



**UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO DE TITULACIÓN

Propuesta Tecnológica, previo a la obtención del Título de:

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

**“Desarrollo de servidor de almacenamiento y servicio de imágenes
utilizando librerías de reducción y escalado en la ‘Escuela Lic.
Angélica Villón Lindao’”**

AUTOR:

CAMATÓN RENDÓN JOHNNY ALBERTO

PROFESOR TUTOR:

ING. SHENDRY BALMORE ROSERO VÁSQUEZ

LA LIBERTAD - ECUADOR

2021

AGRADECIMIENTO

Ante todo, le agradezco a DIOS por haberme guiado y dado fuerzas para seguir adelante.

A mi familia por su apoyo, a mis padres: Johnny, Lourdes por su amor incondicional; a mis hermanas: Genesis, Danna; por ser parte importante en mi vida; a mis otras madres: Vilma, Mayra, las quiero.

A mis amigos-hermanos: Edson, Ximena, Oscar, Jefferson, François y Jonathan.

A IRIS: Andrés, Katherine, José, Alex, Ginger y Matías; amigos con los que compartí dentro y fuera del aula y ahora son mis amigos para toda la vida.

Al Ing. Jaime Orozco, Ing. Marjorie Coronel, por siempre estar dispuestos a ayudarme.

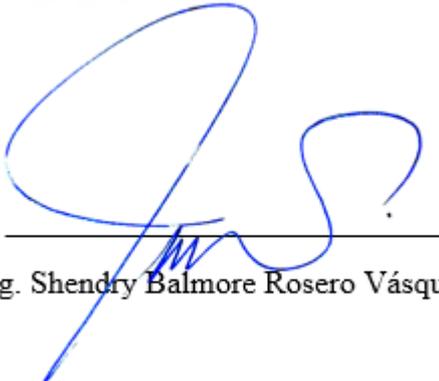
A mi docente tutor, Ing. Shendry Rosero, quien despejó mis dudas sobre la realización de este proyecto a tiempo.

Gracias a todos.

APROBACIÓN DEL TUTOR

En mi calidad de Tutor/Tutora del trabajo de titulación denominado: **“Desarrollo de servidor de almacenamiento y servicio de imágenes utilizando librerías de reducción y escalado en la escuela Lic. Angélica Villón Lindao”**, elaborado por el estudiante **Camatón Rendón Johnny Alberto**, de la carrera de **Tecnologías de la información** de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, lo apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

La Libertad, agosto del 2021

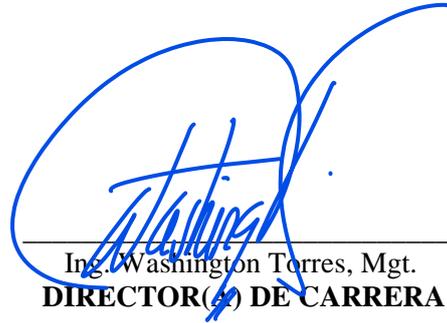


Ing. Shendry Balmore Rosero Vásquez

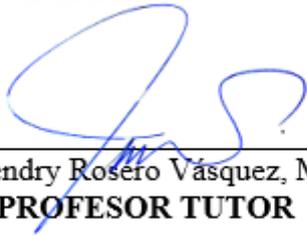
TRIBUNAL DE GRADO



Ing. Iván Sánchez, MSc.
DOCENTE ESPECIALISTA



Ing. Washington Torres, Mgt.
DIRECTOR(a) DE CARRERA



Ing. Shendry Rosero Vázquez, Ms.CC.
PROFESOR TUTOR



Ing. Marjorie Coronel Suárez, MGTI.
PROFESOR DE GUÍA

RESUMEN

La escuela de educación básica "Lic. Angélica Villón Lindao" ubicada en la provincia de Santa Elena, está en constante mejora de sus procesos para garantizar la excelencia académica y los servicios educativos. Uno de ellos es la matriculación, que mantiene registros detallados de los estudiantes de la escuela. Para llevar a cabo el registro, la institución intentó implementar un sistema de matrícula en línea, pero no lo implementó debido a problemas en el procesamiento y almacenamiento de imágenes. Por lo tanto, el registro se realiza en la institución mediante el uso de escritorios, material de oficina y el uso de recursos físicos para almacenar archivos como pagos de matrícula, datos de los estudiantes y fotografías. Sin embargo, debido a la intervención física y el uso de recursos institucionales, este proceso es ineficiente. El propósito de esta propuesta tiene como objetivo resolver el problema de la gestión y el registro de imágenes mediante el desarrollo de un servidor web de almacenamiento y una interfaz de programación para interactuar con el servidor, que permite subir, eliminar y obtener una imagen para el sistema de matriculación en línea. La construcción del servidor web se basa en el modelo de software incremental en el desarrollo de la interfaz de programación, además se reduce el tamaño de la imagen mediante el uso del modelo experimental. El servidor web se creó con las herramientas del *stack* LAMP, haciendo uso de la arquitectura cliente-servidor y patrón de arquitectura MVC. Los resultados obtenidos son: un servidor web Linux para almacenar objetos de imagen y una interfaz de programación (API) para interactuar con el servidor. La integración de estas herramientas permitió solucionar el manejo de las imágenes en el sistema y mitigar la interacción humana.

ABSTRACT

The elementary school "Lic. Angélica Villón Lindao" located in the province of Santa Elena, is constantly improving its processes to ensure academic excellence and educational services. One of them is enrollment, which keeps detailed records of the school's students. To carry out registration, the institution tried to implement an online enrollment system, but did not implement it due to problems in image processing and storage. Therefore, registration is carried out at the institution through the use of desks, office equipment, and the use of physical resources to store files such as tuition payments, student data, and photographs. However, due to the physical intervention and the use of institutional resources, this process is inefficient. The purpose of this proposal aims to solve the problem of image management and registration by developing a storage web server and a programming interface to interact with the server, which allows uploading, deleting and retrieving an image for the online enrollment system. The construction of the web server is based on the incremental software model in the development of the programming interface, also the weight of the image is reduced by using the experimental model. The web server was created with the tools of the LAMP stack, making use of the client-server architecture and MVC architecture pattern. The results obtained are: a Linux web server to store image objects and a programming interface (API) to interact with the server. The integration of these tools allowed solving the handling of the images in the system and mitigating human interaction.

DECLARACIÓN

El contenido del presente Trabajo de Graduación es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.



Camatón Rendón Johnny Alberto

TABLA DE CONTENIDOS

ITEM	PÁGINA
AGRADECIMIENTO	2
APROBACIÓN DEL TUTOR	4
TRIBUNAL DE GRADO	5
RESUMEN	6
ABSTRACT	7
DECLARACIÓN	8
TABLA DE CONTENIDOS	9
ÍNDICE DE FIGURAS	12
ÍNDICE DE TABLAS	13
LISTA DE ANEXOS	15
INTRODUCCIÓN	16
CAPÍTULO I	17
1. FUNDAMENTACIÓN	17
1.1. Antecedentes	17
1.2. Descripción del proyecto	19
1.3. Objetivos del proyecto	20
1.3.1. Objetivo general	20
1.3.2. Objetivos Específicos	20
1.4. Justificación	21
1.5. Metodología del proyecto	22
1.5.1. Metodología de la investigación	22
1.5.2. Metodología de recolección de información	23
1.5.3. Variable del proyecto	23
1.5.4. Beneficiarios del proyecto	23
1.5.5. Metodología de desarrollo	23

1.5.6.	Etapas del desarrollo	24
1.5.7.	Metodología experimental	24
CAPÍTULO II		25
2.	La propuesta	25
2.1.	Marco Contextual	25
2.1.1.	Escuela de Educación Básica “Lic. Angélica Villón Lindao”	25
2.1.2.	Visión	26
2.1.3.	Misión	26
2.2.	Marco Conceptual	26
2.2.1.	Arquitectura LAMP	26
2.2.2.	Arquitectura Modelo-Vista-Controlador	27
2.2.3.	Interfaz de Programación de Aplicaciones (API)	28
2.2.4.	Punto de conexión o punto final.	28
2.2.5.	Protocolo HTTP.	28
2.2.6.	<i>Framework</i>	29
2.2.7.	Linux Ubuntu	29
2.2.8.	Apache	29
2.2.9.	PHP	29
2.2.10.	MariaDB	30
2.2.11.	<i>Diagrams</i>	30
2.2.12.	PhpStorm	30
2.2.13.	Symfony 5	30
2.2.14.	Doctrine 2.8	30
2.2.15.	API Platform	31
2.2.16.	MIME	31
2.2.17.	Identificador Único Global	31
2.2.18.	Almacenamiento de objetos.	31

2.2.19.	JSON	31
2.3.	Marco Teórico	32
2.3.1.	El uso de software libre y código abierto.	32
2.3.2.	La seguridad informática en el desarrollo web.	32
2.3.3.	Microservicios en el desarrollo web	33
2.3.4.	La importancia de los servidores de imágenes y el uso de los CDN	33
2.4.	Componentes de la propuesta	33
2.4.1.	Módulos del sistema	33
2.4.2.	Requerimientos funcionales	35
2.4.3.	Requerimientos no funcionales	37
2.5.	Diseño de la propuesta	39
2.5.1.	Diagrama de procesos	39
2.5.2.	Configuración de servidores	42
2.5.3.	Descripción de alto nivel de la API	43
2.5.4.	Descripción técnica de la API	44
2.5.5.	Librerías de tratamiento de imágenes	46
2.6.	Estudio de Factibilidad	47
2.6.1.	Factibilidad técnica	47
2.6.2.	Factibilidad económica	48
2.7.	Resultados	51
2.7.1.	Pruebas de funcionalidad	51
2.7.2.	Resultados finales	62
2.8.	Conclusiones	66
2.9.	Recomendaciones	67
2.10.	Bibliografías	67
	ANEXOS	72

ÍNDICE DE FIGURAS

ITEM	PÁGINA
Ilustración 1 Modelo incremental del servidor web de almacenamiento y servicio de imágenes. Fuente: autor.	24
Ilustración 2: Mapa de la institución: Fuente: Google Maps.	25
Ilustración 3 stack LAMP: Fuente: autor.	27
Ilustración 4 Modelo-Vista-Controlador. Fuente: autor.	27
Ilustración 5 Interfaz de Programación de Aplicaciones. Fuente: autor.	28
Ilustración 6 Protocolo HTTP. Fuente: autor.	29
Ilustración 7 Ejemplo de objeto JSON. Fuente: autor.	32
Ilustración 8 Diagrama de procesos 'Cargar imagen'. Fuente: autor.	39
Ilustración 9 Diagrama de procesos 'Eliminar imagen'. Fuente: autor.	40
Ilustración 10 Caso de uso 'Obtener imagen'. Fuente: autor.	41

ÍNDICE DE TABLAS

ITEM	PÁGINA
Tabla 1 Beneficiarios de la solución	23
Tabla 2 Requerimientos funcionales.	37
Tabla 3 Requerimientos no funcionales	38
Tabla 4 Particiones de disco duro para Linux.	42
Tabla 5 Descripción de alto nivel de la API: Subir imagen	43
Tabla 6 Descripción de alto nivel de la API: Eliminar imagen	43
Tabla 7 Descripción de alto nivel de la API: Obtener imagen	44
Tabla 8 Descripción técnica de la API: Subir imagen	44
Tabla 9 Descripción técnica de la API: Eliminar imagen	45
Tabla 10 Descripción técnica de la API: Obtener imagen	46
Tabla 11 Factibilidad técnica de hardware	47
Tabla 12 Factibilidad técnica de software	48
Tabla 13 Factibilidad económica de hardware	48
Tabla 14 Factibilidad económica de software	49
Tabla 15 Factibilidad económica de recursos humanos	50
Tabla 16 Factibilidad económica de gastos varios	50
Tabla 17 Factibilidad económica total	50
Tabla 18 Prueba de funcionalidad #1	52
Tabla 19 Prueba de funcionalidad #2	53
Tabla 20 Prueba de funcionalidad #3	54
Tabla 21 Prueba de funcionalidad #4	55
Tabla 22 Prueba de funcionalidad #5	56
Tabla 23 Prueba de funcionalidad #6	57
Tabla 24 Prueba de funcionalidad #7	58
Tabla 25 Prueba de funcionalidad #8	59
Tabla 26 Prueba de funcionalidad #9	60
Tabla 27 Prueba de funcionalidad #10	61
Tabla 28 Comparativa de tiempo en formatos de almacenamiento	62
Tabla 29 Promedio del porcentaje de reducción de imágenes JPEG	62
Tabla 30 Promedio del porcentaje de reducción de imágenes PNG	63
Tabla 31 Promedio del porcentaje de reducción de imágenes GIF	63

Tabla 32 Promedio del porcentaje de reducción de imágenes WEBP	64
Tabla 33 Promedio del porcentaje de reducción total de imágenes	64
Tabla 34 Resultados en la obtención de una imagen JPEG	65
Tabla 35 Resultados en la obtención de una imagen PNG	65
Tabla 36 Resultados en la obtención de una imagen WEBP	65
Tabla 37 Resultados en la obtención de una imagen GIF	66

LISTA DE ANEXOS

Anexo 1: Formato de entrevista	72
Anexo 2: Visión y Misión	73
Anexo 3 Cargar imagen al servidor	73
Anexo 4 Obtener una imagen del servidor	74
Anexo 5 Eliminar una imagen del servidor	74
Anexo 6 Codificación de la función de 'obtener imagen' en la interfaz de programación	75
Anexo 7 Codificación de la función de 'cargar imagen' en la interfaz de programación.	75
Anexo 8 Codificación de la función 'eliminar imagen' en la interfaz de programación.	76

INTRODUCCIÓN

La escuela de educación básica "Lic. Angélica Villón Lindao" es una institución que mejora continuamente sus procesos para buscar una educación eficiente y de calidad. Uno de ellos es la matriculación, que debe realizar un registro detallado de los estudiantes en la institución. Para llevar a cabo este registro, la institución intentó implementar un sistema de registro en línea, pero no lo implementó debido a problemas en el procesamiento y almacenamiento de las imágenes.

Por lo tanto, este registro se ejecuta de manera interna en la institución mediante el uso de escritorios, artículos de oficina y el uso de recursos físicos para almacenar los documentos como los registros de pago de matrícula, datos de los estudiantes y fotografías. En consecuencia, este proceso resulta ser ineficiente por la intervención humana y el uso de recursos de la institución.

