Facultad de Informática

Trabajo de Fin de Grado

Cust-OTEA: Marcado de documentos de forma segura

Por
Samuel Antonio Eugercios Nevado

UNIVERSIDAD COMPLUTENSE
MADRID

Dirigido por

José Luis Vázquez Poletti
David Pacios Izquierdo

MADRID, 2022–2023

Abstract

This work aims to protect PDF documentation under secure and invisible watermarks, to prevent its theft and fraudulent use by websites illegally. In order to do such a thing, it has been used Python 3.8 to create the watermark and create the key-value from the author. On the other hand, LaTeX has been used to develop the document and this document itself. In addition, it has been compared the time to process 1000 documents in a local computer with cloud computing and serverless. It has been checked that the serverless architecture process faster the documents than the other two. Because of that, it has been created a distributed architecture to process the documents. As a future work, the application will be conected with the Moodle.

Keywords: Lambda, serverless, pdf, fraudulent, watermarks and cloud computing.

Resumen

Este trabajo pretende proteger la documentación PDF bajo marcas de agua seguras e invisibles, para evitar su robo y uso fraudulento por parte de sitios web de forma ilegal. Para ello, se ha utilizado Python 3.8 para crear la marca de agua y crear la clave-valor del autor. Por otro lado, se ha utilizado LaTeX para desarrollar el documento y este mismo. Además, se ha comparado el tiempo de procesado de 1000 documentos en un ordenador local frente cloud computing y serverless. Se ha comprobado que la arquitectura serverless procesa más rápido los documentos que las otras dos. Por ello, se ha creado una arquitectura distribuida para procesar los documentos. Como trabajo futuro, la aplicación se conectará con Moodle.

Palabras Clave: Lambda, serverless, pdf, fraudulento, marcas de agua y cloud computing..

Índice general

Chapter 1. Introduction

1.1. Motivation

Many times we think that the world is a fair and nice place, that everyone will respect what has taken you so much time and effort, whether it is to create a document, a piece of writing or a thesis. Until one day you see all that work fraudulently has been stolen by another person or company, to use it for their own benefit and eliminating any trace of the original author, claiming for themself the authorship.

Unfortunately, it happens more often than we think and in many areas of society, from the student who copies the homework done years before by another classmate, to the company that steals documentation and uploads it on their websites for financial gain, without the author being aware of it or being credited with the recognition of authorship.

With the growing technology of document editing and the dozens of programs that serve this purpose, it is logical to seek methods of securing the documentation created. And that the authorship cannot be removed in the event that it is obtained by third parties for possible malicious use.

We are able to create tools and technologies that can be used to protect our work. Researching and developing an efficient, free and free means of protection.

The main problem is the normalization of documentation theft, due to the free access to information at any time and place thanks to the Internet. Causing an endless trickle of document theft, for its fraudulent use or illegal sale without having to face the consequences, being the author who has the obligation to prove that he/she is the owner before the courts.

In conclusion, the rise of this type of theft is an extremely serious and widespread problem, rooted in all levels of society. Every year millions of economic losses are generated to companies, institutions and individuals, with a gradual growth of websites selling fraudulently acquired documentation. The most common cases are digital books and academic notes that are distributed and sold without the corresponding legal permissions.

1.2. Objective to be achieved

It is intended to develop an architecture to create an encrypted watermark in documents and difficult to find by third parties, to avoid the elimination of authorship with the following characteristics:

- Implementation of an encrypted, secure and invisible watermark.
- Modification of metadata.

The following operations will be performed on the software:

- Generation of an encrypted date and time stamp, linked to the author's e-mail address.
- Modification of the metadata of the document to be marked.
- Document marking and generation with secure watermark and encryption security marking.

The tool will consist of two modules:

- Document generation and marking module.
- Metadata modification module.

1.3. Documentation and style

Document license
The document is licensed under the CC-BY-SA license.

License of all software
All software is under MIT license.

Manuals and appendices
All manuals are in the public domain.

Capítulo 1. Introducción

1.1. Motivación

Muchas veces pensamos que el mundo es un lugar justo y bonito, que todas las personas van a respetar lo que tanto tiempo y esfuerzo te ha llevado realizar, ya sea para crear un documento, un escrito o un trabajo de fin de grado. Hasta que un día ves todo ese trabajo robado de forma fraudulenta por otra persona o empresa, para usarlo en su propio beneficio y eliminado cualquier rastro del autor original, reclamando para sí mismo la autoría.

Por desgracia, pasa más a menudo de lo que creemos y en muchos ámbitos de la sociedad, desde el alumno que copia la tarea realizada años anteriores por otro compañero, hasta la empresa que roba documentación y la sube en sus webs para obtener un beneficio económico, sin que el autor sea consciente de ello ni se le acredite el reconocimiento de la autoría.

Con la creciente tecnología de edición de documentos y las decenas de programas que sirven para ese propósito, es lógico buscar métodos de asegurar la documentación creada. Y que la autoría no pueda ser eliminada en el caso de ser obtenida por terceras personas para un posible uso malicioso.

Somos capaces de crear herramientas y tecnologías que pueden ser usadas para proteger nuestro trabajo. Investigando y desarrollando un medios de protección eficiente, gratuito y libre.

El principal problema es la normalización del robo de documentación, debido al libre acceso a la información en cualquier momento y lugar gracias a internet. Provocando un goteo interminable de robo de documentos, para su uso fraudulento o venta de forma ilegal sin tener que afrontar las consecuencias, siendo el autor quien tiene la obligación de demostrar que es el propietario ante los tribunales.

En conclusión, el auge de este tipo de robo es un problema actual extremadamente grave y de gran amplitud, arraigado en todos los estamentos de la sociedad. Cada año se generan millones de perdidas económicas a empresas, instituciones y particulares, creciendo de forma paulatina las webs de venta de documentación adquirida de forma fraudulenta. Los casos más comunes son los libros digitales y apuntes académicos que se distribuyen y venden sin los permisos legales correspondientes.

1.2. Objetivo a realizar

Se pretende desarrollar una arquitectura para crear una marca de agua cifrada en documentos y difícil de encontrar por terceras personas, para evitar la eliminación de la autoría con las siguientes características:

- Implementación de una marca de agua encriptada, segura e invisible.
- Modificación de los metadatos.

Las operaciones se desarrollarán:

- Generación de una marca encriptada con fecha y hora, enlazada con la dirección de correo del autor.
- Modificación de los metadatos del documento.
- Marcado del documento con una marca de agua encriptada y segura.

Esta herramienta está compuesta por dos módulos:

- Generación del documento y marcado.
- Modificación de los metadatos.

1.3. Documentación y estilo

Document license
The document is licensed under the CC-BY-SA license

The document is licensed under the CC-BY-SA license
Todo el software está bajo licencia MIT.

Manuales y anexos
Todos los manuales son de dominio público.

Capítulo 2. Estado del Arte

2.1. Artículos, proyectos y aplicaciones similares

Se ha realizado una búsqueda de artículos, aplicaciones y proyectos similares para la protección de documentación digital.

