



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**DESARROLLO DE SISTEMA DE VISUALIZACIÓN POR COMPUTADORA
PARA SENSORES DE MONITOREO DE CONDICIONES PARA GRÚAS
HORQUILLA BASADO EN INTERNET INDUSTRIAL DE LAS COSAS
(SV-IIOT)**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

JOAQUIN IGNACIO MORAGA VALLEJOS

PROFESOR GUÍA:
LUCIANO RADRIGAN FIGUEROA

MIEMBROS DE LA COMISIÓN:
FEDERICO OLMEDO BERÓN
ANDRÉS MUÑOZ ORDENES

SANTIAGO DE CHILE
2023

DESARROLLO DE SISTEMA DE VISUALIZACIÓN POR COMPUTADORA PARA SENSORES DE MONITOREO DE CONDICIONES PARA GRÚAS HORQUILLA BASADO EN INTERNET INDUSTRIAL DE LAS COSAS (SV-IIOT)

El presente trabajo muestra el desarrollo de la vista de visualización computacional del sistema VIU210006, el cual tiene como objetivo detectar movimientos riesgos de la maquinaria grúa horquilla, además de la detección de los elementos de protección personal (EPP). Esta vista de la plataforma busca mostrar de manera amigable datos y vídeos para que el operador de planta pueda reducir riesgos.

La vista desarrollada cuenta con la parte de *frontend*, además de realizar un trabajo de desarrollo en el servidor remoto en donde se ejecuta el algoritmo de detección. Además se realizó desarrollo en un sistema embebido Raspberry Pi para poder capturar vídeo de manera remota. La comunicación entre estos sistemas se hizo a través del protocolo de comunicación MQTT.

Para la vista de visualización computacional se implementa una forma fácil de elegir que método se prefiere para subir un vídeo al sistema detecciones, ya sea a través del navegador web o capturar un vídeo con el sistema embebido seleccionado. Una vez ya procesado el archivo, esta vista muestra la visualización del vídeo procesado por el sistema de detecciones además de los datos kpis entregados por el modelo de detección.

Se busca validar la facilidad de uso de esta aplicación a través de una encuesta que siga el estándar SUS. Hasta el momento no se pudo realizar la validación del sistema.

Los resultados obtenidos constan de una plataforma funcional que cumple con alguno de los objetivos, esto es, la implementación de las funcionalidades básicas. No se alcanza a implementar la sección de gráficos en base a los kpis entregados debido a un retraso en la implementación de una función de transformación del archivo JSON entregado por el modelo de detección

Este trabajo va dedicado a mi familia desde los más cercanos como mis padres y mi hermana, hasta mi abuela, tíos y tías. Quienes siempre me han ayudado y confiado en mi.

Agradecimientos

A mi mamá, papá y hermana, por ser un apoyo durante todo este proceso universitario y en la vida. A mi abuelita Nilda, mi tía Carmen y mi prima Rocio quienes siempre me han recibido con una sonrisa y una once. A Luciano Radrigan, por ayudarme en la redacción de este reporte y por su excelente labor como guía. A mis amigos de Iquique y Santiago, y a cada una de las personas que me dieron una mano cuando lo necesité.

Tabla de Contenido

1. Introducción	1
1.1. Objetivos	2
1.1.1. Objetivo general	2
1.1.2. Objetivos específicos	2
1.2. Estructura de la solución	3
1.3. Estructura del documento	3
2. Antecedentes	4
2.1. Estado del arte	4
2.1.1. Caso de uso	5
2.1.2. Plataformas Alternativas	7
2.1.3. Tecnologías para el desarrollo de aplicaciones web	9
3. Solución	13
3.1. Funcionamiento de la plataforma	13
3.1.1. Usuarios	13
3.2. Arquitectura de la aplicación	13
3.2.1. Entradas de archivos de vídeo	15
3.2.2. Ambiente AWS	16
3.3. Producto mínimo viable	16
4. Implementación	17
4.1. Ambiente AWS	17
4.1.1. Setup de buckets S3 y CDN Cloudfront	17
4.1.2. Trabajo en servidor EC2	18
4.2. Sistema Embebido	19
4.3. Frontend	21
4.3.1. Selección de inputs de archivo	21
4.3.2. Subida de archivos hacia el servidor	22
4.3.2.1. Vista subida de vídeo desde el navegador	22
4.3.2.2. Vista captura de vídeo con sistema embebido	24
4.3.3. Visualización de los datos recibidos	25
5. Evaluación	28
5.1. Encuesta	28
5.2. Trafico de datos	29
5.3. Análisis de los resultados	30

6. Conclusiones y trabajo a futuro	31
6.1. Trabajo realizado	31
6.2. Aprendizaje del proyecto	32
6.3. Trabajo a futuro	32
Bibliografía	34
Anexos	37
A. Mock-Up de vista a implementar	38
B. Resultados de la encuesta y calculo del puntaje	39

Índice de Ilustraciones

2.1.	Modelo de grúa horquilla	6
3.1.	Flujo de uso de la aplicación para la visualización de datos con archivo subido por usuario	14
3.2.	Flujo de uso de la aplicación para la visualización de datos con vídeo capturado	15
4.1.	Bucket usados en el sistema	17
4.2.	Ilustración de trabajo del cliente dentro del servidor	19
4.3.	Ilustración de trabajo del cliente dentro del sistema embebido	20
4.4.	Selección de Inputs de archivo	21
4.5.	Vista para subida de archivo local	24
4.6.	Vista para captura de vídeo por sistema embebido	25
4.7.	Vista de espera para el proceso de archivo	26
4.8.	Visualización del vídeo procesado por el sistema de detecciones	27
A.1.	Mock Up preliminar de diseño	38
B.1.	Resultados de la encuesta	40
B.2.	Tabla de clasificación SUS	41
B.3.	Resultado del promedio de puntajes	42

Capítulo 1

Introducción

En respuesta a los requisitos en constante crecimiento del entorno empresarial en términos de la calidad de los productos proporcionados y la necesidad de aumentar su competitividad, las empresas buscan métodos y técnicas para la mejora continua de sus procesos. Actualmente, las organizaciones tienen la oportunidad de desarrollar y mejorar su desempeño a través del desarrollo tecnológico y el uso de los dispositivos informáticos [1].

En el contexto de la transformación industrial y digital actual, el término ampliamente discutido de Industria 4.0 tiene un papel importante que desempeñar. Industria 4.0 es un concepto que involucra una variedad de soluciones tecnológicas y cubre actividades y medidas que afectan a todo el entorno industrial [2]. Un componente esencial de la Industria 4.0 es el Internet de las cosas (IoT), que permite la integración de múltiples dispositivos en un solo sistema integrado, recopilando datos de toda la infraestructura y el entorno. El seguimiento de varias máquinas basadas en sistemas IoT es una fuente muy importante de conocimiento sobre el proceso. Con miras a la aplicación de la Industria 4.0 en las minas, el concepto de Internet Industrial de las Cosas (IIoT) es particularmente importante, trayendo los supuestos de la Internet de las Cosas al campo de las operaciones industriales.

La implantación del concepto de Industria 4.0 marca el inicio de un desarrollo industrial con un nuevo enfoque para el desarrollo de una gestión sostenible basada en el conocimiento. El desarrollo sostenible es un factor clave que determina la selección y el despliegue de indicadores de evaluación de procesos, que pueden afectar significativamente no solo el desempeño de las operaciones, sino también el desarrollo sostenible desde una perspectiva empresarial. Desarrollo tecnológico, permitiendo la adquisición de parámetros de proceso medibles, puede ser un punto de partida para hacer decisiones razonables, gestionar adecuadamente los recursos, aumentar el nivel de seguridad y como un resultado, mejorar el desempeño comercial sostenible general en la industria [3].

Sin embargo, a pesar de los considerables beneficios y oportunidades derivados de la implementación de Industria 4.0, el potencial para implementar tecnologías emergentes a menudo permanece sin explotar, y la información basada en sensores rara vez se utiliza por completo para respaldar los sistemas de toma de decisiones. Estos desafíos tienen su origen principalmente en los altos costos de implementación de nuevas tecnologías y la necesidad de almacenar y analizar grandes cantidades de datos. Un prerrequisito para el uso óptimo de la infraestructura existente y el logro de los objetivos de la empresa es la formulación

de medidas y herramientas apropiadas para evaluar la eficiencia de los procesos industriales. Los instrumentos significativos para evaluar el éxito de los objetivos de una organización son indicadores definidos por el término Indicadores claves estratégicos o kpis por su abreviación en ingles.

La mejora de los procesos mineros se ha hecho cada vez más notoria en los últimos años. Empresas mineras buscan nuevas oportunidades para optimizar procesos, implementando nuevas políticas para la automatización, usando tecnologías digitales y utilizando programas informáticos modernos para soportar procesos tecnológicos.

Teniendo en cuenta la progresiva digitalización de la industria minera, es necesario la integración de un sistema inteligente que pueda ser capaz de recolectar datos e indicadores claves de rendimiento de grandes procesos en terreno y posteriormente desplegar esta información de manera clara para que los usuarios puedan analizar estos datos y tomar decisiones que afecten la eficiencia y efectividad de los procesos estudiados.

El caso de aplicación abordado en esta memoria busca integrar procesos tecnológicos para poder crear una plataforma que permita analizar vídeos de movimientos de grúas del tipo horquilla en los cuales se realizaron movimientos riesgosos o se presentó un accidente. Dentro de este análisis se muestran los movimientos riesgosos realizados en conjunto a la detección del equipamiento de protección personal (EPP) presente por el operador de la máquina.

El objetivo que se busca lograr con la vista de este sistema es poder facilitar la tarea de los encargados de planta a la hora de la ocurrencia de un accidente que involucre las grúas horquillas, con el sistema se podrán detectar fácilmente en el vídeo los movimientos realizados y si es que se cumplían con las normas de seguridad personal presentes para el sector de trabajo.

1.1. Objetivos

1.1.1. Objetivo general

Diseñar e implementar un sistema web el que seleccionan distintos modelos de inteligencia artificial y recibe vídeos de grúas horquillas en operación para así poder estimar y visualizar indicadores claves estratégicos (kpis).

1.1.2. Objetivos específicos

Para poder cumplir con el objetivo general de proyecto se plantean los siguientes objetivos específicos:

1. Programar un *backend* que pueda procesar los vídeos a través de modelos de *Machine Learning* para obtener los indicadores claves.
2. Implementar conexión con dispositivos embebidos basados en IoT como fuentes generadoras de datos
3. Diseñar e implementar un *frontend* con una interfaz amigable para la visualización gráfica de los datos obtenidos.

4. Implementar un visualizador de vídeos en el *frontend* para reproducir el contenido subido por el usuario o por los dispositivos embebidos.

En particular, el proyecto busca facilitar el trabajo del encargado de seguridad con respecto a la horquilla, dándole información de los riesgos presentes en forma de indicadores claves con su representación gráfica correspondiente. Además se hace uso de sistemas embebidos los cuales se encontraran conectados al sistema de detecciones para poder capturar vídeos de manera remota en sectores designados.

1.2. Estructura de la solución

La plataforma desarrollada cuenta con 4 vistas principales: La vista Resumen, la cual muestra una tabla con el resumen de los estados de las máquinas; la pestaña de máquinas, la cual permite observar detalles específicos de una máquina seleccionada; la vista de sensores que entrega datos sobre los sensores instalados en una máquina específica; y finalmente la vista a trabajar de *visual computing* en la cual dependiendo del *input* entregado por el usuario se pondrán obtener datos sobre maniobras de riesgo en conjunto a una visualización gráfica de indicadores claves.

Con respecto a la vista de *visual computing* en esta un puede elegir el modelo de inteligencia artificial que se estime conveniente para analizar el vídeo con posibles maniobras peligrosas, el modelo en la parte de *backend* entrega los indicadores claves de rendimiento que finalmente son ilustrados de manera gráfica.

1.3. Estructura del documento

El presente documento se divide en 6 capítulos. En el Capítulo 2 se da a conocer el estado del arte sobre la Industria 4.0 y se da a conocer el caso de uso y como la solución propuesta busca resolver los problemas expuestos. Además se explican todos los conceptos necesarios para entender cómo se desarrolla la vista a implementar en la plataforma.

Dentro del Capítulo 3 se encuentra una explicación detallada de la vista a implementar, la arquitectura a usar y el flujo de uso de la aplicación. También se presenta el producto mínimo viable (MVP) que se busca lograr en este proyecto.

El contenido del Capítulo 4 trata sobre la implementación de la solución, detallando lo implementado en el desarrollo del trabajo y explicando la lógica y funcionamiento de los componentes presentes en la vista implementada.

En el Capítulo 5 se explica como se evaluara la solución final, se da a conocer la encuesta a realizar y se expone el análisis de los resultados.

Finalmente en el Capítulo 6 se pasa a las conclusiones respectivas al trabajo realizado. Esta última sección presenta una pequeña discusión y el trabajo futuro propuesto.

Capítulo 2

Antecedentes

2.1. Estado del arte

Con respecto al estado del arte para la Industria 4.0 en el ámbito minero, en Chile si bien ha ido en crecimiento la integración de practicas de investigación y desarrollo en tecnologías aproximadamente un 32,3% de empresas mineras en el territorio nacional no han invertido en este ámbito, la mayor razón en común que tienen estas empresas para no invertir en tecnología es que no resulta prioritario o necesario para la empresa (43 % de las empresas pertenecientes a este grupo) [4].

En esta línea la mayoría de las empresas que sí invierten en proyecto de investigación y desarrollo de tecnologías son las mineras que generan ventas de más de 1 billón de pesos [4] pues son las que disponen de más recursos y la mayoría de estas son empresas privadas extranjeras.

Durante los últimos años la tendencia a invertir en proyectos de tecnología ha ido aumentando paulatinamente esto debido a los distintos fondos de financiamientos dispuestos por el estado [4] ya sea en fondos Conicyt (Comisión Nacional Científica y Tecnológica), Fondef (Fondo de Fomento al Desarrollo Científico y Tecnológico) y ciertos fondos ligados a proyectos de Corfo (Corporación de Fomento de la Producción). La mayoría de las empresas que postulan a estos financiamientos son empresas privadas nacionales.

Bastantes proyectos de inclusión de tecnologías de IoT se encuentran en los sectores agrónomos, pesqueros, industriales y mineros. Ejemplos de estos pueden ser el uso de sensores de cambio de flujo del agua en estanques de pesca agrícola de salmón, con el monitoreo de datos como flujo de agua, cantidad de peces en el estanque y niveles de contaminación en el agua se pueden tomar decisiones que logren mejorar los factores claves que logren una mayor producción.

En el apartado agrícola, las tecnologías IoT son usadas mayormente para generar alerta de cambios de pH en el suelo o temperatura en el ambiente para así poder tomar medidas en los horarios de riego o algún cambio que beneficie la producción del sembrado.

