



UAI
UNIVERSIDAD ADOLFO IBÁÑEZ

GRAYMATICS

Proyecto: Aumento de la Cartera de Clientes de Graymatics a través del Desarrollo de un VMS

por Kyle Foster Ferigra

Carrera: Ingeniería Civil Informática

Empresa: Graymatics

Área: Desarrollo de Aplicaciones de Escritorio

Profesor: Víctor Nivia Martínez

Fecha: Diciembre, 2023

Tabla de Contenidos

0.a	Resumen Ejecutivo.....	3
0.b	<i>Abstract</i>	4
1	Introducción.....	5
1.1	Respecto a la Empresa	5
1.2	Área de Trabajo del Estudiante	5
1.3	Brecha Encontrada	6
2	Objetivos.....	8
3	Estado del Arte	10
4	Propuestas de Solución	12
4.1	Solución Escogida	13
5	Evaluación Económica	16
6	Metodología.....	17
7	Medidas de desempeño	18
8	Desarrollo y Avance	21
9	Matrices de Riesgo y Mitigaciones	28
10	Resultados Obtenidos	29
11	Conclusiones	31
12	Referencias.....	32
13	Anexos.....	33

0.a Resumen Ejecutivo

Este documento tiene el fin de detallar el progreso del desarrollo de un VMS (*Video Management System*) para la empresa Graymatics, el cual tiene como fin completar los pasos preliminares de la inserción de la compañía en un mercado en el cual no se encuentra actualmente, permitiendo así el incremento en la cartera de clientes, la diversificación de los productos actualmente ofrecidos por Graymatics, y una mayor competitividad con otras empresas similares.

Se consideró y juzgó la viabilidad de varias opciones; un sistema control de acceso, un sistema de coacción personal, un sensor de superficie, y sistema de gestión de videos. Este último corresponde a la solución escogida y, por ende, el enfoque de este informe. La arquitectura implementada corresponde principalmente a una aplicación de escritorio desarrollada a través de Windows Forms con el lenguaje de programación C#, utilizando Visual Studio 2022. Efectivamente, los resultados obtenidos han probado ser favorables y competitivos con relación a otros VMS encontrados actualmente en el mercado. Junto a lo anterior, se realizó la prueba de un modelo de *deep learning* que será añadido al producto final, esto con el fin de permitir el reconocimiento de varios objetos dentro del VMS a través de cámaras de video conectadas a través del protocolo RTSP. Concluyentemente, el desarrollo del proyecto tuvo un desarrollo continuo y estable, y la pasantía en general ha sido una excelente experiencia para una introducción al campo profesional.

0.b Abstract

This document aims to detail the progress of the development of a VMS (Video Management System) for the company Graymatics, with the aim of completing the preliminary steps of the company's insertion into a market in which it is not currently found, thus allowing an increase in the client portfolio, the diversification of the products currently offered by Graymatics, and a greater degree of competitiveness with other similar companies.

The viability of various options were considered and measured; an access control system, a personal alarm system, a surface sensor, and video management system. The latter corresponds to the chosen solution and, therefore, the focus of this report. The implemented architecture corresponds mainly to a desktop application developed through Windows Forms with the C# programming language, using Visual Studio 2022. The results obtained have proven to be favorable and competitive in relation to other VMS products currently found on the market. Alongside this, a deep learning model in order to implement for the final product with the goal of allowing the recognition of various objects within the VMS through video cameras connected by using the RTSP protocol. Conclusively, the development of the project has had a continuous and stable development, and the internship in general has been an excellent experience for an introduction to the professional field.

1 Introducción

1.1 Respecto a la Empresa

Fundada en *Silicon Valley* el año 2011, La empresa Graymatics proporciona una plataforma de análisis cognitivo multimedia capaz de transformar imágenes y videos en datos granulares significativos. Principalmente, ofrecen soluciones de análisis de video basadas en CCTV para dar acceso a los clientes a información crítica de la infraestructura existente a través de paneles que detallan los resultados encontrados. Junto a lo anterior, provee a sus clientes con soluciones en múltiples sectores; seguridad y vigilancia, marketing digital, comercio electrónico, telecomunicaciones e *IoT (Internet of Things)*. Dentro de sus clientes actuales se incluyen empresas tales como Rolls Royce, Kentucky Fried Chicken, y Unimarc.

La misión principal de la empresa es la creación de plataformas altamente escalables para el análisis de video en base a los requerimientos del cliente, convirtiendo el contenido existente en los medios de comunicación en datos legibles y utilizables debido al prevalente potencial que estos presentan, el cual tiende a verse desaprovechado. La sede principal se encuentra en Singapur. Aun así, existen sedes en India, California y Chile.

1.2 Área de Trabajo del Estudiante

Para mantener un consistente trabajo en equipo, el ciclo de vida del desarrollo de software (SDLC por sus siglas en inglés) se emplea para dividir todas las labores que conllevan producir y entregar un software de calidad. Dicho lo anterior, se participó en el equipo de despliegue de Graymatics. La tarea principal que involucra es garantizar la compatibilidad del producto generado con el o los dispositivos del cliente. Desde el inicio de la pasantía, ha habido tres proyectos (sin incluir el proyecto de pasantía) en los cuales se ha trabajado. Uno de estos proyectos consistió en la integración (entendemos “integración” como una exitosa conexión entre los dos o más servicios usados y requeridos por el cliente) del software de monitoreo Avigilon Control Center (el cual corresponde a VMS) con las analíticas propias de Graymatics a través de un cuerpo en formato JSON que representa un ejemplo de una alarma. Esta alarma era llamada a través de un WebSocket (protocolo bidireccional que facilita la comunicación de un cliente a un servidor y viceversa en tiempo real) activo con un script creado en el lenguaje de programación/scripting Python 3.11 para generar un sonido dentro de este programa. Otro proyecto involucra una generación automática de videos conseguidos a través de un REST API creado de forma compuesta en un kit de desarrollo de software entregado por la empresa Intellicene. Todas las pruebas del software han sido generadas en el dispositivo del cliente en forma remota. Por último, se está

asistiendo en pruebas de un middleware para asegurar su funcionamiento en un dispositivo encontrado en Lucknow, India.

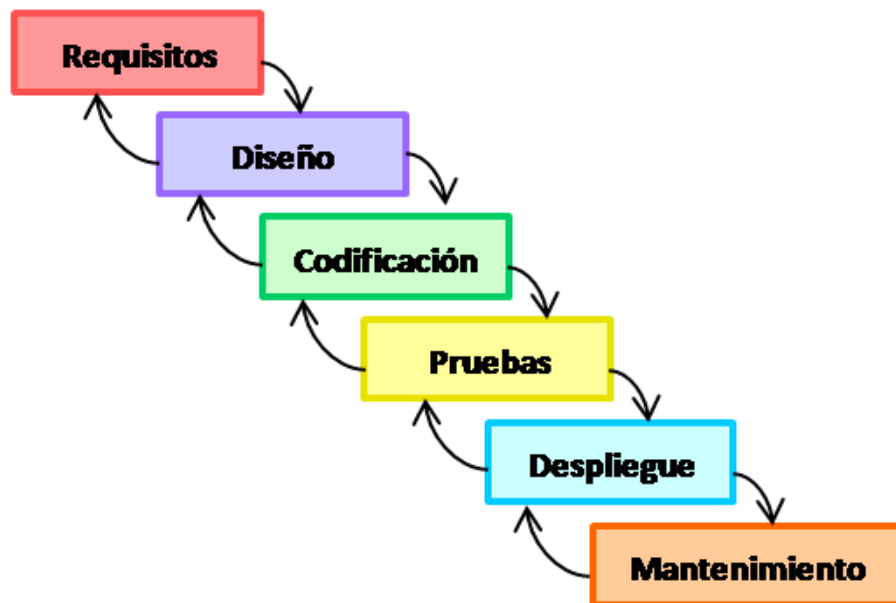


Diagrama N°1: Ciclo de Vida del Desarrollo de Software (SDLC)

1.3 Brecha Encontrada

Actualmente, Graymatics posee analíticas de video que, al integrarlas a programas pertenecientes a terceros, otorgan procesamiento, análisis y visualización a estos programas de manera que permitan la detección de varias ocurrencias en base a las analíticas previamente mencionadas. Estas “ocurrencias” detectadas varían enormemente; actualmente, por ejemplo, existen analíticas para detectar objetos perdidos, uso correcto de vestimenta de trabajo, detección de robos ocurriendo en tiempo real, entre otros. A pesar de lo anterior, empresas similares a Graymatics tales como Senstar, Hikvision, Avigilon, o Gorilla IVAR poseen servicios adicionales a los de esta empresa, los cuales serán analizados en la sección 3 de este informe (el Estado del Arte). Teniendo en mente que Graymatics busca ampliar su potencial clientela: ¿por qué no desarrollar una solución interna con el fin de introducir a la empresa en otro mercado, ampliando significativamente la potencial clientela existente y reduciendo esta brecha actual que hay entre Graymatics y sus competidores que ya poseen diversos productos a la venta? La creación de otra solución interna puede, asimismo, permitir que Graymatics se encuentre por encima de otros competidores que no tengan esta ventaja, tales como Kognition, Cognitec, e IntelliVision, entre otros.

Para cuantificar mejor esta brecha, podemos analizar el número de productos que tienen las empresas en la competencia con el fin de contextualizar el estado actual de Graymatics en relación a estas. La tabla N°1 detalla, a través de una tabla, los productos que tienen ambos Graymatics y su competencia. Los productos señalados son los mismos que se presentarán en la sección del estado del arte:

	Sistema Control de Acceso	Sistema de Coacción Personal	Sensor de Superficie	Sistema de Gestión de Videos
Senstar	si	si	si	si
Hikvision	si	si	si	si
Avigilon	si	no	no	si
Gorilla IVAR	no	no	no	si
Kognit òn	no	no	no	no
Cognitec	no	no	no	no
Intellivision	no	no	no	no
Graymat ìs	no	no	no	no

Tabla N°1: Productos actuales que poseen varias empresas, incluyendo a Graymatics

En base a lo anterior, el enfoque de este proyecto será introducir a Graymatics en un nuevo mercado con una solución de las presentadas.

2 Objetivos

Los Objetivos *SMART* son los siguientes:

S (*Specific*): El desarrollo de una solución para la empresa Graymatics, capaz de identificar varios patrones y alertar a los respectivos usuarios. Se buscará que el proyecto se acerque lo más posible a esta meta durante el transcurso del semestre. Terminado el semestre, se continuará el desarrollo del proyecto hasta alcanzar el objetivo final con el fin de ofrecer una solución completa.

M (*Measurable*): Los resultados serán medidos a través del desempeño de la solución en sí (por ejemplo, la velocidad de respuesta, porcentaje de aciertos en base al contexto de su uso, etc).

A (*Achievable*): Existen varias alternativas para crear software que tenga un impacto en la empresa. Esto se verá a mayor detalle en la sección 3 de este informe (el Estado del Arte).

R (*Relevant*): Implica alta relevancia para la empresa, considerando que ofrece un nuevo servicio para distribuir a clientes que tengan un requerimiento de este servicio. Junto a lo anterior, el término total de este proyecto colocará a Graymatics como un nuevo competidor en un área de productos en la cual actualmente no se encuentra.

T (*Time-bound*): Se estima que la duración total del desarrollo del proyecto es de 4 meses (iniciando desde septiembre).

En base a los puntos anteriores, el objetivo general del proyecto es ampliar la clientela para la empresa Graymatics con un producto que actualmente no posea, esto para reducir la brecha existente al ofrecer un producto adicional. Se reducirá la brecha a través del desarrollo de una solución para la empresa Graymatics, capaz de identificar varios patrones y alertar a los respectivos usuarios. Poseerá una función de lectura de metadatos capaz de reconocer patrones especificados en las analíticas de video. En términos sencillos, si existe una función capaz de detectar objetos perdidos a través del tiempo que estos pasan inmóviles dentro de la visión de una determinada cámara conectada al servicio, entonces la alarma se activará bajo este contexto. Graymatics tendrá un nuevo producto en un mercado en el cual no se encuentra en un plazo de 4 meses desde septiembre.