El propósito de esta propuesta tecnológica tiene como objetivo resolver el problema del procesamiento y almacenamiento de imágenes mediante el desarrollo de un servidor web de almacenamiento para el sistema de matrícula en línea y una interfaz de programación para interactuar con el servidor, con la finalidad de eliminar la intervención humana y mejorar el proceso de matriculación.

Este servidor de almacenamiento web está creado con las herramientas del *stack* LAMP, toda la aplicación está escrita en el lenguaje de programación PHP, utilizando la base de datos relacional MySQL. Para la elaboración de este servidor se tomó en cuenta la arquitectura cliente-servidor, y el patrón de arquitectura MVC en el cual se ejecuta la interfaz de aplicaciones (API) REST, lo que lo convierte en un entorno fácil de desarrollar y desplegar. La integración de estas herramientas permitió solucionar el problema de procesamiento, almacenamiento de las imágenes y mitigar la interacción humana.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1. Antecedentes

Un aspecto negativo que siempre está presente en cualquier desarrollo de software es el tratamiento de archivos que necesitan ser visualizados o utilizados de otra manera en el sistema, un claro ejemplo de ello es el tratamiento de las imágenes en sitios o aplicaciones web [1]. Cabe señalar que el procesamiento inadecuado de imágenes puede tener un impacto significativo en la seguridad y el rendimiento de un sistema informático, lo que puede desencadenar en una vulnerabilidad, como exponer un conjunto de imágenes, archivos del sistema, incluso, exponer credenciales de acceso o información confidencial, destruyendo así, la integridad del sistema o la organización [2].

Bajo este contexto, se encuentra la escuela de educación básica "Lic. Angélica Villón Lindao", actualmente ubicada en el cantón Santa Elena, establecida el 12 de septiembre de 2000, está integrada por un grupo de profesionales de los campos administrativo y docente. En la actualidad, la junta directiva de la institución educativa está compuesta por el coordinador Arq. Mauricio Yunda Villao y la directora administrativa Lcda. Ibelice Tomalá V. (ver Anexo 1). Esta institución se esfuerza por brindar una educación inclusiva y de alta calidad, para así, promover el aprendizaje continuo de los jóvenes de la península, al mismo tiempo que busca desarrollar su potencial en el campo virtual (ver Anexo 1).

La institución intentó implementar un sistema web para la matrícula en línea, en la cual se realizó el diseño de una base de datos y codificación de una aplicación web, sin embargo, el sistema que se desarrolló quedó con inconsistencias, específicamente en el manejo de las imágenes, debido a la falta de recursos económicos y el cambio de administraciones (Ver Anexo 1), lo cual provocó que no se completara el desarrollo de la aplicación de manera oportuna.

La institución necesita guardar una foto de un recibo de pago y una foto del estudiante que se está matriculando, pero la aplicación codificada no puede realizar el proceso de almacenar la imagen de forma segura en el servidor, lo que resulta en no guardar la imagen o guardarla en una ruta pública, por lo que se puede acceder a todas las imágenes del sistema.

En la oficina de la Escuela de Educación Básica "Lic. Angélica Villón Lindao", se realizó una entrevista al Arq. Mauricio Yunda Villao, actual coordinador de la institución, quien expresó que debido a la falta de un sistema de matrícula en línea, por razones de seguridad, debido a la divulgación de los datos y su integridad para su uso posterior, como medida preventiva la institución decidió registrar físicamente a los estudiantes y los archivos que sean posibles receptorlos a través de las redes sociales (Ver Anexo 1).

Es preciso señalar que el manejo inadecuado de las imágenes produciría un impacto significativo en el rendimiento y seguridad de un sistema informático, tales como la exposición de una colección de imágenes, archivos del sistema o incluso llegar a exponer credenciales de acceso o información delicada que pone en peligro la integridad del sistema o las personas.

Ahora bien, también se han explorado otras soluciones para la gestión y el almacenamiento de imágenes en varios lugares, para citar un ejemplo, en Chile, se realizó un sustento de un Sistema de Digitalización y Almacenamiento de Imágenes Radiológicas. Este trabajo tuvo como objetivo principal la obtención del diseño de un Sistema de Información para el Hospital Naval Cirujano Guzmán de Punta de Arenas, creando un repositorio y brindando imágenes Radiológicas a los Doctores en corto tiempo para la obtención de un diagnóstico rápido y eficiente, sin embargo, debido a que este software fue desarrollado para uso privado no se puede acceder a él [3].

Cuba representa otro ejemplo claro en lo que tiene que ver con la gestión de imágenes, en este país se realizó el desarrollo de un sistema para el almacenamiento y transmisión de imágenes médicas, este proyecto se creó como una aplicación de escritorio con herramientas de código abierto, no obstante, el proyecto es utilizado de forma local [4]. En la ciudad de Guayaquil, se realizó el diseño e implementación de una aplicación DICOM cliente/servidor para el transporte de imágenes radiológicas digitales en un ambiente hospitalario, sin embargo, debido a que las herramientas utilizadas son de paga, no representa una solución viable para la institución [5].

Luego de revisar e investigar un conjunto de soluciones relacionadas con este problema, se notó que cumplen parcialmente con los requisitos actuales, pero no complementan todas las necesidades de la institución. Por este motivo, se recomienda utilizar un servidor web para almacenar y servir imágenes con nuevas tecnologías, con el fin de llevar a cabo un registro preciso de los estudiantes matriculados de forma rápida y segura.

1.2. Descripción del proyecto

Debido a la necesidad de gestionar y almacenar las imágenes de los estudiantes en la Escuela de Educación Básica "Lic. Angélica Villón Lindao", se plantea implementar un servidor web para el almacenamiento y servicio de imágenes, en el sistema de matriculación con su respectiva interfaz de programación de aplicaciones (API) REST [6], mediante el protocolo HTTP [7], para poder interactuar y obtener los recursos de manera eficiente, rápida y segura de acorde a la necesidad de la aplicación.

El proyecto utiliza como base el sistema operativo Linux Ubuntu, servidor Apache II, lenguaje de programación PHP, y como gestor de base de datos MariaDB. El desarrollo de este proyecto se divide en tres fases, la primera fase está compuesta por la configuración del entorno de desarrollo. En esta etapa se crea una máquina virtual, se define y se instala el sistema operativo, herramientas, lenguaje, framework [8], librerías, base de datos y el desarrollo del módulo "Cargar imagen".

Posteriormente se desarrolla la API REST HTTP, específicamente un punto final (EndPoint) [9], con el método POST del protocolo HTTP que permite cargar una imagen de manera privada o pública con formatos JPEG, PNG, GIF o WEBP, y obtiene los siguientes metadatos: la dimensión, la extensión en formato MIME [10], y la extensión del archivo. A continuación, se guardará el fichero en el servidor en una ruta indicada en el código y un identificador único global (GUID) [11] para poder acceder al archivo en el servidor. Después se comprimirá la imagen sin comprometer la calidad de esta, y como resultado de este proceso la API devuelve como respuesta un objeto JSON [12], con una URL [13], de acceso, un token para eliminar, y dependiendo de la configuración de la privacidad un hash de acceso.

En la segunda fase se desarrolla el módulo "Eliminar imagen" que, consiste en un punto final (EndPoint) con el método DELETE del protocolo HTTP permitiendo eliminar una imagen. En este proceso primero se verifica el hash de acceso al recurso en caso de tenerlo, luego se verifica que el token de eliminación sea el mismo asociado a la imagen, una vez confirmada la validación, se elimina los datos del fichero de la base de datos, y posteriormente se elimina del servidor. De esta manera, se devuelve como respuesta un objeto JSON con una respuesta de confirmación o error.

La última etapa engloba el desarrollo del módulo "Obtener imagen" consiste en un punto final con el método GET del protocolo HTTP para obtener una imagen, utilizando la URL

que devuelve el EndPoint de la primera etapa. Se puede acceder a la imagen con parámetros personalizables como lo son: el formato, la calidad y la escala. Así mismo para acceder al fichero se necesita la validación del hash de acceso en caso de ser necesario.

Para el desarrollo y despliegue de este proyecto se requieren las siguientes especificaciones base, una máquina con un procesador Intel i3 de 5ta generación, 4GB de memoria RAM, y un espacio de almacenamiento de 500GB. Para la ejecución de estas fases se utilizaron las siguientes tecnologías:

- *Diagrams*: Se emplea para realizar el diagrama UML de la base de datos.
- *PhpStorm*: Es utilizado como el IDE de PHP en el desarrollo de todo el proyecto.
- *PHP, Symfony, Doctrine, y API Platform*: PHP es el lenguaje de programación base utilizado en el desarrollo de todo el proyecto, Symfony como marco de trabajo de PHP, Doctrine como ORM de Symfony y *Api Platform* permite construir webs Api.
- *MariaDB*: Se emplea para almacenar los metadatos de las imágenes de manera estructurada.
- *Apache 2*: Se utiliza como servidor web HTTP para el procesamiento de solicitudes.

El desarrollo de este proyecto para la organización, almacenamiento y optimización de imágenes en el proceso de matriculación en la Escuela de Educación Básica “Lic. Angelica Villón Lindao”, se acopla a la línea de investigación de gestión de desarrollo de software e infraestructura, con el desarrollo de un software de almacenamiento y servicio de imágenes utilizando algoritmos de reducción y escalado [14].

1.3. Objetivos del proyecto

1.3.1. Objetivo general

Desarrollar un servidor web para el almacenamiento, optimización y entrega de imágenes en la Escuela "Lic. Angelica Villón Lindao" mediante el uso de librerías y herramientas del *stack* LAMP.

1.3.2. Objetivos Específicos

1. Implementar un interfaz de programación en PHP, para el ingreso, tratamiento, compresión y entrega de imágenes en el servidor.

2. Desarrollar un repositorio centralizado de almacenamiento de imágenes para el acceso concurrente por parte de múltiples usuarios.
3. Disminuir el espacio de almacenamiento utilizado por los archivos de tipo imagen para optimizar adecuadamente el espacio en el servidor.

1.4. Justificación

A través del tiempo el ser humano ha tenido la necesidad de almacenar, transmitir y preservar sus conocimientos en diferentes soportes, que van desde las antiguas tablas de arcilla, papiros, pergaminos, entre otros instrumentos hasta los medios digitales. En la actualidad el almacenamiento y recuperación de la información representan uno de los problemas a los que la humanidad se ha tenido que enfrentar desde la invención de la escritura. Con la aparición de la computadora este problema se ha resuelto parcialmente con nuevos dispositivos de almacenamiento, diseños de conexión y estructuras de bases de datos [15].

Esto demuestra que con el paso del tiempo el ser humano se ha encontrado con la necesidad de mejorar sus métodos de almacenamiento a causa del incremento de la cantidad de datos producidos, junto con ello también el manejo de la información. Desde las tarjetas perforadas, luego al salto al disquete y así hasta llegar a lo que hoy se conoce como la nube. Dentro del ecosistema digital, el espacio, disponibilidad e integridad son varios de los puntos influyentes a la hora de almacenar información, con ello se han desarrollado diversas técnicas para poder almacenar gran cantidad de archivos en un espacio digital reducido.

La Escuela de Educación Básica "Lic. Angélica Villón Lindao", tiene la necesidad de organizar y mantener gran cantidad de información. Además, la tecnología ha tenido un rol importante en la actualidad, tanto así que la información física de diferentes áreas de la institución ha pasado a convertirse en información digital, requiriendo una mayor organización al momento de manipular los datos.

Ante la necesidad de un registro más preciso, eficiente y seguro para el control de datos e imágenes de los estudiantes, se implementó una solución en base al sistema ya existente para el manejo eficiente y óptimo de imágenes a fin de poder implementar el sistema web de matriculación.

Esta solución proporcionó un almacenamiento centralizado el cual garantiza el correcto almacenamiento, procesamiento, servicio y seguridad de las imágenes para el sistema de matriculación. Mediante la implementación de una interfaz de programación en PHP para la compresión y tratamiento de las imágenes. El sistema brinda un acceso seguro a través de hash y token respectivo para cada una de las imágenes, además la reducción del tamaño de la imagen que se ve reflejado en menos consumo de espacio en disco.

El desarrollo del servidor de almacenamiento y servicio de imágenes permitirá que la Escuela de Educación básica "Lic. Angélica Villón Lindao" lleve un mejor control, seguridad y correcto almacenamiento de las imágenes utilizadas en su sistema informático de matriculación, en el tratamiento de la imagen del estudiante y la imagen de evidencia de pago. Para este proyecto se considerará el uso del algoritmo en el almacenamiento solo de la imagen del estudiante

El tema propuesto está enmarcado a los objetivos del Plan Nacional de Desarrollo, de manera específica se resalta:

Eje 2.- Economía al servicio de la sociedad.

Objetivo 5.- Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria [16].

Política 5.6.- Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades [16].

1.5. Metodología del proyecto

1.5.1. Metodología de la investigación

Para desarrollar esta propuesta tecnológica, se llevará a cabo una metodología de investigación exploratoria [17], en la que se recogerá información de trabajos relacionados con el fin de analizar situaciones similares a servidores de imágenes, servidores de almacenamiento y establecer comparaciones que nos permitan tener una visión clara del alcance de cada uno y el problema a abordar.

Con el fin de conocer el proceso de manejo y almacenamiento de imágenes y obtener información que nos permita realizar esta propuesta tecnológica, se llevará a cabo la

metodología de investigación diagnóstica [17], con el actual coordinador de la Escuela de Educación Básica “Angélica Villón Lindao”.

1.5.2. Metodología de recolección de información

Se utilizarán técnicas de recopilación de información cualitativa [18], para recopilar datos sobre el caso en estudio. Se ejecutará la técnica de observación en las respectivas oficinas de la institución donde se llevan a cabo los procesos de administración y se detallará el almacenamiento y tratamiento de imágenes digitales para el uso en procesos posteriores. Además, se aplicará la técnica de entrevista hacia el coordinador de la institución, Arq. Mauricio Yunda Villao.

1.5.3. Variable del proyecto

La presente propuesta ofrece un almacenamiento rápido, eficiente y centralizado, brindando un acceso eficaz y seguro reduciendo el tamaño, sin comprometer la calidad de la imagen digital. Para cumplir con este objetivo, se evaluará el tamaño de la imagen digital antes y después de la implementación de la propuesta tecnológica.