Dentro de las implementaciones de proyectos de tecnología en la industria minera existen tecnologías para el monitoreo integrado de relaves, monitoreo de condiciones en correas transportadoras y monitoreo en grandes motores mineros entre otras. La mayoría de estas

implementaciones si bien cumplen su cometido principal no aprovechan al máximo el uso de sistemas IoT, ya que la gran parte de los datos obtenidos por estos sistemas se mantienen dentro de la misma planta y no son almacenados en un servicio remoto para un posterior análisis de rendimiento y/o comportamiento.

Con respecto al uso de sistemas basados en redes neuronales o inteligencia artificial dentro del país la implementación de estos sistemas todavía son escasos, pero los instaurados han generado un impacto positivo en la industria, como ejemplo la empresa nacional Codelco actualmente está integrando un modelo predictivo denominado “Miner Plus”, permitiría reducir costos de hasta \$20 millones de dólares en el proceso de construcción de relaves ubicados en Chuquicamata.

Un sistema similar al desarrollar son el sistema de vigilancia eficiente con cámaras MCI de la empresa BOSCH el cual esta especializado para minas a tajo abierto, este sistema captura y almacena imágenes, cubriendo además; perímetro, operaciones y tránsito. Con estos datos multimedia la cámara puede entregar realizar la entrega de datos de espera, clasificación de objetos, aglomeración, velocidad, sentido, entre otros [5].

Para poder desarrollar un sistema de monitoreo la entidad interesada debería invertir en el desarrollo y mantenimiento en el tiempo de una red neuronal la cual se debería ir actualizando con los avances de maquinaria presente en el tiempo, además se debe considerar los costos de *deployment* de un sitio web en el cual se deberán desarrollar distintos tipos de visualizaciones para los datos obtenidos.

Dentro de la industria, se han desarrollado sistemas de procesamiento de vídeo los cuales realizan detecciones de distinta índole gracias al uso de modelos de inteligencia artificial. En el sector médico se encuentran métodos vanguardistas para las detecciones de masas anormales en proceso como la colonoscopia a través del uso de redes neuronales pre entrenadas [6]. Una de las ramas de la industria moderna más destacada en el uso de procesamiento de vídeos es la de videovigilancia, dentro de esta se han usado sistemas de vídeos para detectar de manera automática si se realizaron traspasos no autorizados a sectores peligrosos como lo son los cruces de trenes [7]. También se han usado sistemas de monitoreo en tiempo real en sectores como carreteras para poder detectar la cantidad de automóviles dependiendo de su tipo para así tomar decisiones para poder optimizar el flujo de estos [8].

2.1.1. Caso de uso

En este proyecto se trabajará con una grúa horquilla, esta es un tipo de maquinaria la cual tiene como función primaria levantar, bajar, y mover cargas con muy poco, o sin ningún tipo de esfuerzo manual. Esto es posible usando un mástil en la parte de enfrente de una grúa horquilla.

Se define como un equipo móvil de contrapeso, ya que la carga que manipula está fuera de la distancia entre sus ejes, tiene dos barras paralelas planas en su parte frontal que se introducen en el palé (debajo de la carga), llamadas horquillas (a veces, coloquialmente también uñas), montadas sobre un soporte que se desliza verticalmente por un mástil con lo que se consigue el movimiento de elevación y descenso de la carga. La separación entre horquillas es variable para adaptarse a distintas medidas de palé o cargas.

La elevación de las horquillas, así como la inclinación del mástil y otros movimientos se realizan a través de pistones hidráulicos que forman parte de un sistema óleo-hidráulico accionado por una bomba a la que suministra energía el mismo motor utilizado para el desplazamiento en las de motor térmico y generalmente por un motor aparte en el caso de las eléctricas.

Su uso requiere una cierta capacitación y los gobiernos de distintos países exigen a los negocios que sus empleados tramiten licencias especiales para su manejo.



Figura 2.1: Modelo de grúa horquilla

Dentro de las industrias chilenas el uso de la grúa horquilla se encuentra ligada a varias de estas, una área específica es la industria minera, en esta las operaciones segura de maquinaria crítica depende de muchas variables como lo son diseño adecuado, las condiciones ambientales de operación y las prácticas operativas. Lo más importante es informar adecuadamente a los operadores sobre los eventos de Riesgo para garantizar el funcionamiento seguro de la máquina y el uso correcto por parte del operador. Con esto en cuenta la industria 4.0 a comenzado ha darle suma importancia a la conciencia situacional (situational awarness) presente en muchos programas de seguridad, esto implica recopilar información y señales del entorno para que los comerciantes puedan desarrollar una buena idea de lo que está sucediendo y luego predecir lo que sucederá a continuación [9].

Actualmente algunas normativas nos permiten establecer un funcionamiento seguro de las horquillas, dentro de esta se destaca Osha 3949, la cual cataloga cuáles son las actividades riesgosas en proceso que se deben evitar.

Las operaciones riesgosas identificadas en Osha 3949[10] son las siguientes:

- Sobrecargar las horquillas.
- Levantar carga con una sola parte de la horquilla.
- Usar dos máquinas para levantar una carga.
- Bajar pendientes de frente con la grúa cargada.
- Subir o bajar la carga con la grúa en movimiento.
- Transportar una carga elevada.
- Conducir en pendientes con una inclinación superior a 15°.
- Empujar la carga con la horquilla.
- Subir operadores en las horquillas.
- Conducir a exceso de velocidad.
- No respetar las áreas de tránsito.
- Girar en pendientes.

En 2020, la industria logística reportó más de 151 accidentes en Chile, de los cuales el 39 % fueron declarados con lesionados, el 60 % sin lesionados y el 1 % como fatales.

Con la aplicación a desarrollar se busca reducir al mínimo esta cantidad de accidentes a futuro haciendo uso de la visualización de datos entregada por una inteligencia artificial entrenada para poder detectar estos movimientos riesgosos en accidentes y también observar si es que el operador cuenta con su equipamiento de protección personal.

2.1.2. Plataformas Alternativas

Dentro del mercado existen plataformas que se especializan en soluciones empresariales para industrias, estas principalmente se especializan en la consultoría, manutención de sistemas y visualización de datos. La empresa alemana SAP es la pionera a nivel mundial en el desarrollo de este tipo de *software*, otras empresas como Microsoft y Oracle también ofrecen servicios de planificación de recursos empresariales (ERP).

Este es un listado de algunos servicios ofrecidos por estas empresas.

1. **SAP Bussines One:** *Software* ERP con énfasis para pequeñas y medianas empresas, apunta a la automatización de funciones claves del negocio ya sea en el área de finanzas, operacionales y recursos humanos, se encuentra disponible online. [11]
2. **SAP Bussines ByDesing:** *Software* ERP con énfasis para empresas medianas y de alta magnitud, este servicio cuenta con los mismos servicios que SAP Bussines One pero con *features* especializados para el rubro de la empresa contratante. [12]

3. **Oracle Fusion Cloud ERP:** Es un conjunto de soluciones de ERP en la nube completo y moderno que proporciona a sus equipos capacidades avanzadas, como la IA para automatizar los procesos manuales que los ralentizan, los análisis para reaccionar a los cambios del mercado en tiempo real y las actualizaciones automáticas necesarias para mantenerse actualizados y ganar una ventaja competitiva. [13]
4. **Microsoft Dynamics 365:** Es una suite que integra todas las herramientas necesarias para una gestión completa la compañía. Alojado en la nube, Microsoft Dynamics dispone de un ERP (Microsoft Dynamics 365 Business Central), un Customer Relationship Management (Microsoft Dynamics 365 Sales), Business Intelligence (Power BI) y otras herramientas de ofimática.[14]

La mayoría de estos servicios están diseñados de manera general para ser aceptado por la mayoría de empresas.

Un problema presente en estos sistemas es la falta de integración con modelos de inteligencia artificial, el sistema de Oracle está comenzando con la adición de estos modelos, pero en general los *software* de ERP se encuentran en una fase de desarrollo para integrar estos sistemas, es aquí donde el proyecto presente busca generar una ventaja competitiva sobre los sistemas presentados pues este interactúa de manera directa con los datos entregados por un modelo pre-entrenado de inteligencia artificial.

Existen sistemas específicos para evitar accidentes con grúas horquillas como son los siguientes:

1. **Triomobil Forklift Safety and Tracking System:** Sistema basado en sensores de movimiento los cuales limitan zonas de tráfico de grúas, garantiza la seguridad al reducir automáticamente la velocidad de las grúas horquillas en las zonas limitadas. Y también cuando se encuentren peligrosamente cerca de trabajadores y otros montacargas. [15]
2. **Sewio Forklift Tracking and Monitoring System:** Sistema basado en sensores de movimiento, cada grúa está equipada con una etiqueta que transmite una señal a los sensores adheridos a los techos. Luego, los sensores envían la señal a un servidor cloud y calculan la posición del vehículo en tiempo real y con una precisión de 30 cm. [16]
3. **VIA Mobile360 AI Forklift Safety:** Sistema el cual se conecta directamente a la máquina. A través de las cámaras integradas del sistema se puede detectar movimiento de personas y máquinas en las partes frontal y delantera del equipo, además cuenta con una característica de reconocimiento facial para poder identificar a conductores designados. [17]
4. **SIERA.AI Monitoring and Safety System:** Sistema el cual se conecta directamente a la máquina. Se pueden detectar a través de sensores de proximidad a trabajadores u otras máquinas para poder alertar al operador, también presentan un sistema de cámaras con las cuales se pueden monitorear acciones peligrosas las cuales puedan ocurrir por la parte frontal o trasera de la máquina. Además presenta un sistema de reporte online para estándares de seguridad como OSHA. [18]

La mayoría de los sistemas mencionados anteriormente se basan en la combinación de sensores ya sea de movimiento o captura de imagen como son las cámaras en conjunto a la

inteligencia artificial para poder realizar alertas al operador sobre la proximidad de trabajadores o maquinaria cercana o de traspaso de las áreas de trabajado anteriormente designadas.

Si bien estas opciones de sistemas presentan grandes ventajas en lo que a seguridad y detección de accidentes respecta, estos no se encuentran disponibles para poder ser adquiridos en el territorio nacional, el único sistema que presenta posibilidad de venta hacia el extranjero es SIERRA.IA, pero de por sí es un sistema costoso que además debe ser evaluado por la empresa antes de poder realizar la prestación de servicios.

Debido a los problemas anteriores, se busca crear una alternativa de solución que no dependa de *hardware* de alto costo, pero logre ser útil a la hora de analizar accidentes que involucren grúas horquillas.

2.1.3. Tecnologías para el desarrollo de aplicaciones web

En cuanto a las herramientas, *frameworks* o bibliotecas de desarrollo de aplicaciones web existen varias opciones: Django, React, Angular, Express y Ruby on Rails son algunas de ellas. Debido a que este proyecto ya estaba en curso se usó React para desarrollar las vistas principales de la aplicación en conjunto de Tailwind CSS para agregar estilos distintos a los atributos, además se cuenta con conocimientos previos de la tecnología de React en conjunto a una amplia documentación oficial y no oficial por parte de la comunidad desarrolladora, para el *backend* se trabajara con NodeJS. El lenguaje que se usara con estos *frameworks* es JavaScript debido a que se tienen conocimientos previos del uso de este lenguaje.

En resumen, se opta por React para desarrollar el *frontend* de la aplicación. Para el *backend*, se decide usar NodeJS.

Conceptos tecnológicos

A continuación se presentan conceptos relacionados con la tecnología que se decidió utilizar:

- **Git:** Es un sistema de control de versiones distribuido de código abierto y gratuito diseñado para manejar proyectos con velocidad y eficiencia. [19]
- **Github:** Es una plataforma DevOps basada en Git y que abarca todo el ciclo de vida del desarrollo de *software*. El servicio permite gestionar, administrar, crear y conectar los repositorios con diferentes aplicaciones y hacer todo tipo de integraciones con ellas. [20]
- **Python:** Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. [21]
- **JavaScript:** Es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de *scripting* (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador. [22]

- **NodeJS:** Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono. [23]
- **React:** Es una biblioteca de basada en JavaScript para construir interfaces de usuario. Su principal característica es la componetización.[24]
- **MUI:** MUI es una biblioteca masiva de componentes de interfaz de usuario que los diseñadores y desarrolladores pueden usar para crear aplicaciones React.[25]
- **Tailwind CSS:** Es una biblioteca open-source de herramientas para *frontend* que permite crear sitios rápidos y responsivos. [26]
- **Application Programming Interface (API):** Es un conjunto de procedimientos, definiciones y protocolos que se utiliza para desarrollar e integrar aplicaciones web entre sí. [27]
- **Amazon Web Services:** AWS es una colección de servicios de computación en la nube pública (también llamados servicios web) que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com. En particular se harán uso de los servicios S3 Buckets para almacenar archivos, Cloudfont para realizar streaming de vídeos. [28]
- **Amazon S3 Buckets:** Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Clientes de todos los tamaños y sectores pueden almacenar y proteger cualquier cantidad de datos para prácticamente cualquier caso de uso, como los lagos de datos, las aplicaciones nativas en la nube y las aplicaciones móviles. [29]
- **Amazon EC2:** Amazon Elastic Compute Cloud (Amazon EC2) ofrece la plataforma de computación más amplia y profunda, con más de 500 instancias y la posibilidad de elegir el procesador, almacenamiento, redes, sistema operativo y modelo de compra más reciente para que pueda ajustarla al máximo a las necesidades de su carga de trabajo. Se admiten procesadores Intel, AMD y ARM. [30]
- **Amazon CloudFront:** Amazon CloudFront es un servicio de red de entrega de contenido (CDN) creado para ofrecer un alto rendimiento, seguridad y comodidad para los desarrolladores. [31]
- **AWS SDK:** El SDK de AWS para JavaScript simplifica el uso de los servicios de AWS al proporcionar un conjunto de bibliotecas que son coherentes y habituales para los desarrolladores de JavaScript. Es compatible con la consideración del ciclo de vida de la API, como la administración de credenciales, los reintentos, la transformación de la representación de datos a un formato adecuado, la serialización y la deserialización. El SDK de AWS para JavaScript también admite abstracciones de nivel superior para simplificar el desarrollo.[32]
- **MQTT:** MQTT es un protocolo de mensajería basado en estándares, o un conjunto de reglas, que se utiliza para la comunicación de un equipo a otro. Los sensores inteligentes, los dispositivos portátiles y otros dispositivos de Internet de las cosas (IoT) generalmente

tienen que transmitir y recibir datos a través de una red con recursos restringidos y un ancho de banda limitado.[33]

- **MQTT.js:** MQTT.js es una biblioteca cliente para el protocolo MQTT, escrita en JavaScript para NodeJS y el navegador.[34]
- **Paho MQTT:** Paho Python Client es una biblioteca que proporciona una clase de cliente compatible con MQTT v5.0, MQTT v3.1.1 y v3.1 en Python 2.7 o 3.x. También proporciona algunas funciones de ayuda para que la publicación de mensajes únicos en un servidor MQTT sea muy sencilla.[35]
- **Sistemas Embebidos:** Un sistema embebido es un sistema de computación basado en un microprocesador o un microcontrolador diseñado para realizar una o algunas pocas funciones dedicadas.[36]
- **Raspberry Pi:** La Raspberry Pi es una computadora de bajo costo y con un tamaño compacto, del porte de una tarjeta de crédito, puede ser conectada a un monitor de computador o un TV, y usarse con un mouse y teclado estándar. Es un pequeño computador que corre un sistema operativo Linux. [37]

Modelo predictivo

El presente modelo predictivo a usar esta basado en redes neuronales convolucionales (CNN) y redes neuronales de memoria a corto plazo (LSTM) generando un modelo híbrido que proporciona una alta precisión para la detección de eventos de riesgo instantáneos y temporales [38].

