Escuela Politécnica Superior de Jaén
Máster Universitario en Seguridad Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

HERRAMIENTA PARA ENCRIPTADO DE METADATOS DE IMÁGENES

Alumna: Parras Portillo, Ana Belén

Tutor: Prof. D. Juan José Cubillas Mercado
Tutor: Prof. D. Manuel Carlos Díaz Galiano
Dpto.: Informática

Noviembre, 2021

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don JUAN JOSÉ CUBILLAS MERCADO y Don MANUEL CARLOS DÍAZ GALIANO, tutores del Proyecto Fin de Máster titulado: HERRAMIENTA PARA ENCRIPTADO DE METADATOS DE IMÁGENES, que presenta ANA BELÉN PARRAS PORTILLO, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, NOVIEMBRE de 2021

El alumno:

PARRAS PORTILLO ANA BELEN - 26037975C
Firmado digitalmente por PARRAS PORTILLO ANA BELEN - 26037975C
Fecha: 2021.11.09 12:51:49 +01'00'

ANA BELÉN PARRAS PORTILLO

Los tutores:

CUBILLAS MERCADO, JUAN JOSE (AUTENTICACIÓN)
Firmado digitalmente por CUBILLAS MERCADO, JUAN JOSE (AUTENTICACIÓN)
Fecha: 2021.11.10 15:54:39 +01'00'

D.JUAN JOSÉ CUBILLAS MERCADO

DIAZ GALIANO MANUEL CARLOS - 26032999N
Firmado digitalmente por DIAZ GALIANO MANUEL CARLOS - 26032999N
Fecha: 2021.11.10 13:03:01 +01'00'

D. MANUEL CARLOS DÍAZ GALIANO

Índice

Tabla de Ilustraciones

Tabla de Tablas

Resumen

Este trabajo Fin de Máster –TFM– se fundamenta en el desarrollo de una herramienta para el encriptado de los metadatos que contienen las imágenes.

Por ello, una primera parte del trabajo se basa en entender mejor los métodos de cifrado de información Simétrica, Asimétrica e Híbrida y sus distintas aplicaciones en la protección de la información.

Por otro lado, es necesario comprender los tipos de formatos de las imágenes, el cómo opera la utilidad de la criptografía en este tipo de archivos, en cómo se bloquea la información esencial del archivo, y cómo permitir que los ususarios accedan a ésta.

Dada la importancia de proteger los datos personales para evitar que sean utilizados con una finalidad distinta para la cual se proporcionan, otra parte del proyecto aborda el estudio de todo lo referente a los metadatos en las imágenes, quienes proporcionan una cantidad de información que escapa al control de la administración, así como conocer herramientas que tengan la capacidad de procesarlos para eliminarlos, con el fin de dar mas seguridad en el uso de este tipo de formato.

El TFM está enmarcado en uno de los temas más actuales dentro del mundo de la informática, la Ciberseguridad. Pretende proteger el sistema de ataques digitales así como preservar la privacidad, salvaguardando toda la información contenida en los metadatos de las imágenes digitales.

Palabras claves

Metadatos; EXIF; Criptografía; Python; ASP.NET; Flask; JSON; Exiv2; Metodología ágil; Estándares; Gestión de usuarios; Protocolos; Seguridad; Imagen digital; Aplicación web;

Abstract

This Master Thesis is based on the development of a tool for the encryption of the metadata contained in the images.

Therefore, a first part of the work is based on a better understanding of the Symmetric, Asymmetric and Hybrid information encryption methods and their different applications in the protection of information.

On the other hand, it is necessary to understand the types of image formats, how the usefulness of cryptography operates in this type of files, how the essential information of the file is locked, and how to allow users to access it.

Given the importance of protecting personal data to prevent them from being used for a different purpose for which they are provided, another part of the project deals with the study of everything related to metadata in the images, which provide a lot of information beyond the control of the administration, as well as knowing tools that have the ability to process them to remove them, in order to provide more security in the use of this type of format.

The TFM is framed in one of the most current issues in the world of computing, Cybersecurity. It aims to protect the system from digital attacks as well as to preserve privacy, safeguarding all the information contained in the metadata of digital images.

Keywords

Metadata; EXIF; Cryptography; Python; ASP.NET; Flask; JSON; Exiv2; Agile methodology; Standards; User management; Protocols; Security; Digital images; Web application;

1. INTRODUCCIÓN

Los delitos informáticos avanzan al mismo ritmo que la tecnología sin entender el límite entre la propiedad privada y la lucha por el libre acceso a la información.

Los metadatos de una imagen incluyen información heterogénea, como las coordenadas GPS de la ubicación de la foto, la fecha y la hora en que se tomó, el tipo de cámara y la configuración del obturador, el software utilizado para editar la foto, etc.

Este proyecto pretende realizar un estudio de los metadatos de las imágenes tomadas a partir de drones, imágenes subidas a redes sociales, etc. y que el propietario, decida cuáles serán encriptados u ofuscados y cuáles permanecerán sin modificar para que se pueda realizar un análisis correcto de la imagen.

El foco de la investigación se centra en cómo funcionan los diferentes métodos de cifrado de información en este tipo de formatos para garantizar la integridad del archivo, permitiendo a los usuarios obtener información sin llegar a ser afectada su originalidad.

Otro punto importante es la investigación sobre las opciones de almacenamiento de datos para las aplicaciones web y habilitar los sitios para usar proveedores de identidades sociales para la funcionalidad de autenticación y autorización.

1.1. Agradecimientos

Este proyecto no hubiera sido posible sin el apoyo de varias personas e instituciones a las que quiero mostrar mi agradecimiento.

- En primer lugar, a mis tutores, D. Juan José Cubillas Mercado y D. Manuel Carlos Díaz Galiano por compartir sus conocimientos y el interés mostrado durante todo el proyecto.
- También quisiera agradecer a mis compañeros del Máster entre los que nos hemos apoyado para poder terminar esta etapa con éxito. A los docentes del Máster por compartir sus conocimientos y apoyo durante la realización de éste.
- Gracias a mi familia. A mi hijo por ser mi motor para ser mejor persona cada día, a mi marido por su cariño y apoyo moral para poder cumplir mis sueños y metas, y a mi hermano por el apoyo ofrecido durante esta etapa.

1.2. Justificación

Hoy en día se obtiene una gran cantidad de imágenes tanto con cámaras profesionales, como con teléfonos móviles, o con tabletas; a esto hay que sumar la disponibilidad de los drones, en su mayoría provistos de sensores de distintos tipos que permiten también la captura de imágenes, siendo tecnología emergente con un uso en constante crecimiento.

Uno de los grandes problemas de seguridad al compartir una imagen, es que ésta lleva una serie de metadatos, entre ellos, su geolocalización, que la sitúan en un lugar y día concreto. Como se ha indicado anteriormente, los drones forman parte de una tecnología cada vez más utilizada en ocio e investigación. Es precisamente en este último campo donde en ocasiones se presenta el problema. A menudo se precisa compartir imágenes con fines profesionales donde sus metadatos son fundamentales para poder llevar a cabo un análisis correcto de las mismas, como, por ejemplo: altitud del dron, cámara utilizada, orientación de la imagen, parámetros de vuelo, etc. Todos ellos parámetros claves para poder generar productos con fines métricos, por ejemplo. En este sentido, los metadatos también pueden contener otros parámetros no claves para poder trabajar con estos productos y que, por razones de seguridad y/o privacidad, interesen ser ocultados, sea el caso del día que fue tomada la imagen y su ubicación exacta.

1.3. Estructura del proyecto

La estructura con la que se define el proyecto desarrollado se enmarca en los siguientes bloques:

Tabla 1-1 Estructura del proyecto

Bloque 1	Se corresponde con el bloque actual, la introducción. En él se exponen tanto la justificación del proyecto, como los agradecimientos.
Bloque 2	En donde se ha explicado la motivación.
Bloque 3	Es el bloque que contiene los objetivos generales y específicos del proyecto.
Bloque 4	Se hace un estudio de la planificación del trabajo, del cronograma y del coste asociado que tendrá.
Bloque 5	En el que se explica el estado de la cuestión y la metodología ágil usada.
Bloque 6	En este bloque corresponde al estado del arte, donde se expone la investigación llevada a cabo sobre las metodologías, el procesamiento de los metadatos de las imágenes, las tecnologías y la seguridad en la información y en los sistemas informáticos.
Bloque 7	Se desarrolla con el fin de definir el análisis de requisitos del proyecto, el estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento de dichos requisitos. Es necesario detallar los requerimientos que debe cumplir el proyecto.
Bloque 8	En este apartado se expone el estudio llevado a cabo sobre las tecnologías necesarias para la implementación y seguridad del proyecto. Entre ellas, los lenguajes de programación, las librerías, los frameworks para el desarrollo de APIs y aplicaciones web, los protocolos de seguridad o la gestión de usuarios. También se analizan los resultados obtenios.
Bloque 9	Contiene las conclusiones obtenidas tras la realización del trabajo fin de máster.
Bloque 10	En dicho bloque se especifican los posibles trabajos futuros, así como las mejoras que se pueden aplicar al proyecto.
Bloque 11	Se incluye el manual de usuario y se explica cómo se debe interactuar con la aplicación web y los aspectos a tener en cuenta.
Anexos	En este bloque se expone el manual de instalación del sistema y sus componentes.
Bibliografía	Finalmente se recogen las referencias de todo el material consultado para documentar el proyecto.

Nota: Estructura del proyecto

Fuente: Elaboración propia

2. MOTIVACIÓN

Teniendo en cuenta el creciente aumento de dispositivos electrónicos/digitales, y unido esto a las posibilidades de conexión vía Internet y, por ende, la posibilidad de compartir datos –documentos, imágenes, audios, etc– surge un riesgo potencial de comunicación a través de dispositivos controlados a distancia (drones).

Resulta evidente en la actualidad, la posibilidad de compartir información mediante redes sociales. Como se comentaba anteriormente, con la utilización de estos dispositivos a distancia es posible obtener además información personal ajena.

Por tanto, es necesario trabajar para que no se vulneren derechos fundamentales de las personas y garantizar que se salvaguarden.

Mi vocación científica/investigadora y mi formación en ciberseguridad, me incentivan a trabajar en este proyecto con el fin de que no se vulneren derechos y deberes fundamentales reconocidos en nuestra Constitución art.18 [1] y se preserve el Derecho Informático, disciplina que engloba el conjunto de disposiciones jurídicas para la regulación de las nuevas tecnologías de la información y comunicación.

3. OBJETIVOS DEL PROYECTO

En este punto se establecen los objetivos que se pretenden alcanzar tras el desarrollo del proyecto. El objetivo principal es la implementación de la seguridad informática en un software capaz de permitir al usuario final, decidir qué metadatos de una imagen serán ofuscados y cuáles permanecerán sin modificar para que se pueda realizar un análisis correcto de la imagen.

3.1. Objetivos generales

- Examinar la funcionalidad de los metadatos en las imágenes digitales, qué son y para qué sirven.
- Analizar el funcionamiento del cifrado de información de las imágenes.
- Considerar los servicios de autenticación de usuario.

- Estudio previo de las tecnologías existentes y necesarias justificando la tecnología elegida.
- Investigar sobre los distintos frameworks de desarrollo de aplicaciones web y acreditación del modelo escogido.
- Análisis y diseño de una plataforma web y de una API.
- Elaboración de la documentación y pruebas del sistema completo.
- Generación de los manuales de usuario y de instalación.
- Desarrollo de la memoria del trabajo realizado.

3.2. Objetivos específicos

- Concretar como es el almacenamiento de los metadatos o bien, como se pueden guardar estos datos para poder conservarlos adecuadamente y de manera organizada.
- Entender la importancia de los metadatos en ciberseguridad.
- Comprender los principales tipos de cifrados existentes en criptografía, simétrico y asimétrico y su aplicación en el medio.
- Fijar los diferentes métodos de protección en las imágenes.
- Determinar cómo actúa el cifrado de la información de la imagen.
- Identificar tecnologías que permitan el ocultamiento/borrado de los metadatos.
- Integrar servicios de autenticación a través de la aplicación.
- Diseñar y desarrollar una API con las funcionalidades necesarias que ofrecerá el prototipo.
- Diseñar e implementar una plataforma web que permita seleccionar aquellos metadatos de una imagen con el fin de ofuscarlos o restablecer, según el interés del usuario.
- Redactar los manuales de instalación y de usuario y de la memoria que recogerá todo el trabajo desarrollado.

4. PLANIFICACIÓN DEL PROYECTO

La planificación y duración de las distintas actividades que componen el desarrollo del proyecto, así como las estimaciones de recursos humanos, técnicos financieros y materiales, quedan expuestas en este apartado.

4.1. Definición de tareas

El prototipo conlleva la realización de las siguientes tareas:

- Estudiar las distintas metodologías de ingeniería del software para la toma de decisiones en el desarrollo del software.
- Estudiar los metadatos de las imágenes digitales, su funcionalidad y su almacenamiento de forma segura y organizada.
- Identificar los cifrados existentes en Criptografía.
- Estudiar las distintas tecnologías para ocultar/borrar los metadatos de imágenes digitales.
- Estudio de las tecnologías existentes para el desarrollo de una API y una aplicación web.
- Diseño e implementación de la API.
- Diseño y desarrollo de la aplicación web.
- Integración de los servicios de autenticación a través de la aplicación.
- Testeo para la experiencia de usuario, calidad, estabilidad y seguridad.
- Desarrollo de los manuales de usuario y de instalación.
- Generación de la memoria.

4.2. Estimación de tiempos

Para llevar a cabo el desarrollo del proyecto, se estudia la duración de las diversas tareas necesarias para ello. En la tabla 4-1 se puede ver la estimación de cada una de ellas.

Tabla 4-1 Estimación de tiempos de ImageMetaData

Tarea	Duración/días	Fecha inicio	Fecha fin
Inicio del proyecto	1	24-5-2021	25-05-21
Estudio de metodologías ágiles	6	25-5-2021	31-05-21
Estudio de los metadatos y su correcto almacenamiento	13	31-5-2021	13-06-21
Estudio de los cifrados existentes en Criptografía	10	13-6-2021	23-06-21
Estudio de tecnologías para el ofuscamiento/borrado de los metadatos	7	23-6-2021	30-06-21
Estudio de las tecnologías existentes para la implementación de una API y una aplicación web	12	30-6-2021	12-07-21
Diseño e implementación de la API	27	12-7-2021	08-08-21
Diseño e implementación de la aplicación web	16	8-8-2021	24-08-21
Integración servicios de autenticación a través de la app	5	24-8-2021	29-08-21
Realización de pruebas de usuario	7	29-8-2021	05-09-21
Desarrollo de los manuales de usuario y de instalación	8	5-9-2021	13-09-21
Generación de la memoria	112	24-5-2021	13-09-21

Nota: Estimación de tiempos

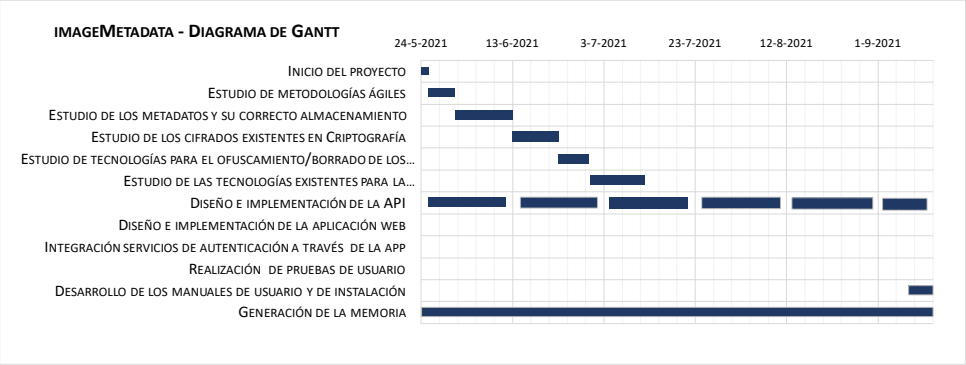
Fuente: Elaboración propia

4.3. Diagrama de Gantt

En el diagrama de Gantt de la ilustración 4.1, se visualizan los plazos del proyecto y las fechas de inicio y fin, tanto de éste, como de los distintos sprints. Se muestran explícitamente los requisitos que se abordarán en el desarrollo iterativo e incremental, del diseño, implementación y pruebas –sin indicar en qué sprint–.

El proyecto tendrá como fecha de inicio el 24 de mayo de 2021 y de finalización el 13 de septiembre de 2021, con lo cual tendrá una duración estimada de 112 días.

Ilustración 4.1 Diagrama de Gantt de ImageMetaData



Nota: Diagrama de Gantt

Fuente: Elaboración propia

4.4. Análisis de costes

Para desarrollar una estimación aproximada de los recursos monetarios necesarios para completar las actividades del proyecto es necesario hacer una distinción entre los costes asociados al hardware, al hardware y al personal.

4.4.1. Hardware

Portátil MSI Summit E14	1800 €

La vida útil del ordenador se estima en seis años y al final de este período el valor será 0 €. Así, el valor estimado será de 300 € al año o 25 € al mes. Supuesta la duración del proyecto ImageMetaData de aproximadamente 4 meses, el coste total ascendería a 100 €.

4.4.2. Software

Sistema Operativo Windows 10: conjunto de programas del sistema informático para coordinar los recursos físicos, software y la interfaz de usuario.

Microsoft Office Word: herramienta de procesamiento de textos utilizada para la creación de la memoria.

Microsoft Office Excel: herramienta de tratamiento de hojas de cálculo usada para la creación de documentación y tablas.

Python: lenguaje de programación usado para la implementación de la API de este proyecto.

C#: lenguaje de programación usado para la implementación de la web de este proyecto.

Exiv2: utilidad de línea de comando para la gestión de metadatos en imágenes.

Microsoft Identity: servicio de autenticación basada en estándares, bibliotecas de código abierto y herramientas de administración de aplicaciones usadas en el desarrollo de este proyecto .

SendGrid: es un servicio en la nube para mandar correos electrónicos.

Visual Studio Code: editor de código fuente para la implementación de la API.

Visual Studio Community 2019: entorno de desarrollo integrado para crear la aplicación web del proyecto con asp.net core.

GitHub: herramienta de gestión de proyectos usada para el desarrollo ágil de software mediante el método visual KanBan.

Google Drive: servicio de alojamiento gratuito de archivos online.

Visual Paradigm online: editor de diagramas de representación case para UML.

Free DNS Hosting: herramienta para la realización de pruebas de usuario del proyecto.

S. O. Windows10 (licencia incluida en el portátil)	0 €
Microsoft Office 365	8,25 €/mes x 4 meses = 33,00 €
Python	0 €
C#	0 €
Exiv2	0 €
Microsoft Identity	0 €
SendGrid	0 €
Visual Studio Code	0 €
Visual Studio 2019	0 €
GitHub	0 €
Google drive	0 €
Visual Paradigm online free	0 €
Free DNS Hosting (servidor)	0 €

4.4.3. Personal

El desarrollo de este proyecto se lleva a cabo por una persona, a la cual se le considera analista de sistemas. El Analista Programador Computacional se desempeña en el área de las tecnologías de información siendo capaz de analizar, diseñar, desarrollar, implementar y asegurar la continuidad de los sistemas computacionales, velando por el correcto funcionamiento de dichos sistemas y aplicaciones, como también de integrar y adaptar sistemas existentes, tal y como se define en [2].

En la resolución del convenio estatal de empresas, en el BOE –Boletín Oficial del Estado– a 10 de marzo de 2021, Núm. 59, Sec III, Pág. 27861, la tabla salarial del grupo profesional I, en [3], la remuneración mensual establecida para un analista programador de sistemas asciende a 1710,28 €. Se ha estimado una duración de 4

meses el diseño e implementación de este proyecto, por lo que el coste total en cuanto a personal se refiere, asciende a 6841,12 €.

4.4.4. Coste total

Para la obtención del coste total se ha de tener en cuenta tanto los costes hardware, software y de personal, como el porcentaje de beneficio. En la tabla 4-2 se muestra el detalle de este cálculo.

Tabla 4-2 Coste del proyecto ImageMetaData

Desglose del coste del proyecto ImageMetaData	
Coste hardware	100,00 €
Coste software	33,00 €
Coste de personal	6.841,12 €
Coste total	6.974,12 €
Beneficio estimado: 30 %	2.092,24 €
Coste total del proyecto	**9.066,36 €**

Nota: Coste del proyecto

Fuente: Elaboración propia

5. METODOLOGÍA

Siguiendo el conjunto de normas internacionales contenidas en la ISO/IEC 3300 [4] “Calidad de los procesos de **desarrollo de software**”, es necesario la creación de un software confiable, eficaz y eficiente, adecuado a las necesidades reales de los usuarios. Así, haciendo uso de la Ingeniería del Software, se lleva a cabo un estudio y posterior elección y justificación de la metodología adecuada para el desarrollo de este proyecto.

Tal y como se explica en [5, pp. 41-42], en la actualidad se pueden diferenciar dos grandes grupos de metodologías de desarrollo de software: las ágiles y las tradicionales.

Las metodologías de desarrollo de *software tradicionales* se caracterizan por:

- Definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de software.

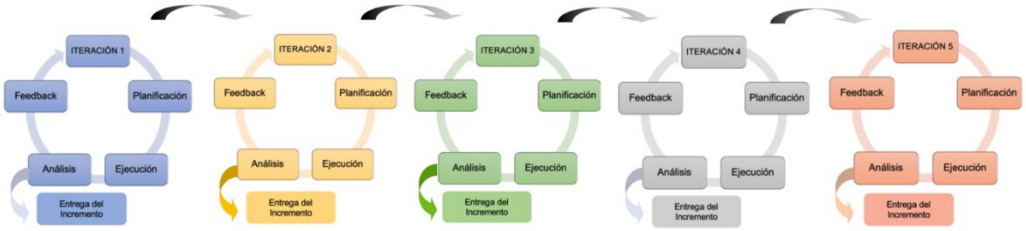
- Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios.
- La organización del trabajo es lineal, es decir, las etapas se suceden unas tras otra y no se puede empezar la siguiente sin terminar la anterior.
- No se puede volver hacia atrás una vez se ha cambiado de etapa, no se adaptan bien a los cambios.