1.5.4. Beneficiarios del proyecto

La propuesta tecnológica beneficiará directa e indirectamente a varios grupos que forman parte de la institución Educativa. Detallados:

Beneficiarios	Cantidad
Directos	
Administradores	5
Indirectos	
Docentes	12
Estudiantes	~160
	Total:
	~177

Tabla 1 Beneficiarios de la solución

1.5.5. Metodología de desarrollo

Debido a la necesidad de considerar la configuración del servidor web y el desarrollo de la API REST, con el fin de brindar soluciones de alta calidad, se recomienda utilizar la adaptación del modelo de desarrollo de software incremental ya que combina elementos del modelo en cascada, pero en base a incrementos [17]. Cada incremento está relacionado con el progreso del incremento anterior, y cada incremento incluye las fases de análisis, diseño, codificación y prueba.

1.5.6. Etapas del desarrollo

- Fase incremental uno: En esta fase se realiza toda la configuración del entorno de desarrollo, se integran todas las librerías, se diseña la base de datos y se codifica el módulo para cargar una imagen.
- Fase incremental dos: En esta fase tiene lugar el desarrollo del módulo para borrar una imagen.
- Fase incremental tres: En esta fase final, se desarrolla el módulo con el fin de obtener una imagen con varias opciones de personalización.

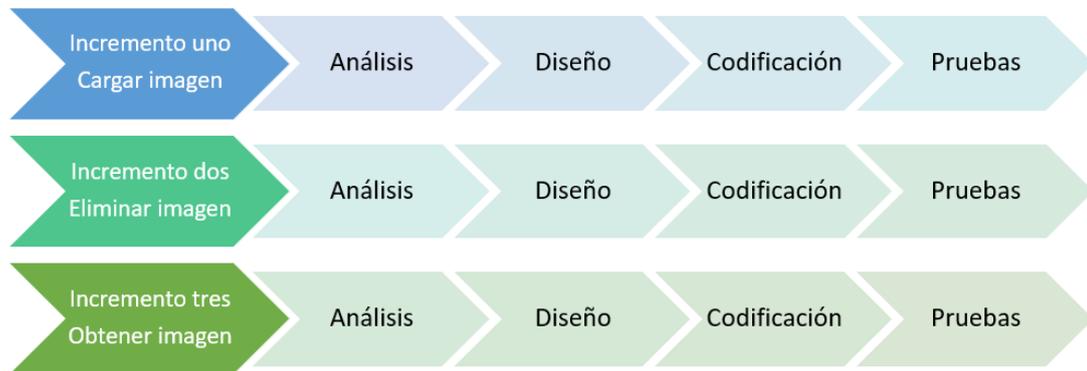


Ilustración 1 Modelo incremental del servidor web de almacenamiento y servicio de imágenes. Fuente: autor.

1.5.7. Metodología experimental

Considerando como variable dependiente el tamaño de la imagen, y variable independiente las bibliotecas de compresión de imágenes de PHP, y definiendo como hipótesis la reducción del tamaño de la imagen desde un 5% hasta un 40%. Se diseñó un experimento con un entorno controlado donde se subirán una colección de imágenes del mismo formato de diferentes fuentes y se obtendrán los resultados en relación con el tamaño de la imagen en el servidor (T_s) y el tamaño original de la imagen (T_o) como índice de reducción de imagen, luego se evaluará el índice de reducción promedio para verificar la reducción de tamaño del archivo de cada formato de imagen [19].

$$I = 100\% - \left(\frac{T_s}{T_o}\right) \times 100$$

Ecuación 1 Índice de reducción

El índice de reducción es igual a cien por ciento menos el tamaño de la imagen en el servidor sobre el tamaño original de la imagen por cien.

CAPÍTULO II

2. La propuesta

2.1. Marco Contextual

2.1.1. Escuela de Educación Básica “Lic. Angélica Villón Lindao”

La escuela de educación básica "Lic. Angélica Villón Lindao" se encuentra actualmente ubicada en la provincia de Santa Elena y fue establecida el 12 de septiembre de 2000. Está conformada por un grupo de profesionales de los campos administrativo y docente. La directiva de la institución está integrada actualmente por el coordinador Arq. Mauricio Yunda Villao y la directora administrativa Lcda. Ibelice Tomalá V. La institución se esfuerza por brindar educación inclusiva y de alta calidad para promover el aprendizaje continuo de los jóvenes en la Península, al mismo tiempo que desarrolla su potencial de manera virtual (Ver Anexo 1).

Se encuentra ubicado en el cantón Santa Elena de la provincia de Santa Elena.

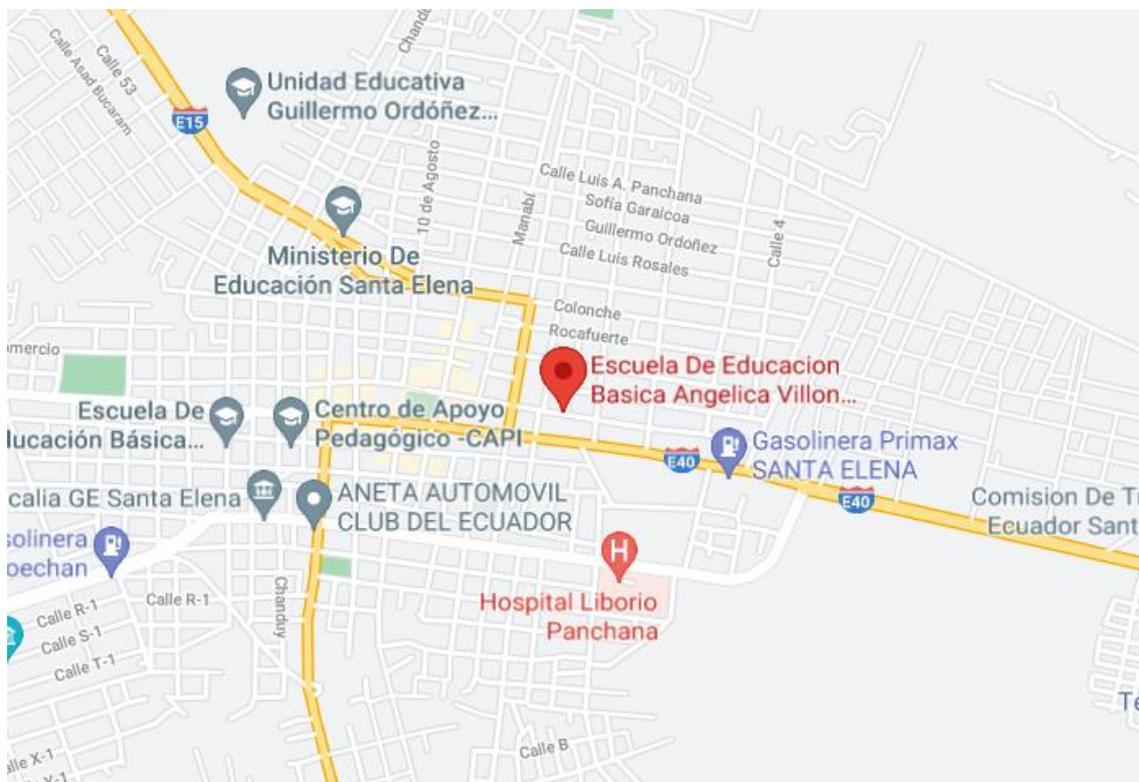


Ilustración 2: Mapa de la institución: Fuente: Google Maps.

2.1.2. Visión

La escuela de Educación Básica “Lcda. Angélica Villón Lindao” ofrece un conjunto de talentos humanos técnicos y pedagógicos al servicio de la educación, desarrollando procesos educativos para los estudiantes con o sin necesidades educativas especiales, atendiendo con absoluta responsabilidad y equidad, sin discriminación alguna y con un enfoque en los derechos (Ver Anexo 2).

2.1.3. Misión

La escuela de Educación Básica “Lcda. Angélica Villón Lindao” anhela brindar una educación actualizada acorde a las exigencias que la sociedad actual demanda, con la atención y diversidad, con docentes capacitados y comprometidos al cambio, para enrumbar a la formación de talentos humanos que se inserten con facilidad de estudios y a la sociedad, que adquieran un pensamiento crítico y reflexivo a través de la investigación científica con una práctica permanente de valores éticos y morales por medio de la inclusión, permitiendo a las personas con necesidades educativas especiales ser parte de la comunidad educativa (Ver Anexo 2).

2.2. Marco Conceptual

2.2.1. Arquitectura LAMP

LAMP es una sigla que hace referencia a una pila de tecnologías que trabajan en conjunto para ofrecer una solución eficiente para sitios o aplicaciones web, el conjunto más utilizado es: Linux, Apache, MySQL, y PHP, siendo este último reemplazable con otros lenguajes como Python o Perl. Estas son utilizadas para la creación de innumerables aplicaciones con un entorno de desarrollo de código abierto que facilita el desarrollo e integración con otras tecnologías.

Estas tecnologías se encargan de trabajar en diversas áreas de una aplicación, Linux siendo el sistema operativo base que se encarga de controlar todos los periféricos del servidor y alojar todos los programas y librerías necesarios, Apache es el software que cumple el rol de servidor, que se encarga de recibir y responder las peticiones, MySQL se encarga de gestionar y almacenar los datos de forma estructurados en el esquema deseado, también puede ser reemplazado por MariaDB, y por último como lenguaje base de la aplicación se tiene a PHP que se encarga de la lógica de la aplicación, también puede ser reemplazado con otros lenguajes como Python o Perl. Para el correcto uso de estas

tecnologías existen ciertos *frameworks* (marcos de trabajo), que definen ciertos patrones de desarrollo, siendo el más utilizado el patrón MVC.

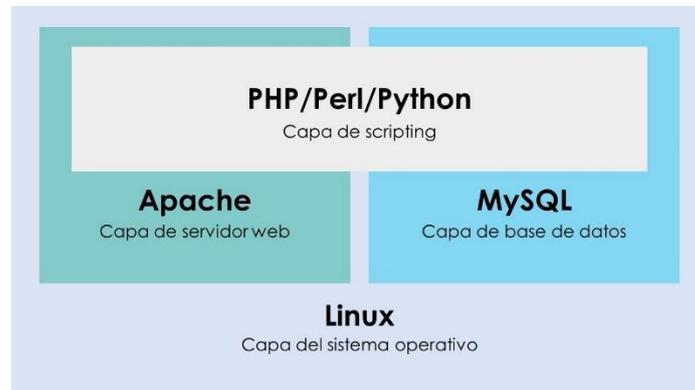


Ilustración 3 stack LAMP: Fuente: autor.

2.2.2. Arquitectura Modelo-Vista-Controlador

El modelo-vista-controlador (MVC) [20] es un patrón de arquitectura que separa una aplicación en tres componentes lógicos principales: el modelo, la vista y el controlador. Cada uno de estos componentes se compila para controlar aspectos de desarrollo específicos de una aplicación. MVC es uno de los marcos de desarrollo web estándar de la industria más utilizados para crear proyectos escalables y extensibles.

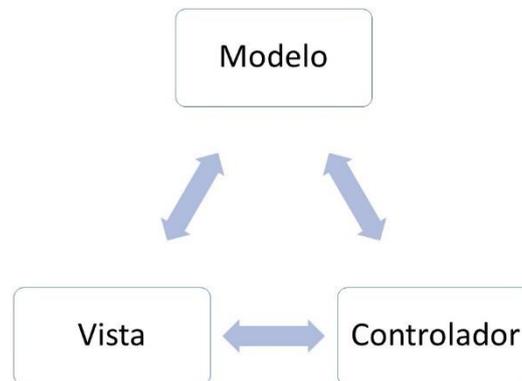


Ilustración 4 Modelo-Vista-Controlador. Fuente: autor.

El modelo corresponde a toda la lógica relacionada con los datos con la que trabaja el usuario. La vista se usa para toda la lógica de la interfaz de usuario de la aplicación. El controlador actúa como una interfaz entre el modelo y la vista para procesar toda la lógica

de negocios y las solicitudes entrantes, manipular los datos mediante el componente modelo e interactuar con las vistas para representar la salida final.

2.2.3. Interfaz de Programación de Aplicaciones (API)

Una interfaz de programación de aplicaciones o mejor conocida en términos básicos como API son un conjunto de funciones y procedimientos que permiten la creación de aplicaciones que acceden a datos y características de otras aplicaciones, servicios o sistemas operativos mediante protocolos definidos [9].

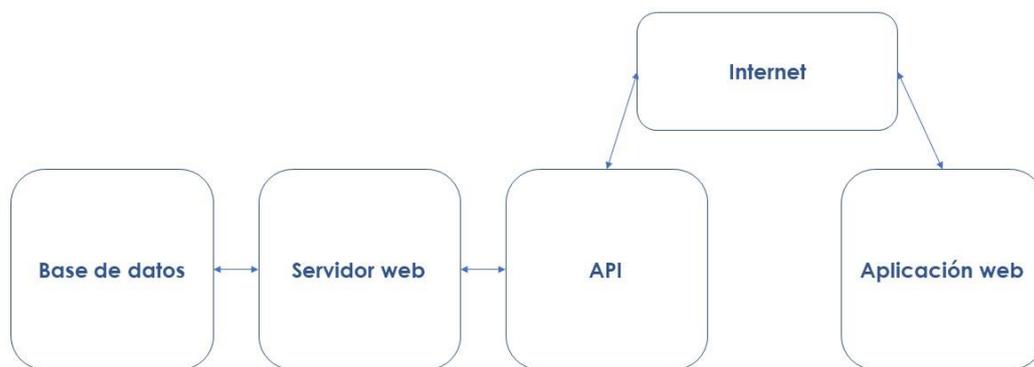


Ilustración 5 Interfaz de Programación de Aplicaciones. Fuente: autor.

2.2.4. Punto de conexión o punto final.

En pocas palabras, un punto final es un extremo de un canal de comunicación. Cuando una API interactúa con otro sistema, los puntos de contacto de esta comunicación se consideran puntos finales. En el caso de las API, un punto final puede incluir una URL de un servidor o servicio. Las API funcionan mediante "solicitudes" y "respuestas". Cuando una API solicita información a una aplicación o servidor web, recibirá una respuesta. El lugar donde las API reciben las solicitudes y donde se encuentra el recurso se llama punto final. [21].