Si bien las técnicas CNN y LSTM son herramientas extremadamente poderosas para el reconocimiento de eventos de riesgo instantáneos o dependientes del tiempo (respectivamente), para casos altamente no lineales o procesos probabilísticos, es necesario recurrir a combinaciones de diferentes técnicas de *Machine learning* que intentan mejorar el rendimiento de los modelos, esto se conoce como métodos de ensamblaje. Existen tipos comunes de conjuntos, entre los que se destacan (i) Bagging, (ii) Boost, y (iii) Stacking [39].

- (i) **Bagging:** Considera procesos de aprendizaje débiles y homogéneos, que los aprenden de forma independiente en paralelo y los combinan siguiendo un proceso de promediación determinista.
- (ii) **Boost:** Considera aprendices débiles homogéneos, los aprende secuencialmente de forma muy adaptativa (un modelo base depende de los anteriores), y los combina siguiendo una estrategia determinista.
- (iii) **Stacking:** Consiste en combinar modelos que difieren en su rendimiento y luego entrenar un metamodelo para generar predicciones basadas en las múltiples predicciones devueltas por estos modelos débiles. Los métodos de apilamiento ayudan al modelo a prestar atención a la diversidad de datos

Para la selección del algoritmo correcto para la detección de eventos de riesgo, es esencial comprender el proceso de generación de datos y sus características, tales como: tiempo de

muestreo, características temporales, señales de comportamiento no lineales o no estacionarias.

Si el proceso a monitorear es altamente probabilístico, como operaciones no estacionarias, en diferentes ambientes y con diferentes operadores, es necesario utilizar técnicas de método de ensamblaje.

El modelo fue implementado usando el *framework* de reconocimiento de visual YoloV5, este fue implementado para poder detectar los movimientos de riesgos de los montacargas o para detectar el uso de EPPs en los operadores, para el caso de las grúas horquillas recibe como parámetro el vídeo para analizar detecciones y entrega como outputs un nuevo archivo de vídeo con las detecciones encontradas y un archivo json el cual debe ser procesado para poder entregar los datos de kpis como lo son tiempo efectivo de uso, tiempo de uso y un contador de actividad de riesgo realizada. Dentro del archivo de vídeo generado se encuentran las kpis del tipo de actividad de riesgo que se efectúa y si el trabajador cuenta con EPPs.

Kpis

Los indicadores claves estratégicos o kpis que genera el programa de detecciones son los siguientes.

- **Tiempo efectivo de uso**
- **Tiempo de uso**
- **Contador de actividad de riesgo realizada**
- **Tipo de actividad de riesgo realizada**
- **Trabajador con elementos de protección personal (EPP)**

Los indicadores de tiempo efectivo de uso y tiempo de uso se diferencian en que el tiempo efectivo de uso se refiere a la cantidad de tiempo en movimiento en el cual se está ejecutando una acción, en cambio el tiempo de uso corresponde a la cantidad de tiempo que la grúa está encendida.

Las actividades de riesgos son las definidas en la sección 2.1.1 y finalmente los elementos de protección personal corresponden a cascos de seguridad, chalecos reflectantes y mascarillas.

Capítulo 3

Solución

Este capítulo explica en detalle la vista para la plataforma propuesta como solución, cómo funciona esta, cuál es la arquitectura del sistema. Además, se presenta el producto mínimo viable (por definir).

3.1. Funcionamiento de la plataforma

Básicamente la plataforma desarrollada cuenta con 4 vistas principales: La vista de Resumen, la cual muestra una tabla con el resumen de información de las máquinas, la vista de máquinas en la cual se pueden obtener datos específicos de cierta máquina; la vista para Sensores que de manera similar a la vista de máquinas en esta vista se obtiene datos de los sensores ligados a la máquina seleccionada; y finalmente la vista a desarrollar de *Visual Computing* en la cual se obtiene información y visualización de datos sobre posibles riesgos presente en un vídeo analizado por un modelo de inteligencia artificial.

3.1.1. Usuarios

Los usuarios a usar este sistema son los encargados de operaciones o de planta correspondientes. También el sistema al poder detectar el uso correcto de equipamiento de protección personal el sistema también puede ser usado por ingenieros en previsión de riesgos.

3.2. Arquitectura de la aplicación

La plataforma presenta una arquitectura del tipo *serverside rendering*, presenta un *backend* desarrollado en NodeJS, para el lado del *frontend* se usa el *framework* React trabajado con JavaScript y para la estilización de las vistas este se acompaña con Tailwind CSS.

Por la parte de *software* se cuenta con distintos modelos de inteligencia artificial predictivos los cuales entregan kpis que miden el riesgo presente en el vídeo entregado.

Con respecto al *hardware* presente se cuenta con una Raspberry Pi 4 con cámara integrada para poder captar vídeos de las operaciones. El servidor de almacenamiento cuenta con un sistema operativo Amazon Linux, además de tener un aproximado de 32 GB de memoria para almacenamiento, finalmente cuenta con GPU Nvidia para poder correr de manera rápida el sistema de detecciones.

El siguiente diagrama muestra el flujo de uso de la aplicación cuando se desea subir un archivo como usuario.

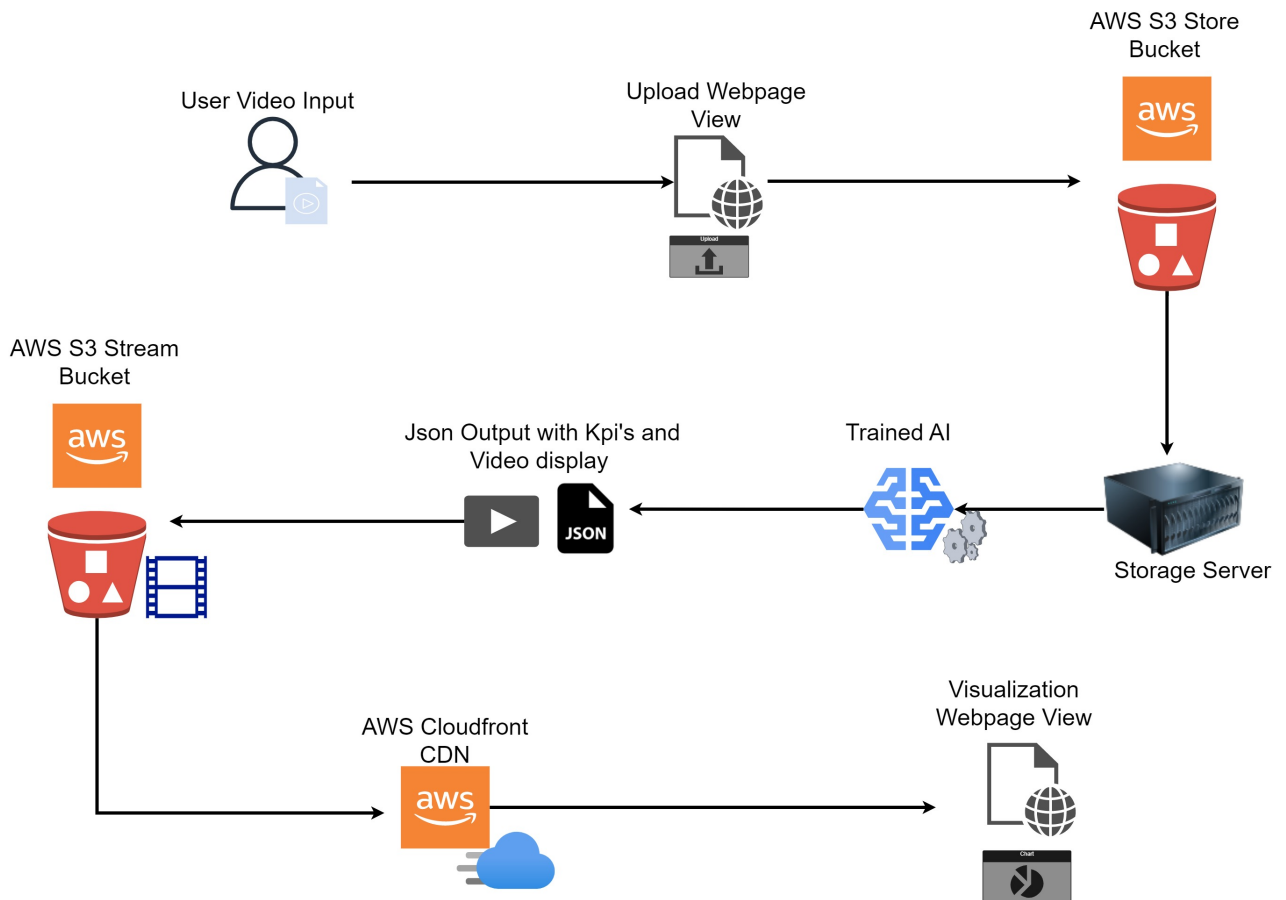


Figura 3.1: Flujo de uso de la aplicación para la visualización de datos con archivo subido por usuario

En este diagrama se logra observar cómo las fuentes de información es un vídeo subido por el usuario, este archivo es subido mediante la vista de la página a un Bucket de almacenamiento de servicio de S3, al terminar la subida del archivo, la vista de *frontend* de la página se comunica con el servidor de almacenamiento mediante comunicación MQTT basada en *websocket* haciendo uso de la librería MQTT.js en la parte de *frontend* del proyecto y Paho-MQTT por el lado del servidor, posteriormente se descarga el archivo en el servidor de almacenamiento y se procesa el vídeo con el modelo de inteligencia entrenada seleccionada por el usuario, después de que la inteligencia genere el archivo JSON con los resultados y vídeo de salida, estos archivos son subidos a otro Bucket del servicio de S3 para que el sistema de Cloudfront pueda realizar stream de vídeo procesado y el *frontend* pueda realizar llamados para obtener el JSON final y así mostrar estos resultados en una pantalla de visualización computacional para poder ser analizados por el usuario.

También existe la opción de capturar vídeo con sistemas embebidos enlazados a la aplicación como en este caso lo es una Raspberry Pi, el flujo de uso es el siguiente:

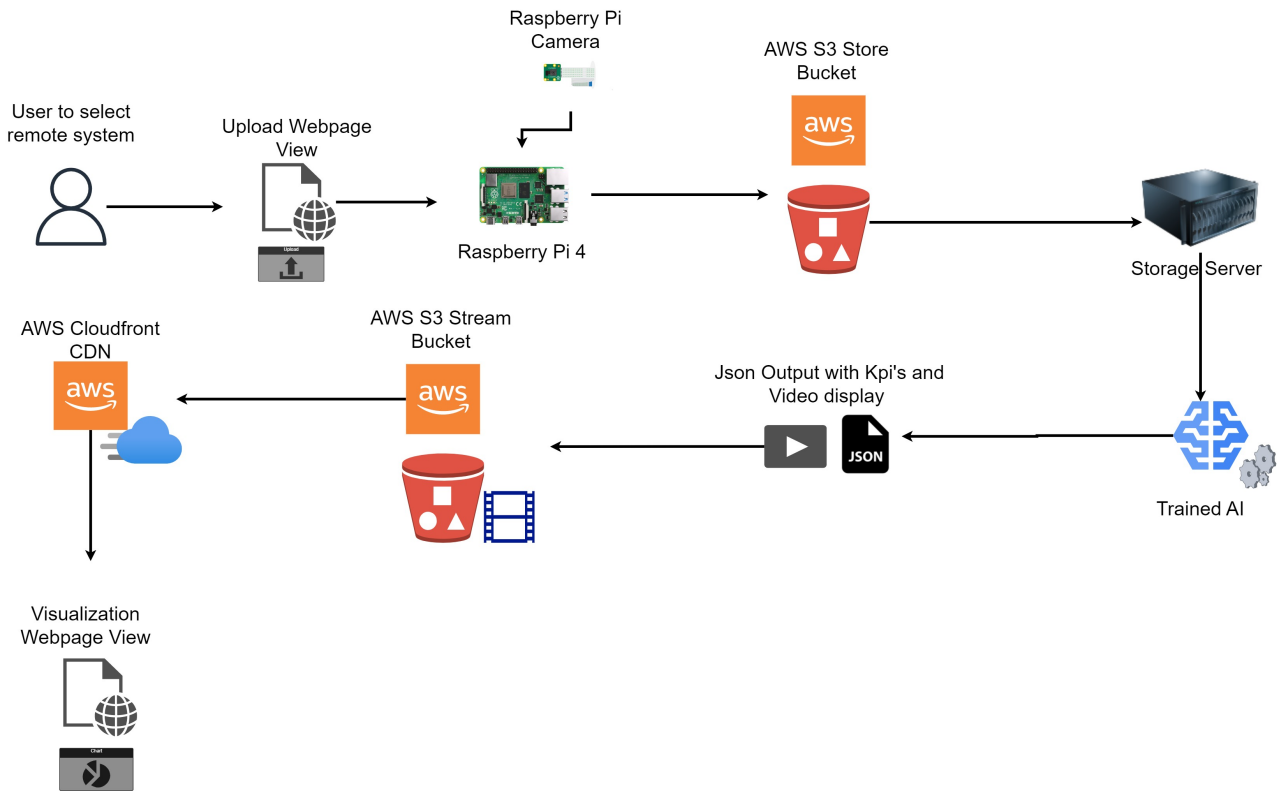


Figura 3.2: Flujo de uso de la aplicación para la visualización de datos con vídeo capturado

Para este caso se manda una señal mediante comunicación MQTT desde la parte del *frontend* de la aplicación hacia el sistema embebido anexo seleccionado por el usuario, el dispositivo captura un vídeo y posteriormente lo sube al Bucket de almacenamiento en el ambiente de AWS, finalmente se comunica con el servidor de almacenamiento EC2 con mensajería MQTT para que este ejecute el mismo proceso de procesar el vídeo y subir los resultados para que puedan ser expuestos en la sección de *visual computing*.