Las metodologías *ágiles de desarrollo de software* se caracterizan por:

- Basarse en la metodología incremental –ver ilustración 5.1–, en la que en cada ciclo de desarrollo se van agregando nuevas funcionalidades a la aplicación final.
- Los ciclos son mucho más cortos y rápidos, por lo que se van agregando funcionalidades en lugar de grandes cambios.
- Permitir construir equipos de trabajo autosuficientes e independientes que se reúnen cada poco tiempo para poner en común las novedades.
- Poco a poco se va construyendo y puliendo el producto final, a la vez que el cliente puede ir aportando nuevos requerimientos o correcciones al poder comprobar cómo avanza el proyecto en tiempo real.

Debido a todas las ventajas expuestas y a su alta flexibilidad y agilidad, se opta por aplicar una metodología ágil para el desarrollo de este proyecto. Es por ello que se lleva a cabo un estudio de las principales metodologías ágiles.

Ilustración 5.1 Desarrollo iterativo e incremental



Nota: Metodología ágil: desarrollo iterativo incremental
Fuente: Elaboración propia

5.1. Metodología ágil empleada

En consonancia con lo expuesto en al anterior apartado, el Manifiesto Ágil –ver en [6]– es la base del estudio de la metodología ágil. Es un documento redactado en 2001 por 17 expertos en programación que supuso un cambio en la forma de desarrollar software frente a los modelos tradicionales –la filosofía que promueve este manifiesto es extensible al desarrollo de cualquier otro producto además del desarrollo de software–.

Los valores definidos en el Manifiesto Ágil no se centran en prácticas, metodologías o procedimientos de trabajo, sino que abogan por un cambio de mentalidad, una nueva cultura organizativa en cuatro pilares:

I. Individuos e interacciones sobre procesos y herramientas.
II. Software funcionando sobre documentación extensiva.
III. Colaboración con el cliente sobre negociación contractual.
IV. Respuesta ante el cambio sobre seguir un plan.

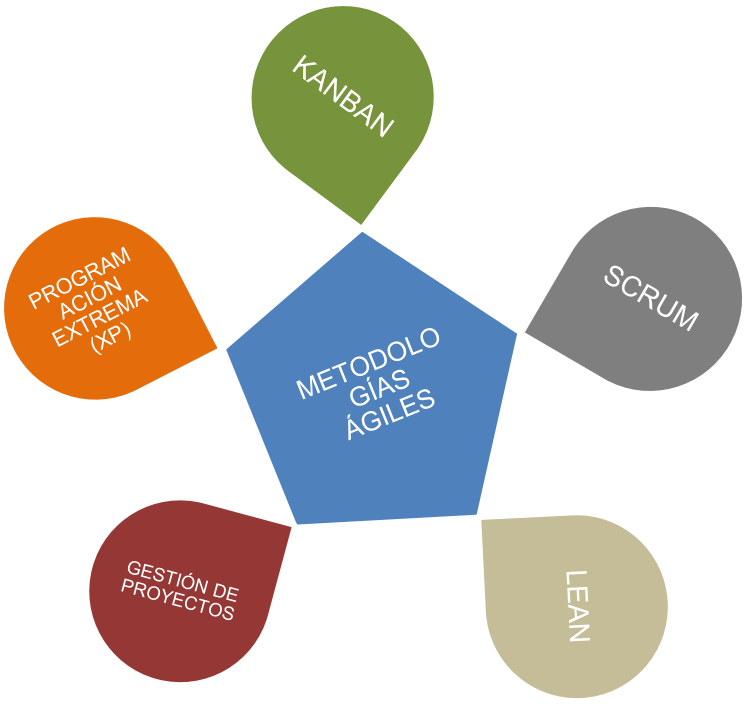
Las principales metodologías ágiles, mostradas en la ilustración 5.2, son:

- *KanBan*: metodología de trabajo inventada por la empresa de automóviles Toyota. Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo dividido en tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual basado en tareas prioritarias e incrementando el valor del producto.
- *Scrum*: es también una metodología incremental que divide los requisitos y tareas de forma similar a Kanban. Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas) para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (planning sprint), ejecución (sprint), reunión diaria (daily meeting) y demostración de resultados (sprint review). Cada iteración por estas etapas se denomina también sprint.
- *Lean*: está configurado para que pequeños equipos de desarrollo muy capacitados elaboren cualquier tarea en poco tiempo. Los activos más importantes son las personas y su compromiso, relegando así a un segundo plano el tiempo y los

costes. El aprendizaje, las reacciones rápidas y potenciar el equipo son fundamentales.

- *Programación extrema (XP)*: es una metodología de desarrollo de software basada en las relaciones interpersonales, que se consideran la clave del éxito. Su principal objetivo es crear un buen ambiente de trabajo en equipo y que haya un feedback constante del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, testing, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación.
- *Gestión de proyectos*: los métodos ágiles suelen cubrir la gestión de proyectos. Algunas herramientas de gestión de proyectos están diseñadas para planificar, hacer seguimiento, analizar e integrar trabajo.

Ilustración 5.2 Metodologías ágiles



Nota: Metodologías ágiles: tipos

Fuente: Elaboración propia

La metodología ágil KanBan es un método visual que se utiliza para controlar las tareas a través de su división por fases hasta su finalización. Por su adecuación al escenario que ofrece este proyecto, se ha optado por utilizarla para su desarrollo.

En la ilustración 5.3 se muestra un ejemplo de tablero KanBan y sus componentes:

1. VISUALIZAR el trabajo. KanBan trata de visualizar en una pizarra la división de tareas. Los equipos escriben sus proyectos y elementos de trabajo en señales visuales como pueden ser adhesivos, tickets, post-it, etc. La pizarra tiene tantascolumnas como estados por los que puede pasar la tarea. Una de ellas suele ser el backlog, donde los clientes y los compañeros de equipo plantean ideas para proyectos.

Por tanto, con esta organización, se consigue que el trabajo a realizar quede claro y se pueda ver la información y prioridad de cada tarea de un vistazo.

2. LIMITAR el trabajo. Es importante definir el número máximo de tareas que se pueden realizar en cada fase. La idea principal de esta regla es que los trabajadores se centren en cerrar tareas, no empezar nuevas –se debe haber finalizado otra previa–.

3. MEDIR el tiempo. Se debe medir el tiempo que se tarda en cerrar cada tarea o “cycle time”, además de tener en cuenta otras métricas importantes como el tiempo desde que se empieza una tarea hasta que se entrega al cliente, “lead time”. Con Es necesario para controlar y mejorar continuamente.

Ilustración 5.3 Componentes de un tablero KanBan

Nota: Ejemplo tablero KanBan

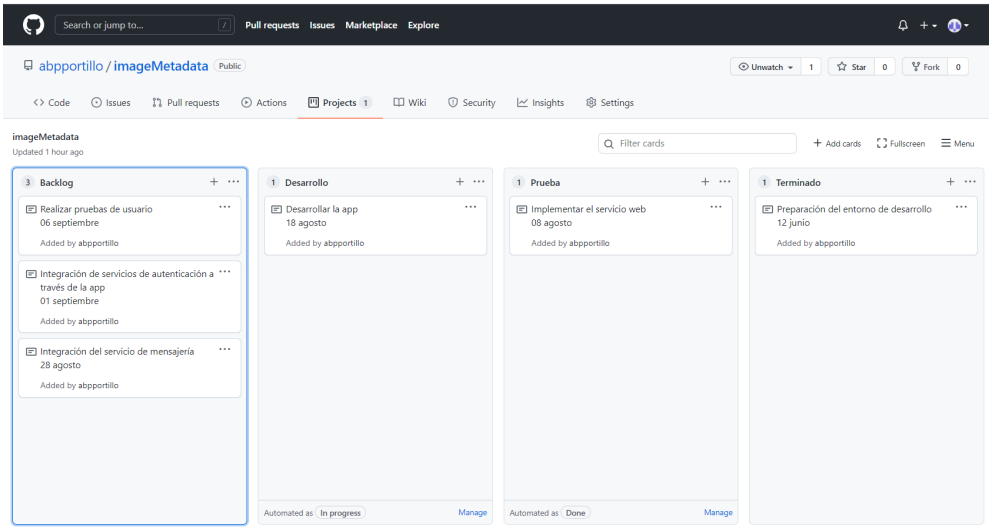
Fuente: Elaboración propia

En resumen, se opta por usar la metodología KanBan por los siguientes motivos:

- Es fácil de usar y aprender.
- Dada su representación a través de tarjetas, es una metodología visual, permitiendo ver el estado de los proyectos y tareas.
- Se puede implantar en el momento, no es necesario empezar un proyecto de nuevo.
- Optimiza el tiempo de implementación de las tareas y la calidad.
- Ofrece una mejora continua: aprovechando la realización de tareas se busca mejorar los procesos, a través de un sistema de mejora continua.
- Organiza y gestiona el trabajo eficientemente.
- Se dispone de capacidad de respuesta ante tareas no previstas, de forma que exista una "cola de espera" de tareas en las que ir priorizando su realización en función de las necesidades de cada momento y de la urgencia de cada una de ellas, dando así, flexibilidad.
- No requiere reuniones diarias con el cliente.

Una de las fases del tablero KanBan, ya que se desplegarán tareas que en este momento quedan ocultas, del prototipo de ImageMetaData, es el que se puede ver en la siguiente ilustración 5.4

Ilustración 5.4 Tablero KanBan de ImageMetaData



Nota: Una de las fases del tablero KanBan de ImagenMetaData

Fuente: Elaboración propia

6. CONTEXTO Y ESTUDIO PRELIMINAR

El estudio se centra en tres cuestiones principales, los metadatos de las imágenes, la seguridad de ellos y los servicios web.

6.1. Metadatos

Lo primero es entender qué son los metadatos. Pese al uso actual de la palabra metadatos en el ámbito de la informática, como se explica en [7], el concepto es anterior a la Internet. El término se acuñó en los años 60 para describir un conjunto de datos y es, a partir de 2004, cuando a causa de la gran diversidad y volumen de las fuentes y recursos en Internet, se hizo necesario establecer un mecanismo para etiquetar, catalogar, describir y clasificar los recursos presentes en la World Wide Web con el fin de facilitar la posterior búsqueda y recuperación de la información.

Este término, comienza a evolucionar hasta convertirse en lo que hoy conocemos, “datos acerca de los datos”. Información sobre los datos que tiene un documento, una imagen, un archivo o una página web. Así, los metadatos son etiquetas que sirven para describir el contenido de un recurso de información.

Los metadatos sólo son posibles en un contexto digital y en red ya que sólo dentro de este contexto se pueden utilizar con la función que les caracteriza, que es la de la localización, identificación y descripción de recursos legibles e interpretables por máquinas.

Hoy en día, los metadatos tienen muchos usos debido a que es un tipo de información muy flexible, pudiéndose ajustar a diversas tareas como, entre otras: facilitar búsquedas y análisis, mejorar el manejo de los datos, ayudar en la integración, facilitar la estandarización, ayudar a generar informes de una manera más eficiente, generar mayor confiabilidad y seguridad en la información, etc. El uso común más frecuente es la refinación de consultas a buscadores. De esta manera, se ayuda al usuario a encontrar unos resultados más precisos evitando perder tiempo buscando de manera manual. Por tanto, los metadatos facilitan el trabajo clasificando, organizando y estructurando los datos disponibles.

Si bien, el objetivo principal de los metadatos es recuperar información relevante y contextualizada, cada vez hay más usuarios más interesados por el metadato en sí que por el dato al que hace referencia.

Este “usuario de datos”, no utiliza los metadatos para encontrar un documento, sino que analiza esos metadatos a gran escala utilizando sus propios métodos e instrumentos de análisis de datos. La reutilización de estos datos sobre datos ofrece muchas posibilidades en diversos ámbitos.

Otra función de los metadatos es mantener una trazabilidad sobre el uso de los documentos. De esta manera, se puede saber quién ha añadido, consultado, editado o eliminado un documento. Estos metadatos, también denominados *trazas de auditoría*, se utilizan en un sistema de gestión documental para proteger datos confidenciales o particularmente sensibles.

Dependerá de los metadatos que se estén consultando que sirvan para una cosa u otra, para identificación, para ordenar y categorizar, para ubicar un documento o un archivo en el espacio.

6.1.1. Tipos de metadatos

Los metadatos se pueden crear automáticamente, de forma manual o de forma semiautomática –si fuera necesario añadir datos extra que no estén–.

Manualmente, puede llegar a ser un procedimiento un tanto complicado, aunque todo depende del formato que se utilice y del volumen que se esté buscando.

Lo normal es que se creen de forma automática, como es el caso de las imágenes o los documentos. El software que se utiliza para crear un archivo o un dron o la cámara o teléfono móvil para la imagen, registran automáticamente los metadatos.

Los modelos de metadatos se aplican, además de a textos, a imágenes: pinturas, fotografías, películas, etc.

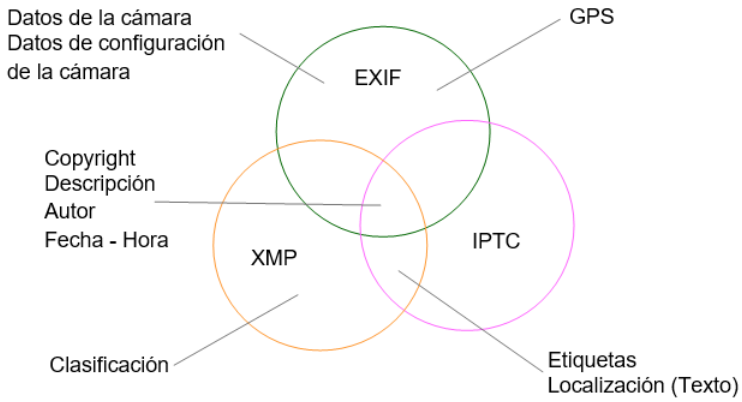
A continuación se indican los principales estándares que existen para aplicar a imágenes y GPS [8].

- EXIF –Exchangeable Image File Format– [9]: se trata de un estándar desarrollado por JEITA [10] – Japan Electronics and Information Technology Industries Association– y es el formato de metadatos más utilizado por las cámaras digitales. Define una serie de etiquetas –tag–, que describen las características de la cámara –fabricante, modelo, software, etc. –, y su configuración en el momento de captura de la imagen. Los metadatos EXIF también contienen las coordenadas de localización en caso de que la cámara disponga de GPS, así como otros metadatos descriptivos como título, autor, copyright, etc.
- IPTC –International Press Telecommunication Council– [11]. IPTC es un consorcio formado por las principales agencias de noticias y empresas de comunicación y se utiliza, sobre todo, para fotografías o noticias. En IPTC las organizaciones de la industria periodística desarrollan y mantienen estándares técnicos, para mejorar y homogeneizar el intercambio de noticias entre las agencias del mundo.
- XMP –Extensible Metadata Platform– [12]. Es un estándar abierto para metadatos en publicaciones que utiliza 3 esquemas específicos para describir fotografías –XML Basic Schema, XML Rights Management Schema y XMP Media Management Schema– . Además, incluye otros esquemas como DC, EXIF, etc. Se puede incluir paquete XMP en los formatos gráficos más conocidos como .jpeg, .gif, .tif, .psd, .eps, .png, etc [13]. XMP es un estándar que define un modelo para la creación y procesamiento de metadatos, basado en etiquetas XML –eXtensible Markup Language–. Este modelo utiliza un esquema de metadatos para almacenar propiedades básicas, y otro para que cada dispositivo o aplicación pueda almacenar su propia información. De este modo, cada aplicación podrá usar este método común para capturar y compartir sus metadatos.

Algunos tipos de metadatos son creados y utilizados en exclusiva por alguno de los estándares. En cambio, otros tipos de metadatos son implementados por más de un estándar.

En la ilustración 6.1 se muestran metadatos compartidos en los tres estándares y metadatos usados en exclusiva por alguno de ellos.

Ilustración 6.1 Metadatos de los distintos estándares



Nota: Metadatos compartidos en los tres estándares y metadatos usados en exclusiva por alguno de ellos

Fuente: Elaboración propia

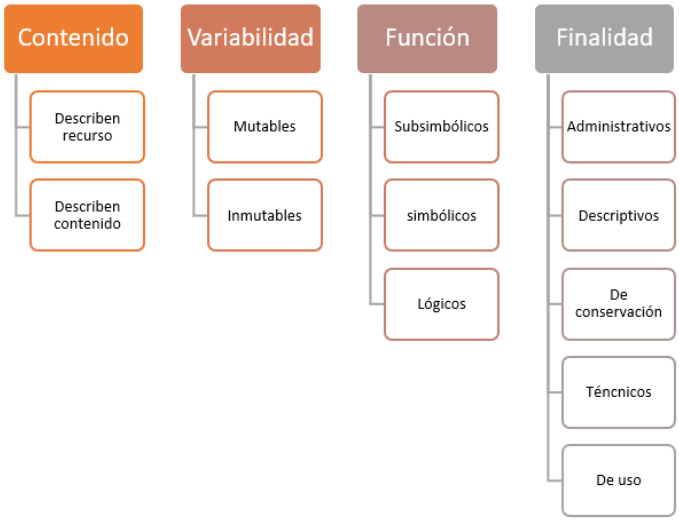
Además de estos estándares, cabe mencionar algunos otros como DIG35 –Digital Imaging Group–, PRISM [14] –Publishing Requeriemnts for Industry Standard Metadata–, JPX [15]–también llamado JPEG 2000, es una extensión del formato JP2 que se prevé que poco a poco irá reemplazando al formato .jpg–, y PHEED –extensión de la especificación RSS2.0–

6.1.2. Clasificación de los metadatos

Principalmente, los metadatos se caracterizan por ser un conjunto de datos altamente estructurados que se encargan de detallar las particularidades de los datos en función de su contenido, información, calidad y otros atributos. Además, éstos presentan diferenciaciones que dependerán de las reglas incluidas en las aplicaciones para establecer la estructura interna de los esquemas de datos. Pero, más allá de eso, los metadatos revelan otras características de suma importancia que facilitan su identificación.

Como se puede apreciar en la ilustración 6.2, las clasificaciones [16] por parte de los diferentes autores e instituciones son muy variadas, y se establecen atendiendo a los distintos aspectos a los que se dé prioridad a la hora de establecer dichas clasificaciones.

Ilustración 6.2 Clasificación de los metadatos



Nota: Clasificación de los metadatos

Fuente: Elaboración propia

En términos generales, los metadatos se definen como una herramienta que proporciona la ayuda requerida para dominar una notable cantidad de información, gracias a que permite organizarla para facilitar el trabajo y acelerar la productividad de los usuarios. Pero, más allá de eso, estos mecanismos se pueden definir de otras maneras, según su clasificación:

➢ Por su contenido: Se cataloga como la clasificación más usual de todas y, en este caso, los metadatos se fraccionan en función de su información.

Metadatos *independientes del contenido*: recogen la información que no depende del contenido del documento –localización, fecha de creación y actualización, seguimiento y control de versiones, etc. –

Metadatos *dependientes del contenido*: recogen la información que depende del contenido, ya sea de forma directa o indirecta. Este tipo de metadatos permite la

interoperabilidad semántica, ya se trate de dominios generales o específicos. Sumado a eso, estos dos grupos se pueden subdividir en otros subgrupos que únicamente dependen de la precisión con la que el usuario desee llevar la clasificación de los datos para cumplir con su cometido.

- Por su variabilidad: Contiene dos grupos específicos. El primero hace referencia a los metadatos que son inmutables y no cambian, independientemente de la parte del recurso que sea visible. Por otro lado, se encuentran los metadatos de tipo mutable que se definen como aquellos que difieren de parte a parte y son diferentes de los demás.

- Por su función: Dependiendo de su función, se conocen tres tipos de 'datos sobre datos' que son los lógicos, simbólicos y subsimbólicos.

  En el caso de los metadatos *lógicos*, se caracterizan por la comprensión y son datos que explican cómo los datos simbólicos pueden emplearse para realizar deducciones de resultados lógicos.

  Los *simbólicos* son todos aquellos que agregan sentido y se ocupan de detallar los datos subsimbólicos.

  Los *subsimbólicos*, sencillamente, no contienen ninguna información acerca de su significado.

- Por su finalidad/cometido: Adicionalmente, se conoce otra clasificación que, aunque es la menos manejada, también es importante considerarla. Esta, secciona los metadatos dependiendo de su finalidad y contienen los siguientes tipos, de uso, de conservación, administrativos, descriptivos y técnicos –como se explica en [17]–

  Administrativos: proporcionan información para ayudar a gestionar un documento, cuándo y cómo fue creado, su archivo y otros aspectos de tipo técnico y quien puede tener acceso. Existen varios subconjuntos para la comprensión de los metadatos. Por ejemplo, los metadatos de gestión y los metadatos de conservación, éstos contienen información necesaria para el archivo y conservación del documento.

Descriptivos: son los metadatos que permiten el descubrimiento y la identificación. Puede incluir elementos como nombre, autor, resumen y palabras clave.

De conservación: para preservar los recursos de información. La preservación constituye un aspecto fundamental en el tratamiento de imágenes y objetos digitales y requiere una continua supervisión –copias de seguridad, reorganización de ficheros, visionado, verificación, etc.– ya que el material digital puede perderse si no se realiza ninguna intervención. Para esto resulta necesario una cuidadosa planificación así como el desarrollo de estrategias y pautas de actuación, sobre todo en lo que se refiere al seguimiento de normas y directrices nacionales e internacionales y la utilización de formatos de archivos y sistemas de almacenamiento estándar para asegurar la vida de la colección. En los últimos años se han puesto en marcha una serie de proyectos con el objeto de desarrollar especificaciones y sistemas de metadatos para apoyar la preservación de una gran variedad de recursos digitales, como son NEDLIB –la red Networked European Deposit Library–, CEDARS –el proyecto CURL Exemplars in digital Archives–, InterPARES y Pandora.

Técnicos: su función es informar sobre los requerimientos técnicos del hardware o software.