2.2.5. Protocolo HTTP.

El Protocolo de Transferencia de Hipertexto (HTTP) es un protocolo de capa de aplicación diseñado para transferir información entre dispositivos en red y se ejecuta sobre otras capas de la pila de protocolos de red. Un flujo típico a través de HTTP implica que un equipo cliente realice una solicitud a un equipo servidor que, por consiguiente, envía un mensaje de respuesta. [7].

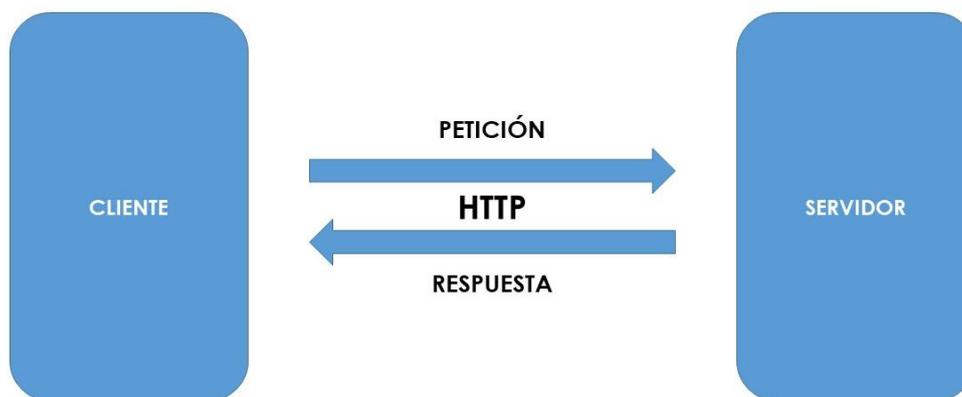


Ilustración 6 Protocolo HTTP. Fuente: autor.

2.2.6. Framework

En programación un *framework* o marco de trabajo se puede definir mejor como una herramienta que permite desarrollar sistemas. El propósito de un marco es mejorar la eficiencia de la creación de nuevo software. Los marcos pueden mejorar la productividad de los desarrolladores y mejorar la calidad, la fiabilidad y la robustez del nuevo software [8].

2.2.7. Linux Ubuntu

Ubuntu es un sistema operativo gratuito de código abierto basado en una distribución Debian GNU/Linux, fue creado como la finalidad de ejecutarse en equipos domésticos, de fácil manejo y totalmente personalizable. Luego fue creada una versión de Ubuntu para servidores por su versatilidad, capacidad de gestión de recursos, cantidad de paquetes disponibles y su gestor de paquetes [22].

2.2.8. Apache

Apache es el servidor web multiplataforma seguro, eficiente y extensible de código abierto que procesa solicitudes y sirve activos y contenido web a través del protocolo HTTP con los últimos estándares establecidos, utiliza al lenguaje de programación PHP como base de respuesta a las peticiones [23].

2.2.9. PHP

PHP es un lenguaje de programación de código abierto del lado del servidor. que se utiliza para desarrollar sitios web estáticos o sitios web dinámicos o aplicaciones web. PHP

significa preprocesador de hipertexto, es potente, fácil y gratis, con un gran rendimiento con cargas elevadas, y con un soporte con un soporte de base de datos integradas [24].

2.2.10. MariaDB

MariaDB es un sistema de administración de bases de datos relacionales (DBMS) [25], de código abierto que es un reemplazo directo compatible para la tecnología de base de datos MySQL ampliamente utilizada. Fue creado como una bifurcación de software de MySQL por desarrolladores que desempeñaron un papel clave en la construcción de la base de datos original [26].

2.2.11. Diagrams

Diagrams es la tecnología de facto para construir software de diagramación. Es una pila de código abierto y una aplicación de usuario final que se usa más ampliamente que cualquier otro software de diagramación de flujo visual, entidad relación, y diagramas UML, utilizado ampliamente para la modelación de diagramas físicos de base de datos basado en tecnologías web [27].

2.2.12. PhpStorm

Es un entorno de desarrollo integrado comercial multiplataforma para PHP, y diversos lenguajes de herramientas que se integran con PHP. Un entorno de desarrollo integrado es un sistema de software para el diseño de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica GUI [28].

2.2.13. Symfony 5

Symfony es un framework para el lenguaje PHP de código abierto para aplicaciones web y un conjunto de componentes PHP reutilizables, y varios patrones de programación definidos, haciendo uso de un conjunto de una gama de tecnologías, como son yaml, json, xml, y sus respectivos patrones para la correcta configuración y uso de marco [29].

2.2.14. Doctrine 2.8

Doctrine ORM es un mapeador relacional de objetos (ORM) para PHP, que proporciona persistencia transparente y eficiente para objetos PHP. Utiliza el patrón *Data Mapper*, con el objetivo de una separación completa de su dominio y lógica comercial de la persistencia en un sistema de administración de base de datos relacional [30].

2.2.15. API Platform

Es un sinónimo de *API Management, Full Lifecycle API Management*, sin embargo, es un framework completo, potente pero fácil de usar dedicado a proyectos impulsados por API. Contiene una biblioteca PHP para crear las API con todas las funciones y normas definidas por los estándares líderes en la industria [31].

2.2.16. MIME

Los MIME es la abreviatura de *Multipurpose Internet Mail Extensions*. Es un estándar desarrollado originalmente para extender los correos electrónicos para poder soportar más formatos como texto no ASCII y archivos adjuntos en forma de archivos de imagen, audio, video o ejecutables. El tipo MIME forma parte del encabezado de MIME y especifica el tipo de medio contenido en un correo electrónico. A menudo también se conoce como tipo de medio o tipo de contenido MIME [10].

2.2.17. Identificador Único Global

Un identificador único global (GUID) es un número de 128 bits creado por Microsoft u otra aplicación de Windows para identificar de forma única componentes específicos, hardware, software, archivos, cuentas de usuario, entradas de base de datos y otros elementos [11].

2.2.18. Almacenamiento de objetos.

El almacenamiento de objetos, también conocido como almacenamiento basado en objetos, es una estrategia para administrar y operar el almacenamiento de datos como diferentes unidades (llamadas objetos). Estos objetos se almacenan en un almacén y no se guardan en archivos de otras carpetas. En cambio, el almacén de objetos combina los elementos de datos que componen el archivo, agrega todos los metadatos relevantes al archivo y agrega un identificador personalizado [32].

2.2.19. JSON

JSON, o notación de objetos JavaScript, es un formato mínimo y legible para estructurar datos que se basa en pares clave-valor y listas ordenadas. Se utiliza principalmente para transmitir datos entre un servidor y una aplicación web, como alternativa a XML [12].

```
{
  "firstName": "Johnny",
  "lastName": "Camatón",
  "loginCount": 4,
  "isWriter": true,
  "worksWith": ["IRIS", "UPSE"],
  "pets": [
    {
      "name": "Black",
      "type": "Dogs"
    }
  ]
}
```

Ilustración 7 Ejemplo de objeto JSON. Fuente: autor.

2.3. Marco Teórico

2.3.1. El uso de software libre y código abierto.

El software de código libre o también conocido como software “open source”, es un software que se distribuye gratuitamente, y su código fuente puede ser modificado por el usuario que lo utiliza [33]. Cada vez es más utilizado en todas las áreas profesionales y no profesionales, junto con el nacimiento de la industria 4.0 [34]. La digitalización gana más terreno de la mano del software libre y herramientas de código libre por las ventajas que la utilización de estas reflejan:

- **Menor costo:** Con el código abierto, el código de un software es gratuito, lo que no representa un costo alguno para las empresas o individuos.
- **Sin dependencia de un solo proveedor:** Esto significa que el usuario puede replicarlo y trasladarlo al lugar donde desea.
- **Flexibilidad:** Dado que el fuerte del open source es la capacidad de modificar. No está obligado a utilizarlo de una manera específica.
- **Transparencia:** Como se tiene acceso al código fuente del programa, se puede verificar los datos que son recopilados.

2.3.2. La seguridad informática en el desarrollo web.

Internet es un sitio que trae consigo muchos beneficios, sin embargo, también es un sitio peligroso. A menudo se encuentran sitios web no disponibles, aplicaciones web que han sido pirateadas, y sus datos y contraseñas han sido vulneradas, todo a causa de un fallo

de despliegue, o pequeños errores de programación que dan acceso a información delicada del sistema [35]. Para evitar estos posibles escenarios, diversas empresas o grupos de personas desarrollan estándares, marcos de desarrollo, patrones de diseños, herramientas y lenguajes de programación de código abierto que pueden ser utilizados por cualquier persona o empresa que ayudan a escanear y mejorar los sistemas de información, a fin de brindar una mejor experiencia y mayor seguridad en los sitios en internet.

2.3.3. Microservicios en el desarrollo web

El uso de la tecnología en diversos procesos de la sociedad va en aumento, tanto así que la arquitectura, el desarrollo y la forma en la que se escalan el software se ha visto alterada por el rápido crecimiento de usuarios, pasando de una arquitectura monolítica a microservicios. Las aplicaciones monolíticas consisten en que todo el sistema está programado en un mismo módulo, si algo falla, todo el sistema colapsa, y los microservicios radican en dividir el sistema en pequeñas aplicaciones independientes, diseñadas para acoplarse a otras [36]. Por consiguiente, el desarrollo, disponibilidad, y escalabilidad son mayores, ya que si un microservicio tiene un fallo el resto del sistema puede seguir funcionando.

2.3.4. La importancia de los servidores de imágenes y el uso de los CDN

Los CDN o también conocidos como *Content Delivery Network*, en español, red de distribución de contenidos, son funcionalidades digitales que permiten distribuir las páginas de un sitio web con más velocidad y seguridad [37]. Los servidores de imágenes proporcionan acceso a datos de imágenes en mapa de bits a través de un servicio sea o no web. Las imágenes se han convertido en parte importante en el desarrollo de internet, así mismo, el saber cómo almacenarlas y tratarlas, por ello se combinan dos herramientas, los CDN y los servidores de imágenes que cumplen el rol de entregar una imagen con un formato y características definidas con la velocidad y seguridad que brindan los CDN.

2.4. Componentes de la propuesta

2.4.1. Módulos del sistema

La implementación de este servidor web permitirá a los integrantes de la Escuela de Educación básica "Lic. Angélica Villón Lindao" la organización y manejo de las imágenes en su sistema de información a través de los siguientes módulos:

- **Módulo de cargar imagen:** Este módulo permite la carga y optimización de la imagen en el servidor a través de los siguientes procesos.
 1. **Guardar registro de la imagen:** Se realiza un barrido en la petición con el fin de extraer el parámetro de privacidad y el archivo para su análisis y validación. En esta etapa se obtienen los metadatos de la imagen, se verifica que la imagen y los metadatos sean los correctos de acuerdo con el formato soportado por el sistema, luego se crea una clase de tipo imagen y se generan los parámetros de identificación y seguridad en base a la configuración de privacidad y se registra en la base de datos del sistema.
 2. **Guardar imagen en el sistema de ficheros del servidor:** Una vez registrados los datos en la base de datos, se obtiene el identificador único global generado en la etapa anterior para colocarlo de nombre del fichero. Luego la imagen pasa de estar en memoria a ser escrita en el disco duro en el directorio especificado en la configuración del marco de trabajo. Para poder guardar la imagen se verifica que exista el directorio, de no existir se crea.
 3. **Optimizar imagen en el servidor:** Se localiza el archivo en el sistema de ficheros para reducir su tamaño en disco, sin comprometer su resolución y calidad. La reducción del tamaño puede variar dependiendo del formato de la imagen o si ya ha sido reducido anteriormente.
- **Módulo de eliminar imagen:**
 1. **Validación de seguridad:** Para eliminar una imagen se envía una solicitud con los parámetros de identificación y seguridad al servidor, luego estas son validadas de acuerdo con la configuración de seguridad de la imagen, si todo es correcto se da paso al siguiente proceso.
 2. **Eliminar registro y fichero del servidor:** Se carga la clase con los datos de la imagen en memoria, luego se genera el directorio de la imagen para eliminar el fichero del sistema de archivos, si la eliminación falla, los datos en memoria de la imagen se vuelven a guardar en el disco.
- **Módulo de obtener imagen:**
 1. **Validación de seguridad:** Para obtener una imagen se envía una solicitud con los parámetros de identificación y seguridad al servidor, luego estas son validadas de acuerdo con la configuración de seguridad de la imagen, si todo es correcto se da paso al siguiente proceso.

2. **Preparación de la imagen:** Se analiza la petición en busca de los parámetros de obtención de la imagen, entre ellos la proporción, formato y calidad, luego se construye la ruta de la imagen para leerla y cargarla en memoria con las modificaciones requeridas.
3. **Construir respuesta:** Una vez cargado el fichero en memoria con las nuevas especificaciones se prepara la respuesta agregando a la cabecera el formato de la imagen, información de cache y directivas de acceso, luego se carga la imagen en el cuerpo de la respuesta, por consiguiente, se devuelve la respuesta.

2.4.2. Requerimientos funcionales

Código	Requerimiento
RF-01	Se subirá la imagen al servidor haciendo uso de los estándares API, mediante el método POST del protocolo HTTP.
RF-02	El sistema tratará con 5 tipos de imágenes: <ul style="list-style-type: none"> • Imágenes con extensión png • Imágenes con extensión jpeg • Imágenes con extensión webp • Imágenes con extensión gif
RF-03	El sistema guardará la imagen con un UUID como nombre del archivo en el sistema de ficheros.
RF-04	El sistema identificará los archivos basándose en los MIME TYPE, y su extensión.

