro que se aplica para extraer ciertas características importantes o patrones de esta y es usado para detectar bordes, enfoque, desenfoques, entre otras características de la imagen y es logrado para la convolución entre la imagen y el kernel.

Este proceso se desarrolla entre la imagen y el kernel, con la finalidad de que este filtro o kernel recorra toda la imagen (pixel por pixel). Por lo general, el kernel es de menor tamaño que la imagen. La convolución permite 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 .

Es una operación que se usa en las redes convolucionales. El padding se aplica agregando píxeles de valor cero alrededor de la imagen original. Tiene dos usos: El primero es para que al realizar la convolución la imagen resultante sea de igual tamaño que la imagen original. El segundo es cuando se tiene información relevante en las esquinas de la imagen por lo que al realizar convolución el filtro pasa más por el centro de la imagen que en las esquinas, por lo que se aplica el padding para tener la información más relevante cerca del centro.

Las tareas comunes de este tipo de redes son: detección o categorización de objetos, clasificación de escenas y clasificación de imágenes en general. La red toma como entrada los píxeles de una imagen.

## 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.

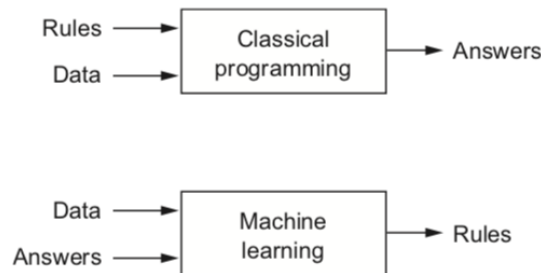


Figura 2.15: Diferencias entre programación clásica y M.L.

Fuente: (Chollet y Allaire, 2018)

En la figura 2.15 se aprecia 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 de esta manera desarrollar reglas que generan respuestas para nuevos datos.

### 2.4.1. Métodos de Machine Learning

En el Machine Learning, las tareas son generalmente clasificadas en grandes categorías, 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, que entrena un algoritmo basado en un ejemplo de entrada y salida el cual esta categorizado por un humano, y el aprendizaje no supervisado, que proporciona el algoritmo sin ningún dato categorizado permitiendo encontrar una estructura dentro de los datos de entrada.

#### Aprendizaje Supervisado

La computadora esta provista con entradas de 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.

¡Posible imagen!

En la ilustración.....

#### 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.

¡Mas conceptos!ref bibliografica!

## 2.5. Deep Learning (Aprendizaje profundo)

Según la figura 2.10, 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, ya que para saber que es lo que hay en el mundo nuestros cerebros deben ser capaces de representar esta información en toda su abundancia de color, forma, movimiento, detalle y belleza. (Briega, 2015)

Por lo tanto, 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, con el objetivo de ser tratadas por 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 decisiones en sus tareas asignadas. La visión por computadora trata de imitar a la visión humana, usando geometría y un enfoque estadístico para tratar 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. Librerías

Una de las librerías más utilizadas para las técnicas de visión por computadora es OpenCV. Es una biblioteca de uso libre para el desarrollo de aplicaciones usando visión artificial desarrollada por Intel. Esta librería reúne diversas características 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 librería, 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 librería  
Fuente: Web

Una de las características más interesantes de OpenCV es el reconocimiento facial. OpenCV, en su extensa biblioteca de funciones, brinda las capacidades para realizar las tareas de preprocesamiento sin ningún problema, así como los algoritmos de predicción. Además de usar el algoritmo de detección de objetos, es posible usar el seguimiento de objetos, para identificar rostros en una transmisión de video. OpenCV incluso posee funciones para configurar fácilmente el modelo en una transmisión en vivo, como en un video pregrabado (TheResearchNest, 2020).