De uso: su función es informar sobre el nivel de utilización, tipo de usuarios, etc.

Estructurales: los metadatos estructurales indican cómo se colocan los documentos compuestos juntos. Por ejemplo, como los documentos están ordenados para conformar un expediente.

6.1.3. Esquemas de metadatos

Los esquemas de metadatos, o simplemente esquema, son un conjunto de elementos diseñados para un fin específico, como puede ser la descripción de un expediente o unidad documental compuesta, un documento o unidad documental simple, documento o unidad documental simple que forma parte de un expediente, o el de firma electrónica. La definición o el sentido de los propios elementos de

metadatos son el contenido. Esto quiere decir que los esquemas de metadatos en general, especifican los nombres de los elementos que incorporan y su significado, es lo que se conoce como el modelo de datos o listado de datos.

Hay diferentes esquemas de metadatos que han sido desarrollados por usuarios en gran variedad de ambientes y disciplinas. No únicamente en el ámbito de la gestión de documentos y archivos, sino en el de geografía, fotografía, bibliotecas, museos, etc.

Muchos esquemas de metadatos incluyen elementos tales como números de estándares que únicamente identifican la gestión o el documento a que los metadatos se refieren. La localización de un documento digital se puede dar usando un nombre de expediente, una URL –Uniform Resource Locator–, pero no es suficiente para garantizar su identificación por lo que es necesaria la combinación de metadatos para conseguir distinguir un documento de otro para su perfecta identificación.

Entre otros, algunos esquemas de metadatos son:

6.1.3.1. Dublin Core [18]

Dicho modelo surgió en 1995 en un taller patrocinado por OCLC[1] –originalmente se llamó Ohio College Library Center y luego pasó a denominarse Online Computer Library Center– y el National Center for Supercomputing Applications –NCSA– en Dublin –Ohio–. En un principio, el conjunto de elementos acordados recibió el nombre de Dublin Core y tras su evolución y desarrollo se gestiona por el Dublin Core Metadata Initiative –DCMI–. Su ámbito de aplicación ha sido principalmente las bibliotecas y los documentos que estas acogen.

Inicialmente el objetivo de Dublin Core fue definir un conjunto de elementos que pudieran ser usados por los propios autores para poder describir sus recursos web. Frente a la proliferación de los documentos electrónicos y la incapacidad por parte de los centros bibliotecarios de catalogar todos estos recursos, el objetivo fue definir unos elementos y unas normas simples que ayudaran a simplificar el trabajo sin

[1] OCLC es una organización cooperativa sin fines de lucro 'dedicada a los propósitos públicos de promover el acceso a la información mundial y reducir los costos de información'. Fue fundado por Frederick Kilgour en 1967.

perder eficacia. Los 15 elementos actuales determinados son –aunque inicialmente fueron 13–: título, creador, asunto, descripción, editor, colaborador, fecha, tipo, formato, identificador, recurso, idioma, relación, cobertura y derechos.

Todos los elementos de Dublin Core son opcionales o repetibles y pueden ser presentados en cualquier orden, aunque recomienda el uso de valores controlados en algunos campos y no permite la implementación particular.

En el contexto de los proyectos de localización geográfica o en descripción de recursos educativos, el uso de Dublin Core Metadata Standard –convertida en la ISO 15836:2003– [19] es frecuente.

6.1.3.2. Encoded Archival Description (EAD) [20]

Su desarrollo comenzó en 1993 con un proyecto iniciado por la Universidad de California, Berkeley. El objetivo era investigar la conveniencia y viabilidad de desarrollar un estándar de codificación común para encontrar información de inventarios, registros, índices y otros documentos creador por archivos, bibliotecas, museos, etc. y apoyar la difusión de sus materiales en la red.

En 1998 se publicó la versión 1.0 de las Directrices de aplicación del EAD y el Repertorio de Etiquetas EAD y el Document Type Definition –DTD– elaboradas por el Grupo de Trabajo de EAD de la Society of America Archivists.

Las Directrices de aplicación de EAD tienen como objetivo aportar a archiveros y a todas aquellas personas relacionadas con la descripción de archivos, información sobre cómo funciona EAD, una guía de asuntos administrativos que tienen que ver con la puesta en práctica de EAD, una visión general de cómo etiquetar con EAD, con información sobre los elementos obligatorios y los elementos clave, resúmenes comparativos de las herramientas y métodos relativos a la creación y publicación, también información básica sobre SGML y XML que está relacionada con EAD e instrucciones sobre el uso de elementos de enlace de la EAD. Disponen además de pautas sobre los elementos mínimos recomendados para obtener un instrumento de descripción eficaz y hace referencia a aquellos necesarios para la validación mecánica de un instrumento de descripción EAD.

También presenta tablas de traducción EAD y tres normas relacionadas con ella como son ISAD(G) –General International Standard Archival Description– [21], MARC [22] y Dublin Core.

En el año 2002 se publicó la versión 2002 del Document Type Definition (DTD) de EAD. El principal avance es que ahora es responsabilidad de cada país el uso de las listas de códigos y la terminología recomendada por las prácticas adoptadas de acuerdo con EAD.

La actualización del estándar EAD está mantenida conjuntamente por el Grupo de Trabajo de EAD de la Society of American Archivists y el Network Development and MARC Standards Office de la Library of Congress.

6.1.3.3. Metadata Encoding and Transmission Standard (METS) [23]

Es un estándar abierto de metadatos para codificar en formato electrónico un documento u objeto desarrollado por la comunidad bibliotecaria. METS está principalmente pensado para el intercambio de archivos audiovisuales que contengan imágenes, vídeos o sonidos ubicados en una biblioteca digital; Se utiliza un esquema de lenguaje XML de la W3C –World Wide Web Consortium–

6.1.3.4. MPEG Multimedia Metadata

MPEG (Moving Picture Experts Group) de ISO/IEC [24], dentro de sus estándares MPEG-7 y MPEG-21. MPEG-7 define una serie de herramientas de metadatos para crear descripciones del contenido audiovisual, tratando de dar solución, entre otros, al problema de la gestión de la ingente cantidad de contenido audiovisual disponible en la red.

6.1.3.5. CSDGM [25]

CSDGM –Content Standard for Digital Geospatial Metadata– son metadatos geoespaciales. Los metadatos geoespaciales comúnmente documentan datos digitales geográficos como archivos del Sistema de Información Geográfica –GIS–, bases de datos geoespaciales e imágenes de la tierra, pero también se pueden usar

para documentar recursos geoespaciales, incluidos catálogos de datos, aplicaciones de mapeo, modelos de datos y sitios web relacionados. Los registros de metadatos incluyen elementos básicos del catálogo de la biblioteca, como título, resumen y datos de publicación; elementos geográficos como la extensión geográfica y la información de proyección; y elementos de base de datos como definiciones de etiquetas de atributos y valores de dominio de atributos.

ISO 19115 es un estándar de la Organización Internacional de Normalización –ISO serie 19100–, cuyo objetivo es proporcionar un procedimiento claro para la descripción de conjuntos de datos geográficos digitales de modo que los usuarios puedan determinar si los datos de una explotación les serán de utilidad y cómo acceder a los datos.

6.2. Metadatos y ciberseguridad

La existencia de sofisticadas herramientas informáticas que permiten almacenar y analizar amplios conjuntos de datos para identificar patrones de comportamiento, hábitos, relaciones y detalles personales, hacen que, al enviar cualquier archivo, los metadatos pueden ser un problema para nuestra privacidad ya que ellos contienen toda la información.

Como se explica en [26], el primer objetivo de un atacante es acceder a los sistemas de su víctima, para lo cual, la extracción y análisis de los metadatos son una de las tareas previas de cualquier atacante que quiera acceder a los sistemas. Con ello se facilita entre otras, la tarea de descubrimiento de contraseñas –utilizando métodos como ataques de diccionario, de fuerza bruta,… –, o para realizar ataque de ingeniería social, o incluso para, en base a las versiones de software detectadas, saber qué exploit utilizar para la explotación de vulnerabilidades –según se expone en el *esquema Nacional de Seguridad de Borrado de Metadatos* [27]–.

Es por ello que se debe tener mucha precaución a la hora de publicar en la web imágenes con metadatos, que de manera individual aparentemente no representan un gran riesgo, pero que si se analizan de forma conjunta pueden representar una seria vulnerabilidad para los sistemas de la organización.

En la ilustración 6.3 se muestran los metadatos de una simple fotografía, y cómo a través de éstos se puede obtener información muy útil para realizar un ciberataque.

Entre muchos otros datos se puede advertir, en primer lugar, la fecha y hora exactas (A), y en segundo lugar, la marca y el modelo del dispositivo con el que fue tomada la fotografía (B).

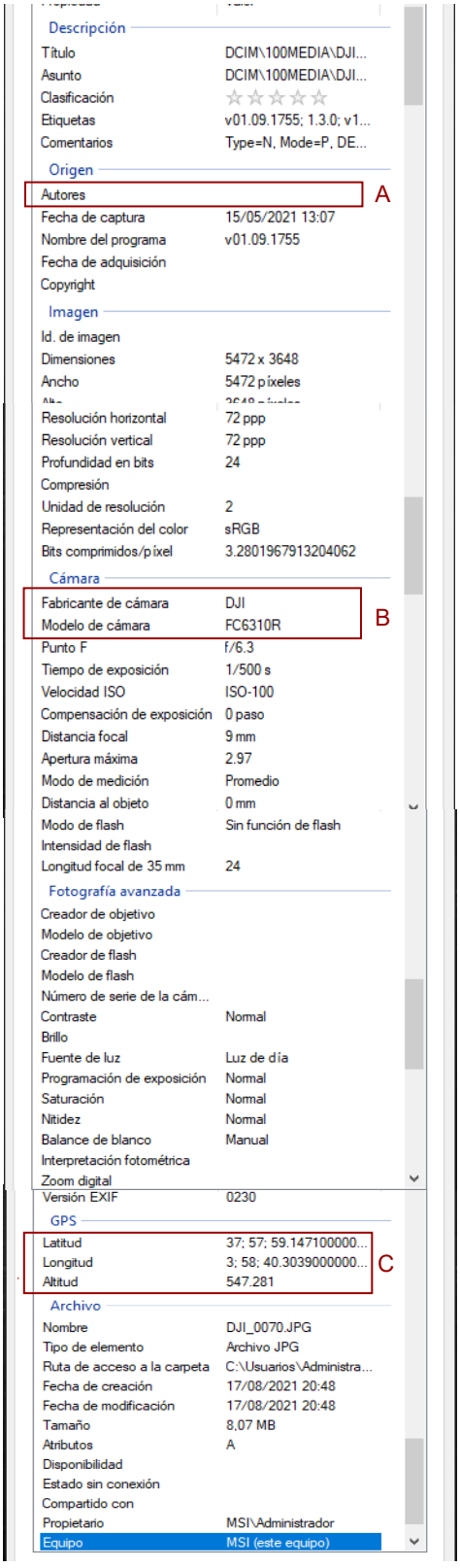
Este tipo de información podría parecer simple y poco sensible, pero podría simplificarle las cosas a un atacante, ya que al saber el tipo de dispositivo que usa su posible víctima, podrá buscar los exploits adecuados, por ejemplo. Además puede observarse la versión del sistema operativo, lo cual conducirá a saber qué vulnerabilidades podrían afectar al equipo.

Sin embargo, la información más sensible en este caso es la ubicación geográfica, que revela la ubicación exacta del usuario (C).

Teniendo en cuenta la amplia conectividad móvil disponible hoy en día, es frecuente que las personas se saquen fotos estando de vacaciones y las suban a redes sociales.

Con esta información de GPS, un cibercriminal podría saber que quien tomó la foto y si está fuera de su casa; Aprovechando los servicios gratuitos disponibles en Internet, como por ejemplo Google Maps, se pueden colocar las coordenadas para encontrar la ubicación en un mapa.

Ilustración 6.3 Metadatos de una fotografía



Nota: Metadatos de una imagen
Fuente: Elaboración propia

6.2.1. Fotografía y normativa de protección de datos.

La imagen personal forma parte de la privacidad/intimidad de cada persona. La existencia de una fotografía digital es inherente a la existencia de un fichero –implica que hay algún tipo de estructuración de fichero–. Cada persona decide, por tanto, dentro de los derechos fundamentales que marca nuestra Constitución qué uso se hace de dicha imagen, aunque esto en algunos casos, suponga un conflicto con otros derechos fundamentales como son, la libertad de expresión, la de información, la propiedad intelectual, etc.

Las plataformas de redes sociales donde se pueden compartir imágenes, pero donde también se pueden sustraer, manejan procesos internos de borrado de metadatos. Considerando las políticas de seguridad de estas compañías, no se menciona que estos datos serán borrados, aunque si se descarga una imagen de cualquiera de estos sitios, se observa que la información relevante ha sido borrada. No es así, lo que realmente hacen es sustraerla, aprovechar lo importante para lanzar publicidad de los productos que puedan interesar y almacenarla en sus bases de datos.

La Ley Orgánica 1/1982 de 5 de mayo [28], de Protección Civil del Derecho al Honor, a la intimidad Personal y Familiar y a la Propia Imagen, regula los derechos de imagen. Se entiende por *dato personal* cualquier información concerniente a personas físicas identificadas o identificables; en este sentido, la imagen es un dato de carácter personal, cuya protección se asegura en la Ley Orgánica 15/1999, de 13 de diciembre [29], de Protección de Datos de Carácter Personal –LOPD– y el Real Decreto 1720/2007, de 21 de diciembre [30], por el que se aprueba el Reglamento de desarrollo de la LOPD.

En el mundo digital, en cuanto a los metadatos, ni el RGPD ni la LOPD hace referencia alguna a ellos [31], pero, en la propuesta del Reglamento Europeo y del Consejo sobre el respeto de la vida privada y la protección de datos personales en el sector de las comunicaciones electrónicas y por el que se deroga la Directiva 2002/58/CE del Parlamento Europeo y del Consejo de 12 de julio de 2002

–Reglamento de E-Privacy– [32], sí que se regulan subsumiéndolos en el concepto de “Datos de comunicación electrónica”.

El Reglamento ePrivacy, que todavía no está vigente [33], dice que “*Entre esos metadatos figuran los números a los que se ha llamado, los sitios web visitados, la localización geográfica o la hora, la fecha y la duración de una llamada, información que permite extraer conclusiones precisas sobre la vida privada de las personas participantes en la comunicación electrónica tales como sus relaciones sociales, sus costumbres y actividades de la vida cotidiana, sus intereses, sus preferencias, etc”.* Es decir, que podrían ser considerados ‘datos’ a los efectos del RGPD.

Mientras el RGPD garantiza la protección de datos personales, el Reglamento E-Privacy amparará como *lex specialis* al tratamiento de datos de comunicaciones electrónicas llevado a cabo en relación con la prestación y utilización de servicios de comunicaciones electrónica.

El metadato, pues, es contemplado por la normativa como relevante jurídicamente en tanto que se encuentre en el ámbito normativo del futuro Reglamento, pero, per se, aunque pudiera identificar a una persona, como tal metadato, parece quedar fuera de la regulación.

6.3. Criptografía y Seguridad en Sistemas Informáticos

El término de seguridad informática, centra la atención en los medios informáticos en los que se genera, gestiona, almacena o destruye la información. La protección de dicha información, mientras se transmite y cuando está almacenada, es imprescindible para las empresas modernas, para los gobiernos y para toda la sociedad ya que los atacantes encuentran formas cada vez más innovadoras de comprometer los sistemas y robar datos.

Desde el nacimiento de las computadoras hasta hoy, ha habido un sustancial crecimiento de la tecnología criptográfica convirtiéndose en un pilar imprescindible sobre el que debe apoyarse la seguridad informática, abarcando la autenticación, las firmas y muchas otras funciones de seguridad elementales.

Teniendo en cuenta que la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales [34] estipula sanciones económicas cuantiosas en caso de ataque cibernético, se tiene la obligación de garantizar la privacidad de los mensajes y de los usos que de una página web, o sistema, haga cualquier persona.

La criptografía se encarga de que toda la información se transmita y almacene de manera segura, ocultándola frente a observadores no autorizados y preservando la integridad [35].

Sin darnos cuenta de ello, la criptografía está presente en muchas de nuestras actividades cotidianas. Por ejemplo, en el momento de acceder a nuestra cuenta de correo en los distintos clientes de correo, se usa criptografía. Lo mismo ocurre al realizar cualquier compra a través de Internet, al reservar un vuelo en una web segura, al realizar pagos con tarjetas o al sacar dinero de un cajero automático.

6.3.1. Criptografía

Un criptosistema se define como el conjunto de procedimientos que garantizan la seguridad de la información y utilizan técnicas criptográficas. Un sistema criptográfico es un método de proteger la información y comunicarse a través de códigos para que solo los usuarios que se enfrentan a la información puedan leer dicha información. Los componentes básicos de un criptosistema incluyen:

- Texto plano: conjunto de mensajes sin cifrar –los datos que se protegen durante la transmisión–.
- Criptogramas: conjunto de mensajes cifrados.
- Conjunto de claves: son los datos o llaves que permiten cifrar los mensajes.
- Conjunto de transformaciones de cifrado: existe una transformación diferente para cada valor de la clave.
- Conjunto de transformaciones de descifrado.

Todo criptosistema debe garantizar la integridad, confidencialidad y disponibilidad de la información para que se pueda considerar seguro a un sistema [36].

Los tipos de criptosistemas, que se clasifican según el método utilizado para cifrar los datos, se exponen en la tabla 6-1 y en la ilustración 6.4 se puede ver el funcionamiento general de ellos.

Tabla 6-1 Tipos de criptosistemas

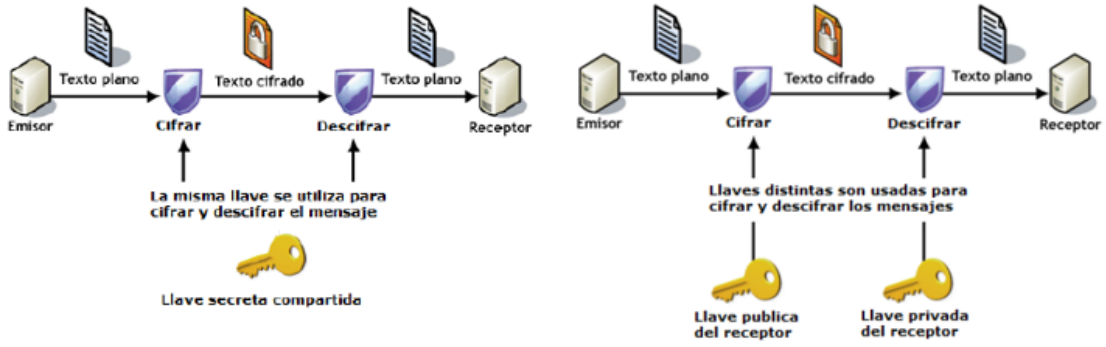
Criptografía simétrica o de clave privada
• Misma clave para cifrar y descifrar • Problema de la distribución de claves, debe estar tanto en el emisor como en el receptor • Dos tipos: cifrados de bloque y cifrados de flujo
Criptografía asimétrica o de clave pública
• La clave k tiene dos partes (k_p, k_P) • Una parte es pública y la otra privada • Una cifra y la otra descifra • Sirve para proteger y garantizar el origen de la información
Funciones resumen -hash-
• No existe función de descifrado • En algunos casos ni siquiera hay claves • Sirven para garantizar la integración de la información • Validar contraseñas sin tener que almacenarlas

Nota: Distintos tipos de criptosistemas

Fuente: Elaboración propia

En la práctica se emplea una combinación de los criptosistemas simétricos y de los asimétricos, puesto que los segundos presentan el inconveniente de ser computacionalmente mucho más costosos que los primeros. En el mundo real se codifican los mensajes –largos– mediante algoritmos simétricos, que suelen ser muy eficientes, y luego se hace uso de la criptografía asimétrica para codificar las claves simétricas –cortas–. Un criterio esencial para la seguridad del encriptado es la longitud en bits de las claves.

Ilustración 6.4 Esquemas generales de un sistema de cifrado simétrico y asimétrico



Esquema de un sistema de cifrado simétrico

Esquema de un sistema de cifrado asimétrico

Nota: Esquemas generales de un sistema de cifrado simétrico y asimétrico

Fuente:

https://virtual.itca.edu.sv/Mediadores/cms/u63_esquema_general_de_un_sistema_de_cifrado_simtrico.html

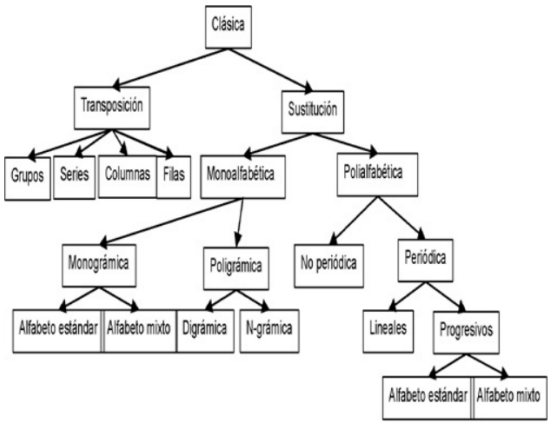
https://virtual.itca.edu.sv/Mediadores/cms/u66_esquema_general_de_un_sistema_de_cifrado_asimtrico.html

6.3.2. Algoritmos Criptográficos

6.3.2.1. Algoritmos clásicos de cifrado

Todos los algoritmos criptográficos clásicos son de carácter simétrico, ya que hasta mediados de los años setenta no nació la Criptografía Asimétrica. A continuación se exponen las técnicas básicas de cifrado que se han utilizado desde el principio de nuestra civilización hasta mediados del siglo pasado. La ilustración 6.5 instruye la clasificación de la criptografía clásica.