2.1.1. Canva

Canva[1] es una aplicación de diseño gráfico tanto para pc como para entornos móviles, que se utiliza en la modificación y edición de imágenes, documentos e introducciones para vídeos. Esta cuenta con muchas plantillas ya predefinidas y su uso se ha popularizado en estos dos últimos años. Especialmente en la plataforma de vídeo Youtube[1], ya que han aumentado los tutoriales sobre la eliminación de marcas de agua con Canva[2] en documentos e imágenes.

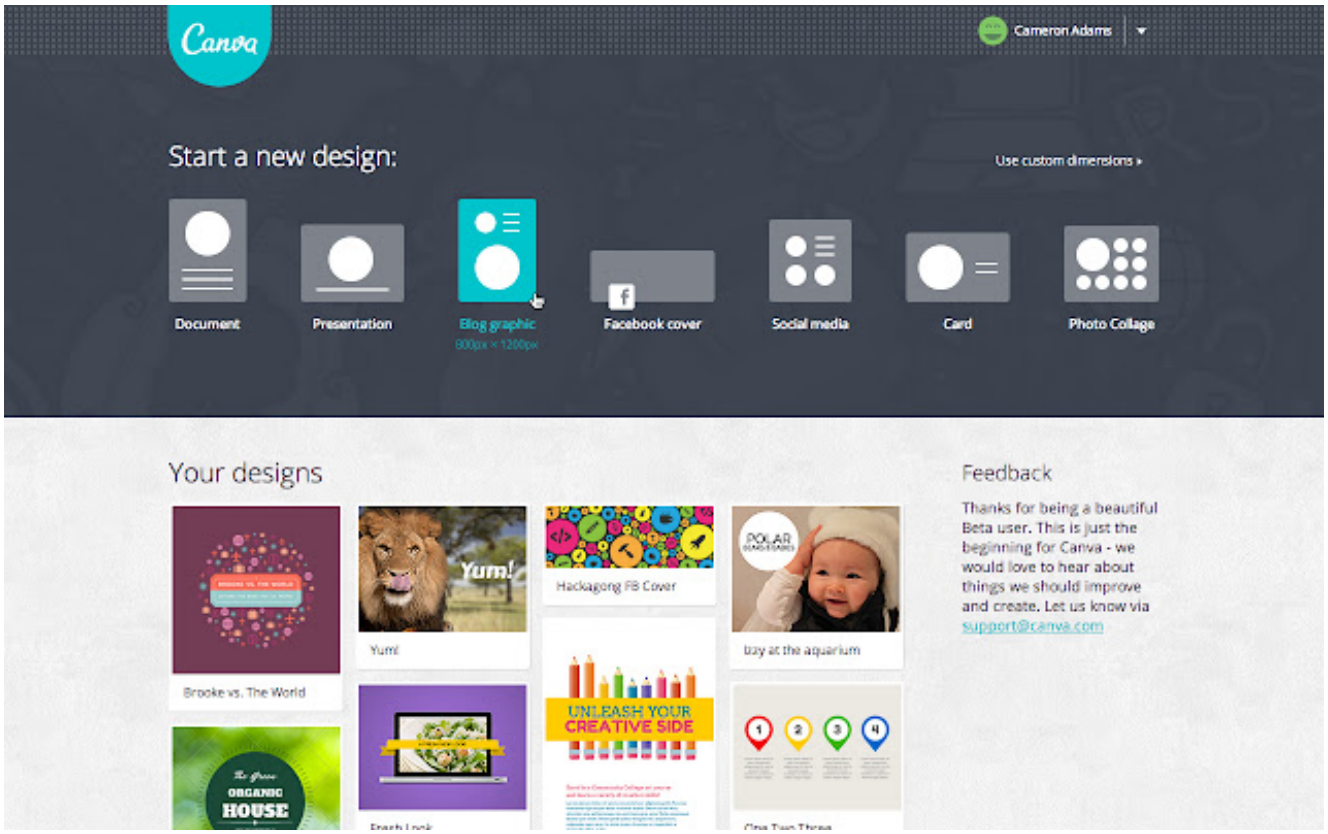

Figura 2.1: Imagen de entorno Canva

[1]Dirección web:`https://www.youtube.com/`
[2]Dirección web:`https://www.canva.com/es_es/`

2.1.2. Acrobat Reader

Acrobat Reader[2] es una aplicación de Adobe y una de las más antiguas en el uso de PDFs del mercado. Tiene dos versiones, la gratuita y la Pro. Dependiendo de la versión que tengamos, podremos tener más funcionalidades:

	Descargar lector de PDF	Probar Acrobat Pro
Ver e imprimir archivos PDF (incluido en pantallas pequeñas con Liquid Mode)	✓	✓
Compartir y comentar archivos PDF	✓	✓
Editar texto e imágenes de archivos PDF	✗	✓
Convertir archivos PDF a tipos de archivos como Word, PowerPoint y Excel	✗	✓
Comparar archivos PDF y redactar información confidencial	✗	✓
Enviar documentos PDF para firmar	✗	✓
Enviar archivos para firmar de forma masiva	✗	✓
Proteger archivos PDF con contraseña	✗	✓
Añadir marcas personalizadas a los contratos	✗	✓
Integraciones con Microsoft 365	✗	✓

Figura 2.2: Imagen de funcionalidades de las versiones de Acrobat Reader

2.1.3. LibreOffice

LibreOffice[3] es una suite de programas libres, entre los que se encuentra su módulo Writer que sirve para crear y modificar documentos de texto, y se puede insertar marcas de agua y transformar el documento de texto a PDF. Es un proyecto gratuito y de código abierto, pero que depende de la comunidad para su mejora y mantenimiento constante.

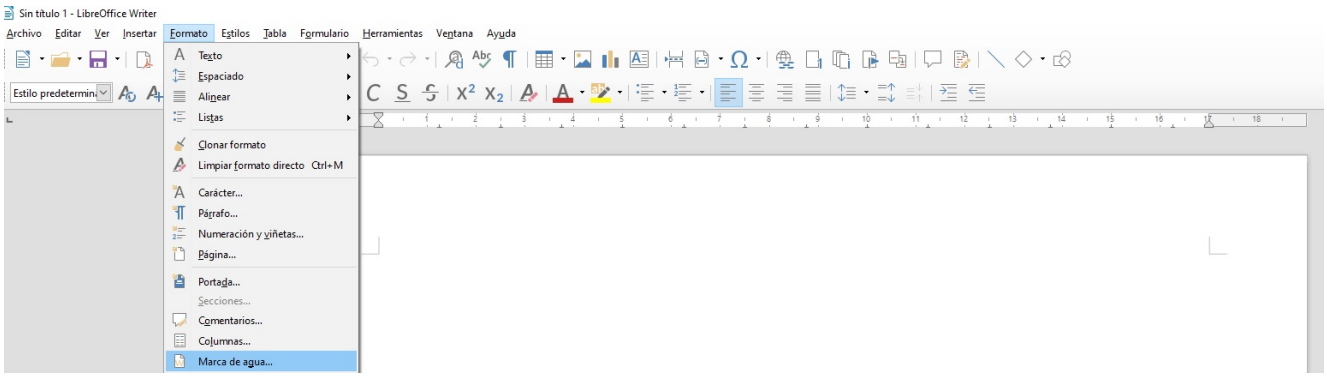

Figura 2.3: Imagen de entorno de Writer de LibreOffice

2.1.4. LaTex

LaTex[4] es sistema de composición de textos, concebido para la creación de documentación de gran calidad tipográfica. Usando su biblioteca *draftwatermark* se pueden crear marcas de agua personalizadas para los documentos en este entorno. El problema es su curva de aprendizaje, ya que, se requiere tiempo para aprender el entorno.

```
\documentclass{article}

\usepackage{draftwatermark}
\SetWatermarkText{\textsc{Confidencial}} % por defecto Draft
\SetWatermarkScale{5} % para que cubra toda la página
\SetWatermarkColor[rgb]{1,0,0} % por defecto gris claro
\SetWatermarkAngle{55} % respecto a la horizontal

\usepackage{blindtext}

\begin{document}
 \Blinddocument
\end{document}
```