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

## 2.7. Transmision de video en vivo

¡figuras¿y conceptos

### 2.7.1. Protocolos

¡figuras¿y conceptos En la figura, 2.17

#### HLS

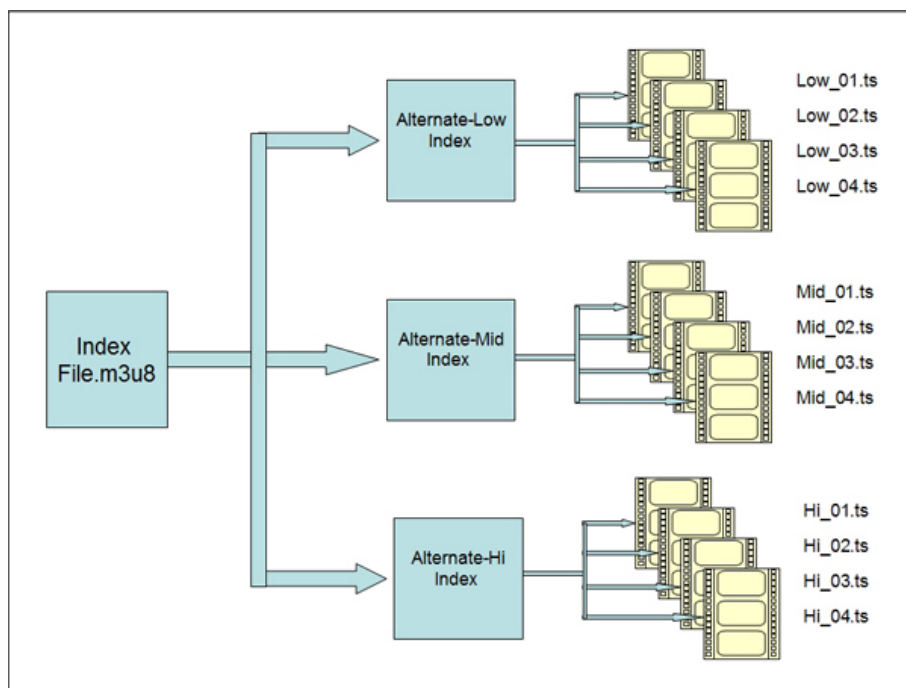


Figura 2.17: HLS.  
Fuente: (Ozer, 2017)

#### DASH

En la figura 2.18

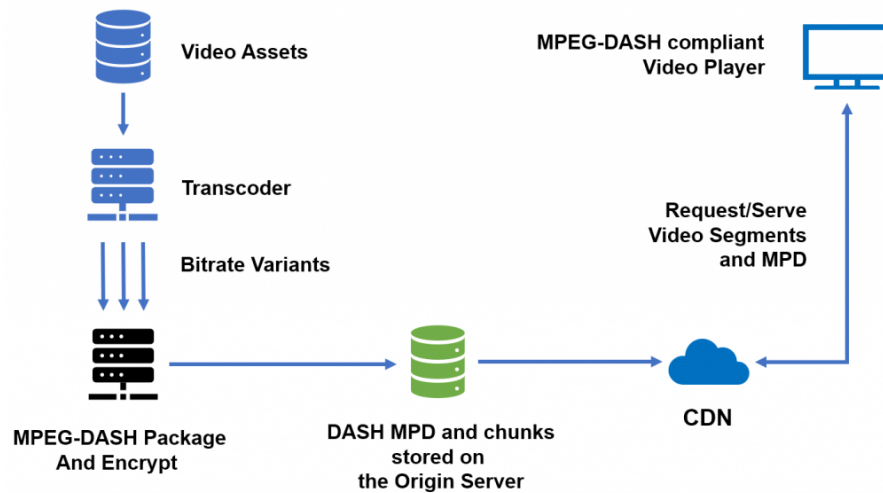


Figura 2.18: Dash.

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.

¡logo!

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).

¡imagen de popularidad! Motivo Es usado en Machine Learning y Deep Learning es el lenguaje líder en la inteligencia artificial

## 2.9. Metodología de desarrollo

El término de ingeniería de software se toma propone por primera vez en el conjunto de conferencias históricas organizadas por el comité de ciencia de la OTAN.<sup>1</sup> en los años 60. Para ese tiempo la ingeniería de software tampoco era conocida ni aceptada donde en ese entonces los proyectos de software complejos eran problemáticos y costosos de completar donde se supuso que sería beneficioso considerar el desarrollo de software como ingeniería (Ganis, 2010).

Los encargados de la codificación del software denominados programadores, en un principio eran ingenieros y 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

<sup>1</sup>Organización del Tratado del Atlántico Norte.

a los equipos de proyectos creen planes lentos y metódicos para la creación de sistemas de software.

### 2.9.1. Modelo Cascada (Waterfall)

En el inicio de su definición como un modelo para el desarrollo de software; 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 basandose en experiencias de desarrollo de software para la planificación de misiones aéreas; el autor describe los fundamentos del desarrollo de software. Gran parte de esos fundamentos aun 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 2.19.