Ilustración 6.5 Clasificación de la criptografía clásica



Nota: Clasificación de la criptografía clásica

Fuente: http://www.revista.unam.mx/vol.7/num7/art55/popups/popup8.htm

- Cifrado por transposición o permutación: los caracteres del criptograma o texto cifrado son los mismos que los del mensaje o texto claro, pero aparecen en posiciones diferentes, es decir, permutados, y su simple lectura no permite deducir el mensaje que esconde, salvo que se conozca una clave que permita reordenar fácilmente dichos caracteres y volverlos a su posición original. Es el caso de la escítala –ver ilustración 6.4–.
- Cifrado por sustitución de caracteres: Los algoritmos de esta familia se basan en cambiar por otros los símbolos del mensaje, sin alterar su orden relativo. Cada uno de ellos vendrá definido por el mecanismo concreto empleado para efectuar dicho cambio. En un cifrado por sustitución *monoalfabético*, cada letra del texto claro se representa por otra letra o signo en el texto cifrado. Es el caso del cifrado de Polybios y el del cifrado César. La sustitución *polialfabética* consiste en que cada letra del texto en claro se va sustituyendo –cifrando– por una letra de un alfabeto de cifrado distinto, de acuerdo a una clave y la posición que ocupan los caracteres en el texto en claro con respecto a dicha clave. Un ejemplo de esta metodología es el cifrado de Vigenère.

6.3.2.2. Cifrados simétricos

Como se recoge en [37] se dividen en:

- *Cifrados por bloques*.

Operan dividiendo el mensaje que se pretende codificar en bloques de tamaño fijo, aplicando sobre cada uno de ellos una combinación más o menos compleja de operaciones de confusión –sustituciones– y difusión –transposiciones–, estructura denominada Red de Sustitución-Permutación.

Los cifrados de bloque son más útiles cuando se conoce la cantidad de datos, como un archivo, campos de datos o protocolos de solicitud/respuesta, como HTTP, donde la longitud del mensaje total ya se conoce.

- *Cifrados de flujo*.

Es un cifrado de clave simétrica en el que los dígitos de texto plano se combinan con un flujo de dígitos de cifrado pseudoaleatorio (flujo de claves) utilizando la operación XOR.

Los cifrados de flujo generalmente se ejecutan más rápido que los cifrados de bloque. En términos de complejidad del hardware, los cifrados de flujo son relativamente menos complejos. Los cifrados de flujo son la preferencia típica sobre los cifrados de bloque cuando el texto sin formato está disponible en cantidades variables (por ejemplo, una conexión wifi segura), ya que los cifrados de bloque no pueden operar directamente en bloques más cortos que el tamaño del bloque.

Una condición esencial para garantizar la seguridad de un cifrado de flujo es que nunca deben cifrarse dos mensajes diferentes con la misma secuencia. Ello permitiría validar posibles suposiciones acerca de los dos mensajes para un atacante, y deducir los fragmentos de secuencia empleados.

Debido a esto, surge en criptografía lo que se conoce como *nonce* –número de un solo uso–. No es más que la incorporación de algún tipo de información que nunca se repita, en la semilla de la secuencia, y suele formar parte de forma explícita en muchos de los algoritmos de cifrado de flujo.

En la tabla 6-2 se resumen las principales características del cifrado simétrico.

Tabla 6-2 Comparativa cifrado de bloque y flujo

Bases para la comparación	Cifrado de bloque	Cifrado de flujo
BASIC	Convierte el texto plano tomando su bloque a la vez	Convierte el texto tomando un byte del texto plano a la vez
Complejidad	Diseño simple	Comparativamente completo
Nº de bits utilizados	64 bits o más	8 bits
Confusión y Difusión	Utiliza tanto la confusión como la difusión	Sólo confía en la confusión
Modos de algoritmos utilizados	- BCE (libro de códigos electrónicos) - CBC (Cipher block Chaining)	- CFB (Cipher Feedback) - OFB (Feedback de salida) - CTR (Modo contador)
Reversibilidad	Invertir texto cifrado es difícil	Utiliza XOR para el cifrado que puede revertirse fácilmente al texto simple
Implementación	Cifrado de Feistel	Vernam Cipher
ALGORITMOS		
	DES, IDEA, AES –Rijindael–	RC4, SEAL, Salsa20, ChaCha, A5

Nota: Comparativa cifrado de bloque y cifrado de flujo

Fuente: Elaboración propia

6.3.2.3. Cifrados asimétricos

La diferencia con respecto a la criptografía simétrica, es que no se comparte una clave privada común, sino que forman pares para cifrar y descifrar los datos.

La criptografía asimétrica basada en aritmética modular emplea generalmente longitudes de clave mucho mayores que la simétrica y las dos aplicaciones inmediatas son, la protección de la confidencialidad, al no transmitir la clave de descifrado, y la autenticación de mensajes, con las funciones resumen MDC.

Entre los algoritmos de cifrado asimétrico más habituales, se pueden encontrar RSA, Diffie-hellman, DSA, ECDSA, edDSA, ElGamal o el algoritmo de Rabin. Las propiedades de los algoritmos están relacionadas con la curva elíptica que usan para su funcionamiento.

6.3.2.4. Funciones resumen

Las funciones resumen –hash, en inglés– es otra de las tecnologías para garantizar la seguridad en el intercambio. Proporcionan, a partir de un mensaje de longitud arbitraria, una secuencia de bits de longitud fija que va asociada al propio mensaje, actuando como una especie de *huella dactilar* del mismo, resultando difícil de falsificar.

La utilidad principal de estas funciones es la de detectar pérdidas de integridad de la información. Además, se utiliza en muchas áreas de la informática , como:

- La comunicación cifrada entre el servidor web y el navegador, así como la generación identificadores de sesión para aplicaciones de internet y el almacenamiento intermedio de datos –caché–.
- La protección de datos confidenciales, como contraseñas, análisis web o formas de pago.
- La firma de mensajes.
- La identificación de registros idénticos o similares en funciones de búsqueda.

Existen fundamentalmente dos tipos de funciones resumen: las que son calculadas sobre el mensaje directamente y cuyo fin es garantizar que éste no ha

sufrido modificaciones, denominadas MDC –Modification Detection Codes–, y las denominadas MAC –Message Authentication Codes– que emplean en sus cálculos una clave adicional para garantizar además el origen del mensaje.

Entre las funciones hash MDC más comunes se encuentran MD5, SHA-1, la familia SHA-2 –compuesta por SHA-224, SHA-256, SHA-384 y SHA-512–, el algoritmo Keccak –SHA-3–

Sobre los algoritmos de las funciones resumen MAC se puede nombrar el algoritmo Poly1305.

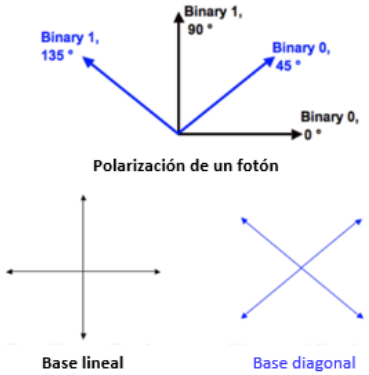
6.3.3. Criptografía Cuántica

En los últimos años se está hablando mucho sobre todo lo cuántico y cómo puede cambiar nuestras vidas. Como se explica en [38], la información protegida por sistemas de encriptación, cuya clave es un producto de dos números primos muy largos, se puede ver afectada por el avance de los ordenadores cuánticos. A medida que evolucionan, parecen más capaces de resolver dichas claves en cuestión de minutos.

Como teoría, la criptografía cuántica apareció en 1970, con Stephen Wiesner y su idea de codificación conjugada, pero no sería hasta 1984 cuando se publicaría el primer protocolo de cifrado cuántico, el denominado *protocolo BB84* de Charles H. Bennett y Gilles Brassard, con el que se considera que nació la criptografía cuántica propiamente dicha. En 1991, Artur Ekert introduciría un nuevo enfoque para la distribución de claves cuánticas basadas en el entrelazamiento cuántico, su protocolo E91 o EPR.

De acuerdo a [39], QKD –Quantum Key Distribution– es una técnica de distribución de claves criptográficas protegidas por las propiedades de la mecánica cuántica cuyos protocolos emplean los fotones, la unidad de energía más pequeña en una onda de luz y que no se pueden dividir, y sus propiedades cuánticas para cifrar los mensajes en los dos estados de polarización de éstos.

Ilustración 6.6 Posibles estados de polarización de un bit en dos bases



Nota: Posibles estados de polarización de un bit en dos bases

Fuente: https://www.techedgegroup.com/es/blog/comunicacion- criptografia-cuantica

Con la QKD lo que se hace es enviar información usando los bits actuales –0 y 1–, pero empleando qubits para cifrar las claves –los qubits o bits cuánticos son la medida básica de información en la informática cuántica tienen como propiedad el tener dos estados a la vez, es decir, ser 0 y 1 al mismo tiempo–. En la ilustración 6.6 se ve cómo se puede codificar un bit con el estado de polarización del fotón. Partiendo de dos posibles bases, la base lineal, +, o la diagonal, x, un 0 binario puede tener una polarización de 0° en la base lineal o 45° en la base diagonal. De la misma forma, un 1 binario, puede tener 90° en la base lineal o 135° en la diagonal, como se expone en la ilustración 6.7

Ilustración 6.7 Posibles estados de polarización para un bit

↕	↔	⤢	⤡
1	**0**	**0**	**1**

Nota: Posibles estados de polarización para un bit

Fuente: https://www.techedgegroup.com/es/blog/comunicacion-criptografia-cuantica

Además, es importante entender el concepto de la física cuántica de que cuando una partícula es observada, su estado se ve alterado, lo que en el caso de que un mensaje o información cifrada mediante protocolos de encriptación cuántica, fuese interceptado, al ser observado, el mensaje o la información se verían alteradas, de manera que no le servirían de nada al atacante.

Aunque la criptografía cuántica lleva varios años implementándose a nivel experimental, todavía está lejos de alcanzar los estándares de aplicabilidad que

tiene la criptografía actual basada en las matemáticas, especialmente porque la emisión de fotones que se emplea en el cifrado cuántico es muy sensible y puede sufrir variaciones durante la transmisión de la información, lo que la invalidaría.

6.3.4. Aplicaciones Criptográficas

Posiblemente, el establecer canales de comunicaciones seguros entre dos puntos, sea considerado como la aplicación más antigua de la Criptografía. Para ello, principalmente se mitigan dos aspectos peligrosos fundamentales:

- El acceso por agentes no autorizados. Esta posibilidad ha de darse por hecho y, por lo tanto, el sistema de protección debe centrarse en garantizar que el mensaje resulte ininteligible a un atacante.
- Las alteraciones en el mensaje. En este sentido, las alteraciones pueden aplicarse tanto sobre el mensaje propiamente dicho, como sobre la información acerca de su verdadera procedencia.

En general, las distintas aplicaciones que necesiten de estas técnicas, tendrán características específicas, habiendo una combinación de algoritmos criptográficos en cada caso. Estas combinaciones de algoritmos se estructuran en forma de protocolos, para proporcionar los mecanismos de comunicación segura normalizados.

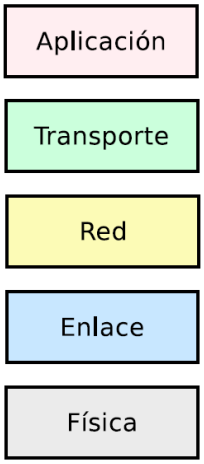
Los protocolos de cifrado proporcionan conexiones seguras, permitiendo que dos partes se comuniquen con privacidad e integridad de datos.

➢ Protocolos TCP/IP –Transmission Control Protocol/Internet Protocol–: como se indica en [40], es el estándar sobre el que se construye la red Internet. TCP e IP son dos protocolos distintos para redes informáticas. IP es la parte que obtiene la dirección a la que se envían los datos. TCP se encarga de la entrega de los datos una vez hallada dicha dirección IP. TCP/IP descompone cada mensaje en paquetes que se vuelven a ensamblar en el otro extremo y divide las distintas tareas de comunicación en capas apiladas –ver ilustración 6.8–. Cada una de ellas se comunica con las capas inmediatamente superior e inferior, logrando diversos

niveles de abstracción, que permiten intercambiar información de forma transparente entre ordenadores.

La consecuencia más importante de este enfoque es que los distintos dispositivos de distintos fabricantes de hardware y software, pueden conectarse entre ellos simplemente con que dispongan de una implementación TCP/IP.

Ilustración 6.8 Esquema del conjunto de protocolos TCP/IP



Nota: Esquema protocolos TCP/IP
Fuente: Elaboración propia

En la práctica se pueden encontrar protocolos encaminados a obtener comunicaciones seguras en prácticamente todos los niveles de este esquema.

➢ Protocolo SSL –Secure Socket Layer–: como afirma Manuel Lucena en [40], se sitúa en la capa de aplicación, directamente sobre el protocolo TCP, permitiendo establecer conexiones seguras a través de Internet. Se usa principalmente para proporcionar seguridad a los protocolos HTTP –web–, SMTP –email– y NNTP –news–.

Este tipo de protocolos utilizan criptografía tanto asimétrica –RS, Diffie-Hellman o DSA–, como simétrica –RC2, RC4, IDEA, TripleDES o AES– y como funciones resumen SHA-1 o MD5. Una comunicación a través de SSL implica tres fases fundamentales:

1. Partiendo del conjunto de algoritmos soportados por cada uno de los interlocutores, se establece la conexión y negociación de los algoritmos criptográficos. El servidor enviará el certificado con la clave pública. Si el servidor no cuenta con un certificado, el navegador mostrará un error.
2. Intercambio de claves, empleando algún mecanismo de clave pública, y autenticación de los interlocutores a partir de sus certificados digitales.
3. Cifrado simétrico del tráfico.

Puede decirse que el emplear un protocolo de comunicaciones en lugar de un algoritmo o algoritmos concretos supone una ventaja, puesto que así, ninguna de las fases del protocolo queda atada a ningún algoritmo, por lo que si apareciesen algoritmos mejores en el futuro, el cambio puede hacerse sin modificar el protocolo.

Las ventajas de protocolos como SSL son evidentes, ya que liberan a las aplicaciones de llevar a cabo las operaciones criptográficas antes de enviar la información.

➢ Protocolo TLS –Transport Layer Security–: el TLS es la siguiente generación del certificado SSL, basado en la versión 3.0 de éste. Una de las ventajas que proporciona sobre SSL es que puede ser iniciado a partir de una conexión TCP ya existente, lo cual permite seguir trabajando con los mismos puertos que los protocolos no cifrados. En este protocolo se emplea una serie de medidas de seguridad adicionales, encaminadas a protegerlo de distintos tipos de ataque, en especial de los de intermediarios.

➢ Protocolos IPsec –Internet Protocol security–: IPsec es un conjunto de protocolos que proporcionan cifrado y autenticación a los paquetes IP, trabajando en la capa de red. IPsec también incluye protocolos para el establecimiento de claves de cifrado (Diffie-Hellman).

6.3.4.1. Autenticación, certificados y firmas digitales

En muchos casos, existe el requerimiento de conocer la identidad del interlocutor al establecerse una comunicación. Con lo que se denomina autenticación –o autentificación–, se evita que el interlocutor no sea suplantado por

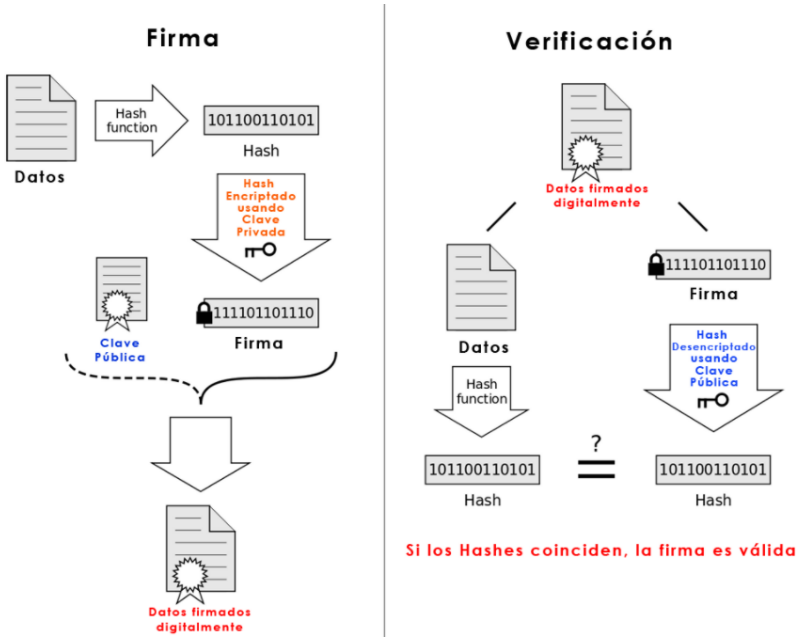
un impostor. Ésta puede realizarse tanto en el mismo momento –por ejemplo, introduciendo una contraseña para acceder al sistema– como cuando se firma digitalmente un mensaje, es decir, fuera de línea.

Una firma digital es una secuencia de bits que se añade a una pieza de información cualquiera y que debe cumplir las siguientes propiedades:

- Una firma digital válida para un documento es única, no puede ser válida para otro distinto.
- Sólo puede ser generada por su legítimo titular.
- En cualquier momento se puede comprobar su autenticidad de forma pública, por cualquiera.

La forma más extendida de calcular firmas digitales, como se puede ver en la ilustración 6.9, consiste en emplear una combinación de cifrado asimétrico y funciones resumen.

Ilustración 6.9 Esquema del funcionamiento de la firma digital



Nota: Funcionamiento de la firma digital

Fuente: Elaboración propia

El certificado digital es un archivo informático que permite la identificación de la persona que ejecuta la firma digital. Tiene una estructura de datos que contiene información sobre la entidad –por ejemplo una clave pública, una identidad o un conjunto de privilegios–. La firma de la estructura de datos agrupa la información que contiene de forma que no puede ser modificada sin que esta modificación sea detectada. Es esencialmente una clave pública y un identificador, firmados digitalmente por una autoridad de certificación, y su utilidad es demostrar que una clave pública pertenece a un usuario concreto.

En criptografía, el *estándar X.509* especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación.

En el caso de que una clave pública pierda su validez –por destrucción o robo de la clave privada correspondiente–, es necesario anularla. Para ello, se hace uso de los certificados de revocación.

6.3.4.2. PGP

PGP –Pretty Good Privacy o privacidad bastante buena– es un programa creado por Phil Zimmermann. Zimmermann diseñó PGP a principios de los 90 y con el paso de los años, PGP se ha convertido en uno de los mecanismos más fiables para mantener la seguridad y privacidad en las comunicaciones, sobre todo a través del correo electrónico.

PGP trabaja con criptografía asimétrica, y su auge se debe a la gran facilidad que ofrece al usuario a la hora de gestionar sus claves públicas y privada.

PGP cifra el mensaje empleando un algoritmo simétrico con una clave generada aleatoriamente –éstos son considerablemente más rápidos que los asimétricos– y posteriormente codifica la clave haciendo uso de la llave pública del destinatario.

Para decodificar el mensaje, PGP busca en la cabecera las claves públicas con las que está codificado y pide una contraseña. La contraseña servirá para que PGP compruebe la clave privada que permitirá decodificar el mensaje.

6.4. Servicios Web

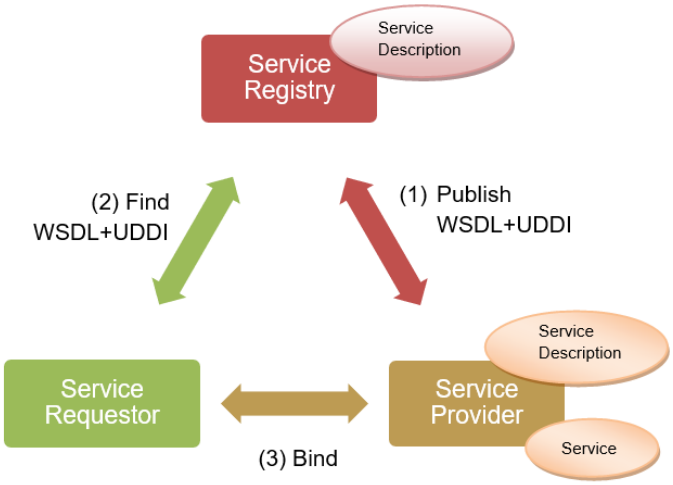
Otro de los objetivos del desarrollo de la herramienta para encriptado de metadatos de imágenes, es la integración de la misma con cualquier aplicación o servicio, de forma simple. Para ello, se lleva a cabo un estudio de las tecnologías existentes para la implementación de una API y de una aplicación web, y seleccionar aquella que mejor se adapte a los requisitos de este proyecto.

Un servicio web es un sistema software diseñado para soportar la interacción máquina-a-máquina, a través de una red, de forma interoperable. La interacción se basa en el envío de solicitudes y respuestas entre un cliente y un servidor, que incluyen datos.

Un servicio web es un tipo de API –Interfaz de Programación de Aplicaciones– pero una API puede ser un servicio web o no. Una API especifica la forma en la que los componentes de software deben interactuar entre sí. Es un conjunto de rutinas y protocolos cuyas respuestas se devuelven haciendo uso de la estructura de un JSON –JavaScript Object Notation– o de un XML para representar los datos. Además, las APIs se caracterizan por poder utilizar cualquier tipo de protocolo de comunicación, sin estar limitadas como lo están los servicios web.

La arquitectura de un servicio web estandarizado se basa en el uso de tres componentes principales, el proveedor del servicio web o Service provider, el solicitante del servicio web o Service requester y el corredor de servicios o Service broker [42].

Ilustración 6.10 Arquitectura de los servicios Web



Nota: Arquitectura de los servicios Web

Fuente: Elaboración propia

Además, como muestra la ilustración 6.10, los componentes son:

- WSDL –Web Services Description Language–. Es un lenguaje basado en XML para describir los servicios web y cómo acceder a ellos. Es el formato estándarpara describir un web Service, y fue diseñado por Microsoft e IBM. WSDL es una parte integral del estándar UDDI, y es el lenguaje que éste utiliza.
- UDDI –Universal Description, Discovery and Integration–. Es un estándar para describir, publicar y encontrar servicios web. Es un directorio donde las compañías pueden registrar y buscar servicios web. Es un directorio de interfaces de servicios web descritos en WSDL que se comunican mediante el protocolo en cuestión –SOAP, HTTP, REST,… –.
- Elemento Bind. Describe los protocolos que se utilizan para llevar a cabo la comunicación en un determinado PortType.