Figura 2.4: Código generador de Marca de Agua en LaTex

2.1.5. Códigos QR cifrados como Marcas de Agua en Patrones de Difracción

En el SOMI[5] (congreso de instrumentación, año 2017) se trató la creación de Marcas de Agua basadas en códigos QR cifrados con patrones de difracción y cifrados mediante el algoritmo DES, dependiendo su visibilidad del patrón de difracción o el medio portador.

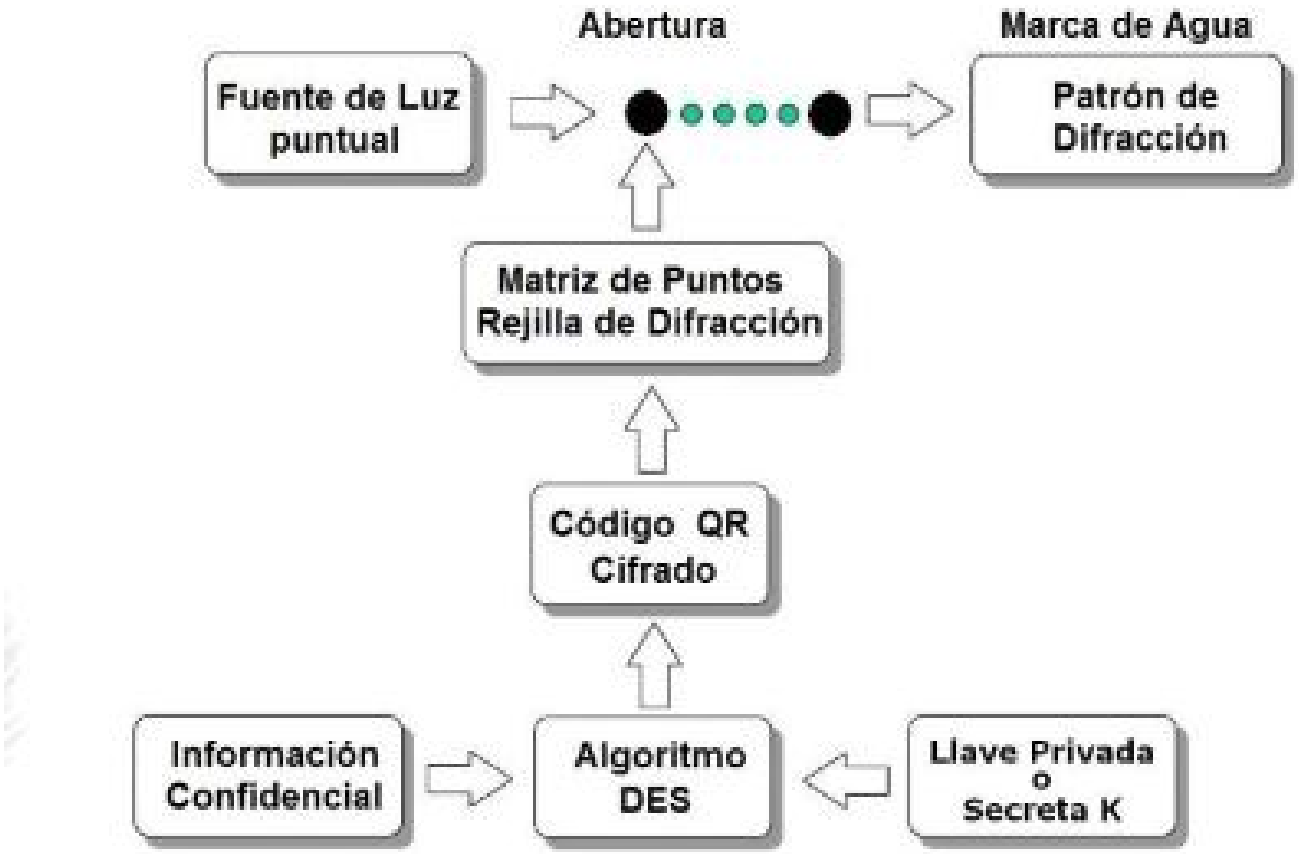

Figura 2.5: Esquema de inserción de la marca de agua para la generación de un patrón de difracción.

En el procedimiento para crear una abertura como una rejilla de difracción se han utilizado dos técnicas, una usando un modelo abertura matemático y haciendo su propagación sobre ella misma. El otro es el código QR cifrado, que es generado tras la inserción de la información cifrada con el algoritmo DES, una imagen en blanco y negro en un mapa de bits de 256 colores.

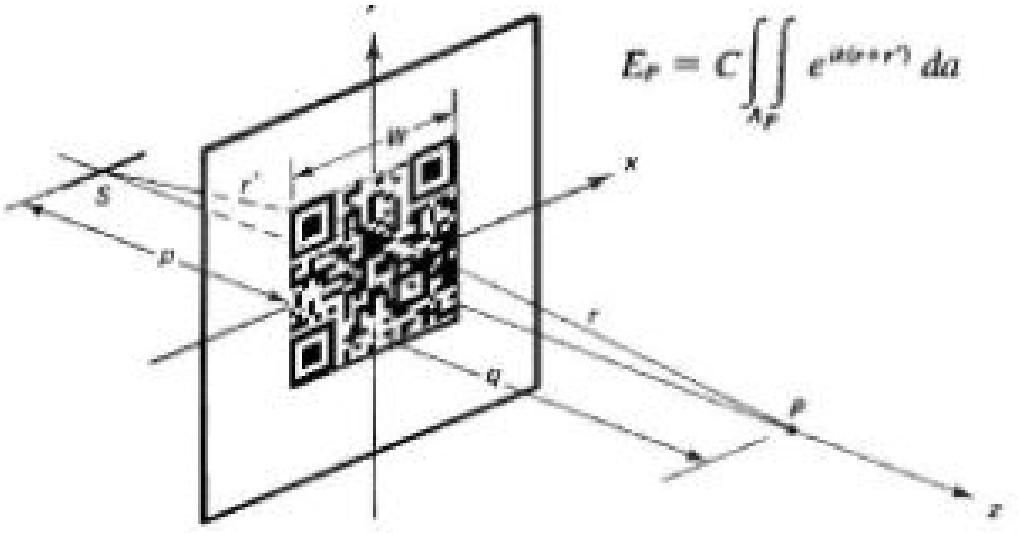

Figura 2.6: Sistema de la difracción de las aberturas para las Marcas de Agua

2.1.6. Ley 3/2014, de 27 de marzo

El día 27 de marzo del 2014 entró en vigor la modificación de Ley de protección del consumidor de contenidos digitales[6], donde se reforma una ley anticuada y de poca validez. Con esta modificación se pretendía proteger tanto a consumidores como a suministradores de contenido digital, donde especificaba con claridad la responsabilidad de ambos en los servicios demandados y adquiridos, además de sus responsabilidades dentro del marco de la protección de derechos de autor.