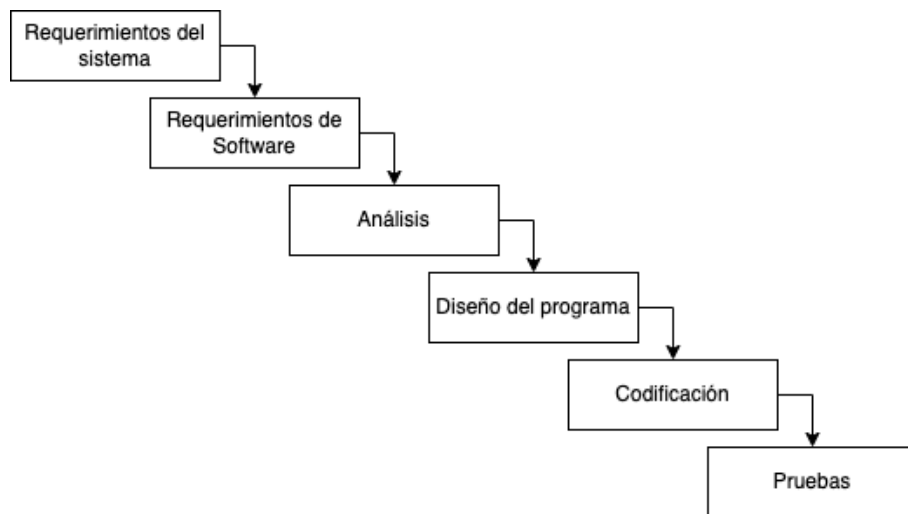


Figura 2.19: Modelo Cascada  
Fuente: Elaboracion Propia

Una vez definida esta secuencia, se creó 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.<sup>2</sup> (Bell y Thayer, 1976). En la figura 2.19 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 dependencia secuencial del producto entregado 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 relacion iterativa con el siguiente paso definido en la metodología que asegura la completitud del producto entregado en la fase anterior. Esta relación esta ilustrada en la figura 2.20. 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

---

<sup>2</sup>Metodología de Desarrollo de Software

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.

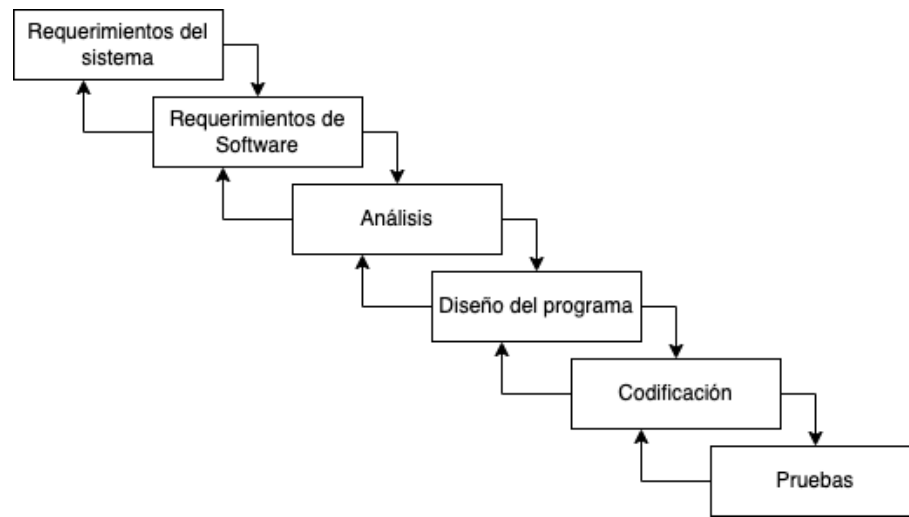


Figura 2.20: Modelo Cascada: Relación iterativa entre las fases sucesivas.

Fuente: Elaboracion Propia

Existen diferentes versiones de las fases del modelo en cascada y según la versión o enfoque, la cantidad de fases puede variar. Sin embargo las principales son las siguientes:

1. Requerimientos del sistema
2. Requerimientos de Software
3. Análisis
4. Diseño del programa
5. Codificación
6. Pruebas
7. Mantenimiento

A continuacion se detalla cada una de las fases y las actividades que implica cada fase:  
¡detalle de cada fase!!!1



# Capítulo 3

## Seguridad en el hogar

### 3.1. Introducción

La seguridad en el hogar es un asunto especialmente sensible y delicado de tratar, principalmente por ser el más sensible de todos los espacios vitales del ser humano. Sea del tipo que sea, en un hogar se fraguan los vínculos más íntimos y personales, contiene a quien más amamos y lo que más deseamos proteger.