Los objetivos específicos son los siguientes:

1. La creación de funciones de lectura de metadatos.
2. Conexión y lectura de cámaras a través de direcciones RTSP (*Real Time Streaming Protocol*).
3. Pasado el semestre y terminado el proyecto, el desarrollo de este proyecto será parte del proceso de la empresa para su uso continuo.

3 Estado del Arte

Esta sección se enfoca en determinar las posibles soluciones para introducir a Graymatics en un área determinada. En base a esto, la solución escogida reflejará el mercado en el cual se situará el producto una vez que se encuentre terminado.

- Sistema Control de Acceso/Administración de entradas y salidas:

Un sistema de control de acceso permite un control granular sobre el flujo de ya sean personas o vehículos para una zona determinada. Existen distintas categorías dependiendo del uso final requerido; sistemas de proximidad, biométricos, y de reconocimiento de matrícula. Las tres categorías previamente mencionadas son las más frecuentemente utilizadas. Los sistemas de proximidad realizan un proceso de autenticación generalmente a través de tarjetas con identificación de frecuencia infrarroja (RFID) o comunicación de campo cercano (NFC, el cual es un derivado del RFID) (Kisi, 2020). Los biométricos utilizan rasgos físicos, tales como las huellas dactilares, para permitir acceso a individuos aprobados. Tiene la ventaja de no necesitar un objeto adicional que puede ser olvidado al momento del control de acceso (Bosch, 2019). Por último, la lectura de matrículas permite que, en base al vehículo que está siendo autenticado, este pueda acceder (Neural Labs, 2022).

- Sistema de Coacción Personal (PAS, por sus siglas en inglés):

El sistema de coacción personal tiene la función de aumentar el grado de seguridad de un trabajador, especialmente cuando se trata de uno que se encuentra trabajando de manera aislada y la supervisión se vuelve complicada. El sistema es generalmente de un pequeño tamaño y es equipado en los correspondientes trabajadores involucrados. Utiliza un sistema ultrasónico para habilitar la comunicación entre el usuario y un sistema de gestión de la seguridad, asimismo habilitando la ubicación con un margen de error ínfimo. Adicionalmente, estos dispositivos tienden a poseer un botón de emergencia para enviar señales por parte del usuario (Senstar, s.f.).

- Sensor de Superficie/Volumétrico:

Un sensor de superficie existe para la protección de perímetros utilizando diversos sistemas, tales como son la tecnología de microondas, infrarroja o una combinación de estos (también llamados híbridos). Mientras se encuentra activo, cualquier entidad transgresora activará una alarma en base a las propiedades físicas del intruso; la misma radiación infrarroja de un ser vivo permite la activación de esta para los sistemas infrarrojos (Hikvision, 2019). Para los sistemas de microondas, es la frecuencia de las ondas producidas las que generan la alarma (Southwest Microwave, s.f.).

- Sistema de Gestión de Videos (VMS dadas las siglas en inglés)

Tal como lo indica el nombre, los VMS (*Video Management System*) son soluciones de gestión de vídeo utilizados en el contexto de aplicaciones de video-vigilancia. Principalmente, facilitan el continuo monitoreo de una multitud de cámaras. Un VMS permite la integración con otros sistemas para mejorar la seguridad, almacenamiento de video, una sistema de búsqueda avanzado, analíticas de video aplicadas a los datos de video, entre otros (Mammoth Security, 2023).

4 Propuestas de Solución

Teniendo en cuenta la información previa del estado del arte, la viabilidad de las propuestas es la siguiente:

- Sistema Control de Acceso/Administración de entradas y salidas:

Efectivamente, Graymatics posee analíticas de video que permiten la lectura de matrículas vehiculares, validando esta como una posible solución. Adicionalmente, este mercado se encuentra en un constante y seguro aumento, medido en términos anuales. Para esta estimación, cabe definir de inmediato que se utiliza la tasa de crecimiento anual compuesta (CAGR por sus siglas en inglés). El CAGR permite, tal como su nombre lo indica, estimar el crecimiento económico respecto a los años anteriores. Se utiliza, bajo este contexto, a través de citas para ver el estado actual de un determinado mercado: “Se proyecta que el tamaño del mercado global de control de acceso crecerá de 9,9 mil millones de dólares en 2023 a 14,9 mil millones de dólares en 2028, y se espera que crezca a una tasa compuesta anual del 8,4% de 2023 a 2028.” (MarketsandMarkets, 2023). Se necesitará de la manufactura física de estos productos, incluyendo a credenciales de ingreso en caso de ser necesarias.

- Sistema de Coacción Personal (PAS, por sus siglas en inglés):

Un ejemplo de una empresa similar a Graymatics que actualmente ofrece este servicio es Senstar. Y, nuevamente, podemos aseverar que existe un crecimiento económico considerable en este mercado, indicando que es una opción viable: “El tamaño del mercado se estima que va a aumentar en un CAGR del 5% de USD 9,36 mil millones a USD 12,54 mil millones para el año 2027.” (Maximize Market Research, 2023). Cabe notar, adicionalmente, que los sistemas de coacción personales también requieren de un proceso de manufactura al ser productos físicos.

- Sensor de Superficie/Volumétrico:

Presentando requerimientos similares a las propuestas mencionadas previamente, el mercado para sensores de superficie encontrará un gran aumento económico a futuro: “Se espera que el mercado de sensores de movimiento tenga un valor de 9.300 millones de dólares para 2025, creciendo a una tasa compuesta anual del 83% durante 2019-2025.” (MarketsandMarkets, 2019)

- Sistema de Gestión de Videos (VMS dadas las siglas en inglés)

Es esencial notar que presenta un tamaño del mercado significativo, y un crecimiento superior a las demás opciones, junto al hecho que no requiere la manufactura física de productos: “El mercado mundial de software de gestión de vídeo (VMS) se valoró en 6.900 millones de dólares en 2021 y se espera que alcance los 32.680 millones de dólares en 2029, registrando una tasa compuesta anual del 21,46% durante el período previsto de 2022-2029.” (DataBridge Market Research, 2022)

4.1 Solución Escogida

Tomando en consideración todos los potenciales proyectos mencionados anteriormente, se ha escogido realizar el sistema de gestión de videos, esto se ve impulsado por varios factores:

- Los sistemas de control de acceso, sistemas de coacción personal, y los sensores volumétricos requieren de hardware y equipamiento que resulta ser dificultoso de adquirir para la empresa. Generalmente, las empresas diseñan y especifican las funcionalidades de los productos, para luego ordenar la manufactura de estos ya sea por parte de terceros o interna. En contrario a lo anterior, Graymatics no posee actualmente ni manufactureros externos ni una solución interna para la producción de hardware. Es por lo anterior que estas opciones muestran encontrarse, a momento, fuera del alcance de lo que desea lograrse con este proyecto, considerando ambas las opciones actuales y el tiempo.
- Como contraparte de lo previo, el desarrollo de un VMS permite ocupar las actuales fortalezas de la empresa a su favor, justificado al considerar que se encuentran a disposición una plétora de modelos de analíticas de video internas que pueden aplicarse a este proyecto, incluso luego de haber terminado el proyecto. Es posible entrenar y utilizar algoritmos tales como YOLO (*You Only Look Once*), SSD (*Single Shot Detector*) y *Faster R-CNN* para la efectiva detección de los objetivos o patrones especificados. A modo de ejemplo, el diagrama N°2 detalla el proceso que realiza este primer algoritmo:

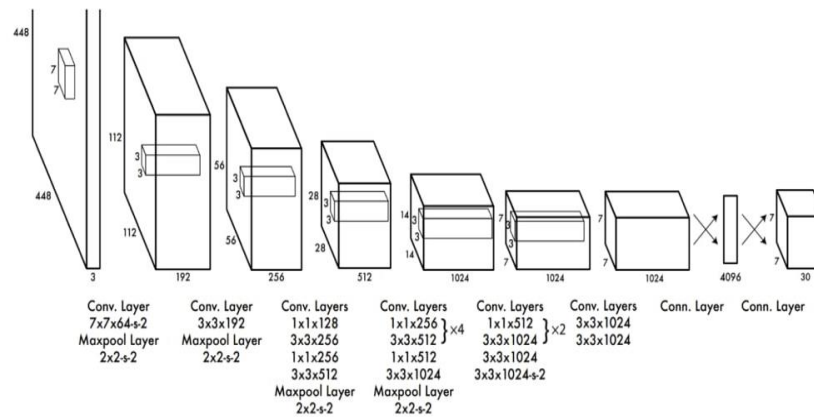


Diagrama N°2: Arquitectura del Algoritmo YOLO (Rohit, 2023)

El diagrama N°2 detalla el proceso realizado para la lectura y reconocimiento en imágenes. Equivalentemente, aplica asimismo para video en tiempo continuo. Consiste en una red neuronal convolucional, la cual en sí extiende las capacidades de un perceptrón multicapa. Tal como su nombre lo indica, contiene varias capas de filtros; la capa de entrada recibe los valores y parámetros para luego ser procesados a través de las varias capas ocultas encontradas (un total de 24 capas convolucionales seguidas de 2 capas totalmente conectadas). En las capas convolucionales, se realizan la mayor suma de los cálculos. La convolución en sí consiste en un detector de características que recorre los campos receptivos de una imagen (o cada marco individual del video) con el fin de encontrar las características especificadas. Una vez realizado lo anterior, se introduce una transformación de unidad lineal rectificada (ReLU). Lo que hace la función ReLU es transformar las características encontradas de forma binaria (0 o 1) en el mapa de características de la imagen/marco. Luego, al pasar a las capas totalmente conectadas, se realiza la clasificación utilizando como contexto las capas anteriores. En contraste con las capas convolucionales, se utilizan funciones SoftMax en vez de ReLU para clasificar los patrones encontrados e indicar un grado de confianza *fuzzy* (simplemente un número decimal) entre 0 y 1. Es esto lo que se envía a la capa de salida para recibir el resultado esperado: la clase del objeto detectado y su confianza.

- Si comparamos los valores estimados del CAGR para cada mercado existente, notaremos de inmediato que existe un valor significativamente mayor de crecimiento para los productos VMS a largo plazo en relación a las demás opciones, junto a un superior valor general del mercado.

- Otro aspecto importante a considerar es el interés que tiene la empresa en la creación de un VMS, inclusive antes de haber iniciado la pasantía. El desarrollo base (por ejemplo, el funcionamiento de inicio de sesión y conexión a vínculos RTSP para ver cámaras en tiempo real) ya se encontraba en curso, y el director ejecutivo periódicamente comenta respecto a los avances entregados. A pesar de lo anterior, existen varios aspectos por añadir y por mejorar. Si irá en detalle de estos puntos en la sección 8 (Desarrollo y Avance).

5 Evaluación Económica

Un flujo de caja detallando las posibles ganancias para la empresa a través de este proyecto se puede encontrar en la tabla N°2:

	Meses transcurridos											
	1	2	3	4	5	6	7	8	9	10	11	12
Detalle de Ingresos												
Nro. Ventas Proyectadas						15	17	21	28	36	47	61
Ingresos por Venta						5133.2	5646.5	7340.4	9542.5	12405.3	16126.9	20964.9
Ingresos Acumulados						5133.2	10779.6	18120.0	27662.5	40067.8	56194.7	72026.5
Detalles de Egresos												
Costo Infraestructura	450.0	450.0	450.0	450.0	450.0	450.0	450.0	450.0	450.0	450.0	450.0	450.0
Costo Pasante/Empleado	400.0	400.0	400.0	400.0	400.0	400.0	3750.0	3750.0	3750.0	3750.0	3750.0	3750.0
Detalle Saldo Neto	-850.0	-850.0	-850.0	-850.0	-850.0	9416.3	12226.1	21260.4	33005.1	48273.1	68121.6	88791.4
Detalle Saldo Acumulado	-850.0	-1700.0	-2550.0	-3400.0	-4250.0	5166.3	17392.4	38652.8	71657.9	119931.0	188052.6	276844.0

Tabla N°2: Flujo de Caja estimado para el proyecto

Para este flujo de caja, se utilizó como moneda el dólar de Singapur (SGD). El número de ventas proyectadas fue generado considerando que esta es una inserción inicial de Graymatics en el mercado, por lo cual se espera una clientela relativamente baja para los primeros meses. Es importante destacar que el costo estimado del producto es de 342,21 SGD (equivalente al momento a 234.289,16 CLP), esto formulado en base a los costos actuales de otros servicios de VMS existentes que tienen precios elevados.