El funcionamiento de un servicio web consiste en:

1. El Service Provider genera el WSDL describiendo el servicio web y registra el WSDL en el directorio UDDI o Service Registry.
2. El Service Requestor o la aplicación del cliente requiere un servicio web y se pone en contacto con el UDDI para localizar el servicio web.
3. El cliente, basándose en la descripción descrita por el WSDL, envía un request para un servicio particular el servicio web listener, que se encarga de recibir

y enviar los mensajes en el formato seleccionado, SOAP –Simple Object Access Protocol– o REST.

4. El servicio web analiza el mensaje del request e invoca una operación particular en la aplicación para procesar el request. El resultado se escribe de nuevo en el formato seleccionado, SOAP o REST, en forma de respuesta y se envía al cliente.

5. El cliente finaliza el mensaje de respuesta SOAP o REST, y lo interpreta o genera un error en caso de que haya habido alguno.

6.4.1. Tipos de servicios Web

Se pueden distinguir dos tipos de servicios Web, servicios Web SOAP –big Web Services–, y servicios Web RESTful.

La única diferencia entre estos tipos es el protocolo o formato que utilizan para intercambiar datos entre aplicaciones, el protocolo SOAP o el protocolo REST. Los servicios Web que utilizan el protocolo REST tienen un funcionamiento prácticamente igual a los de protocolo SOAP.

Sin embargo los servicios Web tipo RESTful tienen algunas diferencias, ya que a diferencia del protocolo SOAP, el protocolo REST no está estructurado bajo estándares definidos, y es más ligero. Además, es mucho más flexible y permite que funciones no solo con lenguaje XML, sino también con JSON, entre otros.

SOAP usa XML como lenguaje de intercambio de datos con una estructura compleja que es capaz de albergar todo tipo de datos sobre la solicitud o respuesta generada.

REST usa el propio *protocolo HTTP* –HyperText Transfer Protocol– para la comunicación entre máquinas. HTTP es ampliamente soportado por todos los sistemas y de hecho para la transferencia de datos en la web se usa HTTP, siendo los más comunes:

- GET: se utiliza para obtener un recurso.
- POST: se utiliza para crear un recurso en el servidor.

- PUT: se utiliza para actualizar un recurso del servidor o cambiarle su estado.
- DELETE: se utiliza para eliminar un recurso del servidor

REST se caracteriza por *no tener estado*. Es decir, el servidor no es capaz de recordar el estado de la anterior solicitud REST que pudo, o no, hacer un cliente. Por ello, el cliente tiene que enviar en cada solicitud todo el estado de su sesión, lo que se suele hacer mediante un token que le 'ayude a recordar' al servidor.

Estas dos características, permiten que la implementación de REST sea realmente fácil y se haya popularizado tanto el tipo de servicio web que se conoce como el nombre de API REST. El servicio basado en estos principios se denomina RESTful. Las ventajas de cada arquitectura [43] se exponen en la tabla 6.3

Tabla 6-3 SOAP vs REST

Ventajas REST	Ventajas SOAP
Pocas operacione con muchos recursos	Muchas operaciones con pocos recursos
Se centra en la escalabilidad y rendimiento a gran escala para sistemas distribuidos hipermedia	Se centra en el diseño de aplicaciones distribuidas
HTTP GET, HTTP POST, HTTP PUT, HTTP DEL	SMTP, HTTP POST, MQ
XML auto descriptivo	Tipado fuerte, XML Schema
Síncrono	Síncrono y Asíncrono
HTTPS	WS SECURITY
Comunicación punto a punto segura	Comunicación origen a destino seguro

Nota: Ventajas de las arquitecturas REST y SOAP

Fuente: Elaboración propia

6.4.2. Estándares empleados

Como se explica en [44], para el intercambio de información, los servicios web utilizan una variedad de estándares y protocolos:

- **Web Services Protocol Stack**: conjunto de servicios y protocolos de los servicios Web.
- **XML**: formato estándar para almacenar los datos que se vayan a intercambiar de forma legible.
- **WSDL**: es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios web publicados. Una definición WSDL indica a un cliente cómo componer una solicitud de servicio y describe la interfaz.
- **SOAP** o XML-RPC –XML Remote Procedure Call–: es una serie de protocolos estándar sobre los que se establece el intercambio de datos mediante XML.
- **UDDI**: protocolo para publicar la información de los servicios web. Permite comprobar qué servicios web está disponibles.
- **WS-Security** –Web Service Security–: protocolo de seguridad aceptado como estándar por OASIS. El protocolo proporciona especificaciones sobre cómo debe de garantizarse la seguridad del intercambio de la información en un servicio Web.
- **REST**: se trata de una arquitectura que haciendo uso del protocolo HTTP, un conjunto de operaciones bien definidas –GET, POST, PUT y DELETE– y una sintaxis universal para identificar recursos, es posible realizar una comunicación entre un servicio Web y el cliente.
- **GraphQL**: se trata de una arquitectura alternativa a REST.

6.5. Conclusiones

El objetivo del desarrollo de esta herramienta es evitar la fuga de información –estudiada en el punto 6.2.1–, pudiéndose borrar y restaurar los metadatos de ficheros de imágenes.

Tras la investigación llevada a cabo sobre los metadatos en imágenes digitales, se concluye que éstas los almacenan en depósito interno. Consiste en consignar 'los datos sobre datos' internamente en el mismo archivo correspondiente a los datos. Es por ello que este proyecto se enfoca en poder exportar los metadatos seleccionados de una imagen a un archivo binario, encriptarlo y almacenarlo en un mismo recurso. Posteriormente, se borrarán para devolver la imagen sin ellos. En esta imagen devuelta, se incluirá el identificador de dicho archivo para, en un futuro, poder devolverle a la imagen 'desnuda', sus metadatos.

Al almacenarlos en depósito externo, es decir, depositarlos externamente en el servidor, y encriptados, se conseguirá una mayor seguridad. Con ello, se presentarán dos escenarios de complejidad para un potencial atacante:

- primero, tendría que hacerse con el servidor, para llegar al fichero de metadatos, y
- segundo, tendría que desencriptar los metadatos.

En cambio, si se optara por dejarlos encriptados en la propia imagen, el único esfuerzo para el atacante sería desencriptarlos.

Gracias a la criptografía se consiguen comunicaciones seguras por Internet a través de protocolos como SSL/TLS. De esta forma, al almacenar y transmitir información, el sistema cumplirá con los requisitos para garantizar la seguridad informática. Los principales objetivos de ambos protocolos consisten en proporcionar confidencialidad, –que a veces recibe el nombre de *privacidad*–, integridad de datos, identificación y autenticación utilizando certificados digitales.

El protocolo HTTP –Protocolo de transferencia de hipertexto– es el utilizado por el navegador y servidores web para comunicarse e intercambiar información. Al cifrar dicho intercambio con SSL/TLS, –HTTPS– se encriptará la información en tránsito proporcionando la seguridad en la comunicación.

En lo referente a los algoritmos de encriptación, de los estudiados en el desarrollo de esta investigación, se opta por AES –Advance Encryption Standard–. De acuerdo a [45] es uno de los algoritmos más seguros que existen hoy en día.

Está clasificado por la National Security Agency –NSA– de EEUU para la más alta seguridad de la información secreta.

Finalmente, en cuanto a los servicios web, se opta por usar un servicio web RESTful, puesto que XML es un formato más voluminoso que genera grandes paquetes de datos, lo que puede crear problemas en las conexiones de red lentas. Además, las API REST ofrecen:

- Separación entre cliente y servidor, ya que no importa el lenguaje del que proviene o el tipo de servidor (Python, Java, Node.js)
- Visibilidad, fiabilidad y escalabilidad.
- Es siempre independiente de la plataforma y el lenguaje.

7. ANÁLISIS DEL PROYECTO

La especificación de este proyecto sigue las directrices del Lenguaje Unificado de Modelado, o UML. Su notación orientada a objetos y los diferentes tipos de diagramas que existen, permiten visualizar, de forma sencilla, el diseño de software haciendo entendible su arquitectura.

Lo que se pretende con este proyecto es obtener una aplicación capaz de tratar los metadatos de una imagen para proteger la privacidad. La herramienta permitirá al propietario de la imagen decidir el uso de los datos personales que contiene dicha imagen eliminándolos o incluyéndolos cuando sea necesario y a través del administrador para garantizar la seguridad.

7.1. Casos de uso

Un caso de uso describe las interacciones entre el sistema y el usuario. Las funcionalidades, normalmente, se corresponden con las funciones declaradas en la especificación de requisitos expuestos en los siguientes apartados.

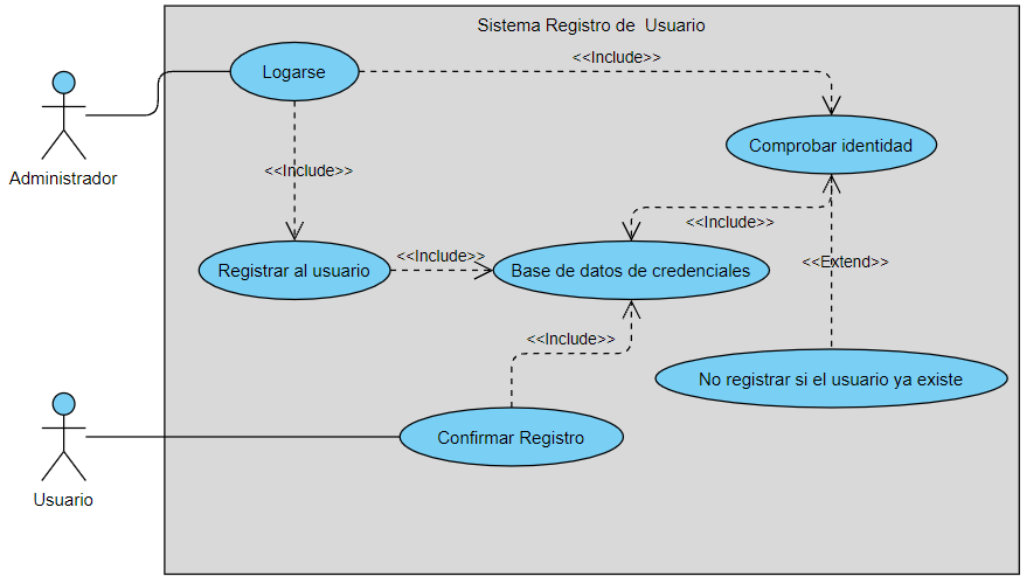
En este prototipo, como se expone en la ilustración 7.1, dos actores interaccionan con el sistema para llevar a cabo el servicio de registro de usuarios a través de la aplicación web. Un primer actor, usuario que representa el rol de

administrador, se debe logar por medio de la aplicación web para poder registrar usuarios –es el papel del segundo actor–. En caso de intentar registrar a un usuario con una cuenta de correo, ya existente, la aplicación no lo permitirá.

El registro de usuarios lo hace única y exclusivamente el administrador. Con ello se pretende dotar de seguridad a este prototipo de herramienta, para que sólo los usuarios autorizados por el administrador, puedan tener acceso a los metadatos, cifrados, para poder devolvérselos a la imagen digital.

Además, el registro de usuarios necesita de la confirmación por parte del usuario, a través del enlace que recibirá por correo electrónico. Nuevamente, el envío de este correo con el protocolo SSL/TLS, asegura la protección, impidiendo que el contenido sea accesible a terceros durante el intercambio de datos, pues para descifrarlo se necesitan las claves correspondientes. Al enviar o recibir mensajes a través de un cliente web en el navegador, se puede afirmar casi con total seguridad que el tráfico de correo está cifrado. Los proveedores serios de correo ofrecen sus servicios web mediante el protocolo HTTPS.

Ilustración 7.1 Diagrama de Caso de Uso del Registro del usuario

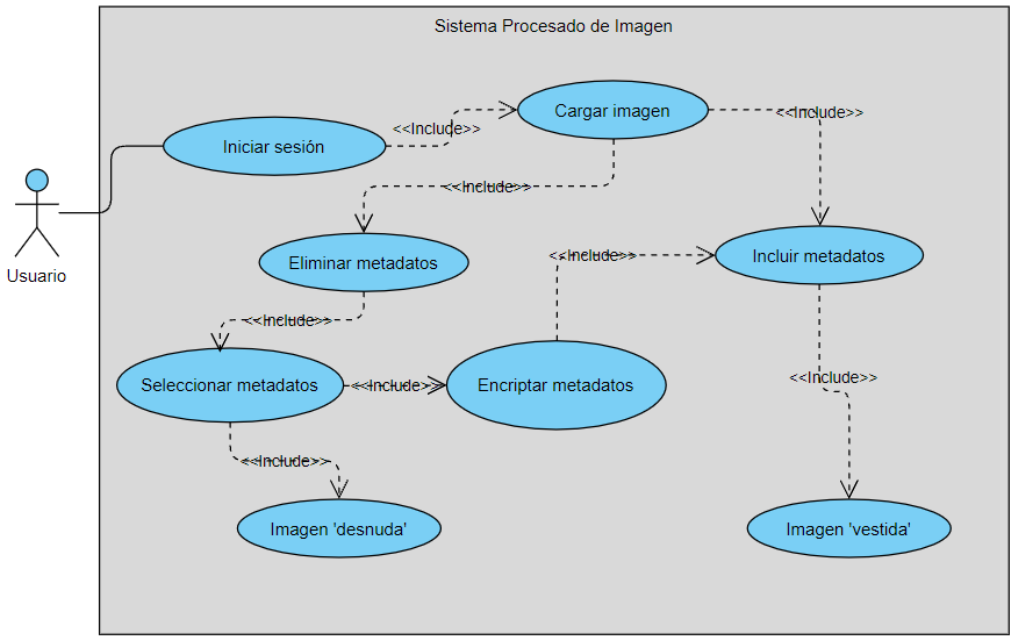


Nota: Diagrama de Caso de Uso del Registro del usuario

Fuente: Elaboración propia

Para el procesado de una imagen digital, la aplicación desarrollada en este proyecto, consta de un solo actor: el usuario de la aplicación. Los usuarios accederán a la herramienta de encriptado mediante una aplicación web que hará uso de los servicios de eliminación e inclusión de metadatos. Los usuarios obtendrán la imagen que desee procesar, bien sin metadatos, o incluyéndoles éstos, como respuesta a la petición. El diagrama de caso de uso se muestra en la ilustración 7.2.

Ilustración 7.2 Diagrama de Caso de Uso de procesado de una imagen



Nota: Diagrama de Caso de Uso de procesado de una imagen
Fuente: Elaboración propia

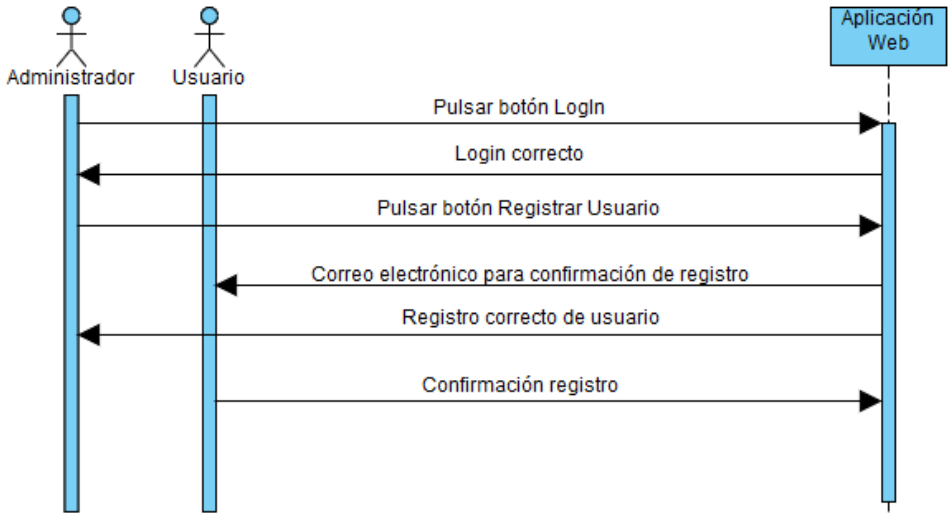
7.2. Diagrama de secuencias

En el diagrama de secuencias mostrado con la ilustración 7.3 se describe la interacción entre los objetos del sistema en el momento del registro de usuario.

En primer lugar, el Administrador se loga pulsando el botón LogIn a través de la aplicación web. A partir de aquí, el administrador registrará al usuario pulsando el botón de Registrar Nuevo Usuario.

El Usuario confirmará el registro a través del enlace que recibirá por correo electrónico en el buzón que se haya especificado. A partir de aquí, el Usuario podrá iniciar sesión en la aplicación.

Ilustración 7.3 Diagrama de secuencias del Registro de usuario



Nota: Diagrama secuencia del Registro de usuario
Fuente: Elaboración propia

Con la ilustración 7.4 se describe la interacción entre los objetos del sistema para procesar una imagen. Como se ve, una vez iniciada sesión, por parte del usuario, éste cargará la imagen y la visualizará a través de la aplicación web.

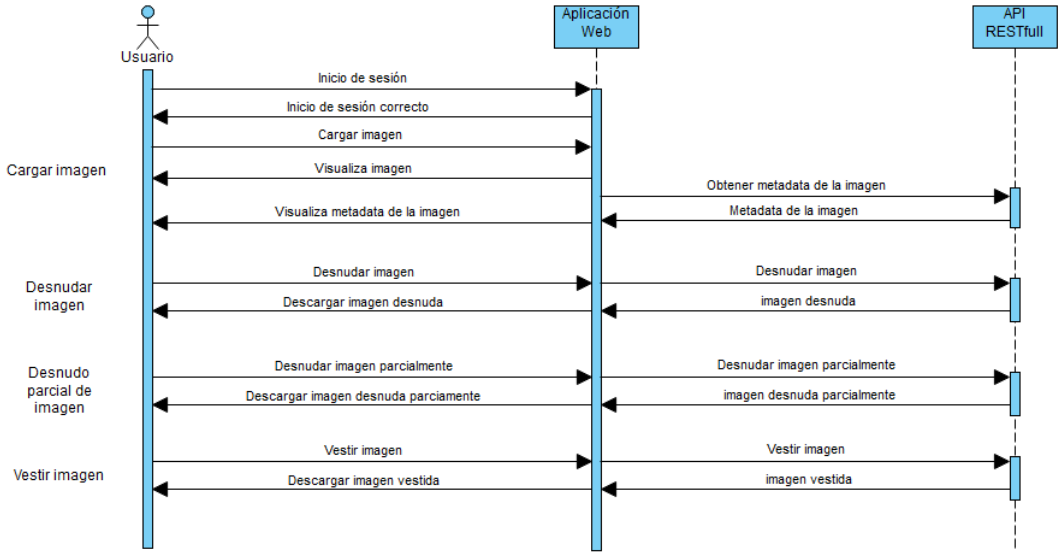
La aplicación enviará la petición a la API para obtener los metadatos de la imagen, y los visualizará.

En este momento, el usuario decidirá si quiere eliminar todos los metadatos de la imagen, si quiere eliminar sólo algunos de ellos, o si se trata de una imagen con los metadatos previamente eliminados, incluirlos nuevamente.

En el primer caso, se usará el servicio web de 'desnudar imagen', donde la API devolverá la imagen sin ningún metadato y la aplicación web permitirá descargarla. En el segundo caso, el usuario seleccionará 'desnudar la imagen parcialmente' a través de la aplicación web –seleccionando aquellos metadatos en cuestión– y se

pedirá a la API RESTful que elimine los metadatos elegidos. Nuevamente, ésta devolverá la imagen procesada y el usuario podrá descargarla. Finalmente, si el usuario necesita el servicio de añadirle los metadatos a la imagen, clicará el botón 'Vestir imagen' a través de la aplicación web y se enviará dicha petición a la API RESTful. Ésta desencriptará los metadatos y devolverá la imagen con todos para poder descargarla a través de la aplicación web.

Ilustración 7.4 Diagrama de secuencias para procesar una imagen



Nota: Diagrama secuencia para procesar una imagen

Fuente: Elaboración propia

La lógica aplicada a este prototipo consiste en encriptar el fichero donde se guardarán los metadatos extraídos de la imagen. En el momento de añadirlos a la imagen, y antes de utilizarlos, se creará una copia de dicho fichero, se desencriptará la copia y se utilizará ésta para incluirlos a la imagen. Por último, se borrará la copia.

7.3. Análisis de requisitos

El análisis de requisitos proporcionará una descripción más minuciosa de los servicios que se proporcionarán, así como sus restricciones. Para ello, se describen tanto los requisitos funcionales como los no funcionales, de este sistema. Cada descripción de requerimiento tendrá la estructura mostrada en la tabla 7-1:

Tabla 7-1 Cómo definir requisitos software

IDENTIFICADOR DE REQUISITO. Texto descriptivo. **(calificación)**	
IDENTIFICADOR DE REQUISITO	Identificador del requisito. Estará formado por las iniciales de la categoría a la que pertenece: **- RF:** Requisito Funcional **- RNF:** Requisito No Funcional **- RFU:** Requisito Funcional de Usuario **- RFO:** Requisito Funcional Operacional **- RFI:** Requisito Funcional de Interfaz **- RFR:** Requisito Funcional de Recursos **- RFP:** Requisito Funcional de Prestaciones Seguidas de un guión y un número natural. Ejemplo **RF-1**
Texto descriptivo	Párrafo que describe el requisito de forma concisa
(calificación)	Determinan la necesidad del requisito. Existen tres calificaciones posibles: **- Esencial:** debe estar presente obligatoriamente **- Deseable:** no es obligatorio pero supone mejoras al producto **- Opcional:** define alternativas que podrían definir productos diferentes

Nota: Cómo se definen los requisitos software

Fuente: Elaboración propia

7.3.1. Requisitos funcionales

Los requerimientos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos.