3.2.1. Entradas de archivos de vídeo

Existen dos maneras de proporcionar archivos de vídeos para ser procesados en el lado del servidor.

- **Subida de archivo de vídeo:** El usuario puede subir un archivo de vídeo directamente desde su computador hacia el sistema.
- **Captura de vídeo con sistema embebido:** Se puede seleccionar la opción de capturar un vídeo a través de un sistema embebido anexo. Actualmente se tiene configurado para que se tome una sola captura de duración de 20 segundos, pero esto puede ir cambiando según lo desee el operador

3.2.2. Ambiente AWS

Se hace uso de los sistemas de S3 Buckets, CDN Cloudfront y EC2 Servers. El sistema de buckets se usa para almacenar los archivos subidos o capturados por el usuario para que después estos puedan ser accedidos por el servidor de procesamiento. Cloudfront es usado para poder realizar streaming de los archivos procesados por el sistema predictivo hacia la sección de visualización computacional en la parte de *frontend*. Finalmente el servicio EC2 Servers es usado para mantener un servidor que pueda correr los sistemas predictivos de detección de manera remota y rápida.

3.3. Producto mínimo viable

En el área de computación se ha visto que un desarrollo progresivo e iterativo es mejor, en el sentido de que permite darnos cuenta de errores y validar cuanto antes la aplicación a través de un MVP. Definimos como MVP de la plataforma un sistema en el cual se tengan presente las siguientes características:

- **Subida de archivo:** Se pueden realizar subidas de archivos de vídeos usando archivos locales del computador o capturados por los sistemas embebidos, se asume que el archivo subido por el usuario muestre un proceso en el cual se ve involucrada una grulla horquilla, además también se asume que los sistemas embebidos anexos se encuentran funcionando para poder capturar vídeo.
- **Visualización de los datos:** Se pueden observar una interpretación gráfica de los kpis obtenidos del procesamiento de vídeos a través del modelo predictivo, además se puede visualizar el vídeo con las detecciones de los movimientos de la grúa horquilla.

Dentro de la visualización del vídeo se podrán observar las detecciones de los kpis de movimiento riesgoso, identificación de uso de EPPs y cada uno de estos es marcado con un cuadro de color.

Para la interpretación gráfica se busca mostrar los kpis de tiempo de uso y tiempo efectivo de uso en conjunto a la cantidad de movimientos de riesgos detectados, todo esto a través de gráficos de tipo radar

Capítulo 4

Implementación

En este capítulo se da a conocer el funcionamiento general de la vista implementada en conjunto una explicación del trabajo realizado para poder lograr el resultado.

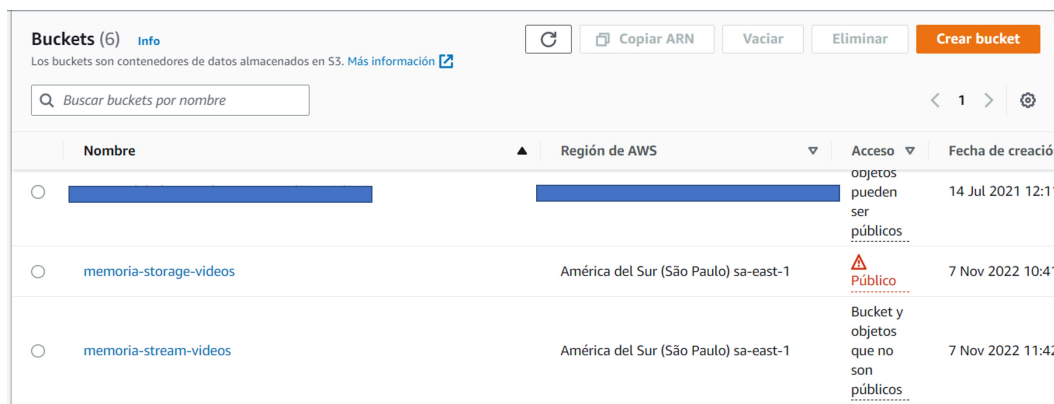
4.1. Ambiente AWS

4.1.1. Setup de buckets S3 y CDN Cloudfront

Para poder llevar a cabo la solución propuesta se comenzó creando los buckets de almacenamiento de archivos, se crearon dos buckets de almacenamiento, el primero tiene como funcionalidad almacenar los archivos subidos por la plataforma ya sea por el usuario o capturados por otros sistemas anexos, estos archivos posteriormente pueden ser accedidos por distintos scripts de ejecución en la parte del servidor EC2. También almacena los archivos json con los kpis generados por los algoritmos de detección.

El otro bucket creado fue el de bucket de stream de vídeo, este almacena los archivos de vídeos procesados por lo algoritmos de detección y esta asociado al servicio Cloudfront para poder realizar stream de los vídeos hacia la parte de *frontend* de la plataforma.

El bucket de almacenamiento fue configurado con acceso publico, pero con restricciones en la política para que solo pueda ser accedido públicamente por sistemas que tengan esta política asociada como lo es la parte del *frontend*.



Nombre	Región de AWS	Acceso	Fecha de creación
[Redacted]	[Redacted]	objetos pueden ser públicos	14 Jul 2021 12:11
memoria-storage-videos	América del Sur (São Paulo) sa-east-1	⚠️ Público	7 Nov 2022 10:41
memoria-stream-videos	América del Sur (São Paulo) sa-east-1	Bucket y objetos que no son públicos	7 Nov 2022 11:42

Figura 4.1: Bucket usados en el sistema

Por otro lado el servicio de CDN Cloudfront fue habilitado y vinculado al bucket de stream para vídeos, el servicio entrega una URL con la cual se puede acceder a stream de los archivos de los vídeos presentes en el bucket para la parte *frontend*.

Cabe destacar que estos buckets fueron diseñados sabiendo que la iteración de esta parte del proyecto corresponde a la fase de *deployment* por lo que estos pueden ser cambiados por buckets con rasgos más restrictivos para poder mantener más seguridad en el programa y sin tener que realizar mayores cambios en otras partes del proyecto para poder reemplazarlos

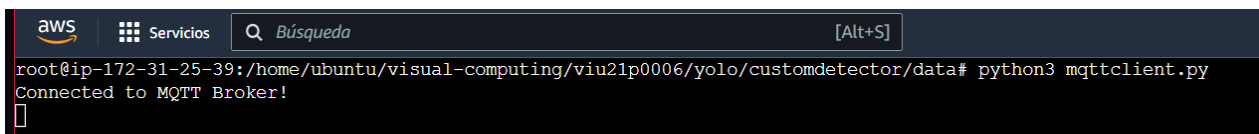
4.1.2. Trabajo en servidor EC2

El servidor de sistema EC2 usado en este proyecto cuenta con un sistema operativo Linux, además de tener Python instalado por defecto.

Dentro de este servidor se ejecutan los modelos de detección disponibles, ambos modelos están escritos en Python por lo que cuentan con el ambiente necesario para poder ejecutarse. Para poder recibir llamados de parte del sistema de *frontend*, se crea un script usando la librería Paho Mqtt para poder generar un cliente el cual esté suscrito al tópico `viu/procesoec2` y que genere un ciclo de instrucciones cada vez que reciba un mensaje. Cuando el cliente recibe un mensaje este rescata de su contenido el nombre del archivo de vídeo a procesar y el sistema de detección seleccionado, dependiendo de estos parámetros se descarga del bucket de almacenamiento el vídeo mencionado y se ejecuta un script asociado al modelo a usar entregando como parámetro el vídeo recién descargado.

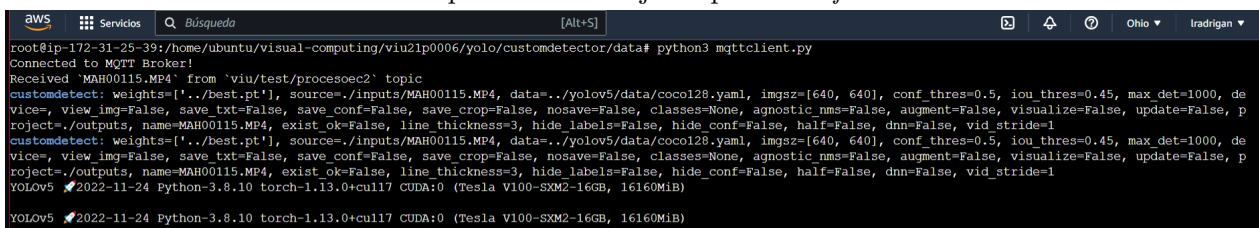
Una vez ya terminado el proceso de detección, el cliente sube el archivo de vídeo generado al bucket de stream y el archivo json al bucket de almacenamiento, a continuación publica en el tópico `viu/resultadoec2` un mensaje con el nombre del archivo de vídeo generado para que los subscriptores en la parte de *frontend* puedan trabajar con los resultados obtenidos.

Al terminar todo este proceso, se realiza una limpieza de los archivos descargados y generados para evitar tener que sobrecargar el almacenamiento del servidor.



```
aws Servicios Búsqueda [Alt+S]
root@ip-172-31-25-39:/home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data# python3 mqttclient.py
Connected to MQTT Broker!
█
```

Cliente esperando mensaje de parte del *frontend*



```
aws Servicios Búsqueda [Alt+S] Ohio Iradrián
root@ip-172-31-25-39:/home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data# python3 mqttclient.py
Connected to MQTT Broker!
Received 'MAH00115.MP4' from 'viu/test/procesoec2' topic
customdetect: weights=['./best.pt'], source=./inputs/MAH00115.MP4, data=./yolov5/data/coco128.yaml, imgsz=[640, 640], conf_thres=0.5, iou_thres=0.45, max_det=1000, de
vice=, view_img=False, save_txt=False, save_conf=False, save_crop=False, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, p
roject=./outputs, name=MAH00115.MP4, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False, vid_stride=1
customdetect: weights=['./best.pt'], source=./inputs/MAH00115.MP4, data=./yolov5/data/coco128.yaml, imgsz=[640, 640], conf_thres=0.5, iou_thres=0.45, max_det=1000, de
vice=, view_img=False, save_txt=False, save_conf=False, save_crop=False, nosave=False, classes=None, agnostic_nms=False, augment=False, visualize=False, update=False, p
roject=./outputs, name=MAH00115.MP4, exist_ok=False, line_thickness=3, hide_labels=False, hide_conf=False, half=False, dnn=False, vid_stride=1
YOLOv5 2022-11-24 Python-3.8.10 torch-1.13.0+cu117 CUDA:0 (Tesla V100-SXM2-16GB, 16160MiB)
YOLOv5 2022-11-24 Python-3.8.10 torch-1.13.0+cu117 CUDA:0 (Tesla V100-SXM2-16GB, 16160MiB)
```

Cliente procesando el archivo recibido en el mensaje

```
aws Servicios Búsqueda [Alt+S] Ohio Iradrigan
10.9ms
video 1/1 (2787/2790) /home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data/inputs/MAH00115.MP4: 384x640 1 grua_abajo, 1 carga, 1 epp_casco, 1 epp_chaleco,
10.9ms
video 1/1 (2788/2790) /home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data/inputs/MAH00115.MP4: 384x640 1 grua_abajo, 1 carga, 1 epp_casco, 1 epp_chaleco,
10.9ms
video 1/1 (2789/2790) /home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data/inputs/MAH00115.MP4: 384x640 1 grua_abajo, 1 carga, 1 epp_casco, 1 epp_chaleco,
10.9ms
video 1/1 (2789/2790) /home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data/inputs/MAH00115.MP4: 384x640 1 grua_abajo, 1 carga, 1 epp_casco, 1 epp_chaleco,
10.9ms
video 1/1 (2790/2790) /home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data/inputs/MAH00115.MP4: 384x640 1 grua_abajo, 1 carga, 1 epp_casco, 1 epp_chaleco,
10.8ms
video 1/1 (2790/2790) /home/ubuntu/visual-computing/viu21p0006/yolo/customdetector/data/inputs/MAH00115.MP4: 384x640 1 grua_abajo, 1 carga, 1 epp_casco, 1 epp_chaleco,
10.8ms
Speed: 0.3ms pre-process, 11.0ms inference, 1.2ms NMS per image at shape (1, 3, 640, 640)
Speed: 0.3ms pre-process, 11.0ms inference, 1.2ms NMS per image at shape (1, 3, 640, 640)
Results saved to outputs/MAH00115.MP4
Results saved to outputs/MAH00115.MP4
procesado video
upload de video a bucket de stream
upload detections a bucket de almacenamiento
clean inputs
datos publicados en topico: viu/test/procesoec2
outputs clean
Connected to MQTT Broker!
```

Cliente finalizando el ciclo del proceso

Figura 4.2: Ilustración de trabajo del cliente dentro del servidor

4.2. Sistema Embebido

En este proyecto se trabajó con un único sistema embebido siendo este una Raspberry Pi 4, dentro de este dispositivo se creó una lógica similar a la expuesta en la sección del servidor EC2.