Cabe mencionar que no existe una inversión inicial por cubrir antes de verse iniciado el proyecto. Adicionalmente, los ingresos comenzarán una vez que el proyecto se encuentre completamente terminado, y se consideran ambos el costo de la infraestructura del servidor para las analíticas de video y el costo del salario de un pasante durante la duración de la pasantía, la cual luego se ve convertida en una estimación del salario promedio posible por ganar en la empresa.

Con una tasa de descuento anual del 10%, el Valor Actual Neto del proyecto viene a ser 103.061,76 SGD (68.114.792,41 CLP) para el décimo segundo mes, indicando un beneficio para la empresa al ser un valor superior a 0.

6 Metodología

El monitoreo y método de autorreflexión para llevar a cabo el proyecto se compone, principalmente, de actualizaciones diarias a varios compañeros de trabajo y superiores de la empresa, incluyendo a la supervisora de esta pasantía. Estas actualizaciones se envían a través de correo electrónico, de manera de generar un continuo registro que se encuentra disponible para todos los involucrados. Por lo tanto, la información respecto al avance actual y por realizar se encuentra disponible de manera continua y presente, facilitando el progreso del proyecto y la transparencia.

Junto con lo anterior, se encuentra habilitado y actualmente en uso el control de versiones a través de Git, el cual habilita un continuo respaldo del código y directorios utilizados. Para cada característica añadida, se genera un nuevo branch que luego es combinado a través de un merge con la rama principal una vez que se encuentre terminada la tarea actual. Adicionalmente, se actualiza un repositorio alojado en GitHub que refleja los cambios previamente mencionados, efectivamente creando respaldos fuera del dispositivo local para una prevención de potenciales riesgos, junto a un grado superior de comunicación reflejado a lo largo de todo el proyecto.

Finalmente, cabe indicar que este trabajo se está realizando entre dos personas, ordenado a través de un plan de iteración ágil, el cual posee las tareas puntuales a realizar por cada integrante y las fechas estimadas en las cuales se deben terminar. Mis cargos específicos se encuentran señalados en la sección 8 (Desarrollo y Avance) con las fechas correspondientes para las tareas restantes. Un 80% del proyecto ha sido realizado durante el semestre, con el porcentaje restante siendo realizado durante el mes de diciembre. Este cambio se sustenta en el hecho que el desarrollo del proyecto es realizado por dos personas en total, y el compañero a cargo de una mitad de las labores tuvo que priorizar otros proyectos durante un periodo aproximado de un mes. Lo anterior causó un desfase general en el plan de iteración, causando que el proyecto de pasantía se viera pospuesto, y otros proyectos tuvieran un enfoque durante este periodo. A pesar de lo anterior, la fecha estimada del término del proyecto previo al desfase se encontraba en el mes de septiembre. Considerando esto, resulta posible la finalización del proyecto durante el mes de diciembre.

7 Medidas de desempeño

En esta sección, se expondrán los KPIs implementados, dando a conocer el buen desempeño o, alternativamente, dar cuenta de posibles falencias y potenciales áreas con espacio para mejorar. Cabe notar que, para la comparación del proyecto con las plataformas ya existentes, se utilizarán los mismos recursos de video cuando sea necesario con el fin de efectuar una comparación justa. Adicionalmente, los 3 primeros KPIs se están realizando en un programa separado al VMS en forma de prueba dado que se encuentran vinculados a las analíticas de video que serán implementadas en el VMS en su debido momento:

- Dado el hecho que contamos con algoritmos y modelos de análisis de video, podemos dar uso de esto y realizar cuenta del número de objetos correctamente reconocidos dentro de un marco individual (definido como una sola imagen encontrada en la composición del video) y calcular un KPI de la siguiente forma:

$$KPI1 = \frac{x}{N} \geq 0.75$$

Para este primer indicador, x corresponde al número de objetos correctamente reconocidos por el modelo dentro de un marco, y N corresponde al número total de objetos que pueden observarse en el mismo marco y que deben ser reconocidos por el dispositivo. Por lo tanto, deseamos que el valor sea lo más cercano a 1 (o derechamente este valor). Un objetivo inicial realista puede consistir en por lo menos un 75% de aciertos totales. Cabe notar que las plataformas existentes hacen todo lo posible para llevar este valor a 1. En base a esto, será esencial que el producto final reconozca correctamente todos los patrones de esta misma manera.

- Junto a lo anterior, el valor de confianza que describe a cada objeto reconocido corresponde a un indicador fundamental para el correcto funcionamiento del software. En este caso, la fórmula pasará a ser la siguiente:

$$KPI2 = \sum_{i=0}^N i \geq 0.8$$

Considerando la suma de todos los objetos encontrados en un marco, una confianza de 0.8 puede probar ser decente como referencia inicial. Sin embargo, es primordial que este número aumente a un rango de $\sim [0.9, 1]$ en una etapa de madurez alta del proyecto. Cabe notar que la confianza en el

reconocimiento de un objeto o patrón depende asimismo de cuán claro se encuentre en el marco actual: si, por ejemplo, el objeto se encuentra parcialmente tapado por otro que se encuentre en frente, esto afectará enormemente la confianza del análisis, simplemente ya que el modelo es previamente entrenado con los objetos completos y no parcialmente tapados. Además, usar datos de entrenamiento que incluyan a los objetos parcialmente tapados sesga los demás resultados y puede afectar negativamente el reconocimiento a gran escala. Con lo anterior de respaldo, incluso los mejores VMS en el mercado presentan niveles de confianza bajos al darse un contexto adecuado.

- Una medida que puede mejorar la experiencia del usuario es aumentar el número de marcos por segundo, o *FPS (frames per second)* del video con las analíticas de video activas. Podemos apuntar a 60 *FPS* como un valor ideal, aunque bien este valor puede verse lejos de ser alcanzado fácilmente considerando que los modelos de reconocimiento de objetos son más bien caros respecto al uso de los recursos de un dispositivo. Aun así, vamos a optar por optimizar este valor de cualquier manera posible mientras no afecte negativamente el reconocimiento de objetos:

$$KPI3 = \frac{x}{60} \geq 0.17$$

Para este caso, x corresponde al número de marcos por segundo. Un valor cercano, igual o mayor a 1 es deseable. Una meta de primera instancia es garantizar como mínimo un valor de $x = 10$, implicando un valor de KPI de aproximadamente 0,17.

- El tiempo de inicio del programa también puede ser determinado y reducido con el objetivo de mejorar la experiencia del usuario. Varios de los VMS actuales demoran significativamente en abrir; Avigilon Control Center Client, un VMS encontrado en el mercado, demora ~22,03 segundos en un promedio de 5 inicios del programa. Al momento, nuestro programa demora ~7,25 segundos bajo las mismas pruebas. Ambos se encuentran instalados en una unidad de disco duro. Es esperable que nuestro valor aumente a medida que el programa es desarrollado:

$$KPI4 = x - 7,25$$

Con nuestro valor inicial de 7,25 en consideración, la variable x en este caso corresponderá al tiempo que demora el programa en cargar durante la prueba. Queremos que KPI4 tenga el menor valor posible, idealmente negativo. Aun así, tener un valor de $KPI4 \leq 2,75$ puede considerarse como bueno.

- Otra métrica por considerar consiste en la precisión de la barra de arrastre al momento de utilizar el cursor en la línea de tiempo de un video. Es decir si, lentamente, la barra de arrastre es movida en torno a una dirección (izquierda o derecha): ¿cuál es la mínima distancia (medida en milisegundos) obtenida entre cada movimiento de la barra? Queremos minimizar este valor con el fin de permitir una búsqueda más específica de eventos que ocurren en el video:

$$KPI5 = x > 0$$

KPI5 requiere ser reducido lo más posible con el fin de permitir un movimiento granular a lo largo de la barra y, simultáneamente, ser mayor a 0 dado que esto implicaría que no hay movimiento y la barra ya no es funcional. Los VMS probados poseen, generalmente, un valor de $x = 500$.

- Cabe recalcar que los FPS pueden ser medidos sin la necesidad de utilizar las analíticas de video simultáneamente para tener una idea del rendimiento general. Podemos ver el promedio de FPS con todas las pantallas para distintos números de pantallas totales; 2x2, 3x3, 4x4 y 5x5.

$$KPI6 = \frac{x}{60} \geq 0.8$$

KPI6 resulta ser una variación de KPI3 que no posee el constante uso de analíticas de video como un factor, permitiendo que las pérdidas de FPS sean menos drásticas. Como peor caso, un FPS de 48 es esperable, el cual es observable en productos similares bajo una carga similar.

8 Desarrollo y Avance

Las especificaciones del computador de escritorio utilizado para ambos el trabajo de la empresa y el proyecto de pasantía deben ser consideradas, ya que el procesamiento de video es una tarea inherentemente intensiva para el dispositivo en término de procesamiento gráfico. Este mismo dispositivo se utiliza para realizar las pruebas de las analíticas de video. Las especificaciones son las siguientes:

- Sistema Operativo: Windows 10 IoT Enterprise LTSC 64-bit
- Procesador: AMD FX-8320E
- Memoria: 20GB DDR3
- Tarjeta gráfica: NVIDIA GeForce GTX 1060 6GB
- Disco sólido (contiene el sistema operativo): KINGSTON SA400S37120G
- Disco duro (usado para almacenamiento): Western Digital WDC WD20EZAZ-22L9GB0

Respecto a las características técnicas del programa en sí, este se verá realizado en Windows Forms, ocupando Visual Studio 2022, inherentemente con disponibilidad para sistemas operativos Windows que aún tienen soporte (Windows 10, Windows 11). La arquitectura anterior implica, adicionalmente, el uso del lenguaje C# para el desarrollo del código.

A diferencia de otros servicios de VMS, este entregará la opción de inicio de sesión a través de ya sea la nube o de forma local, facilitando flexibilidad en su uso y permitiendo una superior accesibilidad. Adicionalmente, tendrá un mecanismo de alarma externo para habilitar un mayor grado de integración de las analíticas de video que quieran implementarse, dado que es frecuente que un VMS actual tenga integraciones limitadas de otros servicios, incluyendo hasta las opciones de modelos de cámaras disponibles.