RF-05	El sistema almacenará los metadatos de una imagen en una base de datos estructurada.
RF-06	La imagen será subida al sistema de archivos en el servidor.
RF-07	El sistema comprimirá el tamaño de la imagen hasta un 75% del su tamaño original
RF-08	El sistema generará una cadena de texto de 40 caracteres que funcionará como un hash para el acceso de una imagen
RF-09	El sistema generará una cadena de texto de 40 caracteres que cumplirá el rol de token para eliminar la imagen del servidor.
RF-10	El sistema eliminará una imagen haciendo uso de los estándares API, mediante el método DELETE del protocolo HTTP
RF-11	El sistema contará con un método de verificación de acceso a la imagen por medio de un hash de 40 caracteres
RF-12	El sistema contará con un método de verificación para eliminar a la imagen por medio de un token de 40 caracteres
RF-13	El sistema realizará un borrado seguro de imagen en el sistema de ficheros

RF-14	Se realizará el borrado de los metadatos de la imagen en el sistema de base de datos.
RF-15	El sistema construye la ruta del fichero para la lectura rápida de la imagen
RF-16	El sistema lee el fichero de imagen en formato blob para convertir la imagen a diferentes formatos definidos cuando se requiera
RF-17	El sistema cuenta con un parámetro de personalización limitada de formatos de la imagen
RF-18	El sistema cuenta con un parámetro de personalización de calidad de la imagen
RF-19	El sistema cuenta con un parámetro de personalización de escalado de la imagen
RF-20	El sistema cuenta con una autenticación de acceso a la imagen por medio de un hash de 40 caracteres

Tabla 2 Requerimientos funcionales.

2.4.3. Requerimientos no funcionales

Código	Requerimiento
RNF-01	El servidor web se ejecutará sobre el sistema operativo Linux Ubuntu 16.04.

RNF-02	Los datos serán almacenados en un Sistema Gestor de Base de Datos SQL, MySQL o MariaDB.
RNF-03	El proyecto llevará a cabo su desarrollo con el lenguaje de programación PHP.
RNF-04	El sistema trabajará bajo los estándares de API REST
RNF-05	El sistema contará con un sistema de excepciones de acuerdo con los códigos de respuesta HTTP
RNF-06	Los tiempos de respuesta tienen un rango de 100ms a 1500ms en las peticiones.

Tabla 3 Requerimientos no funcionales

2.5. Diseño de la propuesta

2.5.1. Diagrama de procesos

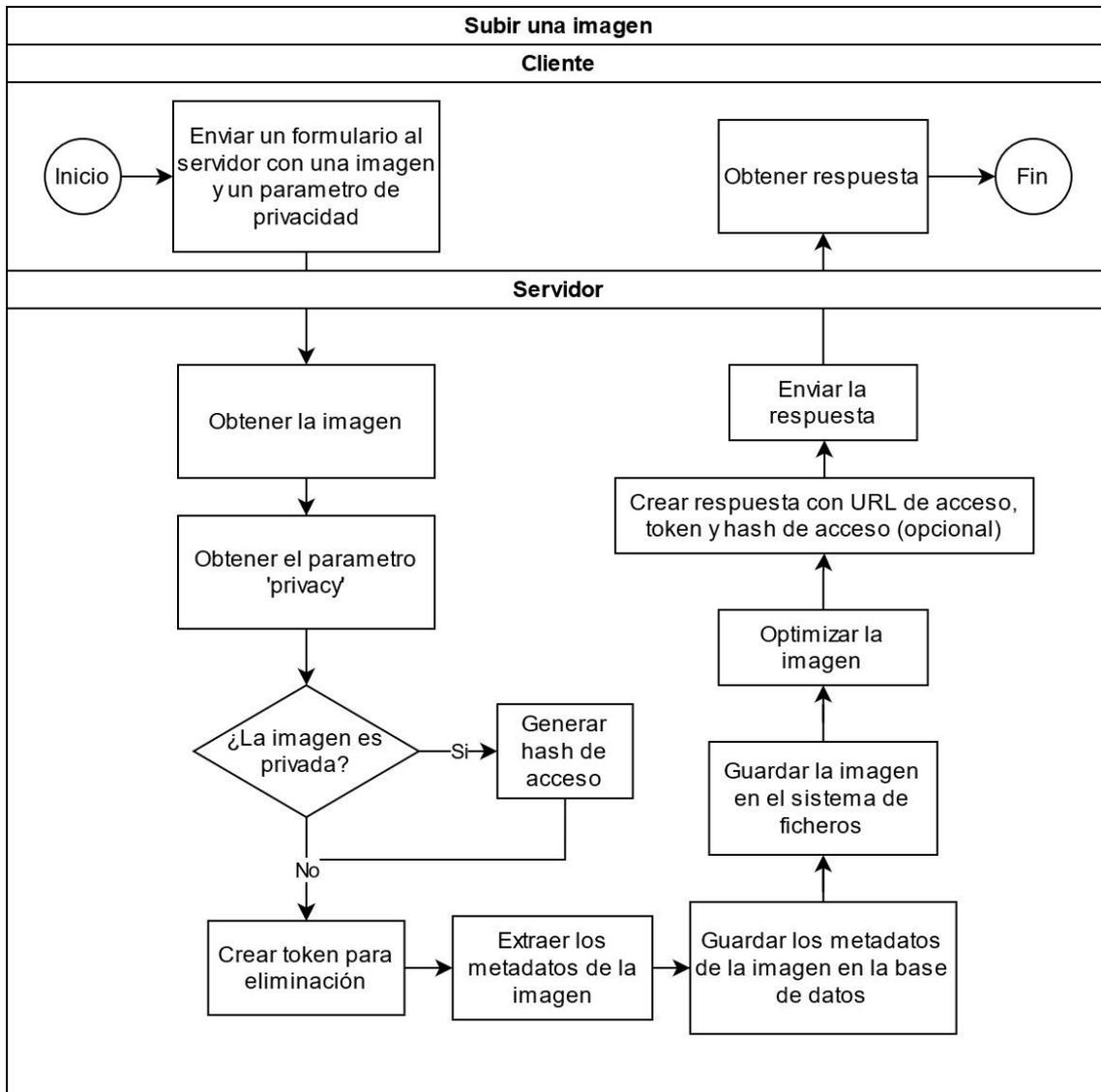


Ilustración 8 Diagrama de procesos 'Cargar imagen'. Fuente: autor.

El la Ilustración 8 muestra un diagrama de proceso que despliega lo que sucede en el sistema después de que el cliente envía un formulario al servidor web. Cuando la solicitud llega al servidor, obtiene la imagen y los parámetros de privacidad del formulario, evalúa los parámetros de privacidad para generar un hash de acceso o no, luego se crea un token de eliminación que se asocia a la imagen, se extrae los metadatos de la imagen y se guarda en la base de datos junto con el hash y el token, luego se escribe la imagen en el disco duro, se optimiza la imagen y se genera una respuesta con la URL para poder acceder al recurso con el token de borrado y, si es necesario, el hash del acceso.

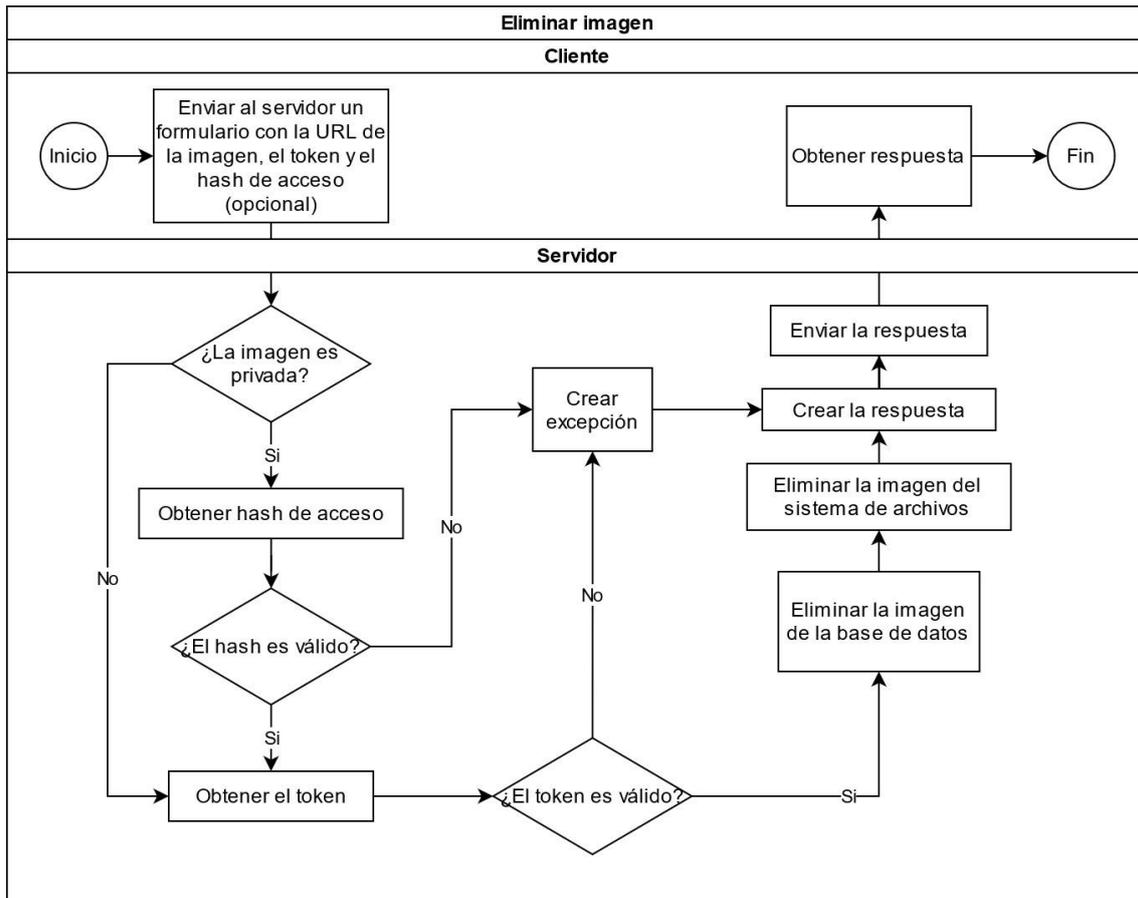


Ilustración 9 Diagrama de procesos 'Eliminar imagen'. Fuente: autor.

La Ilustración 9 muestra un diagrama del proceso de eliminación de una imagen. El cliente envía un formulario que contiene la URL de acceso, el token de acceso y el hash (si la imagen es privada). Si la imagen es privada, el sistema obtendrá y verificará si el hash de acceso es válido. Si no es así, el sistema creará una excepción y generará una respuesta de error para el cliente, si es válido, continuará verificando directamente el token de eliminación, si el token no es válido, se creará una excepción y se generará una respuesta de error para el cliente. Si es válido, la imagen se eliminará de la base de datos, del disco duro y posteriormente se envía una respuesta satisfactoria al cliente.

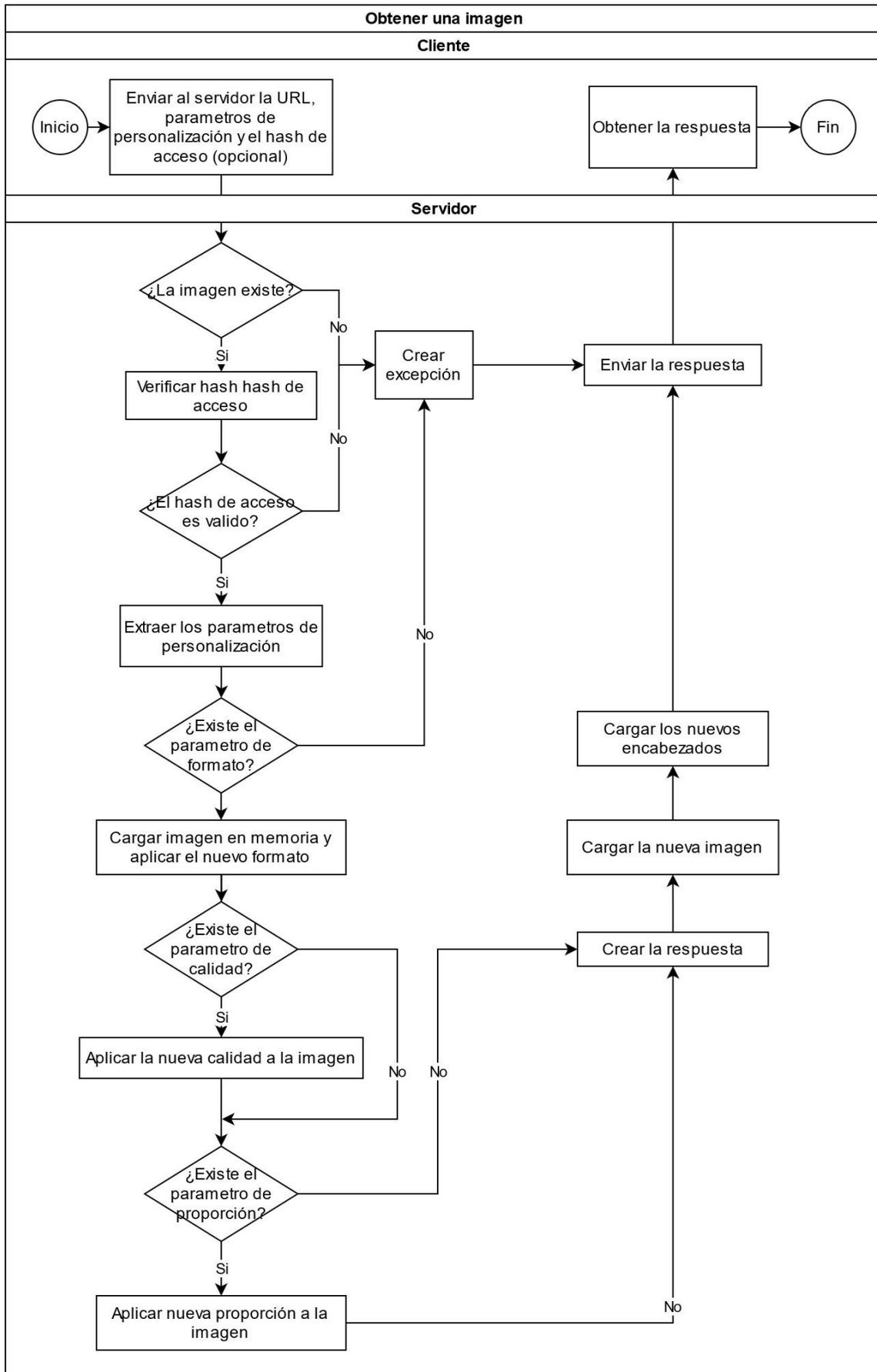


Ilustración 10 Caso de uso 'Obtener imagen'. Fuente: autor.