Primero se crea un cliente para comunicación MQTT usando la librería Paho-Mqtt, este cliente está suscrito al tópic `viu/procesorasp`, cuando se recibe un mensaje se procede a ejecutar un script el cual se graban 30 segundos con la cámara anexa al sistema Raspberry, el archivo obtenido es subido al bucket de almacenamiento y posteriormente se publica un mensaje al tópic `viu/procesoec2` para que comience el procesamiento del vídeo en el lado del servidor.

```
pi@raspberrypi: ~/Desktop/python-video/Raspberry-code-transfer/Iotaws
File Edit Tabs Help
pi@raspberrypi:~/Desktop/python-video/Raspberry-code-transfer/Iotaws $ python3 IoTSub.py
Connected to MQTT Broker!
```

Cliente en sistema embebido esperando mensaje de parte del *frontend*

```
pi@raspberrypi: ~/Desktop/python-video/Raspberry-code-transfer/lotaws
File Edit Tabs Help
pi@raspberrypi:~/Desktop/python-video/Raspberry-code-transfer/Iotaws $ python3 IoTSub.py
Connected to MQTT Broker!
Received `test` from `viu/test/procesorasp` topic
testing
[1:04:41.499751270] [6072] INFO Camera camera_manager.cpp:293 libcamera v0.0.1+21-7c855784
[1:04:41.558097528] [6076] WARN RPI raspberrypi.cpp:1297 Mismatch between Unicam and CamHelper for embedded data usage!
[1:04:41.558957802] [6076] INFO RPI raspberrypi.cpp:1414 Registered camera /base/soc/i2c0mux/i2c@1/imx219@10 to Unicam device /dev/media4 and ISP device /dev/media1
[1:04:41.562872527] [6072] INFO Camera camera.cpp:1026 configuring streams: (0) 1920x1080-XBG R8888 (1) 640x480-YUV420
[1:04:41.563431136] [6076] INFO RPI raspberrypi.cpp:800 Sensor: /base/soc/i2c0mux/i2c@1/imx219@10 - Selected sensor format: 1920x1080-SBGG10_1X10 - Selected unicam format: 1920x1080-pBAA
Track Importing MPEG-4 AVC - Width 1920 Height 1080 FPS 120/4 SAR 0/0
AVC|H264 Import results: 293 samples (305 NALUs) - Slices: 7 I 288 P 0 B - 0 SEI - 5 IDR
0.500 secs Interleaving
vid conv
vid h264 delted
proceso en ec2
video subido
█
```

Cliente en sistema embebido finalizando el ciclo del proceso

Figura 4.3: Ilustración de trabajo del cliente dentro del sistema embebido

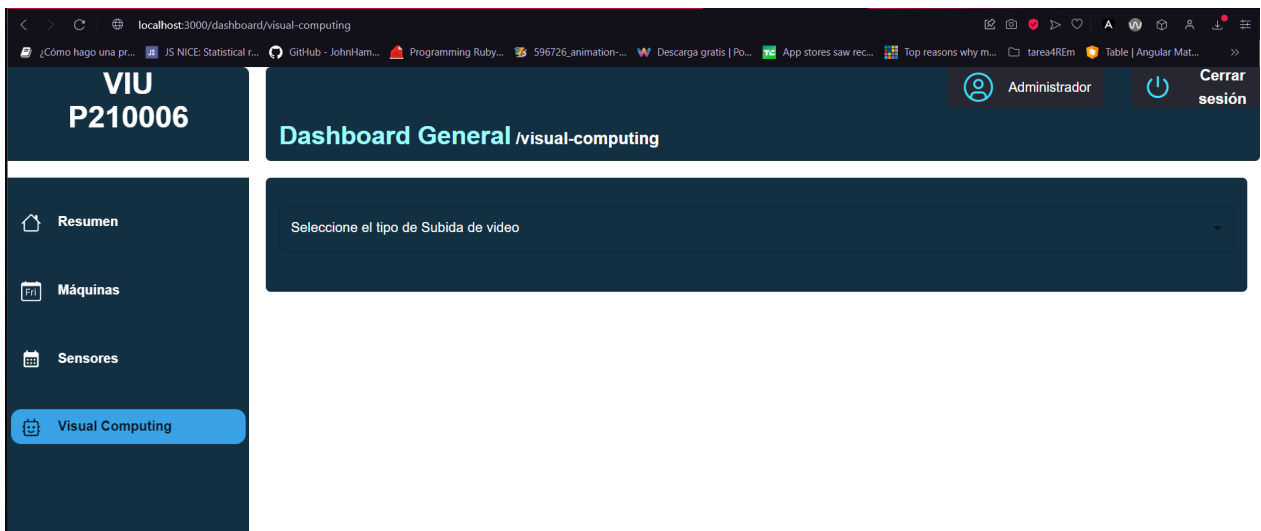
4.3. Frontend

4.3.1. Selección de inputs de archivo

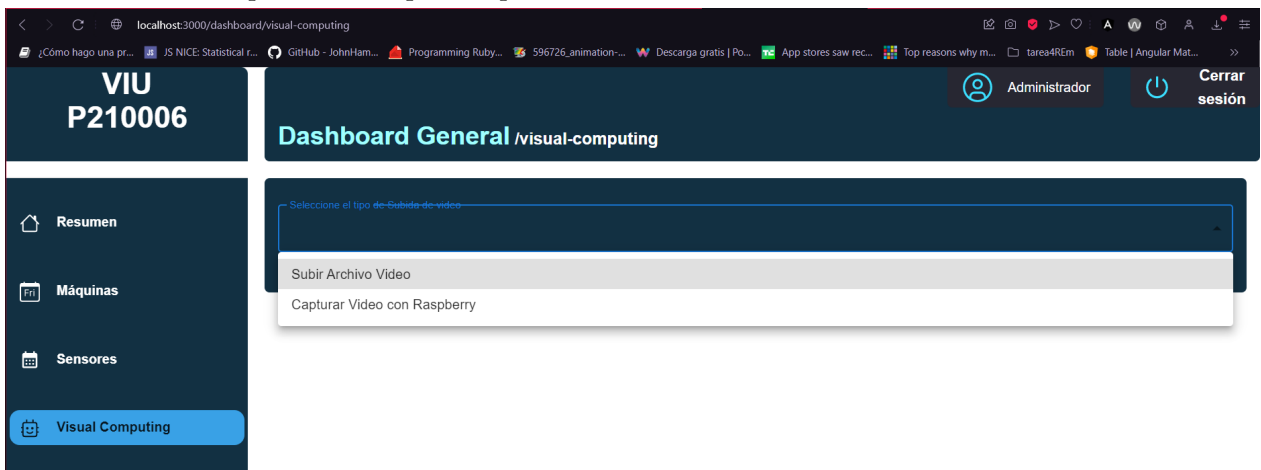
Para esta sección de la vista se hizo uso de los componentes Select, Box, Button, Stack y Dropdown presente en la librería MUI. Se creó un seleccionador del tipo Dropdown en el cual el usuario puede escoger el método de subida de vídeo.

La opción escogida por el usuario es guardada en una variable de estado de la librería React para poder ser usada como referencia en implementaciones de otras partes de la vista.

Por el momento solo se encuentra las opciones de subir archivo mediante captura de vídeo con sistema embebido de Raspberry y subir vídeo directamente desde el navegador web. Se mantiene una lógica de la vista escalable para que a futuro se puedan ir agregando más métodos de subida de vídeo al Dropdown sin tener que realizar mayores cambios a la aplicación.



Componente Dropdown para seleccionar método de subida de archivo



Opciones de subida de archivo presentes

Figura 4.4: Selección de Inputs de archivo

4.3.2. Subida de archivos hacia el servidor

Dependiendo de la opción seleccionada por el usuario para subir archivos de vídeos, se mostrarán distintas vistas dentro de la página.

4.3.2.1. Vista subida de vídeo desde el navegador

Al seleccionar la opción de subir vídeo a través del navegador, la aplicación nos muestra un nuevo componente del tipo Select Dropdown en el cual se encuentran los modelos de detección basados en inteligencia artificial disponibles, por el momento se cuenta con el modelo de detección de procesos con grúas horquillas y otro modelo de detección de equipo de protección personal. Además se añade un componente Button del tipo *input* para poder subir los archivos de vídeo.

Una vez ya seleccionado el modelo de detección a usar en conjunto al archivo a procesar, aparecerán dos componentes creados para el proyecto llamados HookMqttPub y HookMqttSub, el componente HookMqttPub recibe algunos estados ya definidos de React como parámetros además del archivo subido por el usuario, de manera similar el componente HookMqttSub recibe funciones para setear estados de la página.

HookMqttPub

Dentro del componente HookMqttPub se hace uso de la librería MQTT.js, con esta se crea un cliente para comunicación Mqtt con el rol de publicador y se definen las funciones que se disparan a la hora de ocurrir los eventos de conexión, error, desconexión y publicación. Finalmente lo que retorna este componente es otro componente creado para la aplicación llamado ConnectionPub el cual recibe como parámetros las funciones creadas para los eventos anteriormente mencionados además de distintos estados y el archivo seleccionado por el usuario.

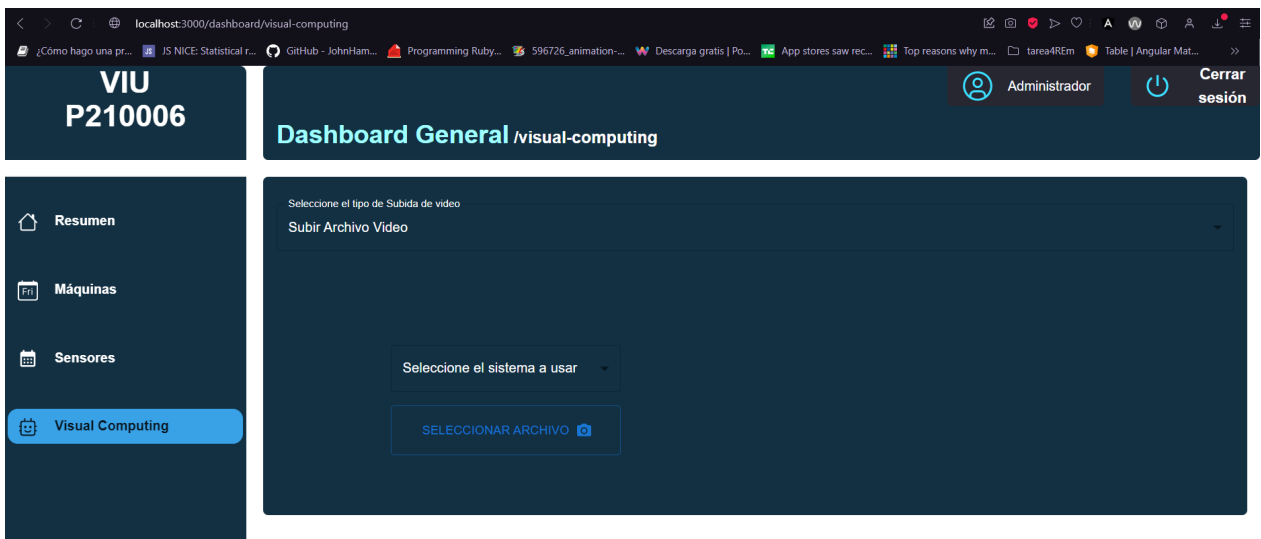
En el componente ConnectionPub se hace uso de la librería AWS-SDK para poder comunicarse con el servicio de S3 Buckets de AWS. Para poder conectarse al bucket de almacenamiento se configura el cliente AWS con el id de la llave de acceso en conjunto a la llave de acceso misma, además se entrega el nombre del bucket en el cual se subirán los archivos y la región geográfica en el cual se ubica este. Luego se configura los datos de conexión para el cliente MQTT, aquí definimos la red online broker de MQTT en el cual se mandaran los mensajes entre clientes, el tiempo de espera de conexión y el id del cliente. Finalmente se realiza la conexión a la red con las opciones definidas.

ConnectionPub retorna un componente del tipo Button el cual tiene la etiqueta de Subir Vídeo, al hacer click en este componente se dispara el evento onClick de JavaScript en el cual se sube el vídeo anteriormente seleccionado por el usuario al bucket de almacenamiento. Si esta operación es realizada con éxito entonces se realiza una publicación MQTT al tópico viu/procesoec2 entregando como contenido del mensaje el nombre del archivo subido y el modelo de detección a usar.

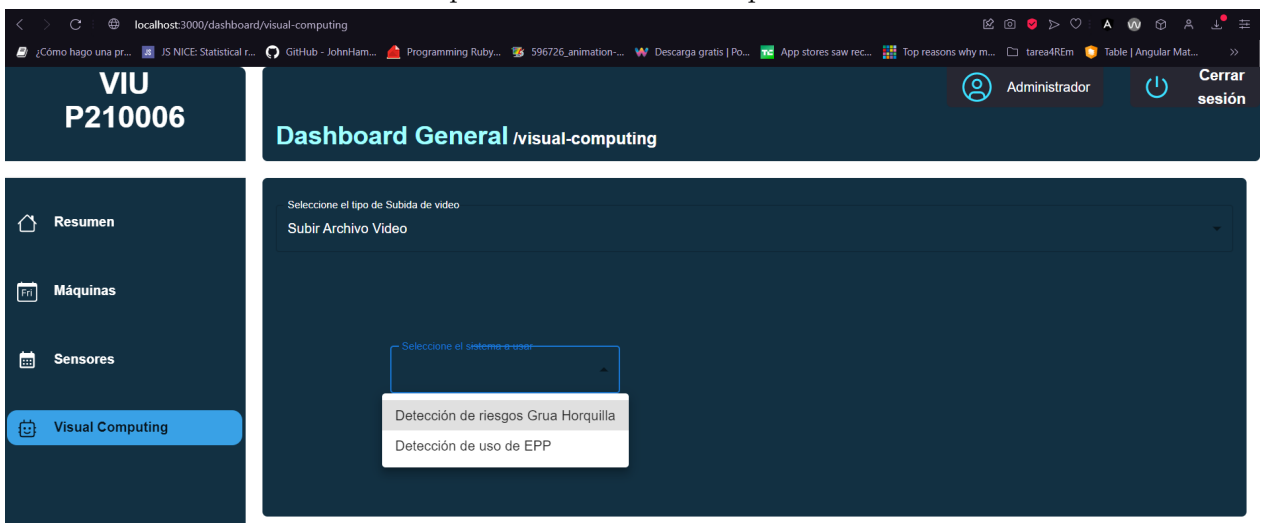
HookMqttSub

De manera similar al componente descrito en la parte anterior en este componente se trabaja con librería MQTT.js para generar un cliente de comunicación MQTT, pero esta vez tendrá el enfoque de subscriber, se definen las funciones para los eventos de conexión, error, desconexión, suscripción y anulación de suscripción. De manera análoga este componente retorna un componente del tipo ConnectionSub, pero este no retorna ningún componente, ya que solo se usara para realizar una suscripción a un cierto tópic. En este se configura los datos de conexión para el cliente MQTT y se suscribe al tópic viu/resultadoec2.

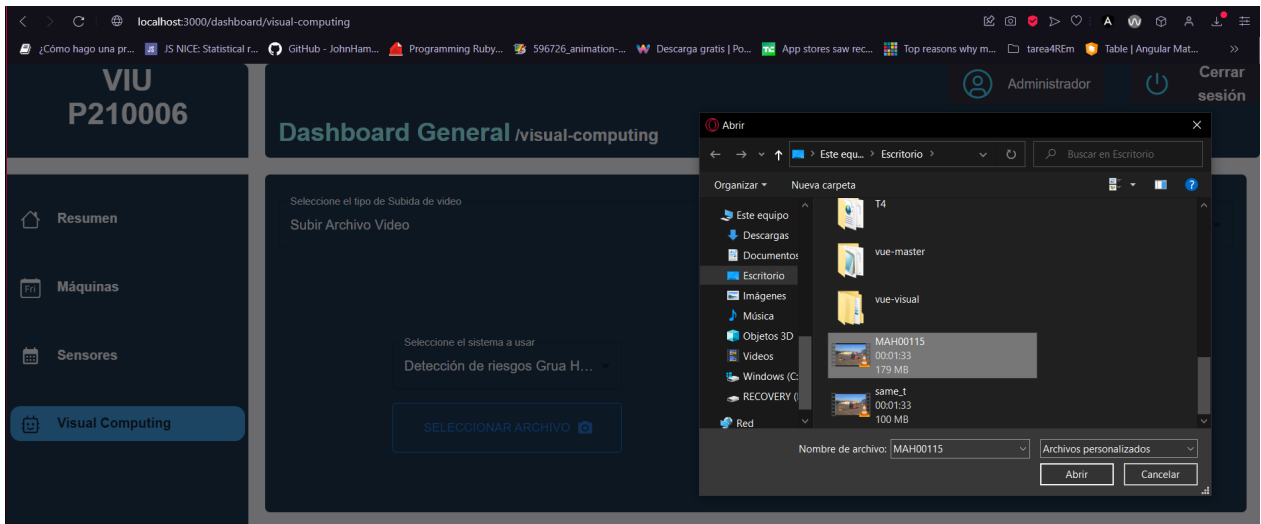
Cuando se recibe un mensaje al tópic suscrito el componente coloca valores a estados de React en el cual se almacena el nombre del vídeo procesado por el modelo de detección que se encuentra el bucket de stream para vídeo además de los kpis entregados por este modelo para ser usados en la parte gráfica.