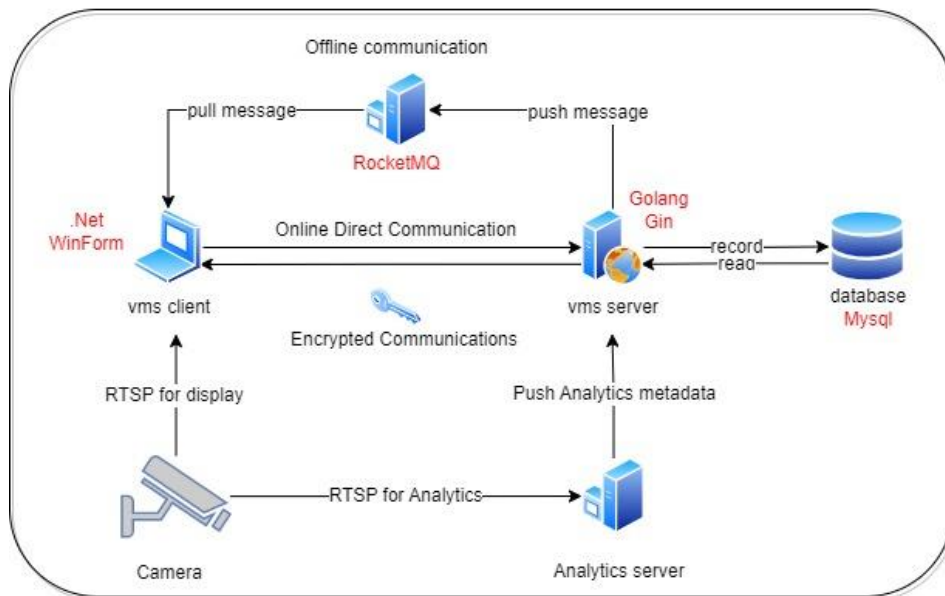


Diagrama N°3: Arquitectura completa del VMS

Considerada el diagrama N°3, el trabajo a desarrollar se encuentra en la aplicación de cliente del VMS. Tendrá un acceso directo a las direcciones RTSP de las cámaras que el usuario requiera, y estas cámaras asimismo se verán conectadas al servidor de analíticas. Para efectos de prueba, las medidas de desempeño en la sección 7 asociadas a las analíticas de video se están probando localmente en una aplicación separada. Estas mismas analíticas luego se añadirán al programa, como podrá observarse en la tabla N°3 en la siguiente página.

Para efectos de la duración de la pasantía, los componentes disponibles al usuario que tendrá el programa son los siguientes:

1. Funcionalidades ya existentes:

- Validación e Inicio de sesión local
- Validación e Inicio de sesión en la nube
- Acceso en vivo a direcciones RTSP ingresadas desde la interfaz gráfica
- Grabación de video de direcciones RTSP, con duración especificada por el usuario
- Acceso a grabaciones y reproducción de estas
- Lista de usuarios que se encuentran en el mismo servidor
- Sistema de notificación de alarmas
- Acceso al historial de todas las alarmas que han sido activadas
- Exploración de la línea de tiempo para el video
- Manejo de múltiples pantallas
- Integración de servidor de respaldo

- Prevención de acceso no-autorizado
 - Desarrollo de API abierta
 - Diseño de tablero
2. La tabla N°3 detalla las funcionalidades por añadir. Como fue previamente mencionado, cabe notar que, si bien la primera implementación al VMS de un algoritmo relacionado a las analíticas de video ocurre el 30 de noviembre, el desarrollo de reconocimiento de objetos y patrones se encuentra en proceso paralelamente a las demás tareas, explicando la existencia de KPIs asociados a estas pruebas en este informe. De este modo, es posible la prueba de estos para luego integrarlos al VMS en su debido momento:

Tarea	Inicio
Diseño de interfaz para dispositivos móviles	22-nov
Implementación de algoritmo de detección de movimiento	30-nov
Soporte para 3 marcas de cámara	06-dic
Vista en el mapa de cámaras	14-dic

Tabla N°3: Plan actual de avance al 14 de diciembre 2023

El diseño de interfaz para dispositivos móviles consiste en la creación de bosquejos de interfaces gráficas que representen el funcionamiento actual del VMS en dispositivos con el sistema operativo Android o iOS, esto para una mirada inicial de lo que viene a un futuro desarrollo multiplataforma del software. Para la implementación del algoritmo de detección de movimiento, cabe considerar que se está utilizando el paquete “Alturos.Yolo”, un wrapper de YOLO para C#, para el reconocimiento de objetos. La justificación para esta elección viene dada su madurez como paquete NuGet de C# en relación a otras alternativas. El soporte para las 3 marcas de cámaras se verá realizado a través del acceso a los correspondientes modelos a través del protocolo RTSP para asegurar el correcto funcionamiento. La vista en el mapa de las cámaras conectadas al VMS se realizará con el uso de librerías nativas de C#, como System.Device.Location.GeoCoordinateWatcher.

Un diagrama de clases de alto nivel que contiene la funcionalidad de inicio de sesión para el VMS es el siguiente:

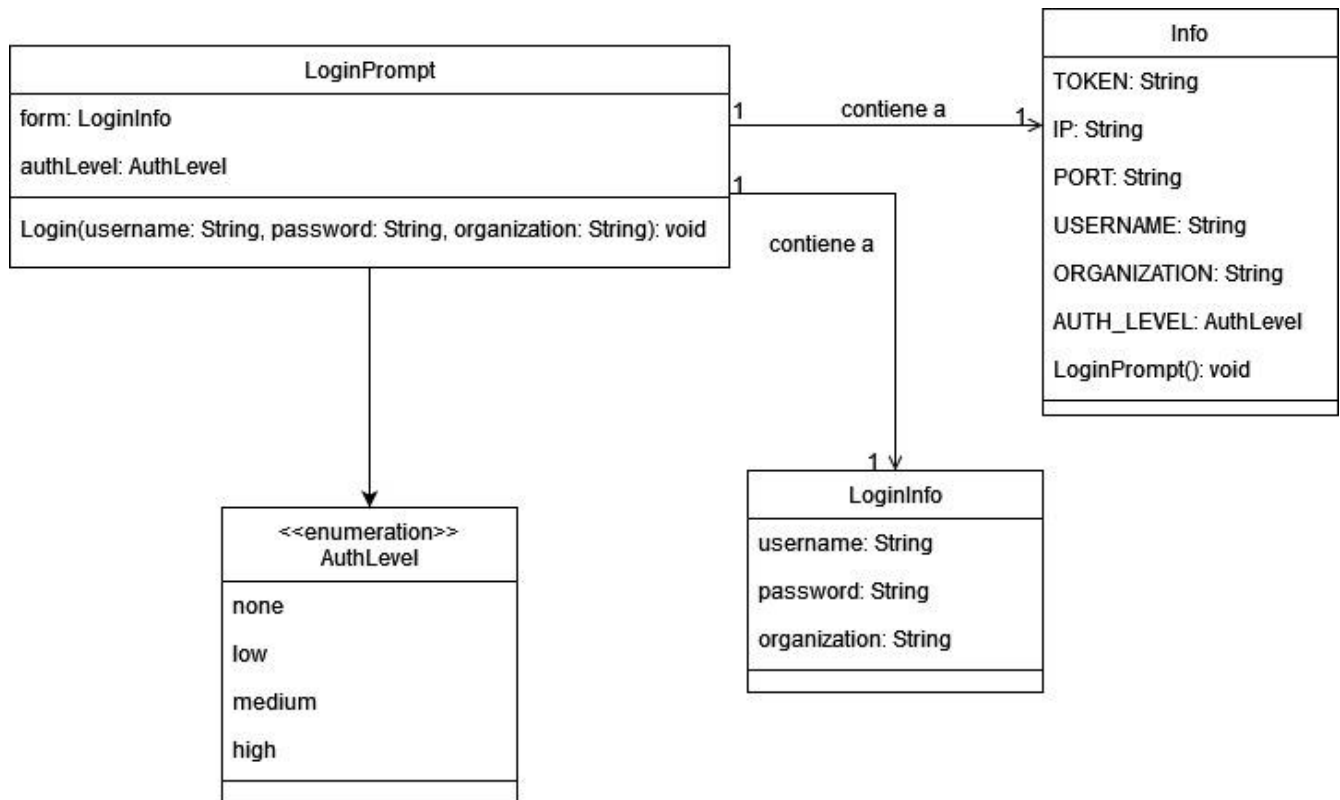


Diagrama N°7: Diagrama de Clases para el Inicio de Sesión del VMS

Cuando el usuario abra el archivo ejecutable del programa, verá la pantalla de inicio de sesión, correspondiente a un form de la clase `LoginPrompt` (véase la imagen N°1). Una clase llamada `LoginInfo` posee los atributos necesarios para el inicio de sesión, siendo estos el nombre de usuario, contraseña y organización del correspondiente usuario. Con una instancia estática de la clase `Info`, se guardan los valores ingresados al momento del inicio de sesión, con un token reemplazando a la contraseña por fines de seguridad. La enumeración `AuthLevel` dicta el nivel de acceso que tiene el usuario a las funcionalidades. Por defecto, tiene un valor bajo/`low`, prohibiendo la grabación y eliminación de grabaciones de video una vez que se haya iniciado la sesión.

Graymatics VMS

GRAYMATICS

Service IP:

106.15.251.66

Port:

5739


Organization:


graymatics

Username:

admin

Password:

 Connect to cloud

 Connect to local


 Exit

Imagen N°1: Inicio de Sesión del VMS

Una vez realizado el inicio de sesión, las funcionalidades encontradas en el VMS son las siguientes:

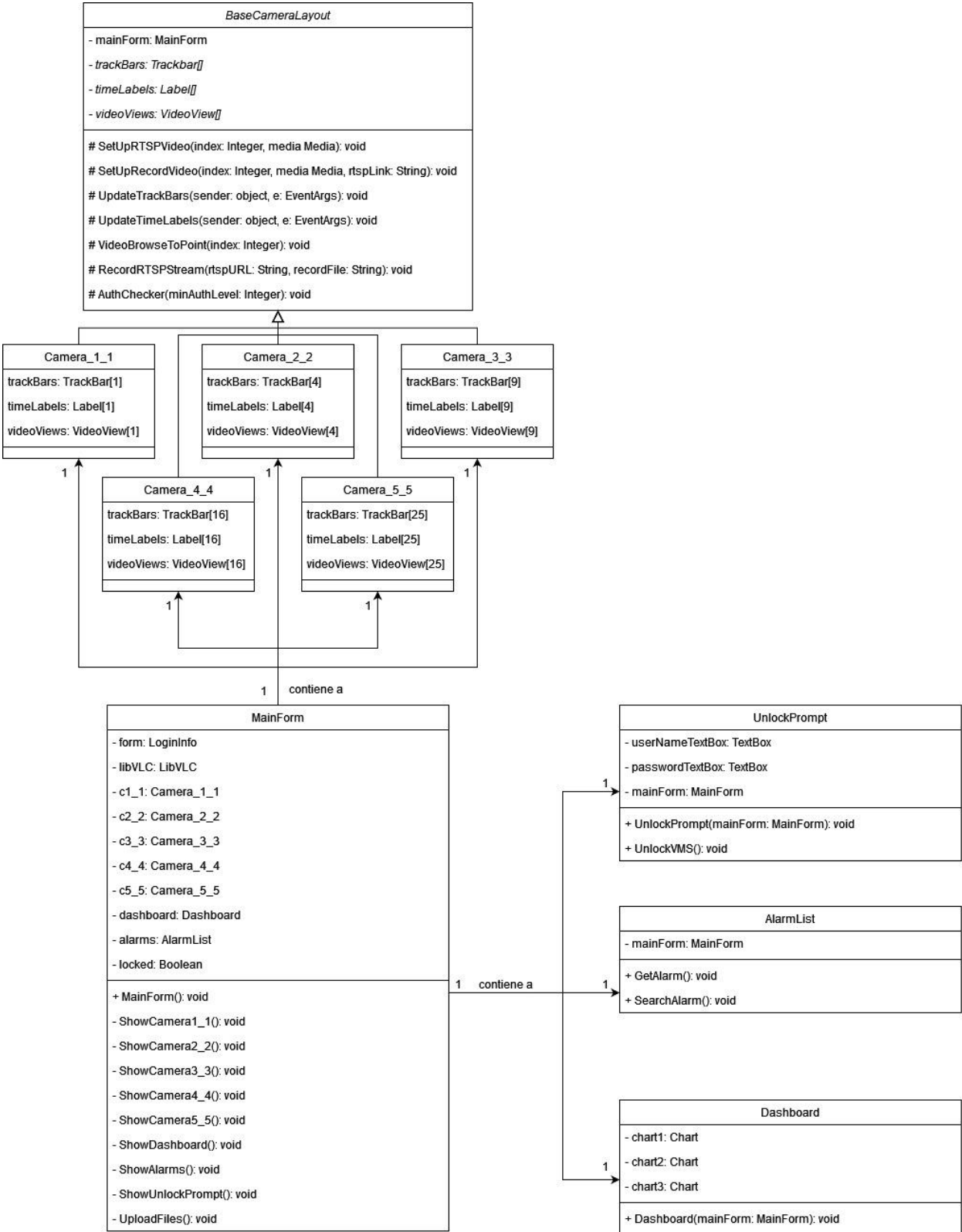


Diagrama N°5: Diagrama de Clases General del VMS

Dada el diagrama N°5, MainForm es el nombre de la clase que alberga las funcionalidades del VMS. Tenemos las vistas de video en vivo/grabaciones que se encuentran en distintas clases en base a la distribución deseada del video (heredando de una superclase abstracta, BaseCameraLayout). Adicionalmente, existe una funcionalidad de bloqueo para mantener en funcionamiento el programa y evitar un acceso no deseado, el cual solamente puede ser accedido con credenciales válidas. Por otra parte, se encuentra una lista de alarmas accesible que permite listar y buscar las alarmas añadidas, y un dashboard que indica el número de sesiones semanales o mensuales junto al espacio de almacenamiento utilizado por grabaciones en el disco y un gráfico circular para categorizar las cámaras por marca.