2.1.7. Sistemas DRM

Los sistemas DRM(Digital Rights Management [7], es decir, sistema de gestión de derechos digitales) sirven para encriptar y distribuir la información a solo aquellos usuarios autorizados por el dueño de la misma, permitiéndoles su acceso.

Compañía	Producto
Adobe/Gassbook	Adobe ebook reader
Alchemedia	Clever Content
Aries Systems	Docurights PDF Store
Content Guard	XrML
Copyright C.C.	Rightslink Republicacion lic.
Digital World Ser.	Secure online del.
Digital Goods	Softlock
Digital Owl	KineticEdge
lntertrust	Metatrust Utility
MediaDNA	Eliminator
Microsoft	MS Reader/DAS
NctLibrary	Ebooks
Reciprocal	Digital Clearing Ser.
SealedMedia	Softseal
Vyou.com	Vyoufirst
DOI	Digital object Identifier
OEB	Open eBook Forum
OPIMA	Open platform initiative
XrML	Extensible Rights Markup Language

Esto conlleva a la restricción de la libre circulación de información y ya que el sistema esta diseñado en post al beneficio económico de las grandes empresas, escudándose en los derechos de autor y provocando en última instancia un perjuicio para el consumidor.

2.1.8. Reconocimiento de marcas de agua embotellada

Este fue un proyecto de fin de master realizado en UPC(Universidad Politécnica de Cataluña)[8] en el año 2018, donde se utilizó una Red Neuronal Convolucional (Convolutional Neural Network), y se comprobó si era factible la creación de un sistema de reconocimiento de imágenes en las que aparece una marca de agua embebida. Debido a que las redes neuronales aprenden tras iteraciones, le es posible al sistema tras varias rondas de prueba y error, aprender a reconocer las etiquetas en las imágenes que hacen referencia a la marca de agua embebida que tienen inscrita.

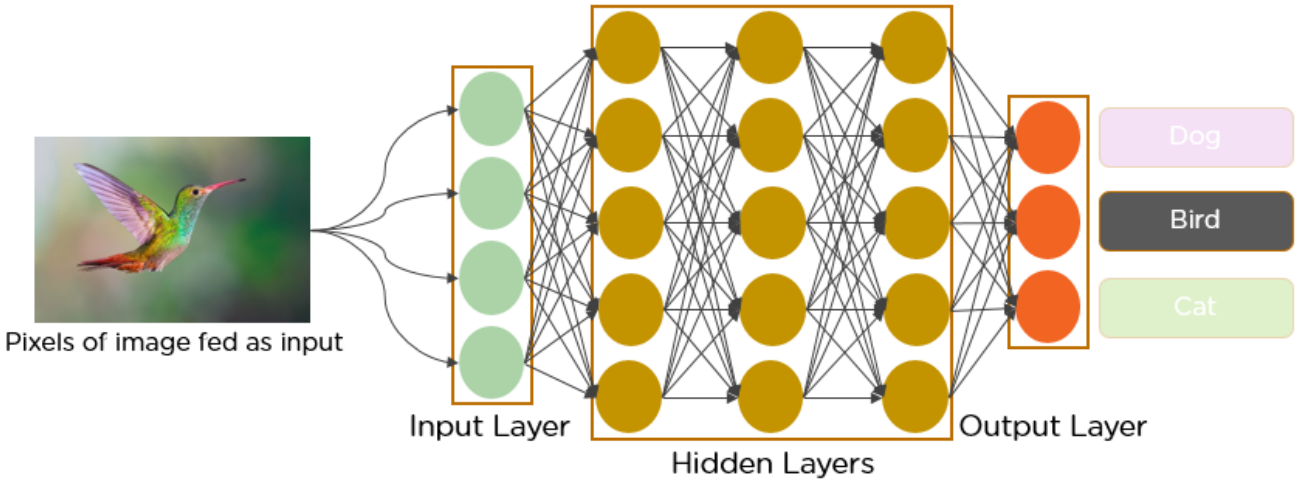

Figura 2.7: Funcionamiento de una red neuronal

2.1.9. Desarrollo de aplicaciones Serverless en entornos distribuidos

En 2021 en la UNPL(Universidad Nacional de la Plata de Argentina)[9], se realizó una conferencia sobre el desarrollo y uso de aplicaciones Serverlees en entornos distribuidos. Se analizó la evolución y uso del mismo, para aplicaciones de corta duración como microservicios, backends IoT móviles, procesamiento de flujo modesto, bots e integración de servicios y reconocimiento de datos e imágenes.

Pese a sus beneficios como la rapidez operacional, sencillez de desarrollo, bajo coste económico y la modularización, también tiene grandes desventajas como su difícil depuración, cierta perdida de control operativo, problemas con APIs a terceros y bibliotecas que no son reconocidas por el sistema, riesgos de seguridad al ser sistemas en los que el desarrollador debe registrarse e identificarse.