Tabla 7-2 Requisitos Funcionales de ImageMetaData

CÓDIGO	DESCRIPCIÓN	EJEMPLOS
RF-1	La herramienta será capaz de eliminar algunos metadatos de una imagen. **(esencial)**	Se deberá habilitar la opción de seleccionar algunos metadatos.
RF-2	La herramienta será capaz de eliminar todos los metadatos de una imagen. **(esencial)**	Se deberá habilitar la opción de seleccionar todos los metadatos.
RF-3	La herramienta será capaz de añadir algunos metadatos de una imagen, no la imagen original. **(esencial)**	Se le añadirán aquellos metadatos que previamente se hayan eliminado.
RF-4	La herramienta será capaz de añadir todos los metadatos de una imagen, no la imagen original. **(esencial)**	Se le añadirán aquellos metadatos que previamente se hayan eliminado.

RF-5	El sistema no permitirá dar de alta a un usuario ya registrado. **(esencial)**	Si el usuario ya existe, cuando el usuario administrador intenta registrarlo, la aplicación le mostrará un mensaje indicándoselo.
RF-6	El sistema deberá garantizar la máxima seguridad de los metadatos extraídos a la imagen. **(esencial)**	Se guardarán en un fichero y se encriptará con cifrado simétrico, AES.
RF-7	El sistema contará con un log de las imágenes que se han procesado. **(esencial)**	Dichas imágenes se guardarán en el servidor donde la API esté instalada.
RF-8	La herramienta permitirá descargar la imagen que se procese. **(esencial)**	
RFU-1	El usuario introducirá la imagen que se procesará. **(esencial)**	
RFU-2	El usuario seleccionará aquellos metadatos de la imagen que quiera eliminar/ofuscar. **(esencial)**	
RFU-3	El usuario seleccionará todos los metadatos de la imagen para eliminarlos. **(esencial)**	
RFU-4	El usuario introducirá la imagen cuyos metadatos fueron previamente eliminados y podrá añadírselos. **(esencial)**	Seleccionará la opción que le permitirá llevar a cabo dicho servicio.
RFU-5	El usuario podrá descargar cualquier imagen procesada. **(esencial)**	
RFU-6	El usuario administrador del sistema debe controlar a quién se permite el acceso al sistema, pues los metadatos contienen información reservada de carácter personal. **(esencial)**	El administrador será única y exclusivamente el que registrará a los usuarios.
RFU-7	El usuario podrá reiniciar la herramienta cuando desee. **(esencial)**	
RFI-1	La herramienta recibirá, por el usuario, la imagen a procesar. **(esencial)**	
RFI-2	La herramienta mostrará la imagen en la pantalla.	

	(esencial)	
RFI-3	La herramienta mostrará, para seleccionar, los metadatos de la imagen en la pantalla. **(esencial)**	
RFR-1	La API no será publicada para que la use el público en general. Será desplegada de forma privada. **(esencial)**	
RFP-1	En la salida por pantalla, la herramienta mostrará los metadatos seleccionados con un tickbox. **(esencial)**	

Nota: Requisitos funcionales de ImageMetaData

Fuente: Elaboración propia

7.3.2. Requisitos no funcionales

No hablan de 'lo que' hace el sistema, sino de 'cómo' lo hace. Representan características generales, no se refieren directamente a las funciones específicas suministradas por el sistema –características de usuario–, sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad.

Tabla 7-3 Requisitos no funcionales de ImageMetaData

CÓDIGO	DESCRIPCIÓN	EJEMPLOS
RNF-1	El sistema será capaz de procesar de forma ágil, reduciendo al mínimo el tiempo de respuesta y de espera del usuario. **(esencial)**	
RNF-2	El sistema debe de ser capaz de atender a la vez a varios usuarios con sesiones concurrentes. **(esencial)**	
RNF-3	El tiempo de aprendizaje de la aplicación por un usuario deberá ser mínimo. **(esencial)**	
RNF-4	La aplicación web debe poseer un diseño 'Responsive' a fin de garantizar la adecuada visualización en múltiples computadores, dispositivos tableta y teléfonos inteligentes. **(esencial)**	
RNF-5	Debe de disponer de memoria y espacio en disco para el procesado de las imágenes y el almacenamiento de los	

	metadatos. **(esencial)**	
RNF-6	El sistema debe poseer interfaces gráficas bien formadas. **(esencial)**	
RNF-7	Todas las comunicaciones a través de Internet han de estar cifradas y ser seguras. Confidencialidad, Autenticación (del servidor) e Integridad con el protocolo SSL/TLS. **(esencial)**	

Nota: Requisitos no funcionales de ImageMetaData

Fuente: Elaboración propia

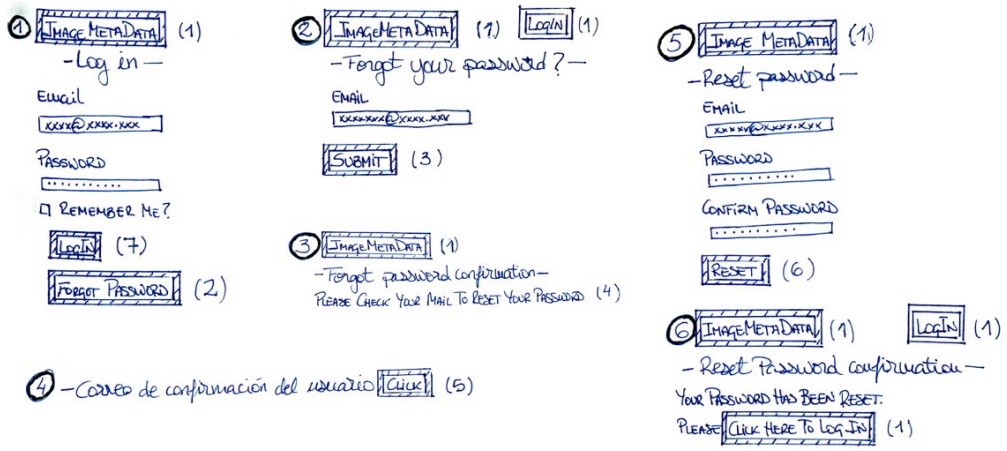
7.4. Prototipo

Para dar una visión de cómo es el comportamiento del sistema desarrollado, en las ilustraciones 7.5 y 7.6, se muestra el prototipo de éste. Se ha optado por diseñar la página web, su funcionalidad, y su navegación, con un prototipo de papel. A tal fin, se consigue hacer participar al usuario en el diseño y evaluarlo en las primeras fases.

Cada uno de los distintos escenarios se refleja en una hoja con el propósito de poder simular las diferentes iteracciones con el sistema.

Para usar la herramienta ImageMetaData es necesario haber sido registrado por el usuario administrador. Partiendo de aquí, la página de inicio de sesión de usuario (1) mostrará la opción para logarse y la opción de restaurar la contraseña en caso de olvido.

Ilustración 7.5 Prototipado de iteracciones con ImageMetaData para iniciar sesión de usuario



Nota: Prototipado de iteracciones con ImageMetaData para iniciar sesión de usuario

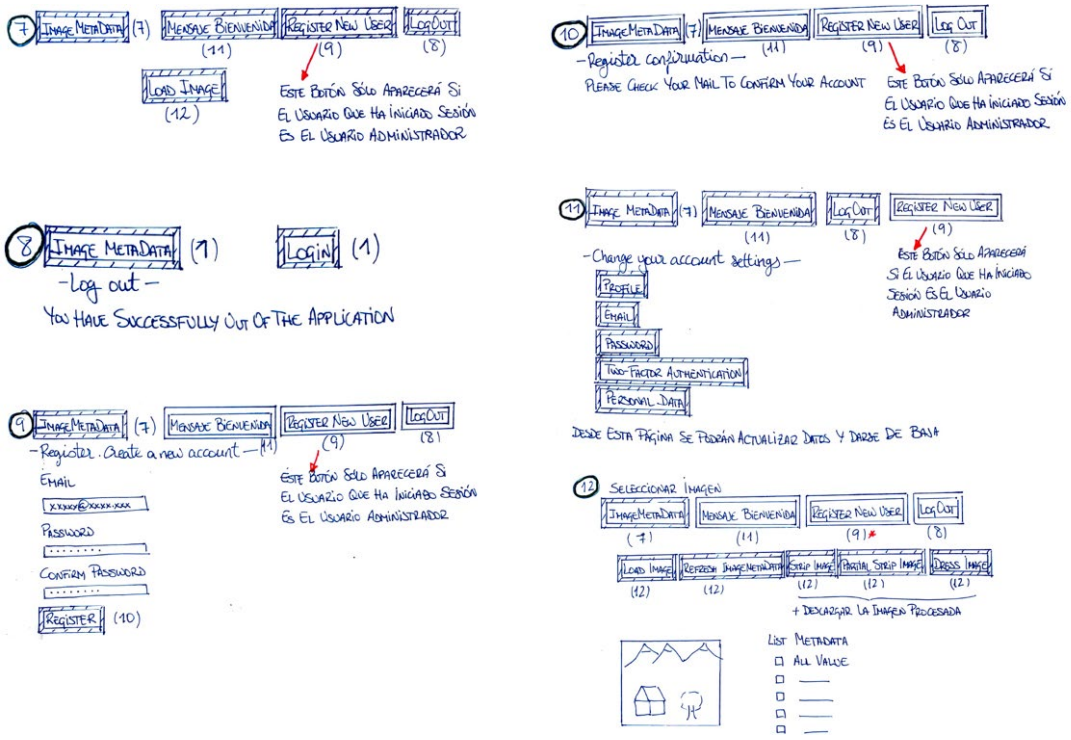
Fuente: Elaboración propia

Como se muestra con la ilustración 7.5, en caso de restaurar la contraseña, la aplicación pedirá una dirección de email(2), invitará a resetear la password y confirmarla a través del buzón indicado previamente(3),(4),(5)y(6). Finalmente, una vez actualizada la password, el usuario se podrá logar desde la página a la que la aplicación remitirá(1).

Una vez logado(7),el usuario podrá usar la aplicación para procesar los metadatos de las imágenes digitales. Con la ilustración 7.6 se plasma esta interacción. Desde esta página, ofrecerán distintos servicios:

- salir de la aplicación (8), desde donde se podrá acceder a ella, logándose nuevamente.
- registrar nuevos usuarios(9). Este servicio sólo estará disponible cuando el usuario logado haya sido el usuario **administrador**. El registro consistirá en solicitar la dirección email, donde posteriormente debe confirmarse(10), y la contraseña.
- cambiar la configuración de la cuenta de usuario, desde el botón de mensaje de bienvenida y nombre de usuario (11). Desde aquí se podrán actualizar datos y darse de baja.
- cargar la imagen para procesar sus metadatos (12). Desde aquí se podrán eliminar todos los metadatos, o sólo aquellos que se quieran, o añadir los metadatos que se hubieran eliminado previamente. En los dos primeros casos, permite descargar al equipo local, una copia de la imagen que se tiene cargada en la aplicación, sin los metadatos.
- una imagen desnudada sólo contendrá como metadatos el nombre de ésta y el nombre del fichero encriptado que contiene los metadatos de dicha imagen. Éstos también aparecerán en caso de que la imagen haya sido parcialmente desnudada.

Ilustración 7.6 Prototipado de iteracciones con ImageMetaData para el tratamiento de los metadatos de las imágenes



Nota: Prototipado de iteracciones con ImageMetaData para el tratamiento de los metadatos de las imágenes

Fuente: Elaboración propia

8. DISEÑO E IMPLEMENTACIÓN

El desarrollo e implementación de todas las partes que conforman este proyecto, se indicarán, pormenorizadamente, en este apartado. Además, se describen las tecnologías utilizadas para ello.

En este punto se especifican todas las tecnologías estudiadas y usadas para el desarrollo del prototipo de ImageMetaData.

8.1. Tecnologías y librerías para el Servicio web (webapi)

8.1.1. Entorno de desarrollo

Se opta por usar VS Code [46], versión 1.59.0 en virtud de ser una herramienta de edición de código fuente multiplataforma y poseer unas características oportunas para el desarrollo del proyecto.

8.1.2. Lenguajes de programación

En cuanto al lenguaje de programación utilizado para el desarrollo del Servicio web se elige Python 3.

Python remonta su origen a principios de los años 90, cuando un trabajador de un centro de investigación holandés –Centrum Wiskunde & Informatica–, Guido Van Rossum, desarrolló este nuevo lenguaje de programación.

Además, es sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Se ha considerado que el uso, conjuntamente, de Python y técnicas ágiles, puede derivar en una potente herramienta de programación. Python es un lenguaje interpretado de alto nivel, con fuerte orientación a objetos, que se utiliza para desarrollar aplicaciones de todo tipo. Por ello, y con el propósito de obtener un código bien organizado con vista a posibles modificaciones futuras y comprensión, se ha creado un fichero .py por cada clase necesaria.

Además de por su sencillez y por sus amplias posibilidades, como se puede leer en [46], sobre todo en el campo que atañe este proyecto, se ha preferido Python frente a otras opciones por los siguientes motivos:

- En primer lugar, una gran cantidad de tecnologías se llevan muy bien con Python, tales como, Data analytics y big data, Data mining, Data science, Inteligencia artificial, Blockchain, Machine learning, Juegos y gráficos 3D o Desarrollo web. A pesar que este lenguaje lleva 30 años en el mercado, su relación con algunos de estos campos con mayor relevancia de la actualidad lo hacen una potente herramienta.
- En segundo lugar, además, de ser portable, las propiedades de este lenguaje son simples y fáciles de aprender, su naturaleza como lenguaje de propósito general hace que disponga de una gran cantidad de librerías y herramientas para programar.
- Tercero. La gran extensión temática de su biblioteca, protocolos de Internet, multimedia, GUI, Internacionalización, etc. hacen que sea una de las herramientas más potentes.
- Y cuarto, permite la creación de entornos virtuales, un ámbito de trabajo aislado al sistema, permitiendo así disponer de un entorno de trabajo personalizado

en un proyecto concreto y pudiendo instalar las librerías o versiones de las mismas sin que afecte al entorno del sistema.

Se ha hecho uso de esta última utilidad mencionada, se ha creado un entorno virtual para la implementación de este proyecto. En él se han instalado todas las librerías necesarias para el correcto funcionamiento del sistema. Así, al instalar el servidor, se instala también dicho entorno virtual facilitando la portabilidad de la aplicación.

8.1.3. Tecnologías web

Hay varias tecnologías clave que se han utilizado en el desarrollo del prototipo de esta API. Se trata de Flask, Flask RESTful y Flask-cors. Estas tecnologías son básicas para poder entender el sistema que se está tratando, ya que proporcionan un marco de trabajo multiplataforma y portable. En un entorno multimedia en el que se busca un acceso universal, una arquitectura basada en estas tecnologías

8.1.3.1. Flask

Flask es un framework web de Python construido para el desarrollo rápido de Aplicaciones web. Está basado en el motor de templates Jinja2.

Como se ve en [48], en un primer estudio de frameworks para este fin, se barajan Flask, Django, Pyramid, turboGears, Falcon, Bottle, Morepath o Sanic, entre otros, –no existen tantos en Python como puede ser el caso de los frameworks de JavaScript–. Se opta por centrar dicho estudio en los dos primeros, –el resto son para casos más concretos–, que son los que más atracción han ido recogiendo de los desarrolladores, por ser proyectos que han ido evolucionando con mejor solera y una mayor frecuencia.

Finalmente se elige el microframework Flask [49] por los siguientes motivos:

- Es un microframework web y ligero para desarrollar de manera muy sencilla aplicaciones web simples, frente a Django que es más adecuado para crear aplicaciones web grandes y complejas.

- Sirve para construir servicios web –como APIs REST– o aplicaciones de contenido estático.
- Es compatible con Python 3 y con WSGI –Web Server Gateway Interface–, protocolo usado por los servidores web.
- Soporta de manera nativa el uso de cookies seguras.
- Debido a que es Open Source, dispone de una buena documentación, código GitHub y está amparado bajo una licencia de software para los sistemas BSD –Berkeley Software Distribution–, un tipo del sistema operativo Unix-like.

8.1.3.2. Flask RESTful

Flask RESTful es la extensión para Flask que se ha empleado para construir la API REST de este proyecto.

Tras profundizar en el estudio de esta tecnología, cabe mencionar que a pesar de que existen procedimientos para proteger una API basada en Flask con contraseña o autenticación basada en token, –una de ellas puede ser la que se explica en [49]–, no se ha hecho uso de ellos a lo largo del desarrollo de este proyecto. La causa es que, en principio, y según los requisitos funcionales, la API no será publicada para que la use el público en general, sino que, será desplegada de forma privada y sólo la utilizará la aplicación web, la cuál será la que autentique y autorice a los usuarios.

8.1.3.3. Flask-CORS

Para poder hacer peticiones HTTP desde otro dominio distinto al que se usa para la aplicación de Flask, es necesario habilitar CORS –Cross Origin Resource Sharing–. Para ello se hace uso de la extensión Flask-CORS [50].

8.1.3.4. PyCryptodome

En cuanto a la seguridad del fichero que contendrá los metadatos extraídos a la imagen, se encripta éste con el paquete de Python, PyCryptodome.

La biblioteca PyCryptodome ofrece implementaciones para AES, Flujo de cifrados como Salsa20, Hash criptográficos como SHA-2, Códigos de autenticación de mensajes como HMAC, o Generación de claves asimétricas RSA, entre otros.

La clave que se utiliza en AES para la implementación del prototipo de ImageMetaData, tiene un tamaño de 128 –16 bytes– y su generación se obtiene aleatoriamente. Con esta clave, se procederá a cifrar el archivo –en el que se almacenarán los metadatos eliminados de una imagen– donde además se almacenará el nonce y la tag para el posterior desencriptado.

8.1.3.5. Exiv2

La biblioteca en C++ y herramientas de metadatos Exiv2 [51] es una utilidad en línea de comandos que permite manipular los metadatos de imágenes (Exif, IPTC, XMP), incluyendo otras funciones adicionales como la importación/exportación a partir de un archivo de texto.

Del estudio llevado a cabo, además de Exiv2, exiftool [53] podría considerarse una de las herramientas más conocidas para este tipo de labores. Al estar escrita, esta última, en Perl, hace que Exiv2 sea más rápida, por lo que se declina a usarla en la implementación de este proyecto, ImageMetaData.

8.2. Tecnologías y librerías para la Aplicación web (webapp)

Se utilizan varias tecnologías para la implementación de la aplicación web.

8.2.1.1. ASP.NET Core

ASP.NET Core [54], es un framework de desarrollo de aplicaciones web de Microsoft. Concretamente, ASP.NET Razor Pages, es un marco de desarrollo web ligero y flexible, del lado del servidor que permite crear sitios web dinámicos. Como parte del marco de desarrollo web ASP.NET Core de Microsoft, Razor Pages admite el desarrollo multiplataforma.

ASP.NET Core y Razor Page, utiliza C# como lenguaje de programación. Arquitectónicamente, RazorPages es una implementación del patrón MVC –Modelo Vista Controlador–. Esta arquitectura separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- El Modelo que contiene una representación de los datos que maneja el sistema, no contiene ninguna lógica que describa cómo presentar los datos a un usuario.
- La Vista, o interfaz de usuario, presenta los datos del modelo al usuario. La vista sabe cómo acceder a los datos del modelo, pero no sabe qué significa esta información o qué puede hacer el usuario para manipularla.
- El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. Escucha los eventos desencadenados por la vista y ejecuta el procedimiento adecuado a estos eventos.

El flujo que sigue el controlador, generalmente, como se ve en la ilustración 8.1 es:

1. El usuario interactúa con la interfaz de usuario de alguna forma –por ejemplo, pulsa un botón, enlace, etc. –
2. El controlador recibe –por parte de los objetos de la interfaz-vista– la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos –handler– o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario –por ejemplo, el controlador actualiza el carro de la compra del usuario–.

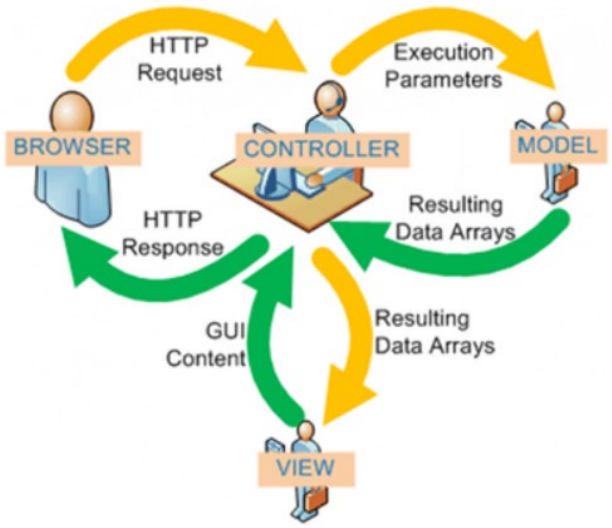


Ilustración 8.1 Flujo del Modelo-Vista-Controlador

4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo –por ejemplo, produce un listado del contenido del carro de la compra–. El modelo no debe tener conocimiento directo sobre la vista.

5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Razor Pages ofrece una manera más fácil de crear características de aplicaciones basadas en páginas. Encapsula la lógica del lado servidor para una determinada 'página' lógica en una aplicación web. En lugar de controlar las solicitudes con los métodos de acción de controlador, se ejecutan los controladores de modelo de página, como 'OnGet()', con lo que se representa la página asociada de forma predeterminada. Razor Pages simplifica el proceso de compilar páginas individuales en la aplicación ASP.NET Core sin dejar de proporcionar todas las características arquitectónicas de ASP.NET Core MVC. Es una buena opción predeterminada para la funcionalidad basada en páginas.

Con el fin de optimizar la experiencia del usuario para que no tengan que esperar mucho tiempo al acceder al contenido, se hace un estudio de las distintas tecnologías y se llega a la conclusión de que la que ofrece mayor ventaja es AJAX (Asynchronous JavaScript and XML). Este conjunto de técnicas de desarrollo web permiten que las aplicaciones web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano –ver [55]–.

Tanto JavaScript como XML funcionan de forma asíncrona en AJAX. Como resultado, cualquier aplicación web que use AJAX puede enviar y recuperar datos del servidor sin la necesidad de volver a cargar toda la página.