La presencia de una persona en el hogar es un factor de seguridad de gran importancia; la mayoría de percances como intrusiones, allanamientos y robos se producen durante su ausencia. Los motivos para dejar un hogar vacío son distintos y variados: desde ausencias prolongadas a salidas más o menos puntuales, o regulares y diarias. Una vivienda vacía es más vulnerable que otra ocupada y esto se debe tomar en cuenta en el diseño de un sistema de seguridad para el hogar que ofrezca las máximas garantías en sus capacidades.

### 3.2. Seguridad

### 3.3. Ausencia en el hogar

Evidentemente, 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, escapadas...).

Hemos tratado de sintetizar estas posibilidades en función del tiempo de ausencia, estableciendo distintos casos con peculiaridades específicas en lo que se refiere a la seguridad y los riesgos que se afrontan, y empezando por exponer las medidas de protección más básicas y elementales que siempre se deberían tomar en consideración, tales como contratar un seguro para la vivienda (especificando algunos elementos importantes que figuran en toda póliza de esta índole para facilitar su elección y contratación).

Referenciando a la figura 4.7.



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

### 3.3.1. Ausencias cotidianas

El primero de los casos supuestos es el más frecuente y cotidiano: las ausencias diarias de horas o minutos que brindan oportunidades a asaltantes atentos. Aquí, se tendrán 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.

Como situación perteneciente a este grupo de supuestos, pero con riesgos añadidos y particularidades propias que obligan a prestarle una atención especial, se tratará aparte el caso de ausencias puntuales dejando en la vivienda a niños, personas mayores o dependientes sin nadie a su cargo. Evidentemente, aquí se tratarán amenazas y riesgos internos de la vivienda, tales como manipulaciones indebidas de instalaciones y componentes de especial peligrosidad, o la atención a emergencias que puedan suceder durante ausencias breves.

### 3.3.2. Ausencias de termino medio

Al salir de casa debemos saber si volveremos al cabo de pocas horas, de unos días o de semanas, ya que cada caso (como hemos comentado) presenta peculiaridades y riesgos específicos que tenemos que afrontar de distintos modos. El segundo supuesto, tras las ausencias cotidianas, será el caso 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.

### 3.3.3. 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).

## **3.4. Situaciones de riesgo**

### **3.4.1. Presencia de intrusos**

Referenciando a la figura 4.7.



Figura 3.2: Ilustración ejemplo de un ladrón  
Fuente: Web

### **3.4.2. Fuego y humo**

Referenciando a la figura 4.7.



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

Referenciando a la figura 4.7.



Figura 3.4: Ilustración de un ladrón

Fuente: Adaptada de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.

### 3.5. Sistemas de seguridad

En el mercado, existen un sinnúmero de posibilidades al alcance para proteger nuestros hogares frente a casi cualquier tipo de amenaza, tanto interna como externa. Los más eficaces y eficientes, sin duda, son los sistemas electrónicos de seguridad, de los que ya hablamos detalladamente en la guía Hogares y negocios seguros

No obstante, sea cual sea la opción elegida a la hora de proteger nuestra vivienda durante ausencias más o menos prolongadas, debemos 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 (avisadores de fugas y escapes) y de emergencia son, para estos casos, 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.

#### 3.5.1. Alarmas

Referenciando a la figura 4.7.



Figura 3.5: Ilustración de un ladrón

Fuente: Adaptada de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.

### 3.5.2. Sensores

Referenciando a la figura 4.7.



Figura 3.6: Ilustración de un ladrón

Fuente: Adaptada de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.

### 3.5.3. Cámaras

Referenciando a la figura 3.7.



Figura 3.7: Variedad de cámaras de vigilancia.

Fuente: Web.

Referenciando a la figura 3.8.

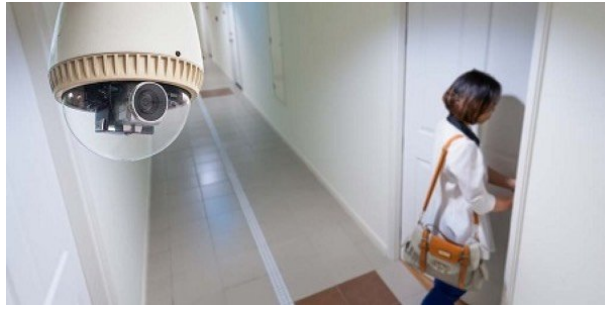


Figura 3.8: Cámara de vigilancia de interiores.

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.

**Sistemas de videovigilancia inteligente** La técnica clave del reconocimiento de la acción humana basada en la visión por computadora consiste en describir y comprender los comportamientos humanos por medio de la visión por computadora.