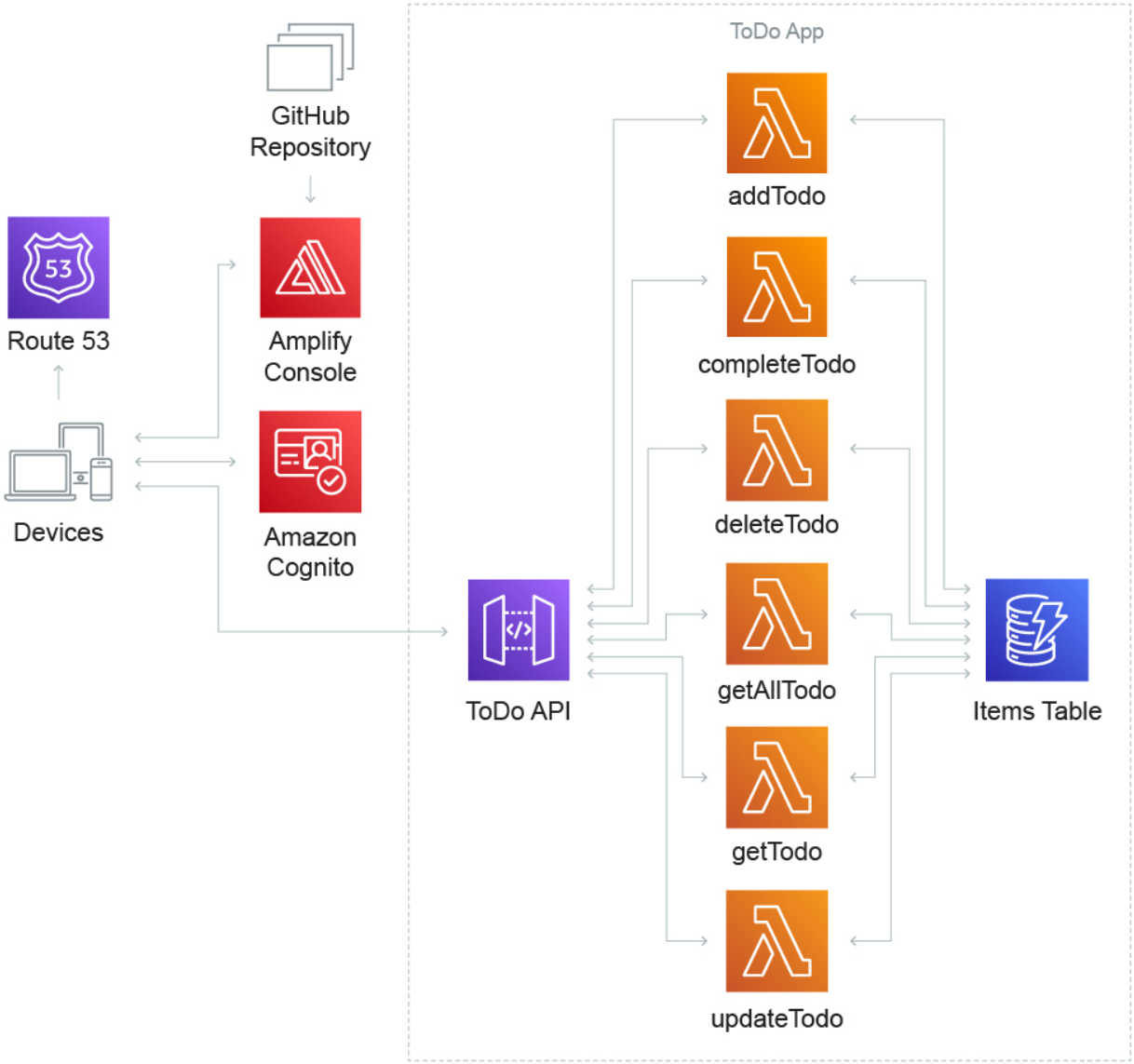

Figura 2.8: Funcionamiento de una arquitectura Serverless

2.1.10. Arquitectura serverless para el procesado de datos y detección de anomalías en el instrumento MARSIS

Este trabajo realizado por David Pacios Izquierdo y otros investigadores[10] en colaboración con la ESA(Agencia Espacial Europea). El objetivo era procesar imágenes obtenidas por la MARSIS, que es una sonda orbital de radar y altímetro de pulso limitado y baja frecuencia creado por la Universidad de Roma.

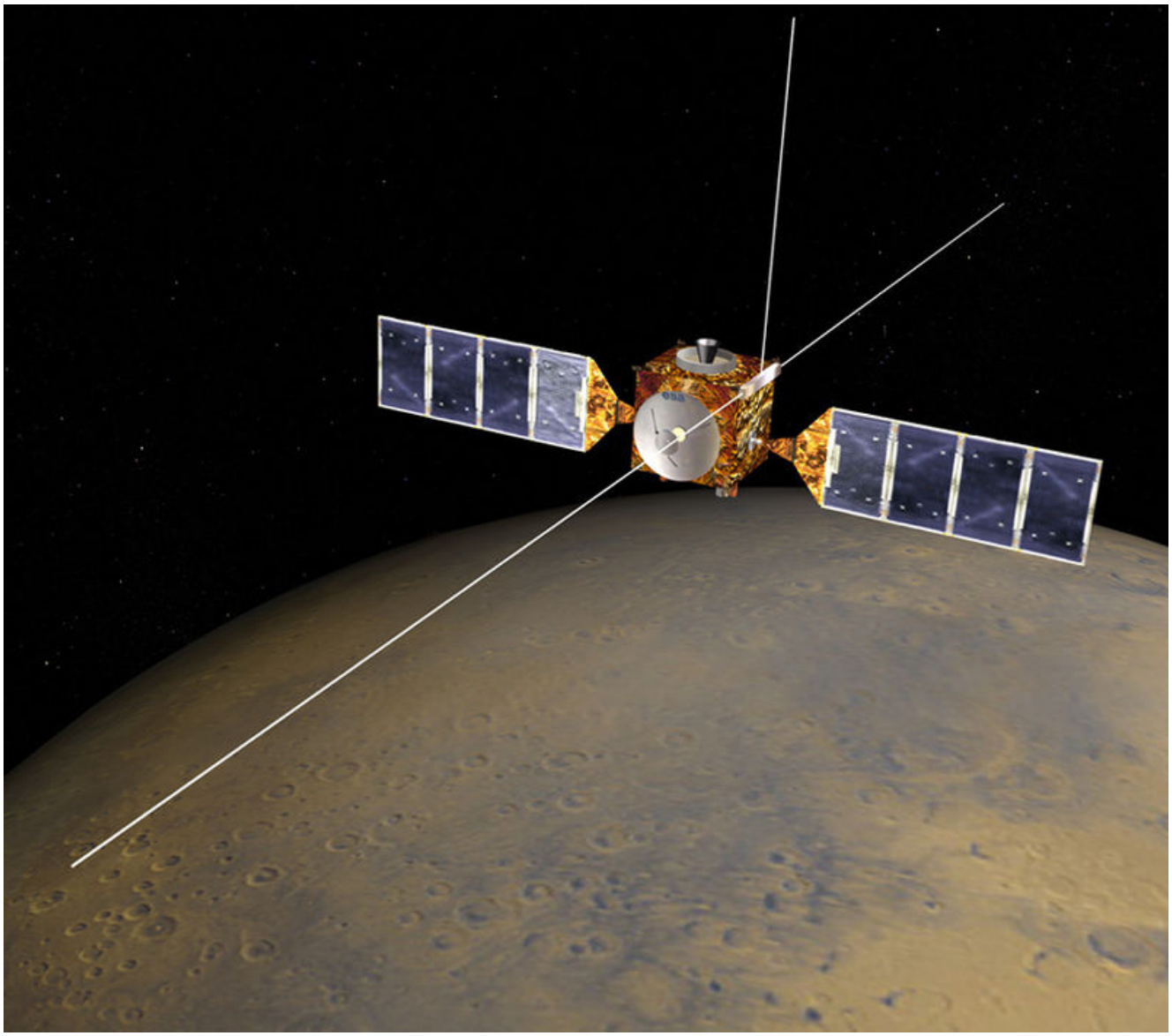

Figura 2.9: Misión ESA Mars Express

Se implementó una arquitectura serverless que analizaba la información enviada por la MARSIS en busca de anomalías. En este proceso se comprobaban un gran número de imágenes en pocos segundos. Al implementarlo se ha obtenido una ganancia de tiempo y coste para la investigación de las anomalías de la superficie marciana.

2.1.11. Computación en Serverless para el análisis de datos de secuencias de ARN

En esta investigación se ha desarrollado una solución para el análisis de datos de secuencias de ARN[11] utilizando serverless. Ha sido realizada por Pietro Cinaglia, José Luis Vázquez-Poletti y Mario Cannataro.