En la ilustración 8.2 se expone el diagrama del funcionamiento de AJAX, consistente en:

Ilustración 8.2 Diagrama de modelo de aplicación web con AJAX

1. El navegador crea una llamada de JavaScript que luego activará XMLHttpRequest.
2. En segundo plano, el navegador web crea una solicitud HTTP al servidor.
3. El servidor recibe, recupera y envía los datos al navegador web.
4. El navegador web recibe los datos solicitados que aparecerán directamente en la página. No es necesario recargar.

Nota: Diagrama de modelo de aplicación web con AJAX

Fuente: https://www.hostinger.es/tutoriales/que-es-ajax

La riqueza en funciones de alto nivel de la biblioteca de JavaScripts, JQuery [56] hace que cosas tales como el recorrido y la manipulación de documentos HTML, el manejo de eventos, la animación y AJAX, sean mucho más simples. JQuery simplifica las cosas complicadas de JavaScript, como las llamada AJAX y la manipulación del DOM –Document Object Model, estructura del documento HTML–. Las características de la biblioteca JQuery son:

- Manipulación HTML/DOM.
- Manipulación CSS.
- Métodos de eventos HTML.
- Efectos y animaciones.
- AJAX.
- Utilidades.

8.2.1.2. Microsoft Identity Web

La gestión de usuarios, en este proyecto, se lleva a cabo con Microsoft Identity. Microsoft Identity Web es un conjunto de bibliotecas de ASP.NET Core que simplifican la compatibilidad con la autenticación y la autorización de aplicaciones.

En la ilustración 8.3 se muestra un diagrama con los tipos de aplicación admitidos y sus argumentos pertinentes.

Ilustración 8.3 Tipos de aplicación admitidos por Microsoft Identity

Scenario	Command	Template		Audience	Call web API (optional)
Web API	dotnet new	webapi2	--auth	SingleOrg	
MVC web app		mvc2			--calls-graph**
Razor web app		webapp2		MultiOrg*	
Blazor Server		blazorserver2			--called-api-scopes "*scopes*" --called-api-url webApi*Url*
Blazor WebAssembly		blazorwasm2 blazorwasm2 --hosted		IndividualB2C	

Nota: Tipos de aplicación admitidos por Microsoft Identity

Fuente: https://docs.microsoft.com/es-es/azure/active-directory/develop/microsoft-identity-web

Los posibles escenarios de aplicación admitidos son:

- aplicación web que permite iniciar sesión a los usuarios.
- aplicación web que inicia la sesión de los usuarios y llama a una API web en su nombre.
- API web protegida a la que solo pueden acceder los usuarios autenticados.
- API web protegida que llama a otra API web –de bajada– en nombre del usuario con sesión iniciada.

El esquema de la ilustración 8.4 muestra la autenticación implementada en este proyecto, con la que la aplicación web permitirá iniciar sesión a los usuarios y llamar a la API web en nombre del usuario con sesión iniciada.

Ilustración 8.4 Esquema de autenticación de la aplicación web de ImageMetaData

Nota: Esquema de autenticación de la aplicación web de ImageMetaData

Fuente: Elaboración propia

8.2.1.3. SendGrid

SendGrid es una plataforma estadounidense para la gestión de correos electrónicos. SMTP –Simple Mail Transfer Protocol– es una forma rápida y sencilla de enviar correo electrónico de un servidor a otro. SendGrid proporciona un servicio SMTP que permite entregar el correo electrónico a través de los servidores en lugar del propio cliente o servidor.

La API SMTP de SendGrid también permite especificar instrucciones de manejo de correo electrónico personalizadas utilizando una lista codificada en JSON llamada *cabecera X-SMTPAPI*. Esta cabecera es analizada por SendGrid para modificar el mensaje de la forma en la que se especifique.

8.3. Modelo de datos

La gestión de usuarios lleva a emplear una implementación de ASP.NET Identity para la base de datos SQLite, utilizando la funcionalidad de Entity Framework en la aplicación web.

SQLite [56] es una librería en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta fiabilidad y con todas las funciones –ver sus ventajas en la ilustración 8.5–. El formato de archivo SQLite es estable, multiplataforma y compatible con versiones anteriores, utilizándose éstos, habitualmente, como contenedores para transferir contenidos ricos entre sistemas y como formato de archivo de datos a largo plazo.

Esta herramienta de software libre cumple con las características ACID –atomicidad, consistencia, aislamiento y durabilidad– y permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware.

Ilustración 8.5 Ventajas de SQLite

Nota: Ventajas de SQLite

Fuente: https://openwebinars.net/blog/sqlite-para-android-la-herramienta-definitiva

8.4. Implementación

En este apartado se analiza la implementación de cada una de las partes que componen el proyecto.

8.4.1. Preparación del entorno de desarrollo

El desarrollo de este proyecto se ha llevado a cabo siguiendo los pasos que se detallan a continuación.

1. Instalación de Visual Studio Community para el desarrollo de la aplicación web con ASP NET Core.

2. Instalación de Visual Studio Code para el desarrollo de la API con Python: Visual Studio Code es un editor de código libre y abierto, diseñado por Microsoft y optimizado para crear y depurar aplicaciones web modernas y aplicaciones en la nube.

Se ha descargado el instalador de la última versión disponible en su página oficial, en [58].

Una vez instalado el editor de código, se han instalado varias extensiones que facilitarán la labor a la hora del desarrollo de la aplicación:

- Python para Visual Studio Code: Una extensión con amplio soporte para el lenguaje Python –para todas las versiones soportadas activamente: 2.7, >=3.5–, que incluye características como IntelliSense, linting, depuración,

navegación de código, formato de código, soporte para notebook Jupyter, refactorización, explorador de variables, explorador de pruebas, fragmentos y mucho más.

- Flask-snippets. Colecciones de fragmentos de código: Inicialmente portado desde PyCharm, TextMate, SublimeText y otros editores / IDEs.
- Jinja2 Snippets Kit: Esta extensión agrupa todos los fragmentos de jinja más utilizados, listos para su utilización en plantillas HTML. También incluye utilidades para tareas como generar números, unir listas y cosas por el estilo.

3. Instalación de Python 3.9

4. Creación Entorno Virtual Python. Se ha hecho uso de un entorno virtual Python para mantener en un espacio separado el proyecto con sus dependencias y módulos. Este entorno es específico para el proyecto y no interferirá, no se verá afectado, con las dependencias de otros proyectos.

- Creación del entorno virtual. Se ha creado el entorno virtual imagemetadata-venv con el siguiente comando `python -m venv imagemetadata-venv`
- Activación del entorno virtual. Antes de empezar a trabajar en el proyecto, siempre se debe de activar el entorno virtual con el comando desde la carpeta del proyecto

```
imagemetadata-venv\Scripts\activate
```

8.4.2. Servicio web API (webapi)

El módulo imagemetadata-webapi.py es el encargado de la creación del servicio web. Este quedará a la escucha de peticiones en el puerto 8080, publicando los siguientes endpoints:

- Obtener metadata – */image/metadata* –: Recibe la imagen, la procesa para extraer los metadatos y los devuelve al peticionario.
- Desnudar imagen – */image/strip* –: Recibe la imagen, la procesa para borrarle los metadatos y la devuelve sin ellos al peticionario.

Almacena los metadatos eliminados en un archivo encriptado, para en un futuro poder devolvérselos a la imagen si así es requerido. Este archivo tendrá un nombre único, que es almacenado en un metadato de la imagen devuelta, estableciendo de esta forma la asociación entre dicha imagen y sus metadatos.

- Desnudar imagen parcialmente – */image/strip_partial* –: Recibe la imagen y una lista de los metadatos a eliminar de ella. Procesa la imagen eliminando dichos metadatos y la devuelve sin ellos al peticionario.

Almacena los metadatos eliminados en un archivo encriptado, para en un futuro poder devolvérselos a la imagen si así es requerido. Este archivo tendrá un nombre único, que es almacenado en un metadato de la imagen devuelta, estableciendo de esta forma la asociación entre dicha imagen y sus metadatos.

- Vestir imagen – */image/dress* –: Recibe la imagen. Si no es una imagen desnudada anteriormente por la aplicación, devuelve un error indicándolo.

Si es una imagen desnudada anteriormente por la aplicación, recupera sus metadatos, que están almacenados en el archivo visto en los puntos anteriores y devuelve la imagen al peticionario.

8.4.3. Aplicación web (webapp)

La aplicación web está desarrollada utilizando ASP.NET Core –con C#, como lenguaje de programación– y se encarga de:

- la gestión de usuarios, utilizando la tecnología de Microsoft Identity, y
- del procesado de los metadatos de las imágenes, apoyándose para ello en los servicios prestados por la api.

8.5. Pruebas y validación

Este estadio es necesario para verificar que el prototipo de sistema software desarrollado cumple con los requisitos especificados y logra su cometido. Por una parte se presenta la necesidad de probar cada una de las funcionalidades de los elementos de la interfaz y por otro, la seguridad y eficacia tanto del borrado de los metadatos de una imagen, como su posterior inclusión.

En el proceso de validación del correcto funcionamiento de la interfaz es necesario la realización de:

- Pruebas de Contenido: se hace una revisión manual de los contenidos del sitio web a través de la navegación de sus páginas, realizando pruebas semánticas –exactitud de la información presentada–, sintácticas –gramática y ortografía del contenido–. Además, se comprueba que la información es concisa, precisa y exacta y que no se infrinjan derechos de autor o marcas registradas.
- Pruebas de Interfaz de Usuario. Como expone Hassan Montero en [59], se ha de diseñar un sitio web para que los usuarios puedan interactuar con él de la forma más fácil, cómoda e intuitiva posible. En consecuencia, se hacen pruebas de HTML dinámico ejecutando la aplicación, pruebas dentro del mayor número de casos posibles para asegurar su compatibilidad, pruebas para la validación de la usabilidad midiendo cuanto de fácil es entender la aplicación con la participación de varios usuarios inexpertos. Se ha prestado especial atención en los casos en los que el sistema debe mostrar mensajes de error o alerta y queden resueltos. En lo referente al diseño, y tal y como expone Jakob Nielsen en sus ’10 heurísticas/principios básicos de usabilidad’ [60], se ha preferido un diseño estético y minimalista.
- Pruebas de semántica de la interfaz para evaluar cuán bien el diseño se ocupa de los usuarios ofreciendo una dirección clara y manteniendo consistencia de lenguaje y enfoque.
- Pruebas de navegación. Su propósito es garantizar que todos los mecanismos que permiten al usuario de la Web app viajar a través de ella sean funcionales. Se comprueba para ello los vínculos de navegación.

Para probar la eficiencia y eficacia del prototipo de la herramienta ImageMetaData, se ha utilizado un conjunto de 8 imágenes, tanto propias como tomadas con drones. En la tabla 8-1 se muestra el resultado, donde algunos metadatos de las imágenes tomadas con drones no son legibles por la herramienta y al usar la funcionalidad de seleccionarlos parcialmente –desnudar parcialmente la imagen– muestra el error de que sólo se pueden eliminar todos. Sin embargo, cuando se prueban los servicios de desnudar –eliminar todos los metadatos– y vestir la imagen –incluir los metadatos–, el proyecto tiene una efectividad del 100%

Tabla 8-1 Porcentajes de aciertos de ImageMetaData

Imagen	Porcentaje eliminar todos los metadatos	Porcentaje eliminar algunos metadatos	Porcentaje inclusión de los metadatos
Imagen 1: dron	100%	98%	99%
Imagen 2: dron	100%	98%	99%
Imagen 3: dron	100%	98%	99%
Imagen 4: propia	100%	100%	100%
Imagen 5: propia	100%	100%	100%
Imagen 6: propia	100%	100%	100%
Imagen 7: propia	100%	100%	100%
Imagen 8: propia	100%	100%	100%

Nota: Pruebas y Validación: porcentajes de aciertos de ImageMetaData

Fuente: Elaboración propia

En general, se pueden considerar satisfactorios los resultados obtenidos, pudiendo destacarse que en el caso de desnudar por completo la imagen, y por consiguiente volverla a vestir, el proyecto tiene una eficacia del 100%.

9. CONCLUSIÓN

Bajo mi punto de vista, uno de los puntos principales por los que se hace necesario el desarrollo de este proyecto, es que a día de hoy, no existe ninguna normativa que regule los metadatos. Además, lo realmente relevante es que se sigue aportando legislación de un ámbito territorial parcial cuando el tráfico de datos y sobre todo de metadatos, es global ya que se realiza a través de la red y ésta es mundial.

El prototipo desarrollado consigue seguridad en los datos y en las comunicaciones debido al uso de la criptografía y de los protocolos SSL/TLS. También la adquiere al cifrar el fichero que contendrá los metadatos, previniendo en caso de que éste sea víctima de un ataque hacker.

Los requisitos para garantizar la seguridad informática que se consiguen con todo ello son:

1. *Seguridad de los canales de comunicación.* Los metadatos tratados quedan protegidos con la encriptación y almacenados en un depósito externo. De esta forma, aquel que tenga la imagen ‘desnuda’ no tiene posibilidad directa de

manipularla e intentar desencriptar los datos y, además, al almacenarse en el servidor, se debería de poder hackear éste para acceder a ellos.

2. *Control de acceso a los datos.* El sistema permite el acceso a la información únicamente a agentes autorizados, siendo el administrador el único que pueda registrar a los usuarios.

3. *Autenticación.* Verificando de forma fiable la autenticidad de los usuarios que accedan a él. La herramienta Microsoft Identity asegura que sólo puedan acceder los usuarios autenticados. Durante el registro, el nuevo usuario recibirá un correo electrónico en el buzón especificado, y no podrá iniciar sesión hasta que confirme dicho registro.

4. *Integridad y no repudio.* Con la encriptación y almacenamiento en un depósito externo de los metadatos, se asegura que éstos no han sido alterados en el momento de recuperarlos para devolverlos a la imagen –para 'vestir' la imagen–. Para este proceso, nuevamente el usuario queda registrado y autenticado puesto que sólo a través del sistema se pueden devolver los metadatos a su imagen correspondiente, y para acceder a éste es necesario la autenticación.

5. *Disponibilidad y control de acceso.* El proyecto desarrollado permite acceder a la información –metadatos de una imagen– únicamente a agentes autorizados, exclusivamente por el usuario administrador.

Una vez finalizado el proyecto y analizando los resultados obtenidos, se puede considerar que los objetivos iniciales se han visto satisfechos.

Si bien es cierto que las técnicas de implementación utilizadas no son las más sofisticadas existentes, sí que son las que mejor se adecúan a las características y requisitos del proyecto. Cabe mencionar además, que el trabajo de investigación y documentación ha sido más arduo que el de implementación.

10. LÍNEAS FUTURAS

En este capítulo, se plantean varias alternativas para investigaciones futuras, que amplíen y mejoren la funcionalidad y seguridad de la aplicación desarrollada:

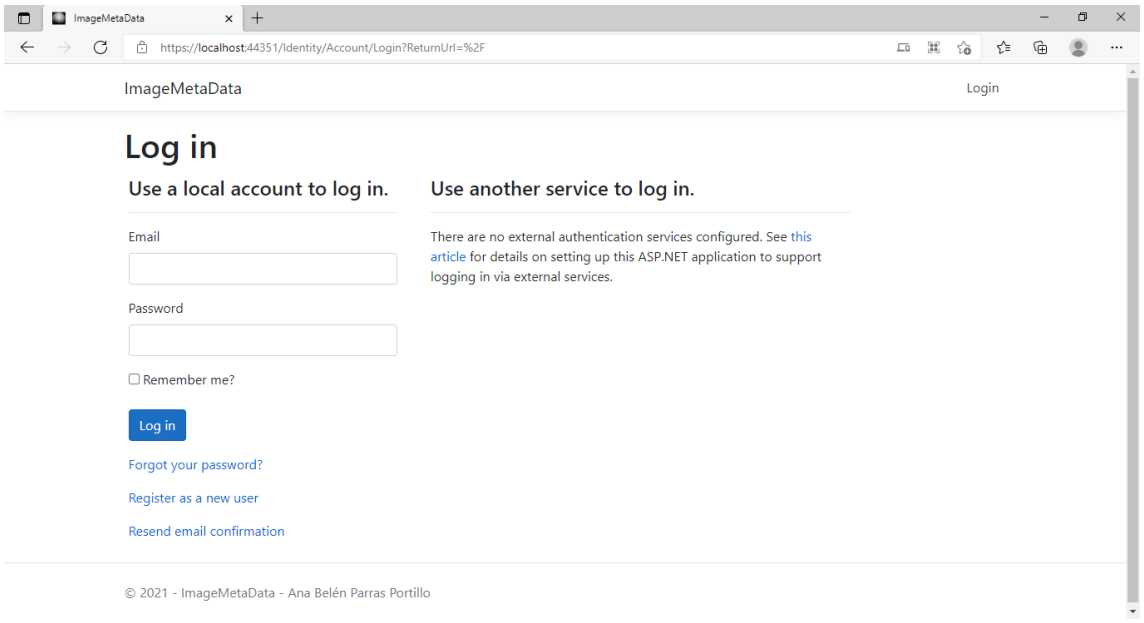
- Ampliar la funcionalidad del administrador para que pueda gestionar y mantener los derechos de acceso a través de un conjunto limitado de perfiles, en lugar de ocuparse de un gran número de usuarios individuales.
- Que el sistema registre, en la pista de auditoría, datos sobre los usuarios que lo usen, la fecha y la hora.
- Implementar el doble factor de autenticación –2FA, 'Two Factor Authentication'– o de autenticación multi-factor –MFA 'Muti-Factor Authentication–, ya que las contraseñas no son tan difíciles de descifrar como se podría pensar; sino que cada vez es aún más fácil hacerlo, dado el avance de la tecnología y de las técnicas de hacking.

11. MANUAL DE USUARIO

El manual de usuario se desarrolla con el objetivo de facilitar la tarea de conocimiento, uso y aprendizaje del sistema desarrollado. Contiene información acerca de todas las operaciones básicas que el sistema ofrece, así como capturas de pantallas útiles para el seguimiento de la explicación. El lenguaje utilizado es el más adecuado al perfil del usuario.

La aplicación web estará disponible a través de la URL https://localhost:<puerto> y presentará la página de inicio de sesión de usuarios –mostrada en la ilustración 11.1–:

Ilustración 11.1 Página de inicio de sesión de usuarios de ImageMetaData



Nota: Página de inicio de sesión de usuarios de ImageMetaData

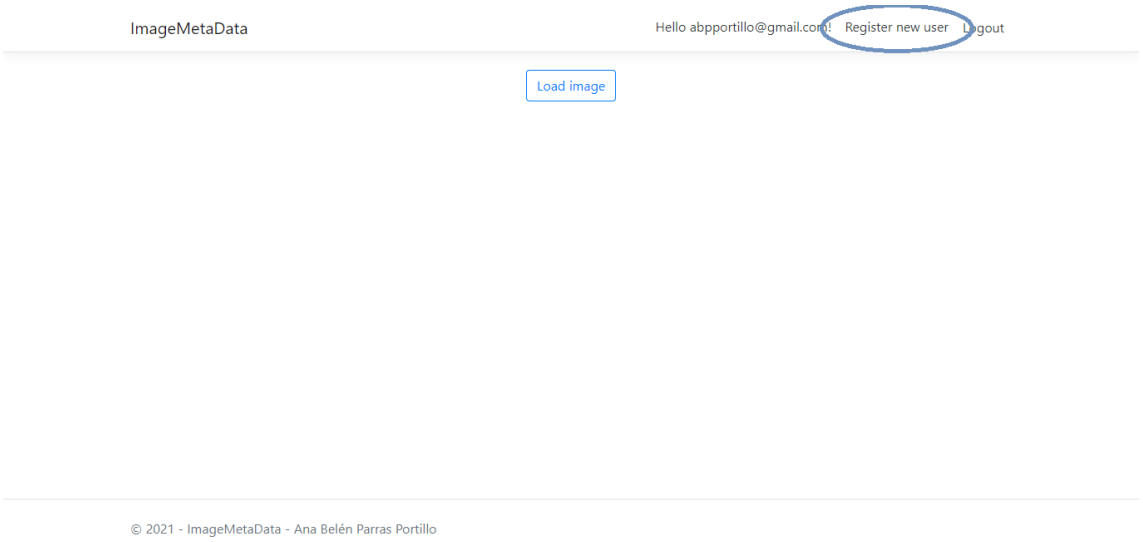
Fuente: Elaboración propia

Podrán iniciar sesión en la aplicación aquellos usuarios que hayan sido previamente registrados. El usuario administrador será el único usuario capaz de registrar a otros usuarios, controlando de esta forma quien podrá utilizar la aplicación.

11.1. Registro de Usuarios

El usuario administrador, tras logarse en la aplicación, podrá registrar nuevos usuarios a través de la opción *Register new user*. Esta opción sólo aparecerá para el usuario administrador –ver ilustración 11.2–.