9 Matrices de Riesgo y Mitigaciones

Las siguientes matrices de riesgo explicitan los problemas que pueden ocurrir durante y luego de la implementación final del proyecto. Cada una posee un porcentaje de probabilidad de ocurrencia y otro de seriedad (una mayor seriedad implica una mayor interrupción del proceso):

Problema	Probabilidad	Seriedad	Mitigación
Atraso por mi parte en una determinada tarea	40%	60%	Mencionar a compañero de trabajo en forma de consulta, añadir a reporte diario para detallar la causa del problema y las soluciones intentadas hasta el momento
Atraso por mi compañero en una determinada tarea	20%	70%	Asegurar que una buena comunicación sea constante, esto para conocer su situación actual y viceversa
Fallo de la conexión al cable Ethernet	20%	80%	Debido a que el computador de escritorio utilizado no tiene una tarjeta de red para conexiones inalámbricas, comunicar esto de inmediato a través del uso de datos móviles en otro dispositivo.

Tabla N°4: Matriz de Riesgo para el desarrollo del proyecto

Problema	Probabilidad	Seriedad	Mitigación
Falta de conocimiento respecto a uso del producto	80%	20%	El programa incluirá su correspondiente documentación. Esta incluirá una introducción al programa junto con una descripción comprehensiva de todas sus funciones.
Error de conectividad a cámara	30%	60%	Si bien es bastante probable que en el producto final este error sea por parte del usuario, se incluirá en una sección "Q&A" las posibles formas de arreglar este error
Incompatibilidad del dispositivo con el software	20%	50%	Asimismo, la documentación/manual de usuario describirá los requerimientos mínimos para un funcionamiento adecuado

Tabla N°5: Matriz de Riesgo una vez realizado el proyecto. Cabe volver a recalcar que el proyecto será distribuido en un periodo posterior al término de la asignatura de pasantía

10 Resultados Obtenidos

Tomando en consideración los KPIs previamente expuestos y el estado actual del proyecto, los resultados obtenidos para efectos de la pasantía son los siguientes. Se han realizado 3 pruebas para los KPI 1 a 3 con el mismo video. Cada prueba corresponde a un marco del video; la prueba 1 fue realizada en el primer marco, la prueba 2 en el marco encontrado al medio de la duración del video, y la prueba 3 en el último marco:

	KPI1	KPI2	KPI3
Prueba 1 Inicial	0.804	0.873	11
Prueba 1 Final	0.913	0.892	10
Prueba 2 Inicial	0.857	0.93	10
Prueba 2 Final	0.928	0.935	11
Prueba 3 Inicial	0.805	0.789	10
Prueba 3 Final	0.861	0.881	10

Tabla N°6: Resultados medidos de los tres primeros KPIs, ambos en las pruebas iniciales y finales

- Las pruebas del KPI1 arrojaron, respectivamente, valores iniciales de 0,804, 0,857 y 0,805. En otras palabras, se obtuvo un valor promedio de 0,822. Esto significa que el porcentaje aproximado de objetos reconocidos correctamente en relación al número total de objetos durante las pruebas iniciales es de un 82%. Una vez que se realizaron las pruebas finales, se consiguieron valores superiores, resultando en un valor de 90%. En base a lo anterior, se ha logrado exitosamente sobrepasar el 75% propuesto inicialmente, antes de iniciar las pruebas.
- Para el KPI2, obtuvimos como valores promediados de confianza 0,873, 0,93 y 0,789 respectivamente. Lo anterior implica un promedio de 0,864, o una confianza del 86% considerando las 3 pruebas. Las pruebas finales dieron como resultado un grado promediado de confianza de un 90,2%. Nuevamente, el valor propuesto de 0,8 logró ser superado.
- Si bien este tiene un grado de importancia menor dado que se encuentra ligado al hardware utilizado, se obtuvo un FPS de, aproximadamente, 10,3 en promedio. Si bien se propuso alcanzar un valor de 15 FPS en la segunda entrega del informe, no se vieron cambios significativos.
- Las pruebas del KPI4 nos dan un valor promedio de 0,221 cuando el VMS es iniciado 10 veces. Previamente, este valor era de 0,198. En base a lo anterior, este KPI ha empeorado. Este es un proceso natural y esperable bajo el contexto del desarrollo de un programa, la adición de más clases y procesos necesarios aumentan la complejidad final del producto. Cabe notar que el aumento del KPI fue ligero al haber una diferencia de meramente 0,023, lo cual es una buena señal.

- El KPI5 tuvo, inicialmente, un valor de $x = 1000$, indicando un espacio de 1 segundo durante el desplazamiento de la línea de tiempo. Dado nuestro fin de reducir lo más posible este valor manteniendo un correcto funcionamiento, se ha logrado llegar a un valor de $x = 200$, equivalente a 0,2 segundos de diferencia.
- Respecto al KPI6, los FPS encontrados al reproducir todas las pantallas para distintos números de pantallas totales fueron favorables. La tabla N°7 lo explicita en una forma estructurada:

Cantidad de pantallas	Estándar	Resultados
2x2	60	60
3x3	60	60
4x4	56	60
5x5	48	55

Tabla N°7: FPS estándar encontrado en productos del mercado y los resultados obtenidos en el proyecto

11 Conclusiones

Desde un punto de vista concluyente, el curso de pasantía ha permitido una mirada de la industria del análisis de vídeo y oportunidades de aprendizaje que han permitido el crecimiento tanto personal como profesional; se aprendieron y perfeccionaron varias habilidades como lo son el uso de herramientas como Visual Studio y el lenguaje de programación C# en el contexto del desarrollo de aplicaciones de escritorio, junto con modelos de reconocimiento de imágenes y videos como YOLO y OpenCV. Visual Studio (y un cambio a Windows 10 para este proyecto en general) ha probado ser dificultoso de utilizar en primera instancia debido ambos a la interfaz comprehensiva y que la mayor parte de los atajos de teclado se desvían bastante del estándar. A pesar de esto, la robusta arquitectura de la herramienta resultó en una gran efectividad una vez entendido su uso general. Los resultados obtenidos con el modelo YOLO han probado ser satisfactorios, a pesar de que existen constantes innovaciones y versiones nuevas que, asimismo, implican generalmente mejoras en rendimiento al aplicar el modelo. Independientemente del proyecto de pasantía, también se adquirieron otras habilidades, tales como implementar servicios según los requerimientos del cliente, pruebas de software, trabajo en equipo para superar problemas encontrados en una manera efectiva y la relevancia del Ciclo de Vida del Desarrollo de Software al aplicarlo a proyectos reales. Por otro lado, también han tenido su papel las carreras universitarias cursadas, como lo son los cursos de Ingeniería de Software y Redes Neuronales, permitiendo, respectivamente, profundizar en la expresión de clases a través de diagramas y el uso de Redes Neuronales Convolucionales.

Por otra parte, se lograron cumplir los objetivos del proyecto, con un 80% del total del trabajo completado y el 20% restante del cronograma del proyecto proyectado a estar terminado antes de que inicie el mes de enero. A grandes rasgos, el VMS se encuentra en estado operativo; las funciones de metadatos se agregarán en su debido tiempo, las conexiones de la cámara RTSP son funcionales y permitirán a la empresa aprovechar su potencial económico y ganar ventaja sobre varias empresas rivales. Con el desarrollo de este software realizado, el desafío futuro de la empresa relacionado al proyecto consistirá en visibilizar este producto al mercado objetivo a través del marketing de una manera efectiva. Considerando que Graymatics no posee actualmente otro producto por distribuir en un nuevo mercado y la inexistencia de otros VMS internos, se utilizará el que fue presentado en este informe.

Mencionado todo lo anterior, se han abierto las puertas a un nuevo mercado para Graymatics.

12 Referencias

- Kisi (Octubre 2020). *A guide to RFID and NFC access control systems*. <https://www.getkisi.com/guides/rfid-access-control>
- Bosch (2019). *Biometric Access Control*. <https://www.boschsecurity.com/xn/en/solutions/access-control-systems/biometric-access-control/>
- Neural Labs (2022). *Neural Server Access Control*. <https://www.neurallabs.net/en/solutions/neural-server/access-control>
- Senstar (s.f.). *Personal Alarm System for Emergency Response*. <https://senstar.com/products/personal-duress-systems/pas/>
- Hikvision (Noviembre 2019). *Intelligent Awareness*. <https://www.hikvision.com/en/products/Thermal-Products/>
- Southwest Microwave (s.f.). *Microwave Sensors*. <https://southwestmicrowave.com/ssd/products/microwave-sensors/>
- Mammoth Security (Noviembre 2023). *What Is a Video Management System? Components of a VMS*. <https://mammothsecurity.com/blog/what-is-video-management-system>
- MarketsandMarkets (Marzo 2023). *Access Control Market by Offering (Hardware-Card-based, Biometric, & Multi-technology Readers, Electronic Locks, Controllers; Software; Services), AaaS (Hosted, Managed, Hybrid), Vertical and Region- Global Forecast to 2028*. <https://www.marketsandmarkets.com/Market-Reports/access-control-market-164562182.html>
- Maximize Market Research (Mayo 2022). *Personal safety alarms Market: Global Industry Analysis and Forecast (2021-2027) Trends, Statistics and Dynamics*. <https://www.maximizemarketresearch.com/market-report/personal-safety-alarms-market/148479/>
- MarketsandMarkets (Marzo 2023). *Motion Sensor Market by Motion Technology (IR, Ultrasonic, Microwave, Dual Technology, & Tomographic), Application (Consumer Electronics, Automotive, Aerospace & Defense, Healthcare, & Industrial), And Geography - Global Forecast to 2028*. <https://www.marketsandmarkets.com/Market-Reports/Motion-Sensor-Market-614.html>
- DataBridge Market Research (Agosto 2022). *Global Video Management Software (VMS) Market - Industry Trends and Forecast to 2029*. <https://www.databridgemarketresearch.com/reports/global-video-management-software-vms-market>
- Rohit, K (17 de enero, 2023) *YOLO: Algorithm for Object Detection Explained [+Examples]*. <https://www.v7labs.com/blog/yolo-object-detection>