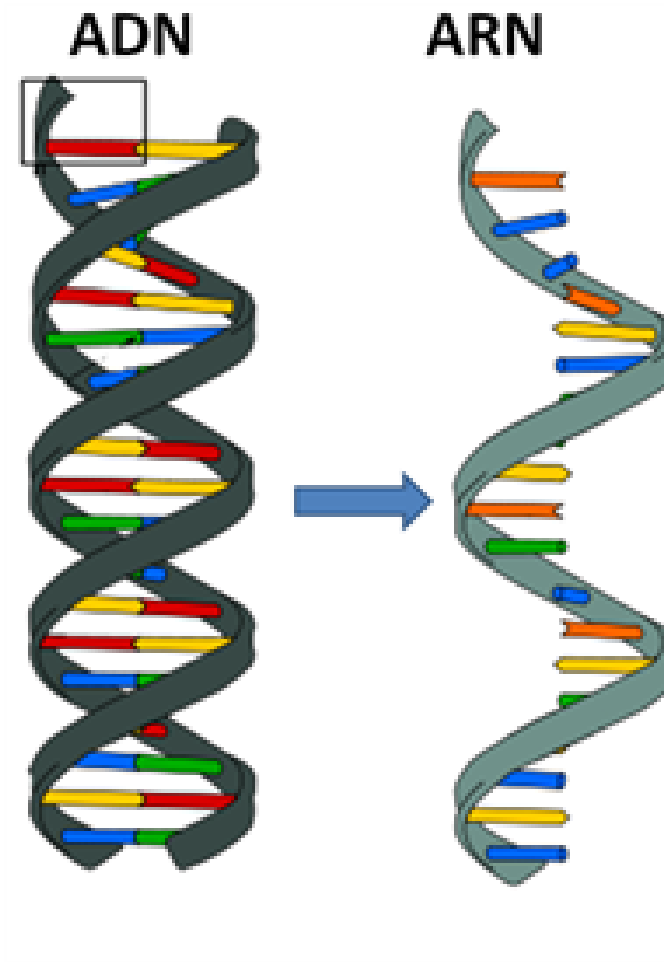

Figura 2.10: Secuencias de ADN y ARN

Esta arquitectura sin servidor desarrollada se centraba en el escaneo de las lecturas de secuenciación del genoma objetivo. Se demuestra que su solución mapea y analiza un gran número de secuencias (hasta 1000 muestras en base 16). Se compara con el entorno local y se muestra que este último requiere de un alto uso de la CPU y grandes cantidades de memoria.

2.1.12. Marca de agua inteligente aplicada al dinero electrónico

En este proyecto[12] se propone aplicar el uso de marcas de agua inteligentes al dinero electrónico, a través de un código ejecutable insertado en la marca para evitar incompatibilidades de funciones y demostrando que se puede emplear en el ámbito financiero digital.

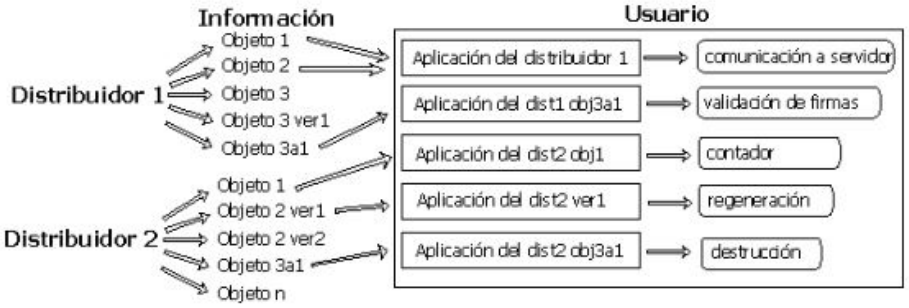

Figura 2.11: Problemas de incompatibilidad del dinero digital

Se creó un escenario donde el usuario puede manejar diferentes tipos de dinero electrónico, mediante una aplicación estándar para su uso con marcas de agua inteligentes. Además, se implementó un cajero automático para pagos, ofreciendo el servicio a aquellos usuarios que no tuvieran cuenta bancaria para poder realizarlos.

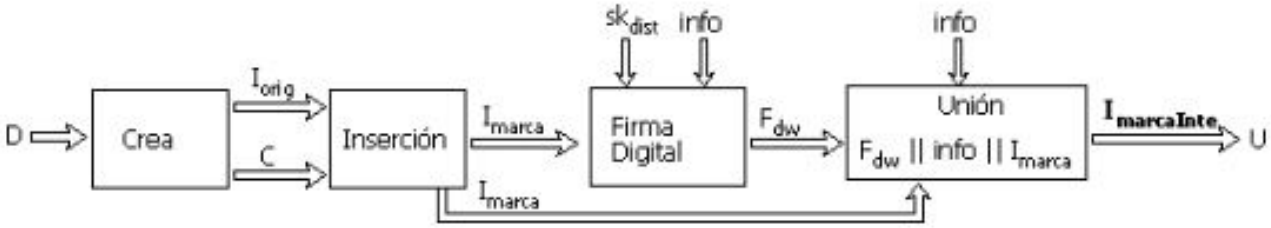

Figura 2.12: Generación de marca de agua inteligente

Capítulo 3. Tecnologías

3.1. AWS

AWS(Amazon Web service)[13] es la colección de servicios en la nube más completa y adaptada del mundo. Al contar con más de 200 servicios, es utilizado para el desarrollo de arquitecturas y software en el ámbito público y privado.

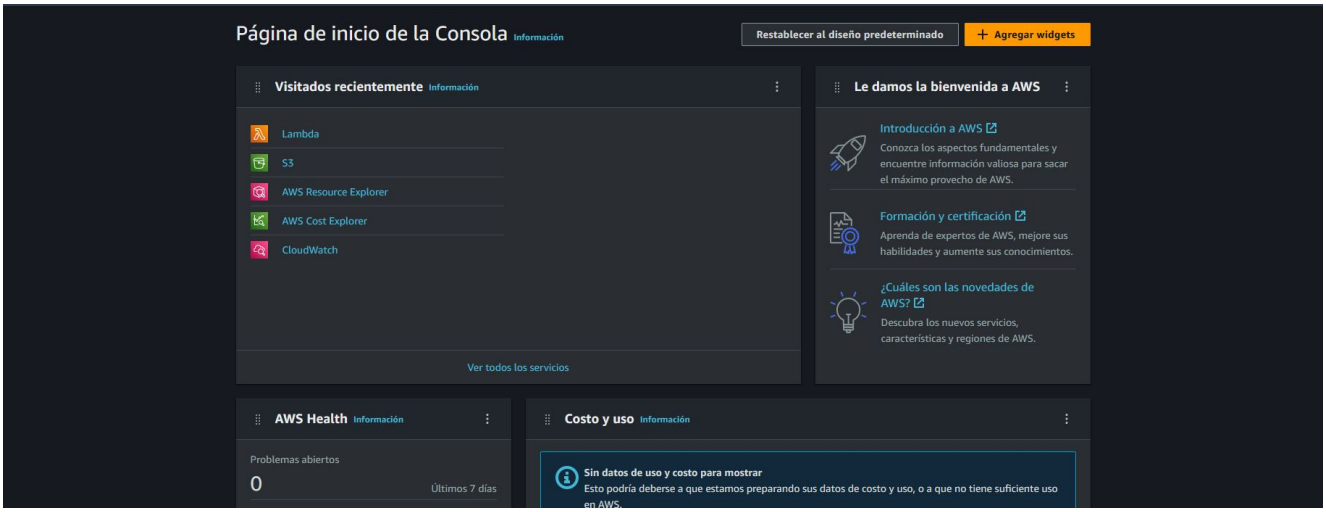

Figura 3.1: Imagen del entorno de bienvenidade AWS

Esta herramienta que requiere previo registro, nos ha servido para utilizar y mejorar el rendimiento del código, pudiendo llegar a gestionar un centenar de hilos de ejecución a bajo coste y capacidad de memoria.