La Ilustración 10 muestra el proceso de cómo obtener una imagen. El cliente envía un formulario que contiene la URL de acceso, el token de acceso y el hash (si la imagen es privada). Comprueba si la imagen existe y, si no existe, devuelve una excepción con una respuesta de error, si existe, obtiene y evalúa el hash de acceso (si la imagen es privada), si el hash de acceso no es válido, se devuelve un mensaje de error. Si es válido, se obtiene los parámetros de personalización, se realiza una verificación para ver si el parámetro de formato está presente, si no, se devuelve un mensaje de error, si está presente, la imagen se carga en la memoria con el nuevo formato especificado. Se verifica si el parámetro de calidad existe, si existe, se aplica la nueva calidad a la imagen cargada en memoria. De igual manera, se evalúa si el parámetro de proporción existe, si existe y es válido, se aplica la nueva proporción a la imagen y luego devuelve una respuesta con la nueva imagen y sus nuevas cabeceras al cliente.

2.5.2. Configuración de servidores

Configuración de las particiones en la instalación del sistema operativo en el servidor.

- Una partición *swap*
- Una partición */boot*
- Una partición */*
- Una partición */home*

Tabla de espacio

Nombre de la partición	Cantidad
<i>swap</i>	Se utiliza 2GB de espacio en disco.
<i>/boot</i>	Se utiliza 500MB de espacio en disco.
<i>/</i>	Se utiliza 7GB de espacio en disco.
<i>/home</i>	Se utiliza 480GB de espacio en disco.

Tabla 4 Particiones de disco duro para Linux.

Esta configuración está basada en un disco duro de 500gb de capacidad. En caso de utilizar un disco duro de mayor capacidad, se debe utilizar las cantidades proporcionales.

Una vez instalado el sistema operativo, se procede a instalar las tecnologías del *stack* LAMP. Para el correcto funcionamiento del servidor se necesitan 6 librerías: *jpegoptim*, *optipng*, *pngquant*, *gifsicle*, *webp*. Estas librerías son necesarias para la optimización de las imágenes en el servidor.

2.5.3. Descripción de alto nivel de la API

SUBIR IMAGEN			
Descripción		Puede cargar imágenes en el servidor.	
Parámetros de entrada			
Dato	Tipo	¿Opcional?	Descripción
<i>privacy</i>	<i>uri</i>	No	Contiene la configuración de privacidad de la imagen.
<i>image</i>	<i>form-data</i>	No	Contiene la imagen que será subida al servidor.

Tabla 5 Descripción de alto nivel de la API: Subir imagen

ELIMINAR IMAGEN			
Descripción		Se puede eliminar una imagen del servidor	
Parámetros de entrada			
Dato	Tipo	¿Opcional?	Descripción
<i>hash</i>	<i>uri</i>	Si	Una cadena de texto de 40 caracteres que da acceso a la imagen.
<i>token</i>	<i>uri</i>	No	Una cadena de texto de 40 caracteres que se puede utilizar para eliminar la imagen del servidor.

Tabla 6 Descripción de alto nivel de la API: Eliminar imagen

OBTENER IMAGEN			
Descripción		Puede obtener una imagen del servidor.	
Parámetros de entrada			
Dato	Tipo	¿Opcional?	Descripción
<i>hash</i>	<i>uri</i>	Si	Una cadena de texto de 40 caracteres que se utiliza para acceder a la imagen.
<i>format</i>	<i>uri</i>	No	Una cadena de texto que especifica el tipo de formato de imagen: jpeg, png, gif, webp

<i>quality</i>	<i>uri</i>	Si	Especifica el número de calidad de imagen: 1-100
<i>proportion</i>	<i>uri</i>	Si	Un rango de números para poder obtener la imagen con una escala menor que su resolución original.

Tabla 7 Descripción de alto nivel de la API: Obtener imagen

2.5.4. Descripción técnica de la API

SUBIR IMAGEN	
Descripción	Se puede cargar una imagen en el servidor.
Información del servicio	
URL	https://XXXX/image?[privacy=XXX]
Requiere autenticación	<i>Authorization: ninguna</i>
Formato petición	json
Formato respuesta	json
Métodos HTTP	GET
Objeto entrada	Formulario
Objeto devuelto	JSON
¿Pública o privada?	Público
Parámetros Uri	
<i>privacy</i>	Tipo de privacidad: <i>public, private</i>
Petición de ejemplo	
POST	https://XXXX/image?privacy=public
Respuesta de ejemplo JSON	
<pre>{ "status": "ok", "url": "/media/8ddc1dab-e69f-4ffc-b2a4-9eecfc7178fe?format=jpeg", "token": "2cd852cc50aaa614d6b82ae8fde6cd4ab50d37fe" }</pre>	

Tabla 8 Descripción técnica de la API: Subir imagen

ELIMINAR IMAGEN

Descripción	Se puede eliminar una imagen del servidor
Información del servicio	
URL	https://XXXX/media/{image_id}?token=XXX[&hash=XXX]
Requiere autenticación	<i>Authorization:</i> ninguna
Formato petición	json
Formato respuesta	json
Métodos HTTP	DELETE
Objeto entrada	
Objeto devuelto	JSON
¿Pública o privada?	Pública
Parámetros Uri	
image_id	Identificador de la imagen
Petición de ejemplo	
DELETE	http://XXX/media/8ddc1dab-e69f-4ffc-b2a4-9eefc7178fe?token=2cd852cc50aaa614d6b82ae8fde6cd4ab50d37fe
Respuesta de ejemplo JSON	
<pre>{ "status": "ok" }</pre>	

Tabla 9 Descripción técnica de la API: Eliminar imagen

OBTENER IMAGEN	
Descripción	Puede obtener imágenes del servidor.
Información del servicio	
URL	https://XXXX/media/{image_id}?format=XXX[&hash=XXX][&quality=XXX][&proportion=XXX]

Requiere autenticación	Authorization: ninguna
Formato petición	json
Formato respuesta	json
Métodos HTTP	GET
Objeto entrada	
Objeto devuelto	JSON
¿Público o privada?	Público
Parámetros Uri	
image_id	Identificador de la imagen
Petición de ejemplo	
GET	http://XXXX/media/8ddc1dab-e69f-4ffc-b2a4-9eefc7178fe?format=webp
Respuesta de ejemplo JSON	
[Imagen]	

Tabla 10 Descripción técnica de la API: Obtener imagen

2.5.5. Librerías de tratamiento de imágenes

En el lenguaje de programación PHP existen dos bibliotecas de tratamientos de imágenes, una llamada *GD* [38] y otra llamada *ImageMagick* [39], ambas son integradas de forma nativa, existen varias librerías de terceros que generalmente se basan en una de las dos, las diferencias de estas librerías son la potencia y disponibilidad.

Nombre	Ventajas	Desventajas
--------	----------	-------------

<i>GD</i>	<ul style="list-style-type: none"> - Mayor disponibilidad - Mayor velocidad - Fácil de usar 	<ul style="list-style-type: none"> - Depende directamente de la memoria de PHP - Soporta menos formatos
<i>ImageMagick</i>	<ul style="list-style-type: none"> - No depende de la memoria de PHP - Soporta más formatos de imágenes - Fácil de usar 	<ul style="list-style-type: none"> - Menos disponibilidad - Mucho más lenta - Menos seguro

Ambas librerías funcionan bien y producen imágenes de la misma calidad y mismo tamaño si configuran y se utilizan correctamente. Para el desarrollo de este proyecto se utiliza la librería de procesamiento de imágenes GD, y varias librerías de soporte.

2.6. Estudio de Factibilidad

2.6.1. Factibilidad técnica

En este apartado se realizará en detalle el estudio de viabilidad técnica y económica para el desarrollo del proyecto.

Hardware

DESCRIPCIÓN	DETALLE	CANTIDAD	COSTO	SUBTOTAL
Computador	Intel i3 de 5ta generación 4GB RAM Disco duro 500GB	1	\$500.00	\$500.00
TOTAL				\$500.00

Tabla 11 Factibilidad técnica de hardware

Software

DESCRIPCIÓN	CANTIDAD	COSTO	SUBTOTAL
-------------	----------	-------	----------

Linux Ubuntu 16.04	1	\$0.00	\$0.00
PHP 7.4	1	\$0.00	\$0.00
Apache 2	1	\$0.00	\$0.00
MySQL	1	\$0.00	\$0.00
<i>PhpStorm</i>	1	\$0.00	\$0.00
<i>Postman</i>	1	\$0.00	\$0.00
<i>Composer</i>	1	\$0.00	\$0.00
PHP Módulos	1	\$0.00	\$0.00
<i>Diagrams</i>	1	\$0.00	\$0.00
XAMPP	1	\$0.00	\$0.00
TOTAL			\$0.00

Tabla 12 Factibilidad técnica de software

El estudio de factibilidad técnica muestra que el desarrollo del proyecto es factible, porque todas sus herramientas son de código abierto, y no reflejan ningún costo excepto por los equipos utilizados para el desarrollo.

2.6.2. Factibilidad económica

Hardware

DESCRIPCIÓN	DETALLE	CANTIDAD	COSTO	SUBTOTAL
Computador	Intel i3 de 5ta generación 4GB RAM Disco duro 500GB	1	\$500.00	\$500.00
TOTAL				\$500.00

Tabla 13 Factibilidad económica de hardware

El costo del hardware para el desarrollo de este proyecto es de \$500.00. Es este proyecto el autor asume cada uno de los componentes, en consecuencia, el costo de los recursos humanos es de \$0.00

Software

DESCRIPCIÓN	CANTIDAD	COSTO	SUBTOTAL
Linux Ubuntu 16.04	1	\$0.00	\$0.00
PHP 7.4	1	\$0.00	\$0.00
Apache 2	1	\$0.00	\$0.00
MySQL	1	\$0.00	\$0.00
PhpStorm	1	\$0.00	\$0.00
Postman	1	\$0.00	\$0.00
Composer	1	\$0.00	\$0.00
PHP Módulos	1	\$0.00	\$0.00
Diagrams.net	1	\$0.00	\$0.00
XAMPP	1	\$0.00	\$0.00
TOTAL			\$0.00

Tabla 14 Factibilidad económica de software

Todas las herramientas de software son libres y de código abierto, por lo tanto, no implica un costo.

Recursos humanos

DESCRIPCIÓN	CANTIDAD	MESES	COSTO	SUBTOTAL
Analista	1	1	\$900.00	\$900.00
Desarrollador Back-end	1	3	\$1000.00	\$3,000.00
Ingeniero DevOps	1	1	\$1000.00	\$1,000.00

TOTAL	\$4,900.00
--------------	------------

Tabla 15 Factibilidad económica de recursos humanos

El costo de los recursos humanos para el desarrollo del proyecto es de \$4,900.00. Es este proyecto el autor asume cada uno de los componentes, en consecuencia, el costo de los recursos humanos es de \$0.00

Gastos varios

DESCRIPCIÓN	CANTIDAD	MESES	COSTO	SUBTOTAL
Energía eléctrica	1	3	\$40.00	\$120.00
Transporte	1	3	\$20.00	\$60.00
Internet	1	3	\$40.00	\$120.00
Suministros de oficina	1	1	\$40.00	\$40.00
TOTAL				\$340.00

Tabla 16 Factibilidad económica de gastos varios

El costo de los gastos varios para el desarrollo del proyecto es de \$340.00. Es este proyecto el autor asume cada uno de los componentes, en consecuencia, el costo de los recursos humanos es de \$0.00

Costo total

DESCRIPCIÓN	SUBTOTAL
Hardware	\$500.00
Software	\$0.00
Recursos humanos	\$4,900.00
Gastos varios	\$340.00
TOTAL	\$5,740.00

Tabla 17 Factibilidad económica total

El costo total para el desarrollo del proyecto es de \$5,740.00. Es este proyecto el autor asume cada uno de los componentes, en consecuencia, el costo de los recursos humanos es de \$0.00

2.7. Resultados

2.7.1. Pruebas de funcionalidad

Para verificar el correcto funcionamiento de la aplicación se realizaron las siguientes pruebas:

- Comprobación de la entrada de imágenes de diferentes formatos al servidor
- Validación de credenciales de acceso a la imagen en el servidor
- Comprobación de la entrega de imágenes del servidor
- Comprobación de eliminación de una imagen en el servidor

PRUEBA DE FUNCIONALIDAD			
Nombre:	Subir una imagen con extensión jpeg al servidor	#	1
Objetivo:	Comprobar si el módulo de cargar imagen funciona correctamente con imágenes de extensión jpeg		
Condiciones:	El usuario debe enviar un formulario con un parámetro de tipo binario con el nombre “ <i>image</i> ” con una imagen de extensión jpeg		
PASOS			
<ol style="list-style-type: none"> 1. Obtener el punto final de ingreso de imagen 2. Crear un formulario con un parámetro binario llamado “<i>image</i>” 3. Seleccionar una imagen con extensión jpeg 			

4. Enviar el formulario al servidor				
Resultados obtenidos				
Resultados esperados		Estado		
El usuario carga la imagen al servidor y debe obtener el enlace de acceso y el token para poder eliminarlo		Éxito	x	Fallo

Tabla 18 Prueba de funcionalidad #1

PRUEBA DE FUNCIONALIDAD			
Nombre:	Subir una imagen con extensión .png al servidor	#	2
Objetivo:	Comprobar si el módulo de cargar imagen funciona correctamente con imágenes de extensión .png		
Condiciones:	El usuario debe enviar un formulario con un parámetro de tipo binario con el nombre “ <i>image</i> ” con una imagen de extensión .png		
PASOS			
<ol style="list-style-type: none"> 1. Obtener el punto final de ingreso de imagen 2. Crear un formulario con un parámetro binario llamado “<i>image</i>” 3. Seleccionar una imagen con extensión png 4. Enviar el formulario al servidor 			
Resultados obtenidos			
Resultados esperados		Estado	
El usuario carga la imagen al servidor y debe obtener el enlace de acceso y el token para poder eliminarlo	Éxito	x	Fallo

Tabla 19 Prueba de funcionalidad #2

PRUEBA DE FUNCIONALIDAD			
Nombre:	Subir una imagen con extensión .gif al servidor	#	3
Objetivo:	Comprobar si el módulo de cargar imagen funciona correctamente con imágenes de extensión .gif		
Condiciones:	El usuario debe enviar un formulario con un parámetro de tipo binario con el nombre “ <i>image</i> ” con una imagen de extensión .gif		
PASOS			
<ol style="list-style-type: none"> 1. Obtener el punto final de ingreso de imagen 2. Crear un formulario con un parámetro binario llamado “<i>image</i>” 3. Seleccionar una imagen con extensión gif 4. Enviar el formulario al servidor 			
Resultados obtenidos			
Resultados esperados		Estado	
El usuario carga la imagen al servidor y debe obtener el enlace de acceso y el token para poder eliminarlo	Éxito	x	Fallo