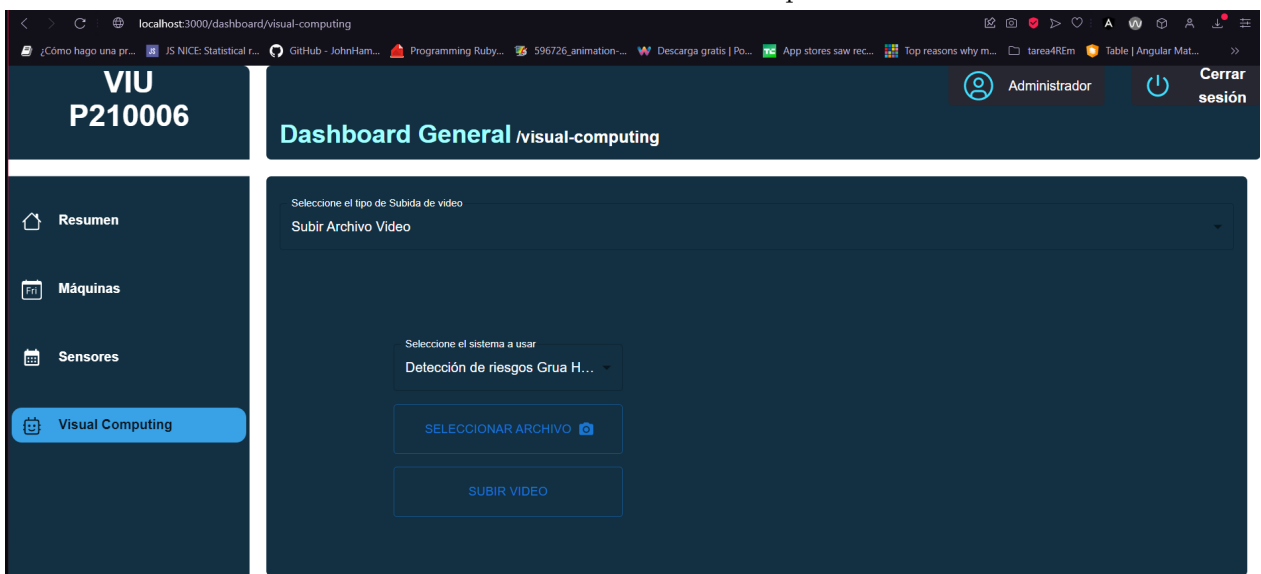
Vista para subida de archivo por usuario



Selección del modelo de detección a usar



Selección del archivo de vídeo a procesar



Vista completa previa a realizar la subida de vídeo

Figura 4.5: Vista para subida de archivo local

4.3.2.2. Vista captura de vídeo con sistema embebido

De manera similar a la vista para subir vídeo desde el navegador, la aplicación nos muestra un nuevo componente del tipo Select Dropdown en el cual se encuentran los modelos de detección en conjunto a dos componentes extra siendo estos HookMqttPubIOT y el anteriormente descrito HookMqttSub.

HookMqttPubIOT

Este componente es análogo en funcionalidad a HookMqttPub, aquí usando MQTT.js se define el cliente y las funciones a ejecutar. Finalmente se retorna el componente ConnectionPubIOT.

ConnectionPubIOT recibe que argumentos distintas funciones de seteo de estados, además del sistema embebido seleccionado, en esta iteración del proyecto solo se encuentra el

sistema de Raspberry. El componente retorna un componente del tipo Button el cual tiene la etiqueta de Capturar Vídeo, al hacer click en esta se realiza una publicación al tópico vii/test/procesorasp entregando un string vacío como payload.

Mientras se realizan todas las acciones en la parte del sistema embebido y servidor EC2, el componente HookMqttSub se encuentra esperando el mensaje de éxito para poder guardar los datos de detección.

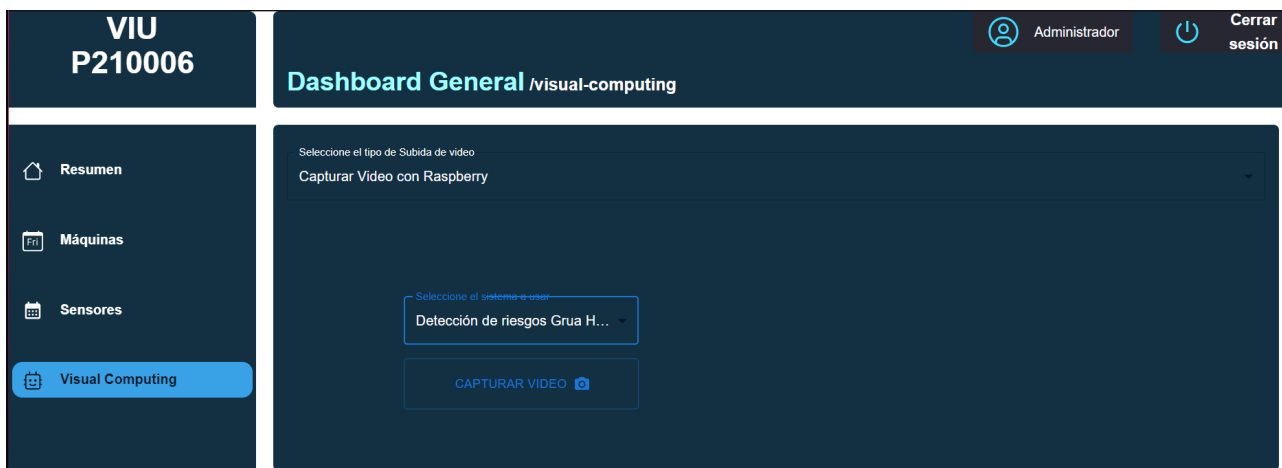
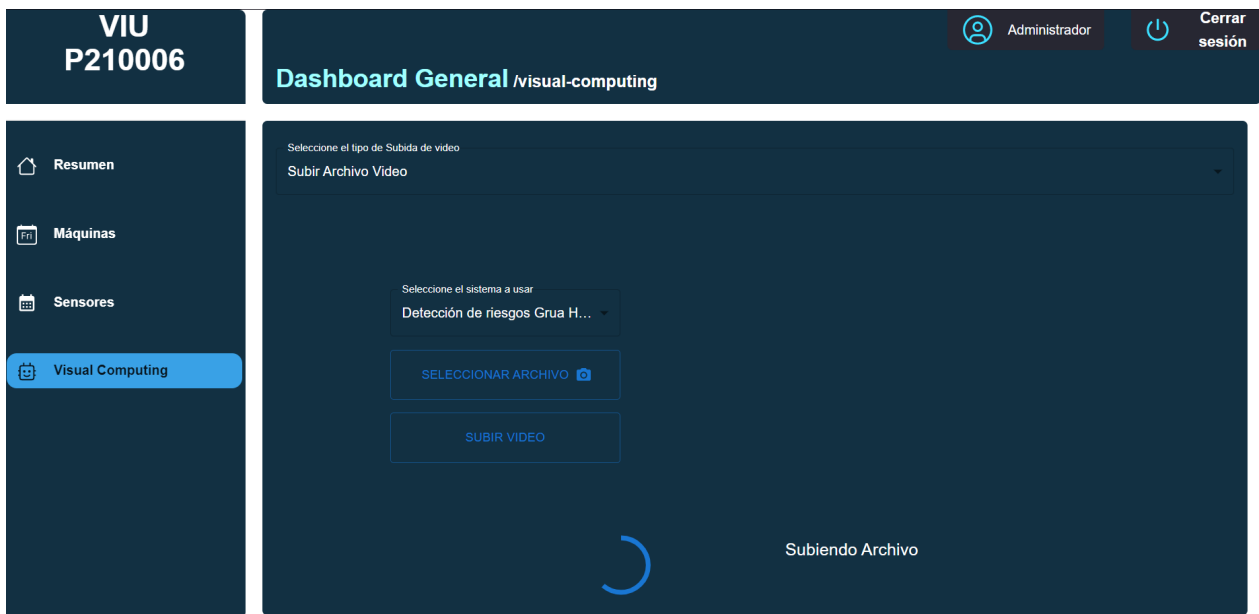


Figura 4.6: Vista para captura de vídeo por sistema embebido

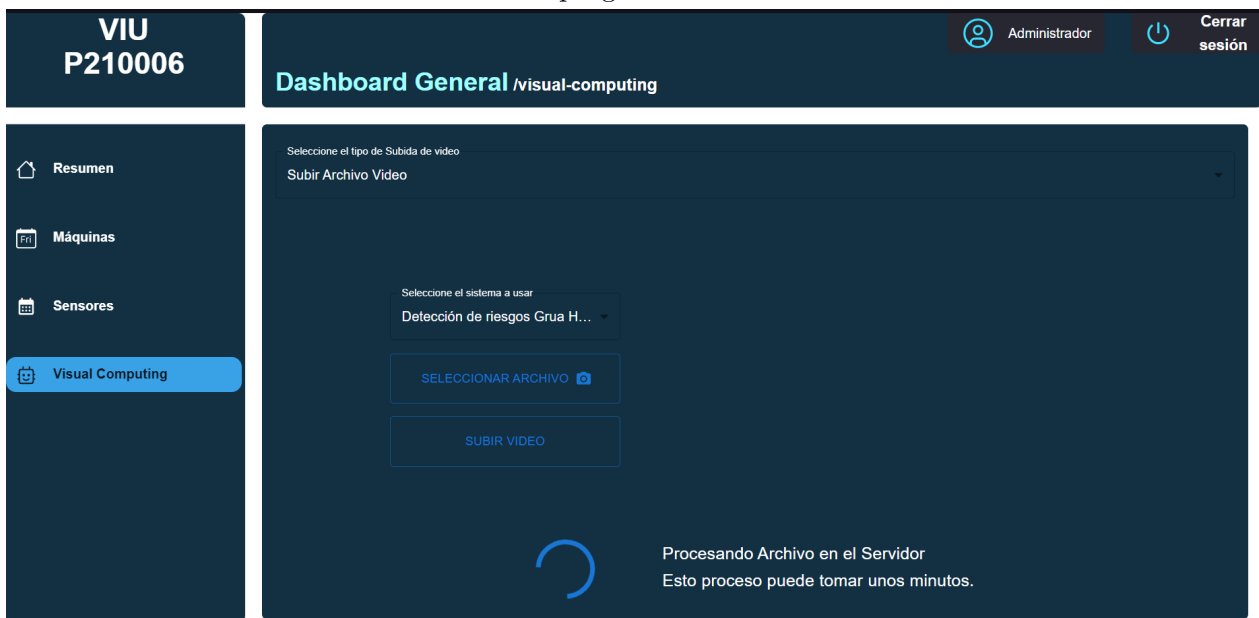
4.3.3. Visualización de los datos recibidos

Una vez se realiza la solicitud de proceso de archivo de vídeo en la parte de *frontend* del sistema, se comienza mostrar la vista de espera.

En esta vista se muestra una barra circular de progreso en la cual primero se expone el progreso de subida del archivo hacia el bucket de almacenamiento. Una vez el archivo se sube al ambiente AWS, se cambia la barra de progreso por una de progreso indefinido en la cual se muestra el mensaje de que el archivo se está procesando en el servidor EC2, dependiendo del tamaño y largo del archivo el proceso puede tomar unos minutos.



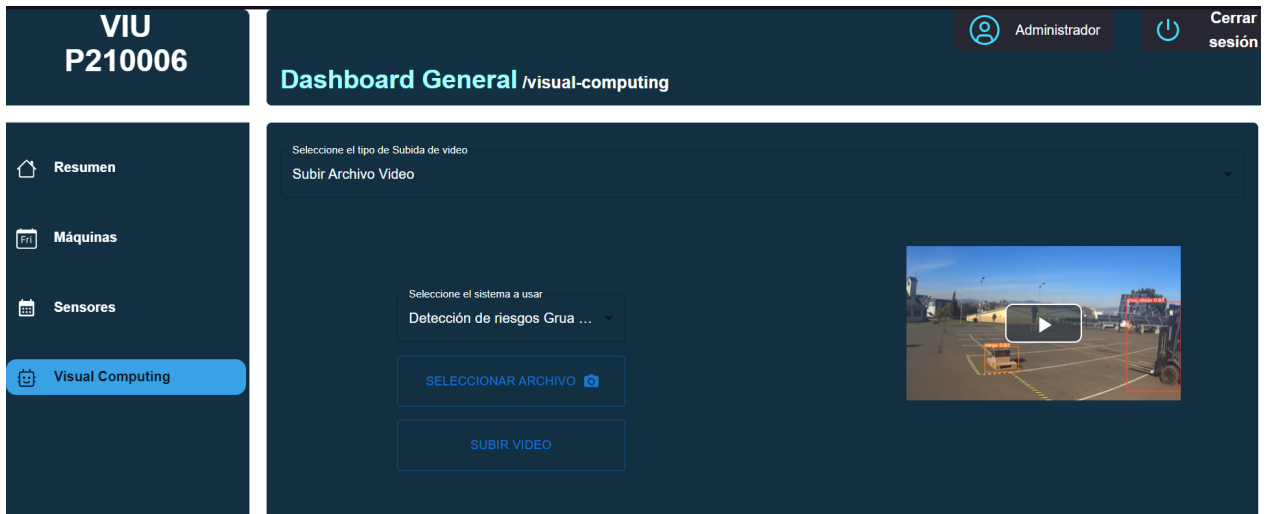
Barra indicando el progreso de la subida del vídeo



Barra de progreso indefinida indicando que el archivo está siendo procesado en el servidor

Figura 4.7: Vista de espera para el proceso de archivo

Una vez el archivo es procesado y se sube el vídeo con las detecciones a los buckets, en la parte del *frontend* se muestran los resultados. Se tiene una visualización en miniatura del vídeo con las detecciones encontradas además de un sector gráfico con la interpretación de estas detecciones.