Este proceso es una tarea complicada e integra algunos campos de investigación que incluyen el procesamiento de imagen, aprendizaje automático, reconocimiento de patrones, etc.

La detección de un objeto móvil consiste en separar las áreas de cambio en el video es decir en las imágenes de fondo que comprenden el video, dicho de otra manera, separar correctamente las áreas y contornos del objeto móvil. Es crítico para el siguiente procesamiento la segmentación efectiva

# Capítulo 4

## Inicialización

### 4.1. Planificación General

El presente proyecto esta planteado para ser desarrollado en 20 dias

### 4.2. Identificación de Requerimientos

#### 4.2.1. Requerimientos del sistema

Tabla 4.1: Tabla de planificación de las diferentes fases del modelo Cascada

Num.	Fase	Fecha inicial	Fecha final	Duración (días)
1.	Fase de requerimientos	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.

Referenciando a la figura 4.1.

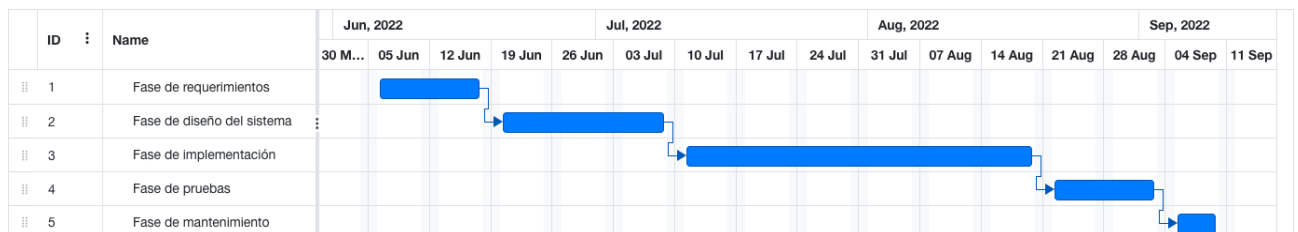


Figura 4.1: Diagrama de Gannt.

Fuente : Elaboración propia

Referenciando a la figura 4.2.

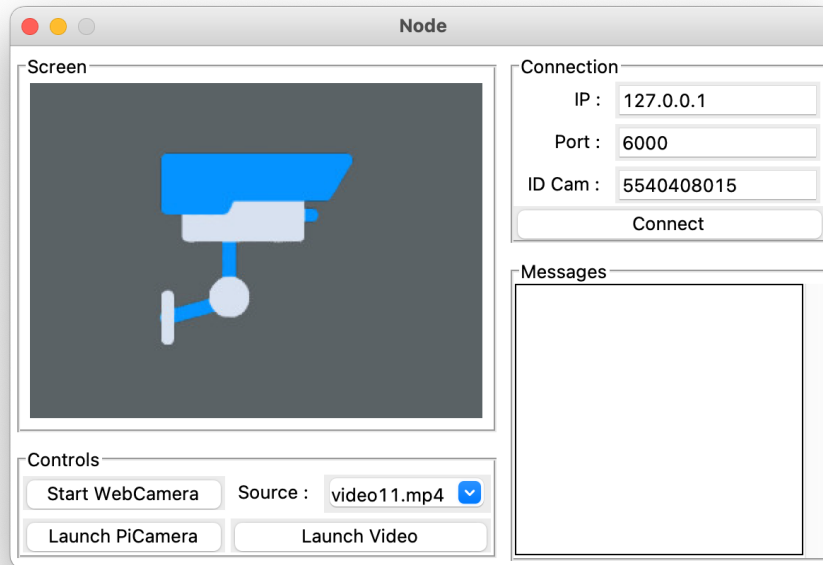


Figura 4.2: Diagrama de Gannt.  
Fuente : Elaboración propia

Referenciando a la figura 4.2.