13 Anexos

Código del Inicio de Sesión:

```
using MaterialSkin.Controls;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using RestSharp;
using System;
using System.Windows.Forms;

namespace graymatics_vms
{
    public partial class Form1 : MaterialForm
    {
        class Login_Struct
        {
            public string username;
            public string password;
            public string organization;
        }

        public enum AuthLevel
        {
            none = 0,
            low = 1,
            med = 2,
            high = 3
        }

        public Form1()
        {
            InitializeComponent();
        }

        private void login_button_exit_Click(object sender, EventArgs e)
        {
            System.Environment.Exit(0);
        }

        private void login_button_local_Click(object sender, EventArgs e)
        {
            do_login("127.0.0.1", textBox_port.Text);
        }

        private void login_button_cloud_Click(object sender, EventArgs e)
        {
            do_login(textBox1.Text, textBox_port.Text);
        }

        private void do_login(string ip, string port)
        {
            Info.IP = ip;
            Info.PORT = port;
            Info.TOKEN = login(textBox_username.Text, textBox_password.Text, textBox_organization.Text);
            if (Info.TOKEN != "" || (textBox_username.Text == "test" && textBox_password.Text == "test"))
            {
                this.Visible = false;
                Info.USERNAME = textBox_username.Text;
                Info.ORGANIZATION = textBox_organization.Text;
            }
        }
    }
}
```

```

        Info.AUTH_LEVEL = 1;
        new MainForm().Show();
    }
    else
    {
        MessageBox.Show("Account password mismatch", "Wrong account or password", buttons: MessageBoxButtons.OK,
icon: MessageBoxIcon.Exclamation);
    }
}

private string login(string username, string password, string organizatiopon)
{
    Login_Struct form = new Login_Struct();
    form.username = username;
    form.password = password;
    form.organization = organizatiopon;
    string body = JsonConvert.SerializeObject(form);
    string url = string.Format("http://{0}:{1}/user/login", Info.IP, Info.PORT);
    var client = new RestClient(url);
    var request = new RestRequest("/", Method.Post);
    request.AddHeader("Content-Type", "application/json");
    request.AddParameter("application/json", body, ParameterType.RequestBody);
    RestResponse response = client.Execute(request);
    if (!response.IsSuccessStatusCode)
    {
        return "";
    }
    JObject jbl = (JObject)JsonConvert.DeserializeObject(response.Content);
    if (jbl["token"] == null)
    {
        return "";
    }
    return jbl["token"].ToString();
}

private void Form1_Load(object sender, EventArgs e)
{
}

}

public static class Info
{
    public static string TOKEN;
    public static string IP;
    public static string PORT;
    public static string USERNAME;
    public static string ORGANIZATION;
    public static int AUTH_LEVEL = 0;
}
}

```

Código del Menú Principal:

```
using Amazon.S3;
using Amazon.S3.Model;
using LibVLCSharp.Shared;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using RestSharp;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Diagnostics;
using System.IO;
using System.Net;
using System.Threading;
using System.Windows.Forms;
using WebSocketSharp;
using MaterialSkin.Controls;

namespace graymatics_vms
{
    public partial class MainForm : MaterialForm
    {
        public LibVLC _libVLC;
        public MediaPlayer[] _mp;
        public MediaPlayer recorder;
        public rtsp_struct[] rtsp_list;
        public camera2_2 c2_2;
        public camera3_3 c3_3;
        public camera4_4 c4_4;
        public camera5_5 c5_5;
        public Web web;
        public AlarmList alarmList;
        public Dashboard dashboard;
        public Dictionary<string, string> rtsp_dic = new Dictionary<string, string>();
        public Dictionary<string, string> record_dic = new Dictionary<string, string>();
        public AmazonS3Client S3Client;
        public bool locked = false;
        public UnlockPrompt UnlockPrompt;
        public FTP_Settings ftp_settings;
        public string ftpIP;
        public string ftpUsername;
        public string ftpPassword;
        public class rtsp_struct
        {
            public string name;
            public string link;
            public string path;
            public MediaPlayer video;
            public MediaPlayer record;
        }
        public class Rtsp_Struct
        {
            public string rtsp_name;
            public string rtsp_link;
        }
        public class record_struct
        {
            public string name;
            public string path;
        }
    }
}
```

```

public MainForm()
{
    InitializeComponent();
    InitializeVariables();
    CheckForIllegalCrossThreadCalls = false;

    LoginSumBasedOnDay();
}

private void MainForm_FormClosed(object sender, FormClosedEventArgs e)
{
    if (recorder != null) recorder.Stop();
    notifyIcon1.Visible = false;
    Environment.Exit(0);
}

private void InitializeVariables()
{
    _libVLC = new LibVLC();

    c2_2 = new camera2_2(this);
    c3_3 = new camera3_3(this);
    c4_4 = new camera4_4(this);
    c5_5 = new camera5_5(this);
    web = new Web(this);
    alarmList = new AlarmList(this);
    dashboard = new Dashboard(this);

    get_rtsp();
    get_record();
    InitAWS();
}

private void InitAWS()
{
    string accessKeyId = ConfigurationManager.AppSettings["accessKeyId"];
    string secretAccessKey = ConfigurationManager.AppSettings["secretAccessKey"];
    S3Client = new AmazonS3Client(accessKeyId, secretAccessKey, Amazon.RegionEndpoint.APNortheast1);
}

public void put_s3_object(string file_path)
{
    string bucket = ConfigurationManager.AppSettings["bucket"];
    var putRequest = new PutObjectRequest
    {
        BucketName = bucket,
        Key = Path.GetFileName(file_path),
        FilePath = file_path
    };
    try
    {
        S3Client.PutObject(putRequest);
    }
    catch (Exception e)
    {
        MessageBox.Show("s3 key invalid:\n" + e.Message.ToString());
    }
}

private void show_camera2_2()
{

```

```

        c2_2.Dock = DockStyle.Fill;
        c2_2.Show();
        panel1.Controls.Clear();
        panel1.Controls.Add(c2_2);
    }
    private void show_camera3_3()
    {
        c3_3.Dock = DockStyle.Fill;
        c3_3.Show();
        panel1.Controls.Clear();
        panel1.Controls.Add(c3_3);
    }

    private void show_camera4_4()
    {
        c4_4.Dock = DockStyle.Fill;
        c4_4.Show();
        panel1.Controls.Clear();
        panel1.Controls.Add(c4_4);
    }
    private void show_camera5_5()
    {
        c5_5.Dock = DockStyle.Fill;
        c5_5.Show();
        panel1.Controls.Clear();
        panel1.Controls.Add(c5_5);
    }
    private void show_web()
    {
        web.Dock = DockStyle.Fill;
        web.Show();
        panel1.Controls.Clear();
        panel1.Controls.Add(web);
    }
    private void show_alarm()
    {
        alarmList.Dock = DockStyle.Fill;
        alarmList.get_alarm();
        alarmList.Show();
        panel1.Controls.Clear();
        panel1.Controls.Add(alarmList);
    }

    private void show_dashboard()
    {
        dashboard.Dock = DockStyle.Fill;
        dashboard.Show();
        panel1.Controls.Clear();
        panel1.Controls.Add(dashboard);
    }

    private void MainForm_Load(object sender, EventArgs e)
    {
        treeView1.ExpandAll();
        show_camera2_2();
        toolStripStatusLabel2.Alignment = ToolStripItemAlignment.Right;
        toolStripStatusLabel1.Text += Info.USERNAME;
        toolStripStatusLabel1.Text += "   Authorization Level: " + Info.AUTH_LEVEL;
        notifyIcon1.BalloonTipTitle = "Graymatics VMS";
        Thread t = new Thread(() => ws_start());
        t.Start();
    }

```

```

        treeView1.BeforeExpand += checkForLock;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        notifyIcon1.ShowBalloonTip(50, "Tip", "test", ToolTipIcon.Info);
        notifyIcon1.ShowBalloonTip(50, "Tip", "test2", ToolTipIcon.Info);
    }

    private void notifyIcon1_MouseDoubleClick(object sender, MouseEventArgs e)
    {
        this.Show();
        if (this.WindowState == FormWindowState.Minimized)
        {
            this.WindowState = FormWindowState.Normal;
        }
    }

    private void MainForm_SizeChanged(object sender, EventArgs e)
    {
        if (WindowState == FormWindowState.Minimized)
        {
            this.Hide();
            this.notifyIcon1.Visible = true;
            notifyIcon1.ShowBalloonTip(500, "Info", "Working in background", ToolTipIcon.Info);
        }
    }

    private void treeView1_ItemDrag(object sender, ItemDragEventArgs e)
    {
        if (locked && UnlockPrompt != null) UnlockPrompt.Focus();
        else
        {
            TreeNode node = (TreeNode)e.Item;
            treeView1.DoDragDrop(node, DragDropEffects.Move);
        }
    }

    private void treeView1_DragEnter(object sender, DragEventArgs e)
    {
        e.Effect = DragDropEffects.Move;
    }

    class WS_Struct
    {
        public string username;
        public string organization;
    }

    private void ws_start()
    {
        Console.WriteLine("start");
        string ws_url = string.Format("ws://{0}:{1}/ws/alarm", Info.IP, Info.PORT);
        using (var ws = new WebSocket(ws_url))
        {
            ws.OnMessage += (sender, e) =>
            ws_msg_handler(e.Data);
            ws.Connect();

            WS_Struct loginStruct = new WS_Struct();
            loginStruct.username = Info.USERNAME;
            loginStruct.organization = Info.ORGANIZATION;
        }
    }

```

```

        string ws_init = JsonConvert.SerializeObject(loginStruct, Formatting.Indented);
        Console.WriteLine(ws_init);
        if (ws.ReadyState == WebSocketState.Open)
        {
            ws.Send(System.Text.Encoding.Default.GetBytes(ws_init));
        }
        while (true);
    }
}

private void ws_msg_handler(string message)
{
    Console.WriteLine(message);
    if (message == null)
    {
        return;
    }
    if (message == "\"Alarm websocket connection established\\n\"")
    {
        toolStripStatusLabel2.Text += "Alarm connection established";
        return;
    }
    if (alarmList.Visible)
    {
        alarmList.get_alarm();
    }
    JObject jb1 = (JObject)JsonConvert.DeserializeObject(message);
    if (jb1["alarm_name"] != null && jb1["description"] != null)
    {
        notifyIcon1.ShowBalloonTip(500, jb1["alarm_name"].ToString(), jb1["description"].ToString(), ToolTipIcon.Info);
    }
}

public void get_rtsp()
{
    string url = string.Format("http://{0}:{1}/rtsp/get", Info.IP, Info.PORT);
    var client = new RestClient(url);
    var request = new RestRequest("/", Method.Get);
    request.AddHeader("Authorization", "Bearer " + Info.TOKEN);
    RestResponse response = client.Execute(request);
    if (!response.IsSuccessStatusCode)
    {
        return;
    }
    JArray ja = (JArray)JsonConvert.DeserializeObject(response.Content);
    TreeNode server_node = treeView1.Nodes[0];
    TreeNode rtsp_node = server_node.Nodes[1];
    rtsp_node.Nodes.Clear();
    rtsp_dic.Clear();
    foreach (JObject jb in ja)
    {
        rtsp_dic.Add(jb["rtsp_name"].ToString(), jb["rtsp_link"].ToString());
        rtsp_node.Nodes.Add(new TreeNode(jb["rtsp_name"].ToString()));
    };
}

private void treeView1_DoubleClick(object sender, EventArgs e)
{
    if (locked && UnlockPrompt != null) UnlockPrompt.Focus();

    else
    {
        switch (treeView1.SelectedNode.Text)
        {