Visualización en miniatura del vídeo con las detecciones



Vista en pantalla completa del vídeo procesado

Figura 4.8: Visualización del vídeo procesado por el sistema de detecciones

El sistema de detecciones produce además del vídeo un archivo json con datos de cada frame del vídeo, estos datos debe ser transformados para poder ser interpretados debido a que solo muestran información presente los ejes x, y, z del vídeo. Lamentablemente la función de transformación del archivo json no logro ser implementada a tiempo por la contra parte encargada del modelo de detección por lo que no se pudo implementar la sección de vista para los gráficos

Capítulo 5

Evaluación

Para evaluar el sistema implementado se tiene contemplado realizar una encuesta entre distintos usuarios que usen la aplicación, dentro de esta encuesta se evaluarán tópicos como facilidad de navegación a través de las vistas en la interfaz, calidad de la representación gráfica de los datos e indicadores establecidos por el usuario, tiempo de carga de los datos a visualizar y comentarios libres en los cuales los usuarios podrán expresar otras características que les gustaría mejorar o que no se encuentran presentes dentro del sistema.

Esta encuesta se basará en el estándar global System Usability Scale (SUS), esta encuesta se caracteriza por ser fácil de administrar los participantes y da buenos resultados para un tamaño pequeño de personas [40].

También se planea evaluar el tráfico de datos presente en todo este sistema para así poder optimizar ciertos procesos de transferencia de datos.

5.1. Encuesta

Se realizó la encuesta basada en el estándar SUS con el objetivo de poder estudiar la facilidad de uso de la aplicación en conjunto a poder obtener opiniones de los usuarios sobre que características podrían ser modificadas o añadidas a futuro.

Dentro de la encuesta cada individuo debía seleccionar cuán de acuerdo estaba con la afirmación presentada, el rango establecido fue de 1 a 5 donde 1 corresponde a Completamente en desacuerdo y 5 corresponde a Completamente de acuerdo, cada puntaje obtenido en las afirmaciones fue usado para obtener el puntaje SUS de la encuesta correspondiente, este puntaje va de un rango de 0 a 100 donde los puntajes más cercanos al máximo representan la facilidad de uso de la aplicación.

Se consideraron las siguientes afirmaciones:

- **Creo que me gustaría usar el sistema con frecuencia.**
- **Encontré el sistema innecesariamente complejo.**
- **Encontré el sistema de la página fácil de usar.**

- Creo que necesitaría el apoyo de una persona técnica para poder usar el sistema.
- Encontré que las diversas funciones en el sistema estaban bien integradas.
- Pensé que había demasiada incoherencia en el sistema usado.
- Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente.
- Encontré el sistema muy engorroso de usar.
- Me sentí muy seguro usando el sistema.
- Tuve que aprender muchas cosas antes de poder comenzar con el uso de este sistema.

Finalmente además de estas afirmaciones se añadió una última pregunta en donde se le indicaba al individuo a dejar un comentario sobre qué le pareció el sistema probado, indicando qué características podrían ser mejoradas o añadidas en el futuro, para así poder considerar un feedback más directo de los usuarios.

5.2. Trafico de datos

Se realizaron pruebas para medir el tráfico de datos en este sistema, para realizar las pruebas se usaron dos archivos de vídeo para las pruebas de detecciones, estos archivos tienen una calidad de vídeo de 720p a 30 cuadros por segundos, el primer archivo MAH00115 muestra la operación de una grúa horquilla, tiene una duración de 1 minuto con 33 segundos y tiene un peso de 179 Mega bytes. El segundo vídeo MAH00119 muestra operaciones riesgosas usando la grúa horquilla, tiene una duración de 2 minutos con 17 segundos y tiene un peso de 264 Mega bytes.

Para la evaluación se subieron y procesaron los archivos 10 veces y se tomó el promedio de tiempo en cada operación. Los resultados se encuentran presentes en la siguiente tabla.

Nombre del Archivo	MAH000115	MAH000119
Peso del archivo [Mb]	179	264
Tiempo de subida [s]	27.5	42.6
Velocidad de subida [Mbps]	6.5	6.2
Tiempo total de proceso [s]	243.1	322.6

Los tiempos de procesos totales fueron en promedio 4 minutos con 4 segundos para el vídeo MAH000115 y 5 minutos con 37 segundos para el vídeo MAH000119.

Cabe destacar que para la parte de procesamiento de datos se genera un fenómeno de cuello de botella pues el vídeo con las detecciones producido por el modelo debe ser recodificado debido a que en su forma original la codificación no es aceptada por los reproductores de

navegadores web por lo que se tiene que realizar un proceso extra.

También se realizaron pruebas subiendo los mismos vídeos de operaciones con la grúa, pero con una calidad más baja para ver como influía el peso del archivo a la hora de ser procesado. Los tiempos obtenidos no fueron muy distintos al promedio obtenido, esto se puede deber a que el modelo de detecciones analiza los vídeos según los frames presentes en estos, por lo que es más influyente la duración del vídeo sobre el peso de esta.

5.3. Análisis de los resultados

Se montó una página de testing para que los usuarios pudieran probar el sistema y posteriormente respondieran la encuesta de usabilidad del sistema. En total fueron 16 usuarios los cuales respondieron la encuesta, de esta se obtuvo un promedio de puntaje SUS de 72.03 puntos, este puntaje se encuentra en el rango de una buena percepción de la usabilidad del sistema.

Dentro de la sección de comentarios presente en la encuesta solo 5 usuarios dejaron comentarios sobre que le pareció el sistema, dentro de estos se mencionan problemas con la compatibilidad de dispositivos móviles, problemas de visualización de las opciones debido a la paleta de colores usada, falta de mensajes de errores con archivos corruptos y finalmente se sugiere añadir una pequeña descripción sobre el sistema de detección seleccionado y pequeñas descripciones sobre las acciones de los elementos al pasar el puntero sobre estos.

Con el puntaje promedio obtenido se puede interpretar que la presente implementación logra un sistema con una buena usabilidad, pero que claramente puede mejorar tomando en cuenta los comentarios de los usuarios de prueba. Los resultados de la encuesta y cálculo del puntaje se encuentra dentro de la sección B del anexo de este informe.

El tráfico de datos mostró que no hay mucho problema a la hora de subir los archivos al servidor, ya que el promedio de velocidad de subida estándar para vídeos de calidad HD (720p) se encuentra dentro del rango aproximado de 5[Mbps] a 6[Mbps] [41][42], pero si puede llegar a tomar minutos realizar el proceso total si es que se desea analizar un archivo de larga duración, sin embargo el largo de los vídeos que muestran operaciones de grúas o detección de equipamiento se encuentra en un rango duración de 10 segundos a los 2 minutos como máximo.

Finalmente se logró casi todo lo planteado del producto mínimo viable a excepción de la parte de interpretación gráfica de los kpis entregados por el sistema de detección, esto por lo explicado anteriormente en la sección de implementación.

Capítulo 6

Conclusiones y trabajo a futuro

En este capítulo se revisan los objetivos logrados, realizando una autocrítica y dando a conocer ciertas reflexiones tras lo trabajado en este proyecto. Además, se presentan perspectivas para la continuación de este proyecto.

6.1. Trabajo realizado

Finalizando las 16 semanas de trabajo en este proyecto se logró el desarrollo de una plataforma que cumple la mayoría de los objetivos propuestos, pero falla en la parte de la visualización gráfica de los datos como lo son los kpis, esto debido a un retraso a la hora de integrar una función de transformación del archivo generado por el sistema de detección para que los datos entregados puedan ser interpretados de manera correcta. Esto pues el archivo json entregado por el sistema de detección retorna coordenadas espaciales para cada frame del vídeo, por lo que no son simples de interpretar. Esto culminó en no poder llevar el sistema a un periodo de evaluación para distintos operadores potenciales.

Sin embargo se lograron gran parte de los otros objetivos planteados, uno de estos es la conexión del sistema dentro del servidor con la interfaz de *frontend* del proyecto, este a mi parecer era el objetivo más crucial, ya que así se pueden procesar los vídeos deseados, además integrar la parte faltante del proyecto se hace más fácil debido a que solo basta ejecutar la función de transformación en el servidor y mandar los datos al *frontend* a través de la conexión implementada.

Durante el trabajo se gastó mucho tiempo aprendiendo sobre las tecnologías de AWS y la comunicación MQTT, pues se tenía cierto dominio en lo que es trabajar en *frontend* con React, pero también se tenía conocimiento nulo sobre AWS y MQTT. Particularmente se gastaron 2 semanas en poder realizar una conexión MQTT exitosa entre el *frontend* y el servidor.

Sin quitarle el peso a lo mencionado anteriormente, se aprendió un montón sobre tecnologías que se usan en la industria actual, además de técnicas de programación en *frontend* y se extrajeron hartos conocimientos sobre lo que es realizar un proyecto con orientación hacia el área de IoT.

6.2. Aprendizaje del proyecto

A modo de reflexión personal y mirando en retrospectiva todo el trabajo realizado, siento que subestime el tiempo de 16 semanas para poder realizar el trabajo, hubo momentos en los cuales sentía que gastaba mucho tiempo aprendiendo o buscando como implementar cierta funcionalidad sin obtener avances significativos, esto pasó debido a que también no pedí demasiada ayuda en estos momentos cuando me sentí estancando.

Sin embargo siento que lo implementado en este proceso de memoria es un gran avance en comparación a como se recibió el proyecto en primera instancia debido a que al comienzo de la iteración, solo se tenía una vista en blanco de la sección de *visual computing* del *frontend* y no se tenía ningún avance en la parte de sistemas embebidos y dentro del servidor de procesamiento.

Si bien no se logró el objetivo completo del proyecto, de todas maneras me siento contento con lo desarrollado pues con lo poco que se tenía al inicio del proyecto se pudo desarrollar un sistema en el cual se realizan comunicaciones entre distintos dispositivos para poder brindar detecciones de riesgos o uso correcto de equipamiento para la industria, lo cual se busca estandarizar a futuro para poder reducir la cantidad de peligros laborales. Siento que lamentarse en no lograr por completo el objetivo final es un error y que en verdad el gran logro de este trabajo fue sentar las bases para la parte de la visualización computacional de este proyecto mientras se aprendía sobre las tecnologías presentes en el mundo laboral.

6.3. Trabajo a futuro

Al finalizar el desarrollo de este trabajo quedaron funcionalidades por implementar además de que durante el desarrollo de este también surgieron posibles ideas para añadir a este sistema a futuro para poder lograr una versión completa, funcional y amigable para el usuario.

Se proponen las siguientes metas a futuro para implementar en la vista de visualización computacional del sistema VIU210006:

1. Añadir la sección de la visualización gráfica para los kpis procesados en el sistema de detección
2. Realizar validación al archivo subido desde el navegador web
3. Detectar que sistemas embebidos se encuentran disponibles para la captura de vídeo
4. Realizar pruebas al sistema con múltiples peticiones de detección al mismo tiempo
5. Posibilidad de migrar el sistema al Bundle Webpack

Con respecto a la sección de visualización gráfica de kpis, debe ser implementada con un enfoque que busque lograr que sea fácil de entender para el operador del sistema, evitando gráficos o visualizaciones muy técnicas, con esto se lograría por completo el objetivo principal del proyecto.

A la hora de subir el archivo al servidor remoto de detección, la única validación presente es el componente de *input* dentro del *frontend* del sistema, a este componente se le asignó que aceptara solo archivos del tipo vídeo, pero esto no es seguro para nada, debido a que cualquier usuario malicioso fácilmente podría subir un archivo modificado para hacerlo pasar como archivo de vídeo, además se debe limitar el tamaño y largo del vídeo para evitar que el proceso de detección pueda tomar un tiempo más que considerable.

Como en este proyecto se trabaja con un solo sistema embebido, se trabajó con que a la hora de realizar llamados este siempre se encuentre conectado y funcionando, pero dentro de un entorno real de operación, los sistemas pueden fallar o simplemente no se encuentren disponibles por distintos motivos, por lo que se estima conveniente que se implemente una forma de detectar si el sistema embebido seleccionado para realizar una captura de vídeo se encuentra disponible previamente a realizar la solicitud de captura de vídeo.

También sería una gran idea probar la capacidad de respuesta del sistema implementado ante muchas consultas de detecciones, ya que un sistema como este en el entorno industrial puede recibir múltiples consultas en periodos cortos de tiempo, esto ayudaría a saber la capacidad de respuesta del sistema y serviría para colocar ciertos límites a la cantidad de consultas recibidas.

Finalmente en la última etapa del proceso se logró observar problemas con el sistema de bundle utilizado en el proyecto inicial, este es Vite uno de las tecnologías más recientes en la índole de los sistemas *frontend*, si bien esta tecnología permite realizar el set up inicial del proyecto de manera más fácil, esta tiene problemas con librerías como lo son aws-sdk la cual es primordial para poder subir archivos al sistema, debido a esto cuando se intentó montar la página de prueba para la parte de evaluación se presentaron muchas trabas debido a los problemas de compatibilidad con librerías, por lo que se decidió cambiar el sistema de bundle a Webpack, esta tecnología es el estándar para montar servicios de *frontend* que si bien es más difícil de realizar el set up inicial, una vez ya resuelto no presenta problemas mayores y deja una aplicación bien armada. Con lo anteriormente mencionado se recomienda el realizar un cambio de tecnología de bundle para así evitar los problemas anteriormente mencionados y otros problemas de compatibilidad con futuras librerías que puedan llegar a ser utilizadas en el futuro.