3.1.1. Lambda

Lambda o Amazon Lambda[1] es un servicio sin servidor y que se basa en eventos, permitiendo ejecutar código de cualquier aplicación o backend, sin la necesidad de reservar o gestionar servidores.

[1]Dirección de lambda y de la imagen https://aws.amazon.com/es/lambda/

Figura 3.2: Funcionamiento de Lambda para procesamiento de archivos

3.1.2. Docker

Es una plataforma que permite la creación, gestión y prueba de aplicaciones o arquitecturas de manera rápida. Docker[2] es una máquina virtual que simula en el servidor un entorno de ejecución para crear y probar la aplicación o arquitectura.

3.1.3. S3

S3 o Amazon S3[3], es un servicio de almacenamiento de objetos(en nuestro caso documentos en txt y pdf). Ofreciendo escalabilidad, seguridad y disponibilidad de los datos para ser usados en las aplicaciones Lambda.

Figura 3.3: Esquema de funcionamiento de S3 de Amazon

[2]Dirección de la web https://aws.amazon.com/es/docker/
[3]Dirección de la web y de la imagen de S3 https://aws.amazon.com/es/s3/

3.2. Python 3.8

Se ha elegido este lenguaje[14] para el desarrollo de la arquitectura porque tiene una gran cantidad de librerías. Es un lenguaje potente y de fácil manejo.

3.2.1. Pillow

Librería de manejo de imágenes de Python, es utilizada para la generación de las marcas de agua.

3.2.2. PyPDF2

Librería de manejo, creación y modificación de PDFs en Python, es utilizada para añadir la marca de agua y generar el documento seguro.

3.2.3. Hashlib

Librería utilizada para la encriptación de la fecha y la hora en Sha256, formará parte de la marca de agua segura.

3.2.4. Img2pdf

Librería de Python que sirve para la conversión de imágenes en PDFs, para su posterior uso como marca de agua y fusión con PyPDf2.

3.2.5. Datetime

Librería que utiliza el tiempo de sistema, que permite obtener el día y la hora para posteriormente crear una clave de encriptación.

3.3. LaTex

LaTeX[4] es sistema de composición de textos, concebido para la creación de documentación gran calidad tipográfica.

3.3.1. Overleaf

Overleaf[4]es un entorno de trabajo on line en LaTeX, donde se ha escrito en su totalidad la documentación del TFG.

[4]Web de Overleaf https://es.overleaf.com

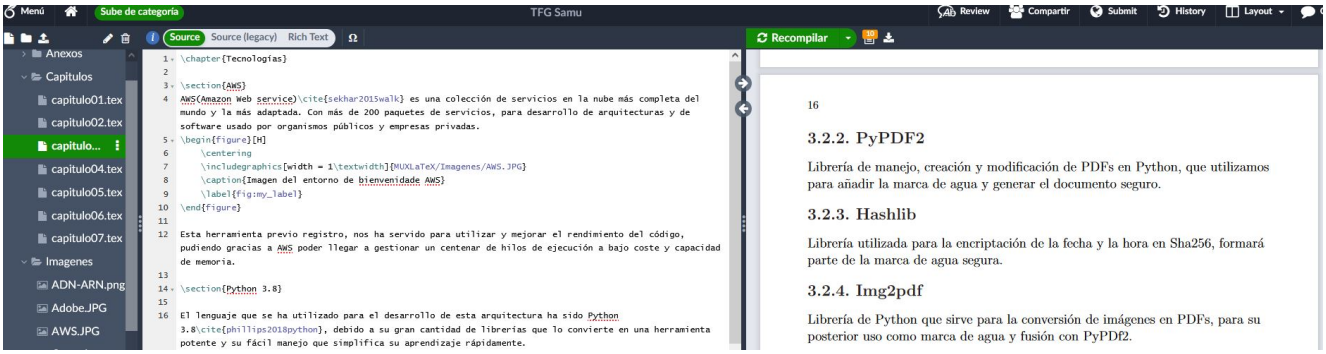

Figura 3.4: Imagen del entorno de Overleaf

3.4. API Gateway

API Gateway[5] es un gestor que interactúa con el envío y recepción de datos, aplica políticas, control de acceso y autenticación a las llamadas de una API para la protección de datos.

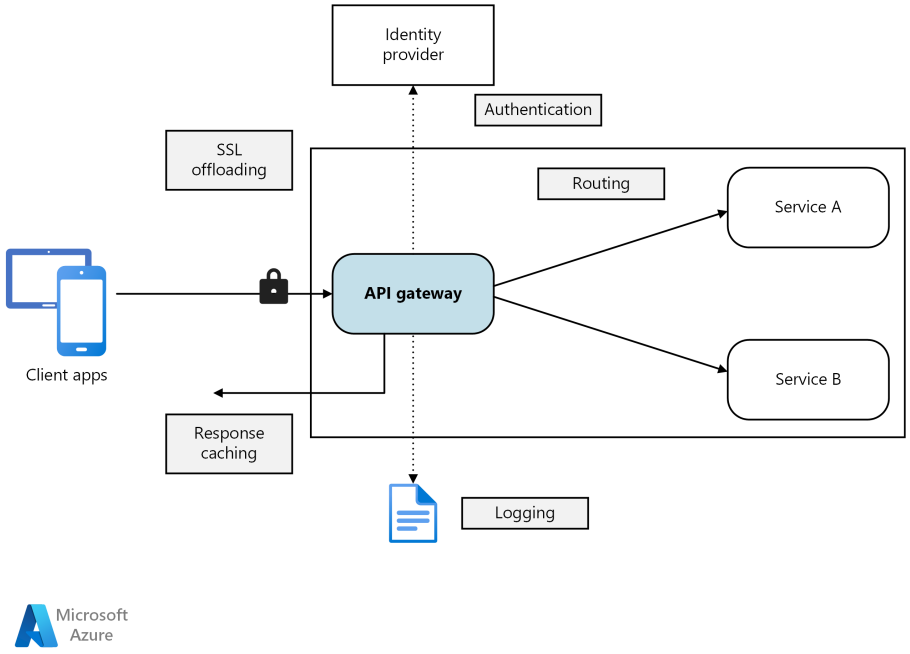

Figura 3.5: Ejemplo de funcionamiento de una API Gateway de Microsoft

[5]Dirección de información sobre API Gateway https://www.tibco.com/es/reference-center/what-is-an-api-gateway

Capítulo 4. Diseño de solución

4.1. La marca de agua

4.1.1. ¿Qué es una marca de agua?