```

```

        case "Server": show_camera2_2(); break;
        case "Webpages": show_web(); break;
        case "Alarm": show_alarm(); break;
        case "Dashboard": show_dashboard(); break;
        default: break;
    }
}
}
private MediaPlayer RecordRTSPStream(string rtspURL, string recordFile)
{
    //Start recording using LibVLCSharp
    var mediaOptions = new[]
    {
        ":sout=#transcode{vcodec=h264,acodec=mpga,ab=128,channels=2,samplerate=44100}:duplicate{dst=file{dst=" +
recordFile + "}}"
    };
    var media = new Media(_libVLC, rtspURL, FromType.FromLocation, mediaOptions);
    recorder = new MediaPlayer(_libVLC);
    recorder.Play(media);
    return recorder;
}

private void treeView1_NodeMouseClick(object sender, TreeNodeMouseClickEventArgs e)
{
    if (locked && UnlockPrompt != null) UnlockPrompt.Focus();

    else if (isUnlocked())
    {
        if (e.Button == MouseButtons.Right)
        {
            treeView1.SelectedNode = e.Node;
            if (e.Node.Text == "RTSP input")
            {
                contextMenuStrip2.Show(treeView1, e.Location);
            }
            else if (e.Node.Parent != null && e.Node.Parent.Text == "RTSP input")
            {
                contextMenuStrip3.Show(treeView1, e.Location);
            }
            else if (e.Node.Text == "Alarm")
            {
                contextMenuStrip4.Show(treeView1, e.Location);
            }
        }
        else if (e.Button == MouseButtons.Left)
        {
            if (e.Node.Text == "2*2") show_camera2_2();
            else if (e.Node.Text == "3*3") show_camera3_3();
            else if (e.Node.Text == "4*4") show_camera4_4();
            else if (e.Node.Text == "5*5") show_camera5_5();
        }
    }
}

private void playToolStripMenuItem_Click(object sender, EventArgs e)
{
}

private void pauseToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (rtsp_struct rs in rtsp_list)

```



```

    {
        if (rs != null && rs.name != "" && rs.video != null && treeView1.SelectedNode.Text == rs.name)
        {
            rs.video.Pause();
            return;
        }
    }
}

private void startRecordingToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (rtsp_struct rs in rtsp_list)
    {
        if (rs != null && rs.name != "" && rs.link != "" && treeView1.SelectedNode.Text == rs.name)
        {
            var folder_name = string.Format("record/{0}", rs.name);
            Directory.CreateDirectory(folder_name);
            rs.path = folder_name + "/" + DateTime.Now.ToString("yyyy-MM-dd-HH-mm") + ".mp4";
            var record = RecordRTSPStream(rs.link, rs.path);
            rs.record = record;
            MessageBox.Show("Recording start");
        }
    }
}

public void get_record()
{
    record_dic.Clear();
    var camera_list = Directory.GetDirectories("record");
    TreeNode record_node = treeView1.Nodes[1];
    record_node.Nodes.Clear();
    foreach (string camera_path in camera_list)
    {
        var camear_name = Path.GetFileName(camera_path);
        var camera_node = record_node.Nodes.Add(new TreeNode(camear_name));
        var video_list = Directory.GetFiles(camera_path);
        foreach (string video_path in video_list)
        {
            var video_name = Path.GetFileName(video_path);
            record_node.Nodes[camera_node].Nodes.Add(new TreeNode(video_name));
            record_dic.Add(video_name, video_path);
        }
    }
}

private void stopRecordingToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (rtsp_struct rs in rtsp_list)
    {
        if (rs != null && rs.name != "" && rs.record != null && treeView1.SelectedNode.Text == rs.name)
        {
            rs.record.Stop();
            Random random = new Random();
            int randomNumber = random.Next();
            string key = DateTime.Now.ToString("yyyy-MM-dd-HH-mm") + "_" + randomNumber.ToString() + ".mp4";
            upload_file(key, rs.path);
            get_record();
        }
    }
}

private void toolStripMenuItem1_Click(object sender, EventArgs e)
{

```

```

        Add_RTSP add_RTSP = new Add_RTSP(this);
        add_RTSP.Show();
    }

    private void toolStripMenuItem2_Click(object sender, EventArgs e)
    {
        delete_rtsp(treeView1.SelectedNode.Text);
    }

    public void delete_rtsp(string name)
    {
        Rtsp_Struct rtsp = new Rtsp_Struct();
        rtsp.rtsp_name = name;
        string body = JsonConvert.SerializeObject(rtsp);
        string url = string.Format("http://{0}:{1}/rtsp/delete", Info.IP, Info.PORT);
        var client = new RestClient(url);
        var request = new RestRequest("/", Method.Post);
        request.AddHeader("Authorization", "Bearer " + Info.TOKEN);
        request.AddParameter("application/json", body, ParameterType.RequestBody);
        RestResponse response = client.Execute(request);
        if (response.IsSuccessStatusCode)
        {
            get_rtsp();
        }
    }

    private void toolStripMenuItem3_Click(object sender, EventArgs e)
    {
        if (locked && UnlockPrompt != null) UnlockPrompt.Focus();
        else
        {
            Add_Alarm a = new Add_Alarm(alarmList, this);
            a.Show();
        }
    }

    public string exec_command(string cmd)
    {
        Process process = new Process();

        process.StartInfo.FileName = "cmd.exe";
        process.StartInfo.Arguments = "/c " + cmd;
        process.StartInfo.UseShellExecute = false;
        process.StartInfo.RedirectStandardOutput = true;
        process.Start();

        string output = process.StandardOutput.ReadToEnd();
        process.WaitForExit();
        return output;
    }

    public string upload_file(string key, string file_dir)
    {
        string cmd = string.Format("aws s3api put-object --bucket graymatics --key {1} --body {2} --profile adminuser --endpoint https://s3.ap-southeast-1.lyvecloud.seagate.com",
            key, file_dir);
        return exec_command(cmd);
    }

    private void contextMenuStrip1_Opening(object sender, System.ComponentModel.CancelEventArgs e)
    {
    }

```

```

string host = "ftp://172.17.48.1";
string UserId = "";
string Password = "";

public bool CreateFolder()
{
    string path = "/Index";
    bool IsCreated = true;
    try
    {
        WebRequest request = WebRequest.Create(host + path);
        if(!DoesFtpDirectoryExist(host + path))
        {
            request.Method = WebRequestMethods.Ftp.MakeDirectory;
            request.Credentials = new NetworkCredential(UserId, Password);
            using (var resp = (FtpWebResponse)request.GetResponse())
            {
                Debug.WriteLine(resp.StatusCode);
            }
        }
    }
    catch (Exception ex)
    {
        IsCreated = false;
    }
    return IsCreated;
}

public bool DoesFtpDirectoryExist(string dirPath)
{
    bool isexist = false;

    try
    {
        FtpWebRequest request = (FtpWebRequest)WebRequest.Create(dirPath);
        request.Credentials = new NetworkCredential(UserId, Password);
        request.Method = WebRequestMethods.Ftp.ListDirectory;
        using (FtpWebResponse response = (FtpWebResponse)request.GetResponse())
        {
            isexist = true;
        }
    }
    catch (WebException ex)
    {
        if (ex.Response != null)
        {
            FtpWebResponse response = (FtpWebResponse)ex.Response;
            if (response.StatusCode == FtpStatusCode.ActionNotTakenFileUnavailable)
            {
                return false;
            }
        }
    }
    return isexist;
}

private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{

```

```

var cameraList = Directory.GetDirectories("record");
foreach (string cameraPath in cameraList)
{
    string destPath = Path.Combine(Application.UserAppDataPath, cameraPath);
    Directory.CreateDirectory(destPath);

    var lastFolder = new DirectoryInfo(cameraPath).Name;

    var videoList = Directory.GetFiles(cameraPath);

    foreach (string videoPath in videoList)
    {
        var videoName = Path.GetFileName(videoPath);
        var videoDirectory = Path.Combine(destPath, videoName);

        // Accessing and uploading files to the machine's AppData folder:
        if (!File.Exists(videoDirectory)) File.Copy(videoPath, videoDirectory);

        // Method for accessing and uploading files to an FTP server:
        if (ftpIP != null || ftpUsername != null || ftpPassword != null) UploadFile(videoPath, ftpIP, ftpUsername, ftpPassword,
lastFolder, videoName);
    }
}
}

```

// Uploads file to the specified FTP server:

```

public void UploadFile(string from, string ftpAdress, string username, string password, string lastFolder, string videoName)
{
    // The FTP address is obtained, a login is attempted:
    WebRequest request = WebRequest.Create("ftp://" + ftpAdress + "/" + lastFolder);
    request.Method = WebRequestMethods.Ftp.MakeDirectory;
    request.Credentials = new NetworkCredential(username, password);

    try
    {
        using (var resp = (FtpWebResponse) request.GetResponse())
        {
            // A directory was generated:
            Debug.WriteLine(resp.StatusCode);
        }
    }
    catch (WebException e)
    {
        // An exception is caught when the directory already exists:
        Debug.WriteLine(((FtpWebResponse)e.Response).StatusDescription);
    }
    using (WebClient client = new WebClient())
    {
        client.Credentials = new NetworkCredential(username, password);
        client.UploadFile("ftp://" + ftpAdress + "/" + lastFolder + "/" + videoName, from);
    }
}

```

```

private void lockToggle_Click(object sender, EventArgs e)
{
    if (!locked)
    {
        treeView1.CollapseAll();
        locked = true;
    }
    else ShowUnlockPrompt();
}

```

```

        lockToggle.Text = locked ? "Unlock" : "Lock";
    }

    private bool isUnlocked()
    {
        if (locked)
        {
            ShowUnlockPrompt();
        }
        return !locked;
    }

    private void checkForLock(object sender, TreeViewCancelEventArgs e)
    {
        if (locked) e.Cancel = true;
    }

    private void ShowUnlockPrompt()
    {
        if (UnlockPrompt == null || !UnlockPrompt.Visible)
        {
            UnlockPrompt = new UnlockPrompt(this);
            UnlockPrompt.Show();
        }
        else
        {
            UnlockPrompt.Focus();
        }
    }

    private void ftpServerSettings_Click(object sender, EventArgs e)
    {
        if (ftp_settings == null || !ftp_settings.Visible)
        {
            ftp_settings = new FTP_Settings(this);
            ftp_settings.Show();
        }
        else
        {
            ftp_settings.Focus();
        }
    }

    // Used by the Dashboard:
    private void LoginSumBasedOnDay()
    {
        if (DatesAreInTheSameWeek(Properties.Settings.Default.lastLogin, DateTime.Today))
        {
            Properties.Settings.Default.lastLogin = DateTime.Today;

            switch (DateTime.Today.DayOfWeek)
            {
                case DayOfWeek.Monday: Properties.Settings.Default.mondayCount += 1; break;
                case DayOfWeek.Tuesday: Properties.Settings.Default.tuesdayCount += 1; break;
                case DayOfWeek.Wednesday: Properties.Settings.Default.wednesdayCount += 1; break;
                case DayOfWeek.Thursday: Properties.Settings.Default.thursdayCount += 1; break;
                case DayOfWeek.Friday: Properties.Settings.Default.fridayCount += 1; break;
                case DayOfWeek.Saturday: Properties.Settings.Default.saturdayCount += 1; break;
                case DayOfWeek.Sunday: Properties.Settings.Default.sundayCount += 1; break;
            }
        }
    }

```

```

else
{
    Properties.Settings.Default.mondayCount = 0;
    Properties.Settings.Default.tuesdayCount = 0;
    Properties.Settings.Default.wednesdayCount = 0;
    Properties.Settings.Default.thursdayCount = 0;
    Properties.Settings.Default.fridayCount = 0;
    Properties.Settings.Default.saturdayCount = 0;
    Properties.Settings.Default.sundayCount = 0;

    Properties.Settings.Default.lastLogin = DateTime.Today;

    LoginSumBasedOnDay();
}

Properties.Settings.Default.Save();
}

private bool DatesAreInTheSameWeek(DateTime date1, DateTime date2)
{
    return date1.Date.AddDays(-((int)date1.DayOfWeek - 1)) == date2.Date.AddDays(-((int)date1.DayOfWeek - 1));
}
}
}