Bibliografía

- [1] Zhou, C., Damiano, N., Whisner, B., y Reyes, M., “Industrial internet of things (iiot) applications in underground coal mines,” *Mining Engineering*, vol. 69, no. 12, pp. 50–59, 2017.
- [2] Xu, L., Xu, E., y L, L., “Industry 4.0: State of the art and future trends,” *Int. J. Prod*, vol. 56, pp. 2941—2962, 2018.
- [3] Gackowiec, P., Podobińska-Staniec, M., Brzywczy, E., Kühnbach, C., y Özver, T., “Review of key performance indicators for process monitoring in the mining industry,” *Energies*, vol. 13, p. 5169, 2020, doi:[10.3390/en13195169](https://doi.org/10.3390/en13195169).
- [4] del Cobre, C. C., “Investigación, desarrollo e innovación en la minería chilena,” 2020.
- [5] S.A, B., “Vigilancia eficiente,” 2022, <https://www.bosch.cl/noticias-e-historias/mineria/sistema-de-monitoreo-para-rajos-abiertos/>.
- [6] Ishiyama, M., ei Kudo, S., Misawa, M., Mori, Y., Maeda, Y., Ichimasa, K., Kudo, T., Hayashi, T., Wakamura, K., Miyachi, H., Ishida, F., Itoh, H., Oda, M., y Mori, K., “Impact of the clinical use of artificial intelligence–assisted neoplasia detection for colonoscopy: a large-scale prospective, propensity score–matched study (with video),” *Gastrointestinal Endoscopy*, vol. 95, no. 1, pp. 155–163, 2022, doi:<https://doi.org/10.1016/j.gie.2021.07.022>.
- [7] Zhang, Z., Trivedi, C., y Liu, X., “Automated detection of grade-crossing-trespassing near misses based on computer vision analysis of surveillance video data,” *Safety Science*, vol. 110, pp. 276–285, 2018, doi:<https://doi.org/10.1016/j.ssci.2017.11.023>. Railway safety.
- [8] Nguyen, M. T., Truong, L. H., Tran, T. T., y Chien, C.-F., “Artificial intelligence based data processing algorithm for video surveillance to empower industry 3.5,” *Computers Industrial Engineering*, vol. 148, p. 106671, 2020, doi:<https://doi.org/10.1016/j.cie.2020.106671>.
- [9] Aqueveque, P., Radrigan, L., Pastene, F., Morales, A., y Guerra, E., “Data-driven condition monitoring of mining mobile machinery in non-stationary operations using wireless accelerometer sensor modules,” 2021, <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9324826>.
- [10] OSHA, “Safe forklift operation,” 2019.
- [11] SAP, “¿qué es sap business one?,” 2022, <https://www.google.com/search?q=sap+business+one&oq=sap+bussines&aqs=chrome.1.69i57j0i10i433i512j0i10i512l8.3067j0j4&sourceid=chrome&ie=UTF-8>.
- [12] Wikipedia, “Sap business bydesign,” 2022, https://es.wikipedia.org/wiki/SAP_Busine

ss_ByDesign.

- [13] Oracle, “Oracle enterprise resource planning (erp),” 2022, <https://www.oracle.com/cl/erp/>.
- [14] Microsoft, “¿qué es dynamics 365?,” 2022, <https://dynamics.microsoft.com/es-mx/what-is-dynamics365/>.
- [15] Triomobil, “Forklift safety, accident and collision prevention system,” 2022, <https://www.triomobil.com/en/forklift-safety-and-tracking>.
- [16] Sewio, “Forklift tracking and monitoring system,” 2022, <https://www.sewio.net/forklift-tracking-monitoring-system/>.
- [17] Viatech, “Via mobile360 forklift safety system,” 2022, <https://www.viatech.com/en/automotive/mobile360-forklift-safety-system/>.
- [18] SIERA.IA, “Forklift monitoring system,” 2022, <https://www.siera.ai/forklift-monitoring-system/>.
- [19] Git, “Git,” 2022, <https://git-scm.com/>.
- [20] Wikipedia, “Github,” 2022, <https://es.wikipedia.org/wiki/GitHub>.
- [21] Wikipedia, “Python,” 2022, <https://es.wikipedia.org/wiki/Python>.
- [22] Mozilla, “Javascript,” 2022, <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- [23] NodeJS, “About node,” 2022, <https://nodejs.org/en/about/>.
- [24] Wikipedia, “React,” 2022, <https://es.wikipedia.org/wiki/React>.
- [25] UXPin, “What is mui and what do you need to know about it?,” 2022, <https://www.uxpin.com/studio/blog/what-is-mui/#:~:text=MUI%20is%20a%20massive%20library,foundational%20and%20advanced%20UI%20elements>.
- [26] TailwindCss, “Tailwind,” 2022, <https://tailwindcss.com/>.
- [27] Hat, R., “¿qué es la api?,” 2022, <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>.
- [28] Wikipedia, “Amazon web services,” 2022, https://es.wikipedia.org/wiki/Amazon_Web_Services.
- [29] Amazon, “Amazon simple storage service,” 2022, <https://aws.amazon.com/es/s3/>.
- [30] Amazon, “Amazon elastic compute cloud,” 2022, <https://aws.amazon.com/es/ec2>.
- [31] Amazon, “Amazon cloudfront,” 2022, <https://aws.amazon.com/es/cloudfront/>.
- [32] Amazon, “Sdk de aws para javascript,” 2022, <https://aws.amazon.com/es/sdk-for-javascript/>.
- [33] MQTT.org, “Mqtt: The standard for iot messaging,” 2022, <https://mqtt.org/>.
- [34] “Mqtt.js,” <https://www.npmjs.com/package/mqtt>.
- [35] Foundation, E., “Paho mqtt,” 2022, <https://www.eclipse.org/paho/index.php?page=clients/python/index.php>.
- [36] "Wikipedia", “Sistema embebido,” 2022, https://es.wikipedia.org/wiki/Sistema_embebido.

- [37] Raspberry, “¿que es raspberry pi?,” 2022, <https://raspberrypi.cl/que-es-raspberry/>.
- [38] Mulyanto, M., Faisal, M., Prakosa, S. W., y Leu, J.-S., “Effectiveness of focal loss for minority classification in network intrusion detection systems,” *Symmetry*, vol. 13, no. 1, 2021, [doi:10.3390/sym13010004](https://doi.org/10.3390/sym13010004).
- [39] Wen, L. y Hughes, M., “Coastal wetland mapping using ensemble learning algorithms: A comparative study of bagging, boosting and stacking techniques,” *Remote Sensing*, vol. 12, no. 10, 2020, [doi:10.3390/rs12101683](https://doi.org/10.3390/rs12101683).
- [40] Brooke, J., “Sus: A quick and dirty usability scale,” *Usability Eval. Ind.*, vol. 189, 1995.
- [41] UK, A., “What is a good download and upload speed in 2022,” 2021, <https://www.airband.co.uk/what-is-a-good-download-and-upload-speed/>.
- [42] Team, R., “What is a good upload speed for livestreaming,” 2022, <https://restream.io/blog/what-is-a-good-upload-speed-for-streaming/#:~:text=Required%20upload%20speeds%20are%20between,is%204%2C500%20to%209%2C000%20Kbps>.

Anexos

Anexo A. Mock-Up de vista a implementar

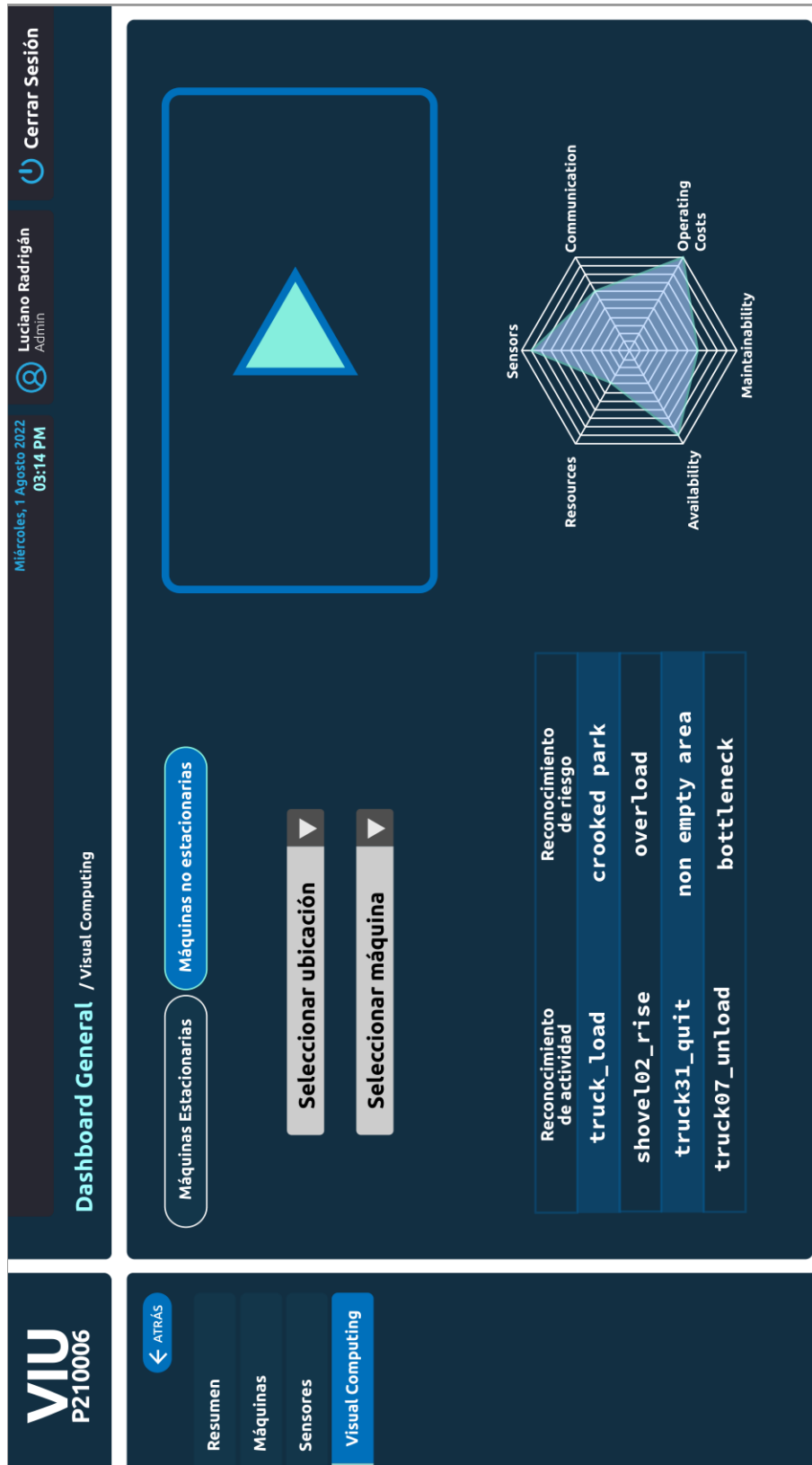


Figura A.1: Mock Up preliminar de diseño

Anexo B. Resultados de la encuesta y calculo del puntaje

Si bien la encuesta fue creada en la plataforma Google Forms, los resultados fueron obtenidos a través de una hoja de cálculo debido a que los usuarios mostraron problemas a la hora de intentar subir el formulario.

Los resultados obtenidos fueron los siguientes:

	A	B	C	D	E
5					
6	Creo que me gustaría usar el sistema con frecuencia.				
7	1	2	3	4	5
8					
9	Encontré el sistema innecesariamente complejo.				
10	1	2	3	4	5
11					
12	Encontré el sistema de la página fácil de usar.				
13	1	2	3	4	5
14					
15	Creo que necesitaría el apoyo de una persona técnica para poder usar el sistema				
16	1	2	3	4	5
17					
18	Encontré que las diversas funciones en el sistema estaban bien integradas				
19	1	2	3	4	5
20					
21	Pensé que había demasiada incoherencia en el sistema usado				
22	1	2	3	4	5
23					
24	Me imagino que la mayoría de la gente aprendería a usar este sistema muy rápidamente				
25	1	2	3	4	5
26					
27	Encontré el sistema muy engorroso de usar				
28	1	2	3	4	5
29					
30	Me sentí muy seguro usando el sistema.				
31	1	2	3	4	5

	A	B	C	D	E
30	Me sentí muy seguro usando el sistema.				
31	1	2	3	4	5
32					
33	Tuve que aprender muchas cosas antes de poder comenzar con el uso de este sistema.				
34	1	2	3	4	5
35					
36					
37	Deje algún comentario sobre que le pareció el sistema probado, indicando qué características podrían ser mejoradas o añadidas en el futuro				
38					
39					
40	C1: La página tiene problema de visualización en teléfonos celulares y				
41	Tablet.				
42	C2: La paleta de colores no ayuda a visualizar bien las opciones.				
43	C3: Que al poner el mouse sobre los botones muestre una pequeña explicación.				
	C4: Al adjuntar un archivo corrupto no entrega un mensaje de error.				
	C5: Tener la opción de leer descripción de los modelos que se aplicará, para no utilizar un modelo con un video de otra operación.				
44					

Figura B.1: Resultados de la encuesta

La presente hoja de cálculo tiene todas las respuestas de los encuestados juntas por lo que no es posible identificar las respuestas de manera individual de cada usuario, esto provoca un problema que será explicado más adelante.

Cada afirmación cuenta con los puntajes de la escala de 1 al 5 donde uno equivale a un desacuerdo total de la afirmación y 5 a un acuerdo total con lo afirmado. Cada línea bajo el puntaje corresponde a una respuesta de un usuario

Para calcular los puntajes SUS de las encuestas se deben sumar los puntajes obtenidos de las afirmaciones con enumeración impar (afirmaciones de connotación positiva) y de las afirmaciones con enumeración par (afirmaciones de connotación negativa), luego definimos la variable G como:

$$G = \sum_{i=1}^5 x_{2i-1} - 5 \quad (\text{B.1})$$

Donde x_u corresponde al puntaje de la pregunta del índice correspondiente. Posteriormente definimos la otra variable B como:

$$B = 25 - \sum_{i=1}^5 x_{2i} \quad (\text{B.2})$$

Finalmente el puntaje SUS (P) se define como:

$$P = 2.5 * (G + B) \quad (\text{B.3})$$

Dependiendo del puntaje obtenido se pueden clasificar la usabilidad del sistema en los siguientes rangos:

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
< 51	F	Awful

Figura B.2: Tabla de clasificación SUS

Para evaluar el proyecto se decidió usar el promedio de los puntajes SUS, pero debido a como se presentaron los datos dentro de la hoja de cálculo no se pueden obtener los puntajes SUS de los encuestados de manera individual y con estos obtener el promedio, por lo que se tuvo que generar una nueva fórmula para obtener el promedio de los puntajes.

De la fórmula B.3 podemos notar que la variable P se puede reducir a la siguiente forma

$$P = 2.5 * (20 + R_{imp} - R_{par}) \quad (\text{B.4})$$

Al sumar las variables G y B se obtiene un valor constante de 20 y una suma de los resultados de la sumatoria de los puntajes obtenidos de las afirmaciones con enumeración impar (R_{imp}) con el resultado de la sumatoria de los puntajes para las afirmaciones enumeración par (R_{par}) en valor negativo.

Con esta nueva forma de representar el puntaje, el promedio total se escribiría como:

$$P_{mean} = \frac{2.5}{N} * (20 * N + \sum_{u=1}^N R_{imp_u} - \sum_{u=1}^N R_{par_u}) \quad (\text{B.5})$$

Donde N corresponde al total de personas encuestadas y R_{imp_u} con R_{par_u} corresponden a los resultados de la suma de los puntajes de las afirmaciones impares y pares correspondientemente para el encuestado número u .

En la hoja de cálculo con los datos de las respuestas de la encuesta se implementó la nueva fórmula B.5 obteniendo el promedio de puntajes SUS

	Suma total de puntajes	Suma preguntas impares	Suma preguntas pares	Promedio de puntajes SUS
P1	62	298	157	72.03125
P2	31			
P3	62			
P4	46	Numero total de encuestados		
P5	58	16		
P6	28			
P7	70			
P8	27			
P9	46			
P10	25			

Figura B.3: Resultado del promedio de puntajes