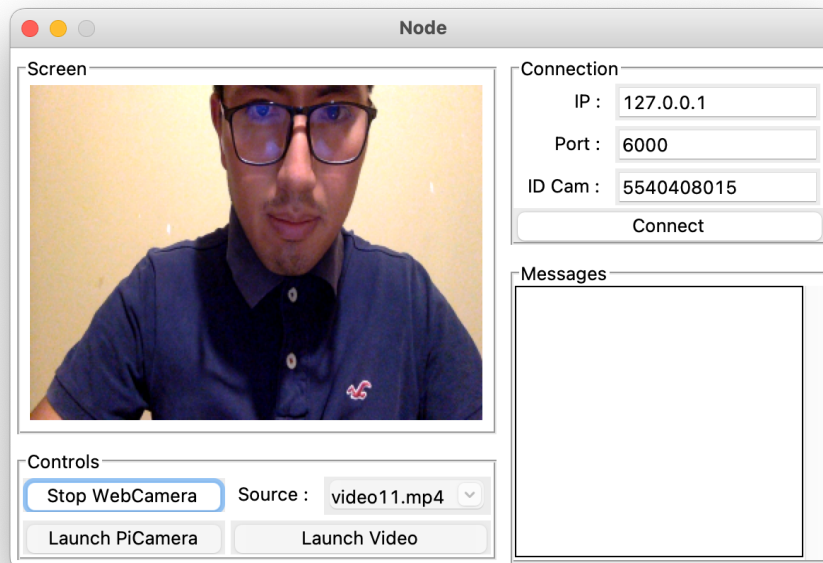


Figura 4.3: Diagrama de Gannt.  
Fuente : Elaboración propia

Referenciando a la figura 4.4.

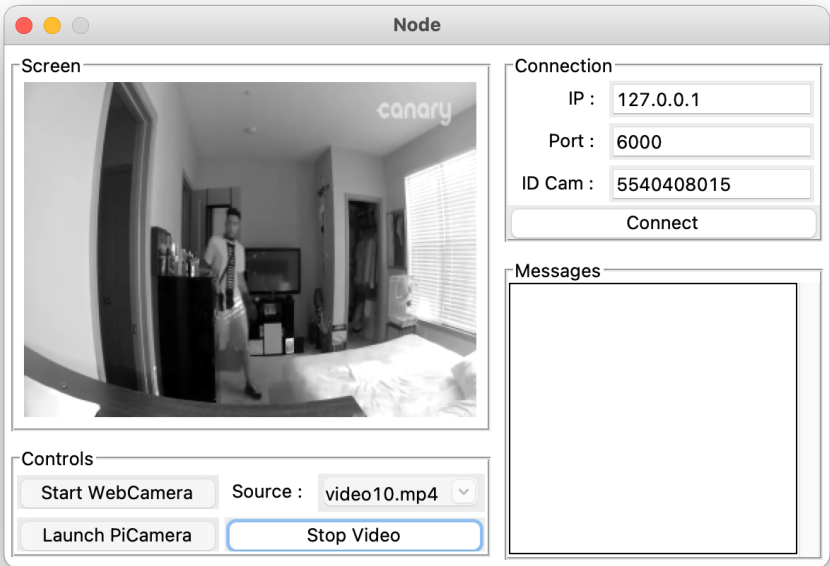


Figura 4.4: Diagrama de Gannt.  
Fuente : Elaboración propia

Referenciando a la figura 4.5.

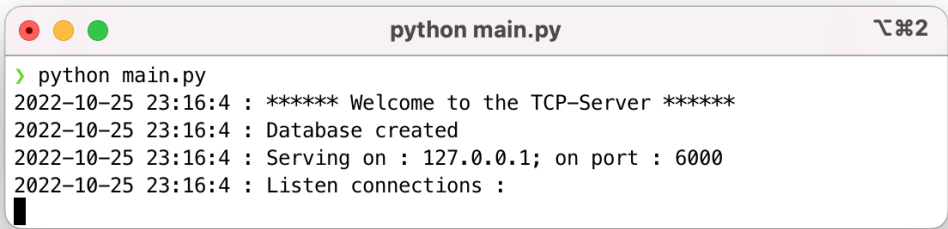


Figura 4.5: Ejecución del servidor TCP.  
Fuente : Elaboración propia

col1	col2	col3
Multiple row	cell2	cell3
	cell5	cell6
	cell8	cell9

1.	Requerimiento uno
2.	Requerimiento dos
3.	Requerimiento tres

4.2.2. Requerimientos del software

4.3. Análisis

4.4. Diseño de Módulos

Tabla 4.2: Detalle de las pruebas realizadas

	Columna 1	Columna 2	Columna 3
Fila 1	item	item	item
Fila 2	item	item	item
Fila 3	item	item	item

Nota. Extraída de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.