```

Código de Lista de Alarmas:

```
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using RestSharp;
using System;
using System.Windows.Forms;

namespace graymatics_vms
{

    public partial class AlarmList : UserControl
    {
        MainForm MainForm;
        public AlarmList(MainForm form)
        {
            InitializeComponent();
            MainForm = form;
        }
        public void search_alarm(string hardware_name)
        {
            Alarm_Struct rtsp = new Alarm_Struct();
            rtsp.hardware_name = hardware_name;
            rtsp.start_time = dt2ts(dateTimePicker_begin.Value);
            rtsp.end_time = dt2ts(dateTimePicker_end.Value);
            rtsp.description = textBox_des.Text;
            rtsp.alarm_name = textBox_title.Text;

            string body = JsonConvert.SerializeObject(rtsp);
            string url = string.Format("http://{0}:{1}/alarm/search", Info.IP, Info.PORT);
            var client = new RestClient(url);
            var request = new RestRequest("/", Method.Post);
            request.AddHeader("Authorization", "Bearer " + Info.TOKEN);
            request.AddParameter("application/json", body, ParameterType.RequestBody);
            RestResponse response = client.Execute(request);
            if (!response.IsSuccessStatusCode)
            {
                return;
            }
            JArray ja = (JArray)JsonConvert.DeserializeObject(response.Content);
            dataGridView1.Rows.Clear();
            int i = 1;
            foreach (JObject jb in ja)
            {
                string start_time = "";
                string end_time = "";
                if (jb["start_time"].ToString() != "")
                {
                    start_time = ts2str((long)jb["start_time"]);
                }
                if (jb["end_time"].ToString() != "")
                {
                    end_time = ts2str((long)jb["end_time"]);
                }
                string[] data =
                {
                    i++.ToString(),
                    jb["alarm_name"].ToString(),
                    start_time,
                    end_time,
                    jb["description"].ToString(),
                }
            }
        }
    }
}
```

```

        jb["hardware_type"].ToString(),
        jb["hardware_name"].ToString()
    };
    dataGridView1.Rows.Add(data);
}
}
public static string ts2str(long timestamp)
{
    DateTime local_time = TimeZone.CurrentTimeZone.ToLocalTime(new DateTime(1970, 1, 1));
    DateTime s = local_time.AddSeconds(timestamp);
    return (s.ToString("yyyy/MM/dd HH:mm:ss"));
}
public static long dt2ts(DateTime dt)
{
    DateTime local_time = TimeZone.CurrentTimeZone.ToLocalTime(new DateTime(1970, 1, 1));
    return (long)(dt - local_time).TotalSeconds;
}
public void get_alarm()
{
    string url = string.Format("http://{0}:{1}/alarm/get", Info.IP, Info.PORT);
    var client = new RestClient(url);
    var request = new RestRequest("/", Method.Get);
    request.AddHeader("Authorization", "Bearer " + Info.TOKEN);
    RestResponse response = client.Execute(request);
    if (!response.IsSuccessStatusCode)
    {
        return;
    }
    JArray ja = (JArray)JsonConvert.DeserializeObject(response.Content);
    dataGridView1.Rows.Clear();
    int i = 1;

    foreach (JObject jb in ja)
    {
        string start_time = "";
        string end_time = "";
        if (jb["start_time"].ToString() != "")
        {
            start_time = ts2str((long)jb["start_time"]);
        }
        if (jb["end_time"].ToString() != "")
        {
            end_time = ts2str((long)jb["end_time"]);
        }
        string[] data =
        {
            i++.ToString(),
            jb["alarm_name"].ToString(),
            start_time,
            end_time,
            jb["description"].ToString(),
            jb["hardware_type"].ToString(),
            jb["hardware_name"].ToString()
        };
        dataGridView1.Rows.Add(data);
    }
}

private void button1_Click(object sender, System.EventArgs e)
{
    search_alarm(comboBox_hardware.Text);
}

```



```

private void button2_Click(object sender, System.EventArgs e)
{
    get_alarm();
}

private void AlarmList_Load(object sender, System.EventArgs e)
{
}

private void comboBox_hardware_Click(object sender, System.EventArgs e)
{
    comboBox_hardware.Items.Clear();
    comboBox_hardware.Items.Add("");
    foreach (string k in MainForm.rtsp_dic.Keys)
    {
        comboBox_hardware.Items.Add(k);
    }
}
}

class Alarm_Struct
{
    public string hardware_name;
    public string description;
    public string alarm_name;
    public long start_time;
    public long end_time;
}
}

```

Código del Dashboard:

```
using System;
using System.IO;
using System.Windows.Forms;

namespace graymatics_vms
{
    public partial class Dashboard : UserControl
    {
        private MainForm MainForm;

        public Dashboard(MainForm form)
        {
            InitializeComponent();
            MainForm = form;
        }

        private void Dashboard_Load(object sender, EventArgs e)
        {
            chart1.Series["Logins"].Points.Clear();

            chart1.Series["Logins"].Points.AddXY("Monday", Properties.Settings.Default.mondayCount);
            chart1.Series["Logins"].Points.AddXY("Tuesday", Properties.Settings.Default.tuesdayCount);
            chart1.Series["Logins"].Points.AddXY("Wednesday", Properties.Settings.Default.wednesdayCount);
            chart1.Series["Logins"].Points.AddXY("Thursday", Properties.Settings.Default.thursdayCount);
            chart1.Series["Logins"].Points.AddXY("Friday", Properties.Settings.Default.fridayCount);
            chart1.Series["Logins"].Points.AddXY("Saturday", Properties.Settings.Default.saturdayCount);
            chart1.Series["Logins"].Points.AddXY("Sunday", Properties.Settings.Default.sundayCount);

            Chart3Setup();
        }

        private void Chart3Setup()
        {
            chart3.Series["StorageUse"].Points.Clear();

            string[] cameraList = Directory.GetDirectories("record");

            // get each folder name:
            for(int i = 0; i < cameraList.Length; i++)
                chart3.Series["StorageUse"].Points.AddXY(Path.GetFileName(cameraList[i]), GetSpaceCountForFolder(i));
        }

        private double GetSpaceCountForFolder(int index)
        {
            var cameraList = Directory.GetDirectories("record");

            var videoList = Directory.GetFiles(cameraList[index]);

            double spaceCount = 0;

            foreach (string videoPath in videoList)
            {
                // var videoName = Path.GetFileName(videoPath);
                FileInfo file = new FileInfo(videoPath);

                spaceCount += file.Length;
            }

            return spaceCount * .000001;
        }
    }
}
```


Código de Bloqueo del Sistema:

```
using System;
using System.Drawing;
using System.Windows.Forms;
using MaterialSkin.Controls;

namespace graymatics_vms
{
    public partial class UnlockPrompt : MaterialForm
    {
        MainForm mainForm;

        public UnlockPrompt(MainForm form)
        {
            InitializeComponent();
            mainForm = form;

            usernameTextBox.GotFocus += new EventHandler(RemoveUsernameText);
            passwordTextBox.GotFocus += new EventHandler(RemovePasswordText);

            usernameTextBox.LostFocus += new EventHandler(AddUsernameText);
            passwordTextBox.LostFocus += new EventHandler(AddPasswordText);
        }

        public void UnlockVMS()
        {
            if(usernameTextBox.Text == "test" &&
                passwordTextBox.Text == "test")
            {
                mainForm.locked = false;
                mainForm.lockToggle.Text = mainForm.locked ? "Unlock" : "Lock";
                Close();
                mainForm.UnlockPrompt = null;
            }
            else MessageBox.Show("Wrong username or password", "System Lock Response", buttons: MessageBoxButtons.OK,
icon: MessageBoxIcon.Warning);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            UnlockVMS();
        }

        public void RemoveUsernameText(object sender, EventArgs e)
        {
            if (usernameTextBox.Text == "Enter your username") usernameTextBox.Text = "";
            usernameTextBox.ForeColor = Color.Black;
        }

        public void AddUsernameText(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(usernameTextBox.Text))
            {
                usernameTextBox.Text = "Enter your username";
                usernameTextBox.ForeColor = SystemColors.InactiveCaption;
            }
        }

        public void RemovePasswordText(object sender, EventArgs e)
        {
            if (passwordTextBox.Text == "Enter your password") passwordTextBox.Text = "";
        }
    }
}
```

```
        passwordTextBox.ForeColor = Color.Black;
    }

    public void AddPasswordText(object sender, EventArgs e)
    {
        if (string.IsNullOrEmpty(passwordTextBox.Text))
        {
            passwordTextBox.Text = "Enter your password";
            passwordTextBox.ForeColor = SystemColors.InactiveCaption;
        }
    }

    void UnlockPrompt_FormClosing(object sender, FormClosingEventArgs e)
    {
        if(mainForm.locked) e.Cancel = true;
        return;
    }
}
```

Código Para Añadir un Directorio FTP de Respaldo:

```
using System;
using System.Drawing;
using System.Windows.Forms;
using MaterialSkin.Controls;

namespace graymatics_vms
{
    public partial class FTP_Settings : MaterialForm
    {
        Main_form mainForm;

        public FTP_Settings(Main_form form)
        {
            InitializeComponent();

            mainForm = form;

            ftpIPBox.Text = Properties.Settings.Default.DefaultFTP_IP;
            ftpUsernameBox.Text = Properties.Settings.Default.DefaultFTP_Username;
            ftpPasswordBox.Text = Properties.Settings.Default.DefaultFTP_Password;

            ftpIPBox.GotFocus += new EventHandler(RemoveIPText);
            ftpUsernameBox.GotFocus += new EventHandler(RemoveUsernameText);
            ftpPasswordBox.GotFocus += new EventHandler(RemovePasswordText);

            ftpIPBox.LostFocus += new EventHandler(AddIPText);
            ftpUsernameBox.LostFocus += new EventHandler(AddUsernameText);
            ftpPasswordBox.LostFocus += new EventHandler(AddPasswordText);
        }

        public void RemoveIPText(object sender, EventArgs e)
        {
            if (ftpIPBox.Text == "Enter the IP address") ftpIPBox.Text = "";
            ftpIPBox.ForeColor = Color.Black;
        }

        public void AddIPText(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(ftpUsernameBox.Text))
            {
                ftpUsernameBox.Text = "Enter the IP address";
                ftpUsernameBox.ForeColor = SystemColors.InactiveCaption;
            }
        }

        public void RemoveUsernameText(object sender, EventArgs e)
        {
            if (ftpUsernameBox.Text == "Enter your username") ftpUsernameBox.Text = "";
            ftpUsernameBox.ForeColor = Color.Black;
        }

        public void AddUsernameText(object sender, EventArgs e)
        {
            if (string.IsNullOrEmpty(ftpUsernameBox.Text))
            {
                ftpUsernameBox.Text = "Enter your username";
                ftpUsernameBox.ForeColor = SystemColors.InactiveCaption;
            }
        }
    }
}
```

```

public void RemovePasswordText(object sender, EventArgs e)
{
    if (ftpUsernameBox.Text == "Enter your password") ftpUsernameBox.Text = "";
    ftpUsernameBox.ForeColor = Color.Black;
}

public void AddPasswordText(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(ftpUsernameBox.Text))
    {
        ftpUsernameBox.Text = "Enter your password";
        ftpUsernameBox.ForeColor = SystemColors.InactiveCaption;
    }
}

private void button1_Click(object sender, EventArgs e)
{
    configureFTP();
}

public void configureFTP()
{
    if (ftpIPBox.Text == "Enter the IP address"
        && ftpUsernameBox.Text == "Enter your username"
        && ftpPasswordBox.Text == "Enter your password"
        ||
        (ftpIPBox.Text == ""
        || ftpUsernameBox.Text == ""
        || ftpPasswordBox.Text == ""))
    {
        mainForm.ftpIP = null;
        mainForm.ftpUsername = null;
        mainForm.ftpPassword = null;
    }
    else
    {
        mainForm.ftpIP = ftpIPBox.Text;
        mainForm.ftpUsername = ftpUsernameBox.Text;
        mainForm.ftpPassword = ftpPasswordBox.Text;
    }
    Properties.Settings.Default.DefaultFTP_IP = mainForm.ftpIP;
    Properties.Settings.Default.DefaultFTP_Username = mainForm.ftpUsername;
    Properties.Settings.Default.DefaultFTP_Password = mainForm.ftpPassword;
    Properties.Settings.Default.Save();

    Close();
    mainForm.ftp_settings = null;
}
}
}

```