Tabla 20 Prueba de funcionalidad #3

PRUEBA DE FUNCIONALIDAD			
Nombre:	Subir una imagen con extensión webp al servidor	#	4
Objetivo:	Comprobar si el módulo de cargar imagen funciona correctamente con imágenes de extensión webp		
Condiciones:	El usuario debe enviar un formulario con un parámetro de tipo binario con el nombre “ <i>image</i> ” con una imagen de extensión webp		
PASOS			
<ol style="list-style-type: none"> 1. Obtener el punto final de ingreso de imagen 2. Crear un formulario con un parámetro binario llamado “<i>image</i>” 3. Seleccionar una imagen con extensión webp 4. Enviar el formulario al servidor 			
Resultados obtenidos			
Resultados esperados		Estado	
El usuario carga la imagen al servidor y debe obtener el enlace de acceso y el token para poder eliminarlo	Éxito	x	Fallo

Tabla 21 Prueba de funcionalidad #4

PRUEBA DE FUNCIONALIDAD			
Nombre:	Comprobar el acceso a un recurso protegido (imagen privada) con token	#	5
Objetivo:	Verificar el funcionamiento del módulo de autenticación al acceso de un recurso protegido		
Condiciones:	El usuario debe ingresar al enlace devuelto por el punto final de “ingresar” una imagen		
PASOS			
<ol style="list-style-type: none"> 1. Abrir el navegador 2. Ingresar al enlace generado 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario ingresa al enlace en el navegador, si el hash de acceso la imagen la devuelta, de lo contrario, no carga	Éxito	x	Fallo

Tabla 22 Prueba de funcionalidad #5

PRUEBA DE FUNCIONALIDAD			
Nombre:	Comprobar el acceso a un recurso protegido (imagen privada) con token	#	6
Objetivo:	Verificar el funcionamiento del módulo de autenticación al acceso de un recurso protegido		
Condiciones:	El usuario debe ingresar al enlace devuelto por el punto final de “ingresar” una imagen		
PASOS			
<ol style="list-style-type: none"> 1. Abrir el navegador 2. Ingresar al enlace generado 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario ingresa al enlace en el navegador, si el hash de acceso la imagen la devuelta, de lo contrario, no carga.	Éxito	x	Fallo

Tabla 23 Prueba de funcionalidad #6

PRUEBA DE FUNCIONALIDAD			
Nombre:	Comprobar el acceso a una imagen con el formato jpeg	#	7
Objetivo:	Verificar el funcionamiento del módulo “obtener imagen” con el formato jpeg		
Condiciones:	El usuario debe ingresar al enlace devuelto por el punto final de ‘ingresar una imagen’, con el parámetro ‘format’.		
PASOS			
<ol style="list-style-type: none"> 1. Abrir el navegador 2. Ingresar al enlace generado y adiciona la variable <i>query</i> ‘format’ con contenido ‘jpeg’. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario ingresa al enlace en el navegador, si el formato de la imagen en soportado la imagen es devuelta, de lo contrario, no carga.	Éxito	x	Fallo

Tabla 24 Prueba de funcionalidad #7

PRUEBA DE FUNCIONALIDAD			
Nombre:	Comprobar el acceso a una imagen con el formato png	#	8
Objetivo:	Verificar el funcionamiento del módulo “obtener imagen” con el formato png		
Condiciones:	El usuario debe ingresar al enlace devuelto por el punto final de ‘ingresar una imagen’, con el parámetro ‘format’.		
PASOS			
<ol style="list-style-type: none"> 1. Abrir el navegador 2. Ingresar al enlace generado y adiciona la variable <i>query</i> ‘format’ con contenido ‘png’. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario ingresa al enlace en el navegador, si el formato de la imagen en soportado la imagen es devuelta, de lo contrario, no carga.	Éxito	x	Fallo

Tabla 25 Prueba de funcionalidad #8

PRUEBA DE FUNCIONALIDAD			
Nombre:	Comprobar el acceso a una imagen con el formato gif	#	9
Objetivo:	Verificar el funcionamiento del módulo “obtener imagen” con el formato gif		
Condiciones:	El usuario debe ingresar al enlace devuelto por el punto final de ‘ingresar una imagen’, con el parámetro ‘format’.		
PASOS			
<ol style="list-style-type: none"> 1. Abrir el navegador 2. Ingresar al enlace generado y adiciona la variable <i>query</i> ‘format’ con contenido ‘gif’. 			
Resultados obtenidos			
Resultados esperados	Estado		
El usuario ingresa al enlace en el navegador, si el formato de la imagen en soportado la imagen es devuelta, de lo contrario, no carga.	Éxito	x	Fallo

Tabla 26 Prueba de funcionalidad #9

PRUEBA DE FUNCIONALIDAD			
Nombre:	Comprobar el acceso a una imagen con el formato webp	#	10
Objetivo:	Verificar el funcionamiento del módulo “obtener imagen” con el formato webp		
Condiciones:	El usuario debe ingresar al enlace devuelto por el punto final de ‘ingresar una imagen’, con el parámetro ‘format’.		
PASOS			
<ol style="list-style-type: none"> 1. Abrir el navegador 2. Ingresar al enlace generado y adiciona la variable <i>query</i> ‘format’ con contenido ‘webp’. 			
Resultados obtenidos			
Resultados esperados		Estado	
El usuario ingresa al enlace en el navegador, si el formato de la imagen en soportado la imagen es devuelta, de lo contrario, no carga.	Éxito	x	Fallo

Tabla 27 Prueba de funcionalidad #10

Comparativa de tiempo en formatos de almacenamiento.

Se realizaron las pruebas necesarias de los tiempos de respuesta del formato de almacenamiento de archivos en relación con el formato de almacenamiento de objetos con las mismas imágenes.

N.º	T. con almacenamiento de archivos (ms)	T. con almacenamiento de objetos (ms)	% ahorro de tiempo
1	53	33	38

2	48	24	50
3	59	43	27
4	63	23	63
5	53	25	53
6	59	38	36
7	57	37	35
8	58	44	24
9	64	30	53
10	48	34	29
Promedio de ahorro de tiempo:			41

Tabla 28 Comparativa de tiempo en formatos de almacenamiento

2.7.2. Resultados finales

Subir una imagen

Una vez definido el problema que la institución de educación básica "Lcda. Angélica Villón Lindao", se definió los requerimientos y se propuso soluciones. Se realizaron las pruebas necesarias para asegurar el funcionamiento del proyecto, y se obtuvieron los siguientes resultados.

N.º	Rendimiento con formato JPEG		
	Tamaño original (kB)	Tamaño en el servidor (kB)	Índice de compresión (%)
1	2510 kB	866 kB	65%
2	582 kB	199 kB	66%
3	2550 kB	560 kB	78%
4	418 kB	104 kB	75%
5	995 kB	864 kB	13%
Promedio de índice de reducción			59.5%

Tabla 29 Promedio del porcentaje de reducción de imágenes JPEG

En la Tabla 29 muestra los resultados del índice de reducción de las imágenes con formato JPEG y su media de índice de reducción, indica que cumple con creces la hipótesis de reducción de tamaño.

N.º	Rendimiento con formato PNG		
	Tamaño original (kB)	Tamaño en el servidor (kB)	Índice de compresión (%)
1	444 kB	440 kB	1%
2	5340 kB	1990 kB	64%
3	28 kB	6 kB	79%
4	259 kB	137 kB	47%
5	33 kB	32 kB	3%
Promedio de reducción			39%

Tabla 30 Promedio del porcentaje de reducción de imágenes PNG

En la Tabla 30 muestra el índice de reducción de la imagen en formato PNG y el resultado del índice de reducción promedio, lo que indica que cumple con la hipótesis de reducción de tamaño.

N.º	Rendimiento con formato GIF		
	Tamaño original (kB)	Tamaño en el servidor (kB)	Índice de compresión (%)
1	2490 kB	2390 kB	4%
2	2530 kB	2510 kB	1%
3	804 kB	799 kB	1%
4	5170 kB	5080 kB	2%
5	96 kB	90 kB	6%
Promedio de reducción			3%

Tabla 31 Promedio del porcentaje de reducción de imágenes GIF

La Tabla 31 muestra los resultados del índice de reducción y el índice de reducción promedio para imágenes en formato PNG, indicando que no cumple con los supuestos de reducción de tamaño, pero, esto no afecta la solución porque las imágenes GIF no se utilizan en la organización.

N.º	Rendimiento con formato WEBP
-----	------------------------------

	Tamaño original (kB)	Tamaño en el servidor (kB)	Índice de compresión (%)
1	28 kB	26 kB	6%
2	268 kB	251 kB	6%
3	16 kB	11 kB	31%
4	32 kB	27 kB	16%
5	96 kB	86 kB	10%
Promedio de reducción			14%

Tabla 32 Promedio del porcentaje de reducción de imágenes WEBP

La Tabla 32 muestra los resultados del índice de reducción y el promedio de índice de reducción refleja que está dentro del supuesto de reducción de tamaño.

Formato	Rendimiento
	Tamaño reducido (%)
JPEG	59.5%
PNG	39%
GIF	3%
WEBP	14%
Promedio de reducción	29%

Tabla 33 Promedio del porcentaje de reducción total de imágenes

Según los resultados de la Tabla 33, el sistema funciona mejor al reducir el tamaño de las imágenes en formato JPEG con una reducción en promedio del 59% en relación con su tamaño original, seguida de una reducción en promedio del 39% del tamaño de las imágenes PNG, además, esto refleja que se puede ahorrar hasta un 29%, en promedio, de espacio de almacenamiento, cumpliendo así con los supuestos de reducción.

Obtener una imagen

N.º 1	Resultados en la obtención de una imagen JPEG			
	Formato de original	Formato de entrega	Tamaño de archivo	Tiempo de respuesta
	jpeg	jpeg	33.87 kB	33 ms

jpeg	png	330.55 kB	490 ms
jpeg	webp	34.01 kB	100 ms
jpeg	gif	180.66 kB	167 ms
Tiempo de respuesta promedio:			198 ms

Tabla 34 Resultados en la obtención de una imagen JPEG

La Tabla 34 muestra los resultados de convertir imagen JPEG de 33,87 kB a diferentes formatos (como png, webp y gif), así como su tiempo de respuesta y tamaño resultante.

N.º 2		Resultados en la obtención de una imagen PNG	
Formato original	Formato de entrega	Tamaño de archivo	Tiempo de respuesta
png	jpeg	94.33 kB	1987 ms
png	png	446.99 kB	54 ms
png	webp	181.73 kB	731 ms
png	gif	651.68 kB	1113 ms
Tiempo de respuesta promedio:			971 ms

Tabla 35 Resultados en la obtención de una imagen PNG

La Tabla 35 muestra los resultados de convertir una imagen PNG de 446,99 kB a diferentes formatos (como jpeg, webp y gif), así como su tiempo de respuesta y tamaño resultante. El resultado de convertir una imagen png a cualquier otro formato perderá su transparencia.

N.º 3		Resultados en la obtención de una imagen WEBP	
Formato original	Formato de entrega	Tamaño de archivo	Tiempo de respuesta
webp	jpeg	23.93 kB	306 ms
webp	png	203.47 kB	259 ms
webp	webp	26.25 kB	51 ms
webp	gif	100.01 kB	141 ms
Tiempo de respuesta promedio:			189 ms

Tabla 36 Resultados en la obtención de una imagen WEBP

La Tabla 36 muestra los resultados de convertir una imagen WEBP de 26,25 kB a diferentes formatos (como jpeg, png y gif), así como su tiempo de respuesta y tamaño resultante.

N.º 4	Resultados en la obtención de una imagen GIF		
Formato original	Formato de entrega	Tamaño de archivo	Tiempo de respuesta
gif	jpeg	2.6 kB	49 ms
gif	png	17.72 kB	49 ms
gif	webp	4.66 kB	38 ms
gif	gif	98.58 kB	37 ms
Tiempo de respuesta promedio:			33 ms

Tabla 37 Resultados en la obtención de una imagen GIF

La Tabla 37 muestra los resultados de convertir una imagen GIF de 98,58 kB a diferentes formatos (como jpeg, png y webp), así como su tiempo de respuesta y tamaño resultante. El resultado de convertir una imagen GIF a cualquier otro formato perderá su animación.

2.8. Conclusiones

Una vez implementado el servidor web de almacenamiento de imágenes y realizadas las pruebas correspondientes, se concluye que:

- Mediante la utilización de la estrategia de gestión y manipulación de almacenamiento basados en unidades diferenciales denominadas objetos en el repositorio, se logra obtener un ahorro de 40% tiempo de respuesta en la lectura de las imágenes, lo que se puede evidenciar en las pruebas de la comparativa de los formatos de almacenamiento.
- Con la metodología empleada se logró determinar que el uso de las librerías: *jpegoptim*, *optipng*, *pngquant*, *gifsicle*, *webp*, logran reducir hasta un 30% del tamaño original de las imágenes con formato JPEG y PNG en el servidor y para las imágenes con formato GIF y WEBP no se pudieron encontrar cambios significativos en la reducción del tamaño de la imagen.
- Al implementar las bibliotecas basadas en *jpegoptim*, *optipng*, *pngquant*, *gifsicle*, *webp*, el tamaño de la imagen se pudo reducir entre un 5% y un 30% sin afectar

la calidad. Sin embargo, después de la reducción del 50%, se determinó que se puede alterar la calidad de la imagen ante un observador humano.

- Las librerías utilizadas eliminaron los metadatos y redujeron la calidad de la imagen. Mientras la imagen generada sea mostrada al usuario final, esta reducción resulta imperceptible al ojo humano. De otra manera, si la finalidad es utilizar las imágenes en un proceso que comprometa el tratamiento de imágenes, la pérdida de calidad y valor de la imagen es evidente.