Tabla 4.3: Detalle de las pruebas realizadas

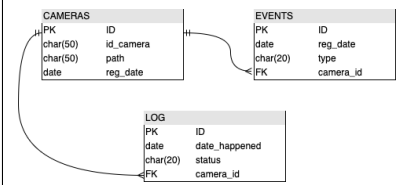
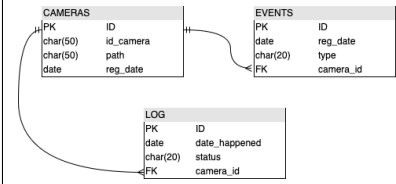
Columna 1	Columna 2	Columna 3
	lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem lorem	<ul style="list-style-type: none"><li>Remote delivery</li><li>Immersive experiences</li><li>text proved</li></ul>
	cell8	<ul style="list-style-type: none"><li>Remote delivery</li><li>Immersive experiences</li><li>text proved</li></ul>



Tabla 4.4: Detalle de las pruebas realizadas

Columna 1	Columna 3
<p><b>CAMERAS</b> PK ID char(50) id_camera char(50) path date reg_date</p> <p><b>EVENTS</b> PK ID date reg_date char(20) type FK camera_id</p> <p><b>LOG</b> PK ID date date_happened char(20) status FK camera_id</p>	<ul style="list-style-type: none"><li>■ Remote delivery</li><li>■ Immersive experiences</li><li>■ text proved</li></ul>
<p><b>CAMERAS</b> PK ID char(50) id_camera char(50) path date reg_date</p> <p><b>EVENTS</b> PK ID date reg_date char(20) type FK camera_id</p> <p><b>LOG</b> PK ID date date_happened char(20) status FK camera_id</p>	<ul style="list-style-type: none"><li>■ Remote delivery</li><li>■ Immersive experiences</li><li>■ text proved</li></ul>

Tabla 4.5: Detalle de las pruebas realizadas

	Columna 1	Columna 2	Columna 3
Fila 1	item	item	item
Fila 2	item	item	item
Fila 3	item	item	item

Nota. Extraída de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.

### 4.5. Identificación de Subsistemas

Referenciando a la figura 4.7.

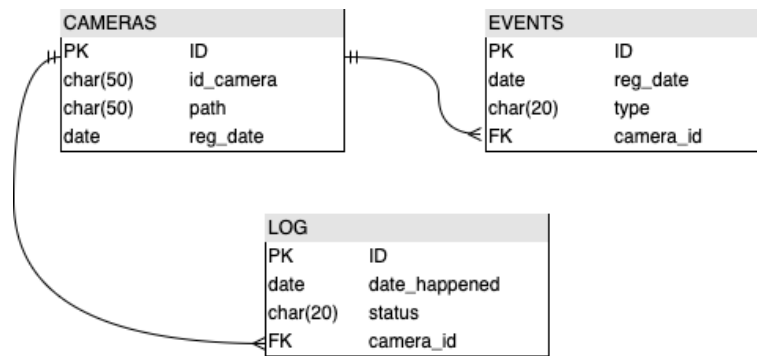


Figura 4.6: Ilustración de un ladrón

Fuente: Adaptada de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.

Referenciando a la figura 4.7.