Ilustración 11.2 Página de inicio de la aplicación

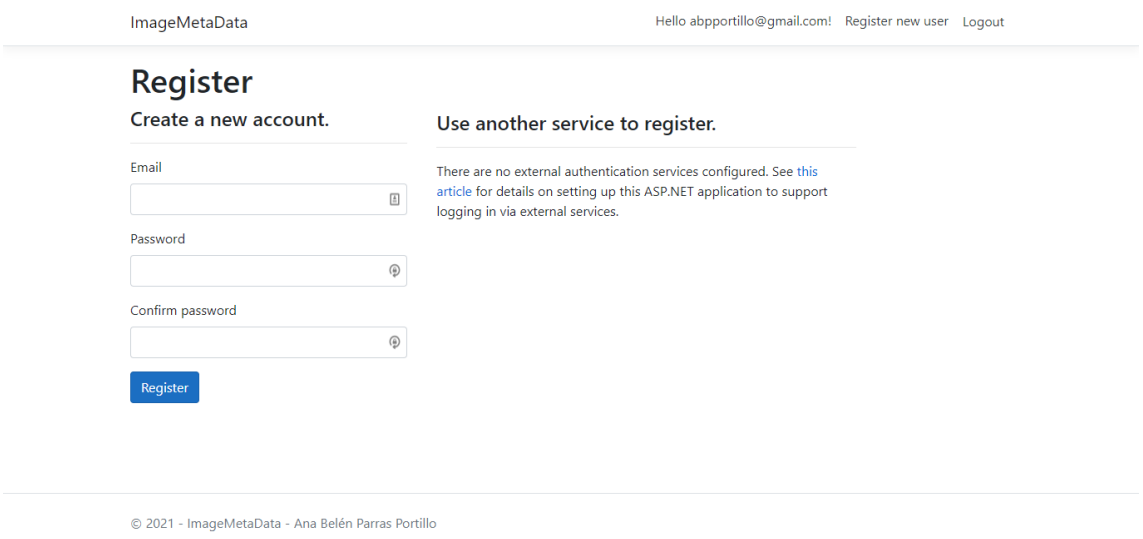


Nota: Página de inicio de sesión de la aplicación

Fuente: Elaboración propia

qué le llevará a la página de registro de usuarios:

Ilustración 11.3 Página de registro de Usuarios de ImageMetaData



Nota: Página de registro de Usuarios de ImageMetaData

Fuente: Elaboración propia

Como muestra la ilustración 11.3, aquí completará los datos requeridos: dirección de correo electrónico y contraseña, y a través del botón *Register* quedará

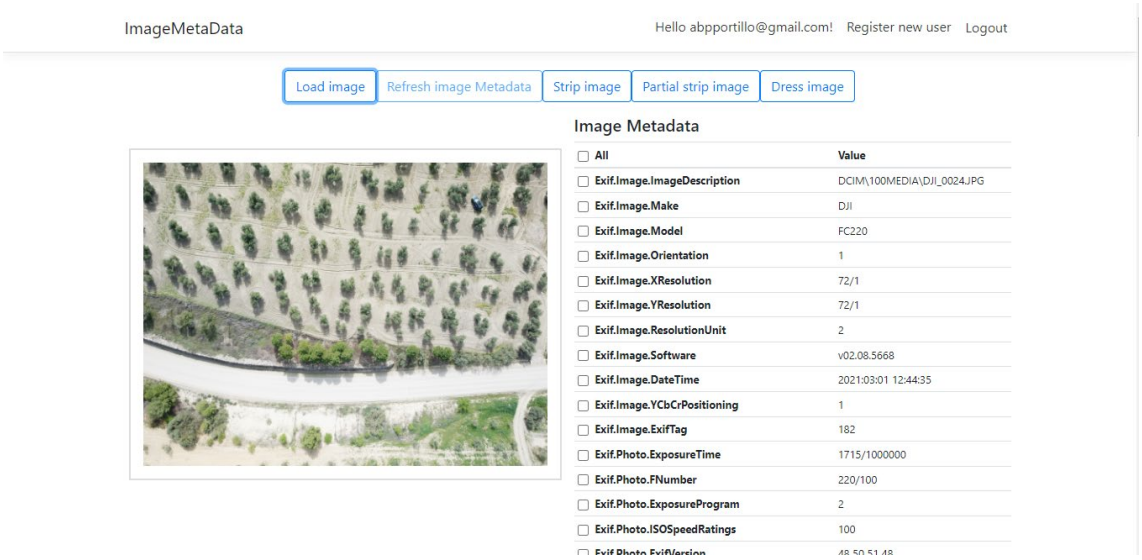
registrado el nuevo usuario en la aplicación. Este nuevo usuario recibirá un correo electrónico en el buzón especificado, y no podrá iniciar sesión hasta que confirme el registro.

11.2. Carga de la Imagen

Recién iniciada sesión, la aplicación muestra una interface muy simple, con un único botón *Load image* –mostrada en la ilustración 11.4–.

Este botón permitirá seleccionar la imagen a procesar y subirla desde nuestro equipo local a la aplicación, que la visualizará junto con el listado de sus metadatos y nuevas opciones de procesamiento para la imagen.

Ilustración 11.4 Load Image



Nota: Load Image de ImageMetaData

Fuente: Elaboración propia

11.3. Refresco de Metadatos

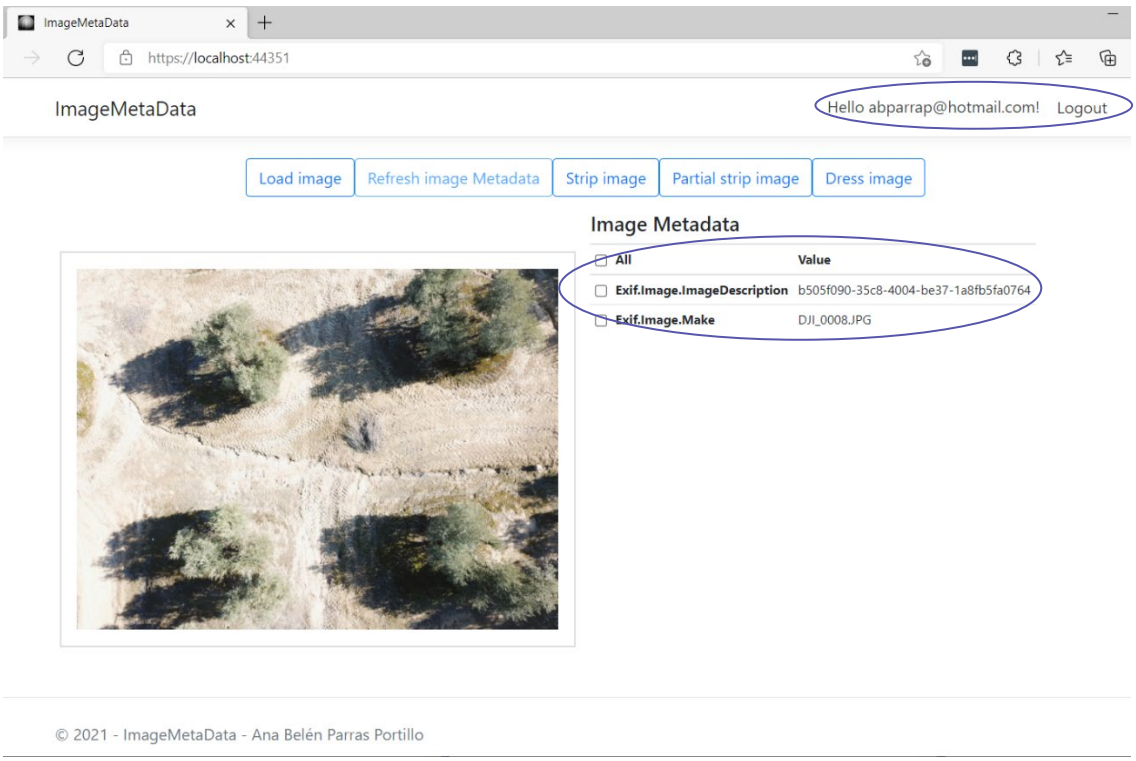
Si por alguna razón la consulta de los metadatos de la imagen fallase, se tendrá la oportunidad de volver a pedirlos a través del botón *Refresh image Metadata.*

11.4. Desnudar Imagen

El botón *Strip image* permite descargar al equipo local, una copia de la imagen que se tiene cargada en la aplicación, sin los metadatos, lo que llamamos una imagen desnuda. Esta nueva imagen contendrá solamente los metadatos correspondientes al nombre de dicha imagen y al nombre del fichero, encriptado, que contendrá dichos metadatos. Este fichero se almacenará en la API, en la carpeta exv.

Este servicio que queda reflejado con la ilustración 11.5, y además, se ha tomado usando la aplicación un usuario no administrador con el propósito de mostrar que no aparece el servicio de registrar a nuevos usuarios.

Ilustración 11.5 'Desnudar' imagen



Nota: 'Desnudar' imagen en ImageMetaData

Fuente: Elaboración propia

11.5. Desnudar Imagen Parcialmente

El botón *Partial strip image* permite descargar al equipo local, una copia de la imagen que se tiene cargada en la aplicación sin los metadatos que hayamos seleccionado previamente en la lista de metadatos. Esta nueva imagen contendrá el resto de los metadatos que no se seleccionaron y el nombre del fichero, encriptado, que contendrá los metadatos ofuscados. Este fichero se almacenará en la API, en la carpeta exv.

11.6. Vestir Imagen

Si se carga una imagen que previamente haya sido desnudada siguiendo alguno de los dos métodos vistos anteriormente, se podrá devolver los metadatos (vestirla) que tenía originalmente, antes de ser desnudada, descargando el resultado en un nuevo archivo a nuestro equipo local.

ANEXO I: MANUAL DE INSTALACIÓN Y EJECUCIÓN DE LA APLICACIÓN

El sistema final ha sido concebido como un sistema distribuido, con dos servicios diferenciados:

Un **Servicio Web RESTful**, que atenderá peticiones para procesamiento de metadatos de las imágenes.

Y una **aplicación Web**, que consumirá la funcionalidad publicada por dicho servicio web, habilitando al usuario para procesar las imágenes.

Anexo I.1. Estructura de la carpeta del proyecto

El contenido de la carpeta raíz del proyecto es el siguiente:

- **imagemetadata-webapi**, carpeta con el código fuente del servicio web y la utilidad exiv2 para la gestión de metadatos en imágenes.
- **imagemetadata-webapp,** carpeta con el código fuente de la aplicación web.

Anexo I.2. Instalación del Servicio web

i.2.1. Creación de un entorno virtual Python

Situándose en la carpeta del proyecto, en la carpeta imagemetadata-webapi, desde el terminal ejecutar el comando:

```
$ python3 -m venv imagemetadata-venv
```

Si se está trabajando con Windows, se hará desde el símbolo del sistema:

```
> python -m venv imagemetadata-venv
```

i.2.2. Activación del entorno virtual

Antes de empezar a trabajar con el proyecto es necesario activar el entorno virtual. Para ello se ejecuta desde el terminal:

```
$ . imagemetadata-venv/bin/activate
```

En Windows: `> imagemetadata-venv\Scripts\activate`

i.2.3. Instalación de los módulos Python requeridos

Una vez activado el entorno virtual, ejecutar los comandos:

```
pip install -r requirements.txt
```

y

```
pip install -t <dir> flask_cors
```

donde <dir> = \...Lib\site-pakages dentro del entrono virtual

i.2.4. Ejecución del Servicio web

Desde un terminal activar el entorno virtual Python `imagemetadata-venv` (ver el punto i.2.2 Activación del entorno virtual en el apartado de Instalación)

Una vez activado dicho entorno virtual, acceder a la carpeta imagemetadata-webapi e iniciar el servidor:

```
$ cd imagemetadata-webapi
$ python image_metadata_webapi.py
```

El servidor quedará a la escucha en http://localhost:8080

Anexo I.3. Publicación de la Aplicación web

Se publicará la aplicación en algún servidor web del que se disponga, como Internet Information Services (IIS), Apache, NGINX entre otros.

Bibliografía

[1] L. C. e. d. 1978, «app.congreso.es,» Congreso de los Diputados, 2003. [En línea]. Available: https://app.congreso.es/consti/constitucion/indice/titulos/articulos.jsp?ini=18&tipo=2. [Último acceso: 02 05 2021].

[2] «duoc.cl,» [En línea]. Available: https://www.duoc.cl/carreras/analista-programador-computacional/. [Último acceso: 01 09 2021].

[3] «boe.es,» [En línea]. Available: https://www.boe.es/boe/dias/2021/03/10/pdfs/BOE-A-2021-3747.pdf. [Último acceso: 01 09 2021].

[4] C. M. Fernández, «AENOR,» [En línea]. Available: http://www.cpiicm.es/wp-content/uploads/sites/3/2019/02/CY18-Ecosistema-Digital-de-AENOR.pdf. [Último acceso: 12 Mayo 2021].

[5] L. N. d. C. d. Software, «Ingeniería del Software: Metodologías y Ciclos de Vida,» Marzo 2009. [En línea]. Available: https://www.academia.edu/9795641/INGENIER%C3%8DA_DEL_SOFTWARE_METODOLOG%C3%8DAS_Y_CICLOS_DE_VIDA_Laboratorio_Nacional_de_Calidad_del_Software. [Último acceso: 18 Mayo 2021].

[6] «Agile-Spain,» [En línea]. Available: https://agile-spain.org/utiles/manifesto-agil/. [Último acceso: 2021 mayo 18].

[7] «PowerData- Metadatos, definición y características,» [En línea]. Available: https://www.powerdata.es/metadatos. [Último acceso: 01 Junio 2021].

[8] S. Chastain, «LifeWire,» 07 Noviembre 2019. [En línea]. Available: https://www.lifewire.com/what-is-metadata-1701735. [Último acceso: 02 Junio 2021].

[9] «EXIF.org,» 2021. [En línea]. Available: https://www.exif.org/. [Último acceso: 02 Junio 2021].

[10] «JEITA,» [En línea]. Available: https://www.jeita.or.jp/english/. [Último acceso: 03 Junio 2021].

[11] IPTC. [En línea]. Available: https://www.iptc.org/. [Último acceso: 02 Junio 2021].

[12] C. Matsuoka y H. J. Carraro, «Extended Module Player,» [En línea]. Available: http://xmp.sourceforge.net/. [Último acceso: 02 Junio 2021].

[13] Adobe, «Adding Intelligence to Media,» [En línea]. Available: https://www.adobe.com/products/xmp.html. [Último acceso: 02 Junio 2021].

[14] «PRISM METADATA,» [En línea]. Available: https://idealliance.org/specifications/prism-metadata/. [Último acceso: 03 Junio 2021].

[15] «JPEG 2000,» [En línea]. Available: https://jpeg.org/jpeg2000/index.html. [Último acceso: 03 Junio 2021].

[16] Marina, «Metadatos. Definición, funciones y ejemplos,» Grupo ATICO34, 11 Febrero 2021. [En línea]. Available: https://protecciondatos-lopd.com/empresas/metadatos/#Segun_su_funcion. [Último acceso: 03 Junio 2021].

[17] «Llevando la Teoría a la Práctica,» Biblioteca de la Universidad de Cornell, 2000-2003. [En línea]. Available: http://preservationtutorial.library.cornell.edu/tutorial-spanish/contents.html. [Último acceso: 04 Junio 2021].

[18] «innovation in metadata design, implementation & best practice,» Dublin Core Metadata Initiative, [En línea]. Available: http://dublincore.org/. [Último acceso: 07 Junio 2021].

[19] I. 15836:2003, «Information and documentation- The Dublin Core metadata element set,» [En línea]. Available: https://www.iso.org/standard/37629.html. [Último acceso: 07 Junio 2021].

[20] «<ead>Encoded Archival Description,» 15 Septiembre 2020. [En línea]. Available: https://www.loc.gov/ead/. [Último acceso: 07 Junio 2021].

[21] c. y. d. gob.es, «Norma Internacional General de Descripción Archivística,» 2020. [En línea]. Available: https://www.culturaydeporte.gob.es/ca/dam/jcr:2700ee49-7b45-40c1-9237-55e3404d3a3f/isad.pdf. [Último acceso: 08 Junio 2021].

[22] «Nomas MARC,» [En línea]. Available: https://www.loc.gov/marc/marcspa.html. [Último acceso: 08 Junio 2021].

[23] «Metadata Encoding & Transmission Standard,» 28 Octubre 2020. [En línea]. Available: https://www.loc.gov/standards/mets/. [Último acceso: 09 Junio 2021].

[24] «MPEG-21,» Wikipedia, [En línea]. Available: https://es.wikipedia.org/wiki/MPEG-21. [Último acceso: 09 Junio 2021].

[25] «GEOSPATIAL STANDARDS,» [En línea]. Available: https://www.fgdc.gov/metadata/csdgm-standard. [Último acceso: 11 Junio 2021].

[26] J. D. Peláez, «Instituto Nacional de Seguridad,» INCIBE, 26 Noviembre 2013. [En línea]. Available: https://www.incibe.es/protege-tu-empresa/blog/metadatos-webs-empresas. [Último acceso: 14 Junio 2021].

[27] «Guía de Seguridad de las TIC CCN-STIC 835,» Marzo 2017. [En línea]. Available: https://www.ccn-cert.cni.es/pdf/guias/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/2031-ccn-stic-835-borrado-de-metadatos-en-el-marco-del-ens/file.html. [Último acceso: 14 Junio 2021].

[28] G. d. España, «Agencia Estatal Boletín oficial del Estado,» [En línea]. Available: https://boe.es/buscar/act.php?id=BOE-A-1982-11196#:~:text=Ley%20Org%C3%A1nica%201%2F1982%2C%20de%205%20de%20mayo%2C%20de,115%2C%20de%2014%2F05%2F1982.%20Entrada%20en%20vigor%3A%2003%2F06%2F1982.%20Departamento%3A. [Último acceso: 15 Junio 2021].

[29] G. d. España, «Ley Orgánica 15/1999,» [En línea]. Available: https://www.boe.es/buscar/act.php?id=BOE-A-1999-23750. [Último acceso: 15 Junio 2021].

[30] G. d. España, «Real Decreto 1720/2007,» [En línea]. Available: https://www.boe.es/buscar/act.php?id=BOE-A-2008-979. [Último acceso: 15 Junio 2021].

[31] Público, «¿Quién nos protege de nuestros metadatos?,» DISPLAY CONNECTORS, SL., [En línea]. Available: https://www.publico.es/sociedad/proteccion-datos-metadatos.html. [Último acceso: 16 Junio 2021].

[32] EUR-Lex. [En línea]. Available: https://eur-lex.europa.eu/legal-content/ES/LSU/?uri=CELEX:32002L0058. [Último acceso: 16 Junio 2021].

[33] «Novedades sobre el Reglamento ePrivacy,» 04 Marzo 2021. [En línea]. Available: https://ecija.com/sala-de-prensa/novedades-sobre-el-reglamento-eprivacy/. [Último acceso: 16 Junio 2021].

[34] G. d. España, «Ley Orgánica 3/2018,» Agencia Estatal Boletín Oficial del Estado, [En línea]. Available: https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673. [Último acceso: 17 Junio 2021].

[35] M. J. L. López, Criptografía y Seguridad en Computadores, 4ª Ed. 2010, 2019.

[36] Ibíd.

[37] «ugr.es,» [En línea]. Available: http://www.ugr.es/~esantos/simetrica.html. [Último acceso: 22 06 2021].

[38] M. Sanz Romero, «computerhoy.com,» 20 02 2021. [En línea]. Available: https://computerhoy.com/reportajes/tecnologia/qkd-criptografia-cuantica-que-es-814885. [Último acceso: 22 06 2021].

[39] B. Varona, «Comunicación y Criptografía Cuántica,» 2020 06 2020. [En línea]. Available: https://www.techedgegroup.com/es/blog/comunicacion-criptografia-cuantica. [Último acceso: 23 06 2021].

[40] «avast.com,» [En línea]. Available: https://www.avast.com/es-es/c-what-is-tcp-ip. [Último acceso: 22 06 2021].

[41] M. J. Lucena López, «Criptografía y Seguridad en Computadores,» 2019, pp. 295-296.

[42] D. Lázaro, «Introducción a los Web Services,» [En línea]. Available: https://diego.com.es/introduccion-a-los-web-services. [Último acceso: 23 06 2021].

[43] Ó. Blancarte, «oscarblanccarteblog.com,» 06 03 2017. [En línea]. Available: https://www.oscarblancarteblog.com/2017/03/06/soap-vs-rest-2/. [Último acceso: 24 06 2021].

[44] «tugesto.com,» 27 04 2018. [En línea]. Available: https://tugesto.com/blog/web-service/. [Último acceso: 24 06 2021].

[45] «MDAnet Archivos,» [En línea]. Available: https://mdanetarchivos.com/que-es-la-encriptacion-de-datos-aes256-grado-militar/.

[46] «code.visualstudio.com,» [En línea]. Available: https://code.visualstudio.com/docs. [Último acceso: 23 06 2021].

[47] Alba, «discoder.tech,» 23 01 2021. [En línea]. Available: https://www.discoder.tech/python-ventajas-desventajas/. [Último acceso: 24 06 2021].

[48] «blog.aulaformativa.com,» 22 06 2021. [En línea]. Available: https://blog.aulaformativa.com/listado-python-frameworks-basicas/. [Último acceso: 24 06 2021].

[49] «flask.palletsprojects.com,» [En línea]. Available: https://flask.palletsprojects.com/en/1.1.x/. [Último acceso: 24 06 2021].

[50] «blog.miguelgrinberg.com,» 28 11 2013. [En línea]. Available: https://blog.miguelgrinberg.com/post/restful-authentication-with-flask. [Último acceso: 26 06 2021].

[51] «pypi.org,» 06 01 2021. [En línea]. Available: https://pypi.org/project/Flask-Cors/. [Último acceso: 27 06 2021].

[52] «exiv2.org,» 15 06 2021. [En línea]. Available: https://exiv2.org/. [Último acceso: 27 06 2021].

[53] [En línea]. Available: https://exiftool.org/. [Último acceso: 27 06 2021].

[54] «dotnet.microsoft.com,» [En línea]. Available: https://dotnet.microsoft.com/apps/aspnet. [Último acceso: 28 06 2021].

[55] «learnrazorpages.com,» 09 04 2019. [En línea]. Available: https://www.learnrazorpages.com/razor-pages/ajax. [Último acceso: 30 06 2021].

[56] «jquery.com,» [En línea]. Available: https://jquery.com/. [Último acceso: 30 06 2021].

[57] «sqlite.org,» [En línea]. Available: https://www.sqlite.org/index.html. [Último acceso: 24 08 2021].

[58] [En línea]. Available: https://code.visualstudio.com/. [Último acceso: 02 07 2021].

[59] Y. Hassan Montero, «nosolousabilidad.com,» [En línea]. Available: http://www.nosolousabilidad.com/articulos/introduccion_usabilidad.htm. [Último acceso: 09 08 2021].

[60] J. Nielsen, «Grupo Nielsen Norman,» [En línea]. Available: https://www.nngroup.com/articles/ten-usability-heuristics/. [Último acceso: 01 08 2021].