2.9. Recomendaciones

- Se recomienda el uso de herramientas de *Deep Learning* como *SRGAN*, combinadas con las librerías antes mencionadas para que el índice de reducción supere el 50% sin obtener pérdida en la calidad de la imagen, e incluso poder revertir la reducción de la calidad.
- Se recomienda el uso de compresión de imágenes de velocidad variable con redes neuronales recurrentes para disminuir el tamaño en disco de una imagen y mantener la calidad de la imagen para el uso en procesos que involucren el tratamiento de imágenes.
- Se recomienda el uso de una herramienta de *Machine Learning* y Vision Artificial como *OpenCV* para realizar la conversión de formatos de imágenes y obtener mejores resultados y tiempos de respuesta en la conversión de un formato de imagen a otro formato de imagen.

2.10. Bibliografías

- [1] WPSCAN, «Carga de archivos arbitrarios no autenticados a RCE,» [En línea]. Available: <https://wpscan.com/vulnerability/8f24e74f-60e3-4100-9ab2-ec31b9c9cdea>.

- [2] Mexis, «¿QUÉ IMPACTO TIENE EN TU EMPRESA QUE TENGAS UNA BRECHA DE SEGURIDAD?,» [En línea]. Available: <https://revista.mexis.net/que-impacto-tiene-en-tu-empresa-que-tengas-una-brecha-de-seguridad/>.
- [3] A. D. G. Solís, *Sistema de digitalización y almacenamiento de imágenes radiológicas*, Punta Arenas: Universidad de Magallanes, 2004.
- [4] C. G. Díaz, *Sistema para el almacenamiento y transmisión de imágenes médicas, versión 3.0*, La habana: Universidad de las Ciencias Informáticas, 2012.
- [5] G. B. C. León, *Diseño e implementación de una aplicación DICOM cliente/servidor para el transporte de imágenes radiológicas digitales en un ambiente hospitalario.*, Guayaquil: Escuela Politécnica Nacional, 2016.
- [6] RockContent, «API rest: conoce la importancia de ese recurso para el desempeño de una página web,» [En línea]. Available: <https://rockcontent.com/es/blog/api-rest/>.
- [7] MDN contributors, «HTTP,» 18 03 2019. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTTP>.
- [8] Sedes de Jesuïtes Educació, «Qué es un framework en programación,» 14 05 2019. [En línea]. Available: <https://fp.uoc.fje.edu/blog/que-es-un-framework-en-programacion/>.
- [9] Red Hat, «¿Qué es una API?,» 03 03 2012. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>.
- [10] Gnome, «¿Qué son los tipos MIME?,» 28 02 2020. [En línea]. Available: <https://help.gnome.org/admin/system-admin-guide/stable/mime-types.html.es>.
- [11] Guid One, «¿Qué es un GUID?,» 03 03 2012. [En línea]. Available: <http://guid.one/guid>.
- [12] MDN Web Docs, «¿Qué es realmente JSON?,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>.

- [13 MDN Web Docs, «¿Qué es una URL?,» [En línea]. Available:
] https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_URL.
- [14 FACSISTEL, «Lineas de investigación,» [En línea]. Available:
] http://facsistel.upse.edu.ec/index.php?option=com_content&view=article&id=58&Itemid=463.
- [15 S. E. Vázquez-Moctezuma, «Tecnologías de almacenamiento de información en el
] ambiente digital,» *Revista de la Universidad de Costa Rica*, vol. 5, nº 2, p. 20, 2015.
- [16 Secretaría Técnica Planifica Ecuador, «Plan Nacional de Desarrollo 2017 – 2021
] Toda una Vida.,» [En línea]. Available: <https://www.planificacion.gob.ec/plan-nacional-de-desarrollo-2017-2021-toda-una-vida/>. [Último acceso: 31 Diciembre 2020].
- [17 R. H. Sampieri, *Metodología de la Investigación*, Mexico: Mc Graw Hil Education,
] 2014.
- [18 M. C. Mora, *Métodos de recolección para enfoques cualitativos*, San José:
] Universidad Estatal a Distancia Costa Rica , 2010.
- [19 J. E. A., «Compresor de Imágenes Médicas con formato DICOM mediante,»
] *REVISTA INFOCIENCIA*, vol. 10, p. 87, 2016.
- [20 Code Academy, «MVC: Model, View, Controller,» [En línea]. Available:
] <https://www.codecademy.com/articles/mvc>.
- [21 Smartbear, «What is an API Endpoint?,» [En línea]. Available:
] <https://smartbear.com/learn/performance-monitoring/api-endpoints/>.
- [22 Ubuntu, «Ubuntu Canonical,» Canonical, [En línea]. Available:
] <https://ubuntu.com/about>.
- [23 Apache, «httpd.apache.org,» 07 08 2020. [En línea]. Available:
] <https://httpd.apache.org/>.
- [24 php, «php.net,» 03 09 2020. [En línea]. Available:
] <https://www.php.net/manual/es/intro-what-is.php>.

[25 Sistemas, «Definición de DBMS,» [En línea]. Available:
] <https://sistemas.com/dbms.php>.

[26 MariaDB Foundation, «MariaDB,» [En línea]. Available: <https://mariadb.org/about/>.
]

[27 diagrams.net, «diagrams.net,» 08 07 2005. [En línea]. Available:
] <https://www.diagrams.net/about.html>.

[28 PhpStorm, «jetbrains.com,» 16 10 2020. [En línea]. Available:
] <https://www.jetbrains.com/es-es/phpstorm/>.

[29 Symfony, «symfony.com,» [En línea]. Available: <https://symfony.com/what-is-symfony>.

[30 Doctrine Project, «doctrine-project.org,» [En línea]. Available:
] <https://www.doctrine-project.org/>.

[31 Api Platform, «api-platform.com,» [En línea]. Available: <https://api-platform.com/docs/distribution/>.

[32 NetApp, «¿Qué es el almacenamiento de objetos?,» [En línea]. Available:
] <https://www.netapp.com/es/data-storage/storagegrid/what-is-object-storage/>.

[33 RedHat, «¿Qué es el open source?,» [En línea]. Available:
] <https://www.redhat.com/es/topics/open-source/what-is-open-source>.

[34 Deloitte, «¿Qué es la Industria 4.0?,» [En línea]. Available:
] <https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html>.

[35 Mozilla, «¿Qué es la seguridad de sitios web?,» [En línea]. Available:
] https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Website_security.

[36 Aplyca, «De sistemas monolíticos a microservicios,» [En línea]. Available:
] <https://www.aplyca.com/es/blog/aplicaciones-monoliticas-o-microservicios>.

[37 rockcontent, «CDN,» [En línea]. Available: <https://rockcontent.com/es/blog/cdn/>.
]

[38 PHP, «Procesamiento de imágenes y GD,» [En línea]. Available:
] <https://www.php.net/manual/es/book.image.php>.

[39 PHP, «Procesamiento de imágenes (ImageMagick),» [En línea]. Available:
] <https://www.php.net/manual/es/book.imagick.php>.

[40 J. V. Jiménez, Servidor de almacenamiento remoto, Madrid: Universidad Carlos III
] De Madrid, 2010.

[41 J. A. L. Cerezo, «Ciencia, Tecnología y Sociedad,» Organización, [En línea].
] Available: <https://rieoei.org/historico/oeivirt/rie18a02.htm>.

[42 Escuela y medios de comunicación, «CTS desde la perspectiva de la educación
] tecnológica,» La Revista Iberoamericana de Educación, [En línea]. Available:
<https://rieoei.org/historico/documentos/rie32a07.htm>.

ANEXOS

ENTREVISTA	
Entrevistador:	Johnny Camatón
Entrevistado:	Arq. Mauricio Yunda Villao
Objetivo:	Determinar el almacenamiento y tratamiento de imágenes digitales.
Fecha:	11/12/2020
Ubicación:	Calle 18 de agosto entre Paquisha y, Fausto Fajardo, Santa Elena
<ul style="list-style-type: none">• Describa brevemente la historia de la institución y como está formada.• ¿Qué objetivo tiene la institución actualmente?• ¿Quiénes son los miembros de la directiva?• ¿Cada cuánto tiempo se elige nueva directiva?• ¿Tienen en sus planes invertir en sistema para agilizar procesos?• ¿Cuál es el proceso que utilizan para archivar documentos?• ¿Cuál es el plan de respaldos para los documentos archivados?• ¿Qué plan de acción tiene si sufren alguna pérdida de documentos?• ¿Qué piensa acerca de almacenar documentos físicos como documentos digitales?• ¿Considera implementar proyectos que ayuden a resolver esta problemática?• ¿Cuál es su perspectiva de la institución a largo plazo?	

Anexo 1: Formato de entrevista



Anexo 2: Visión y Misión

POST http://192.168.0.72... GET http://192.168.0.72... + ... No Environment

http://192.168.0.72/image?privacy=public Save

POST http://192.168.0.72/image?privacy=public Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> image	giphy.gif	
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 201 Created Time: 204 ms Size: 390 B Save Response

```

1  [
2    "status": "ok",
3    "url": "/media/69cda701-2bc7-4f68-9f2b-8ac8f28e35ef?format=gif",
4    "token": "6f2e4831ce52eaa831d3bc8f03d6e6528612c39f5a"
5  ]

```

Find and Replace Console Bootcamp Runner Trash

Anexo 3 Cargar imagen al servidor

POST http://192.168.0.72... GET http://192.168.0.72... + ... No Environment

http://192.168.0.72/media/e9cda7b1-2bc7-4fb6-9d2b-6ac0fd73e3fe?format=gif

GET http://192.168.0.72/media/e9cda7b1-2bc7-4fb6-9d2b-6ac0fd73e3fe?format=gif

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 43 ms Size: 98.56 KB Save Response



Find and Replace Console Bootcamp Runner Trash

Anexo 4 Obtener una imagen del servidor

POST http://192.168.0.72... GET http://192.168.0.72... DEL http://192.168.0.72... + ... No Environment

http://192.168.0.72/media/e9cda7b1-2bc7-4fb6-9d2b-6ac0fd73e3fe?format=gif&token=6fe4a31ce52eaa831d3bc8fb3de6525612c30f6a

DELETE http://192.168.0.72/media/e9cda7b1-2bc7-4fb6-9d2b-6ac0fd73e3fe?format=gif&token=6fe4a31ce52eaa831d3bc8fb3de6525612c30f6a

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> format	gif			
<input checked="" type="checkbox"/> token	6fe4a31ce52eaa831d3bc8fb3de6525612c30f6a			
Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 73 ms Size: 268 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "status": "ok"
3  }

```

Find and Replace Console Bootcamp Runner Trash

Anexo 5 Eliminar una imagen del servidor

```
public function __construct(MediaRepository $mediaRepository)
{
    $this->mediaRepository = $mediaRepository;
}

public function get(string $id, array $params, ?string $hash): Response
{
    /** @var Media $media */
    $media = $this->mediaRepository->findOneByIdOrFail($id);
    $this->checkHash($media->getHash(), $hash);

    $list($format, $quality, $proportion) = $this->sanitizeParameter($params);

    if (!$this->checkFormat($format)) {
        return new Response();
    }

    $image = $this->getImage($media);
    $image = $this->prepareImage($image, $proportion, $format, $media, $quality);

    $response = new Response();
    $response->setContent($image);
    $response->headers->set('Access-Control-Allow-Origin', 'value: *');
    $response->headers->set('Content-Type', 'image/' . $format);
    $response->headers->addCacheControlDirective('max-age', 604800);
    $response->headers->addCacheControlDirective('must-revalidate', true);

    return $response;
}

private function prepareImage($image, $proportion, $format, Media $media, $quality = 75): string
```

Anexo 6 Codificación de la función de 'obtener imagen' en la interfaz de programación

```
// $imageFile = FileService::getFile($request, 'image', true); // form-data
// $image = file_get_contents($imageRequest); // type: string

$privacy = $this->checkPrivacy($privacy);

$image_size = getimagesize($file); // type: array

$mime = explode(separator: "/", $image_size['mime']);
$type = $mime[0];
$format = $mime[1];
$ext = $file->getClientOriginalExtension();
$size = filesize($file);
$width = $image_size[0];
$height = $image_size[1];
$hash = ($privacy == 'private') ? sha1(uniqid()) : null;

$media = new Media($type, $format, $ext, $size, $width, $height, $hash);

$this->saveImageLog($media);
$this->uploadImageToServer($media, $file);
$this->optimizeImageOnServer($media);

$url = (!$media->getHash()) ?
    "/media/{\$media->getId()}:format={\$media->getFormat()}" :
    "/media/{\$media->getId()}:hash={\$media->getHash()}:format={\$media->getFormat()}";

return new JsonResponse([
    "status" => "OK",
    "url" => $url,
    "token" => $media->getToken(),
], status: 201);
```

Anexo 7 Codificación de la función de 'cargar imagen' en la interfaz de programación.

The image shows a screenshot of an IDE (likely PhpStorm) with a project named 'amavis'. The file explorer on the left shows the project structure, including folders like 'bin', 'config', 'orm', 'mapping', 'packages', 'prod', 'test', 'cache', 'doctrine', 'doctrine_migrations', 'framework', 'nelmio_cor.yaml', 'routing', 'routes', 'bundles.php', 'preload.php', 'routes.yaml', 'services.yaml', 'migrations', 'public', 'src', 'Controller', 'Image', 'Upload.php', 'Media', 'DeleteImage.php', 'GetImage.php', 'Entity', 'EventListener', 'Exception', 'Repository', 'Service', 'File', and 'Media'. The main editor window displays the code for the 'DeleteImageService' class in the 'App\Service\Media' namespace. The code includes a constructor and a 'delete' method that returns a 'Response' object.

```
namespace App\Service\Media;

use Symfony\Component\HttpFoundation\Response;

class DeleteImageService
{
    private MediaRepository $mediaRepository;

    private FileUploader $fileUploader;

    public function __construct(MediaRepository $mediaRepository, FileUploader $fileUploader)
    {
        $this->mediaRepository = $mediaRepository;
        $this->fileUploader = $fileUploader;
    }

    public function delete(string $id, string $token, ?string $hash): Response
    {
        /** @var Media $media */
        $media = $this->mediaRepository->findOneByIdOrFail($id);

        $this->checkHash($media->getHash(), $hash);
        $this->checkToken($media->getToken(), $token);

        $this->removeImageLog($media);
        $this->removeImageFromServer($media);

        return new JsonResponse(["status" => "ok"]);
    }

    private function removeImageFromServer(Media $media): void
    {
    }
}
```

Anexo 8 Codificación de la función 'eliminar imagen' en la interfaz de programación.