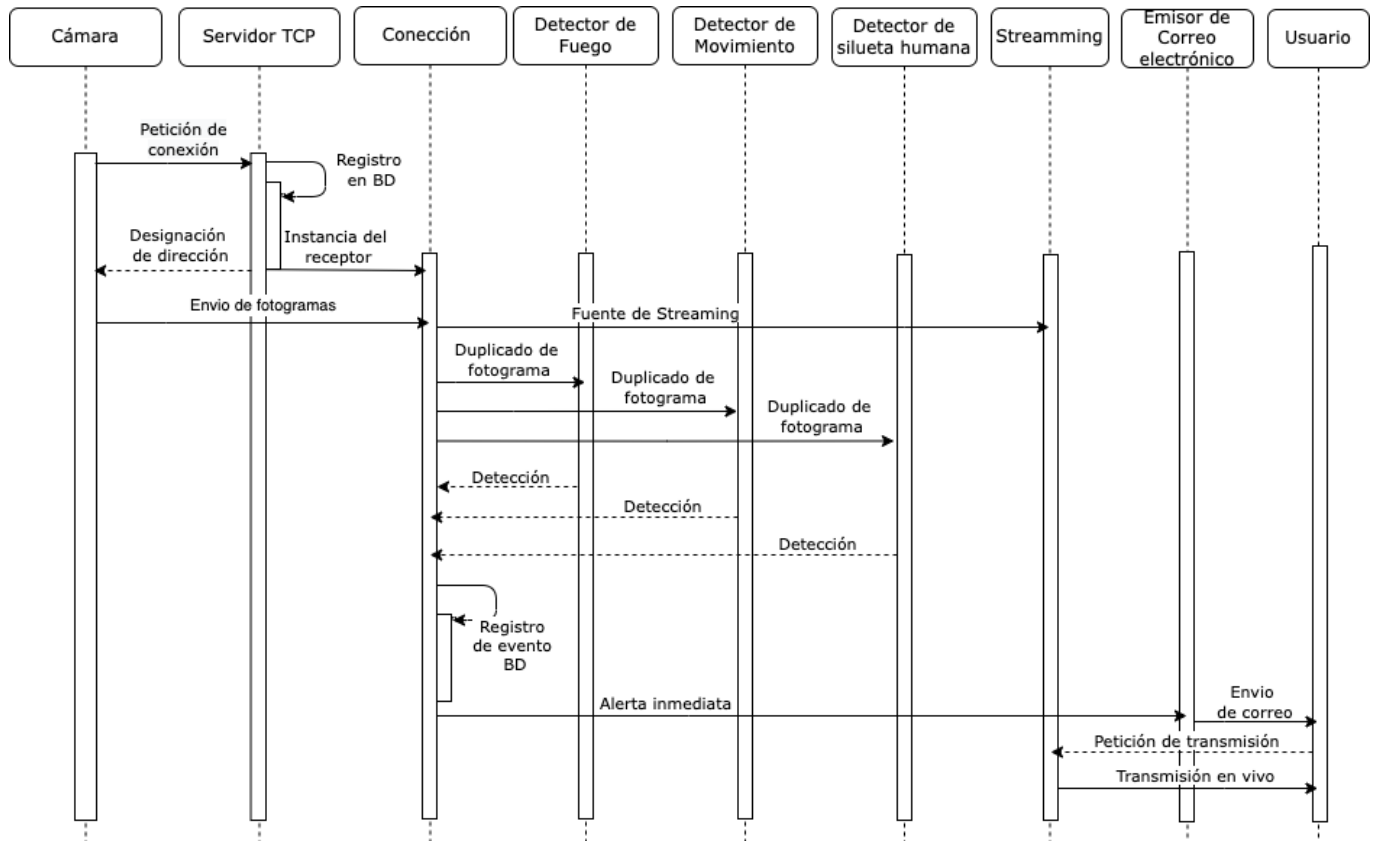


Figura 4.7: Ilustración de un ladrón

Fuente: Adaptada de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.

## **4.6. Comunicación de Sistemas**

### **4.6.1. Sockets**

## **4.7. Planificación**



# Capítulo 5

## Implementación

En este capítulo 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. Diseño de clases
- 5.1.2. Diseño de interfaz de simulador
- 5.1.3. Captura de video
- 5.1.4. Conexión a Servidor TCP
- 5.1.5. Comunicacion y envío de fotogramas

### 5.2. Módulo de Servidor TCP

- 5.2.1. Diseño de clases
- 5.2.2. Manejador de conectores de clientes
- 5.2.3. Streaming HLS
- 5.2.4. Detectores

Detector de movimiento

Detector de intrusos

Detector de fuego/humo

### 5.3. Servidor HTTP

### 5.4. Modulo de envío de correo electrónico

Tabla 5.1: Título de tabla multipágina

[illegible]

Continúa en la siguiente página.

Tabla 5.1 – Continuación de tabla previa

	<b>Columna 1</b>	<b>Columna 2</b>	<b>Columna 3</b>	<b>Columna 4</b>
<b>Fila 11</b>	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.
<b>Fila 12</b>	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Nota. Extraída de Apellido, N. (2000) *Nombre del libro*. Editorial o universidad que lo publicó.





# **Capítulo 6**

## **Pruebas**

- 6.1. Pruebas de integracion**
- 6.2. Prueba de transmision**
- 6.3. Prueba de transmision en vivo**



# Capítulo 7

## Conclusiones

Concluimos que...



# Referencias

- 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](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](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 Editorials. Recuperado de [https://bookdown.org/paul/computational\\_social\\_science/machine-learning-as-programming-paradigm.html#ref-Chollet2018-vf](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/>
- 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](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/>

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, Septiembre). *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 hls (http live streaming)?* 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](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)

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 Editorials. 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>
- TheResearchNest. (2020, Abril). *Computer vision tools and libraries*. Recuperado de <https://medium.com/the-research-nest/computer-vision-tools-and-libraries-52bb34023bdf>
- 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. (2012, Julio). *Perceptor multicapa*. Recuperado de [https://es.wikipedia.org/wiki/Percept%C3%B3n\\_multicapa](https://es.wikipedia.org/wiki/Percept%C3%B3n_multicapa)
- Wikipedia. (2020, Diciembre). *Videovigilancia ip*. Recuperado de [https://es.wikipedia.org/wiki/Videovigilancia\\_IP](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