Una marca de agua[1] es un texto o código, que se inserta en una imagen o documento con la intención de dejar constancia de la autoría o propiedad intelectual.

Es decir, es un sello o marca(logo, nombre de autor o empresa, correo electrónico...) que protege la propiedad digital de una empresa o particular.

Figura 4.1: Ejemplo de imagen protegida y sin proteger por marca de agua

4.1.2. Importancia de la marca de agua

La marca de agua va a proteger los documentos, fotografías y vídeos digitales, siendo un elemento disuasorio a la hora de intentar reproducir o utilizar dichos materiales sin consentimiento del autor y además sirve como marca de autoría de los mismos.

4.2. Problemas de la marcas de agua

4.2.1. El problema de la robustez

Uno de los mayores problemas que sufren las marcas de agua, es la robustez ante programas de edición digital. La aparición de cientos de programas que ya tienen incluidos módulos específicos para insertarlas, modificarlas o eliminarlas, debilitan la seguridad de las actuales marcas de agua.

[1]Información obtenida del articulo: https://www.creativosonline.org/marca-de-agua.html

Algunas aplicaciones utilizadas actualmente para quitar marcas de agua son las siguientes:

- Canva
- Adobe Reader Pro
- Photoshop
- Pixlr
- PDF Filler

4.2.2. Los problemas del tiempo y diseño

Uno de los grandes problemas al implementar una marca de agua es el diseño, es decir, que cumpla con los parámetros de robustez pero sin degradar el documento, evitando que sea ilegible para los usuarios a causa de la marca de agua creada.

Otro problema es el tiempo necesario al diseñar la marca, crearla y aplicarla a los documentos. Este se acentúa cuando el volumen de documentación, vídeos o imágenes es muy elevado, provocando una pérdida de tiempo, esfuerzo y dinero muy elevada para el autor o empresa dueña de dicho contenido digital.

4.3. ¿Como enfrentarse al problema?

Lo primero es ver los problemas que se generan en la creación de marcas de agua y su seguridad. Para ello se ha creado una arquitectura o programa capaz de solucionar cada uno de los inconvenientes paso a paso, hasta poder generar una marca de agua segura e imperceptible para los usuarios.

Las dificultades encontradas al crear esta solución son los siguientes:

- Inserción automática de la marca de agua en un documento.
- Creación de una marca de agua a partir de un correo electrónico.
- Transformación de la marca de agua en una marca segura.
- Modificación de los metadatos como medida de seguridad adicional
- Reducción de tiempo y costes.
- Posibilidad de marcado de grandes cantidades de documentos.

4.4. Fases de resolución

4.4.1. Fase de investigación o estudio

Durante esta fase estudió el problema y su resolución. Además de valorar la viabilidad de las distintas herramientas y lenguajes.

Se decidió usar Python, como lenguaje base por su fácil manejo y versatilidad. Esto permitió que su aprendizaje y uso fuera rápido a la hora generar arquitecturas con las aplicaciones de ámbito local Jupyter y Visual Studio Code para el futuro desarollo del proyecto.

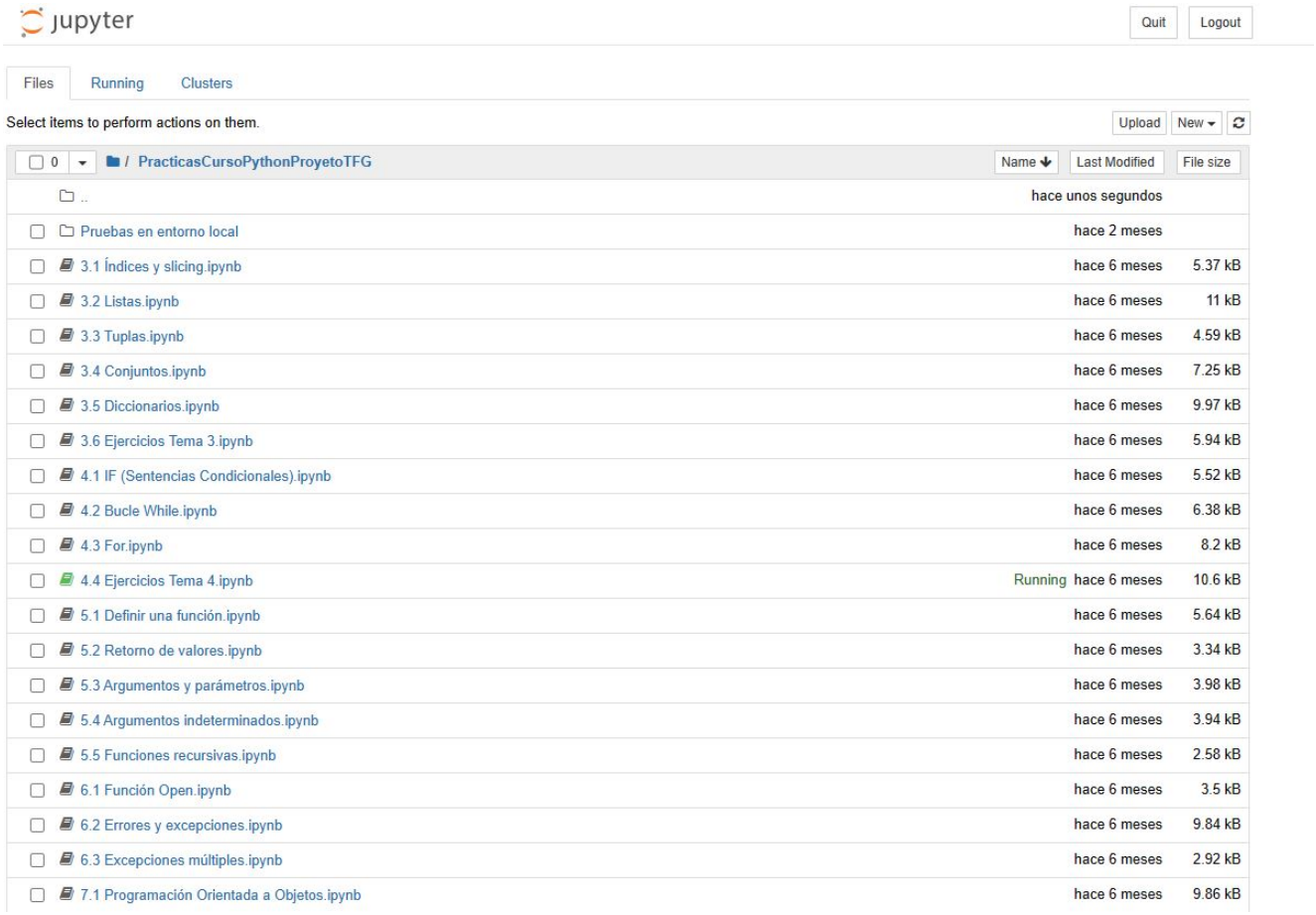

Figura 4.2: Entorno Jupyter de Anaconda

A continuación se procedió al estudio de artículos en Google Scholar[2], comprobando las posibles soluciones previas existentes y si pueden o no solucionar dichos problemas con las marcas de agua.

[2]Google Académico https://scholar.google.es/

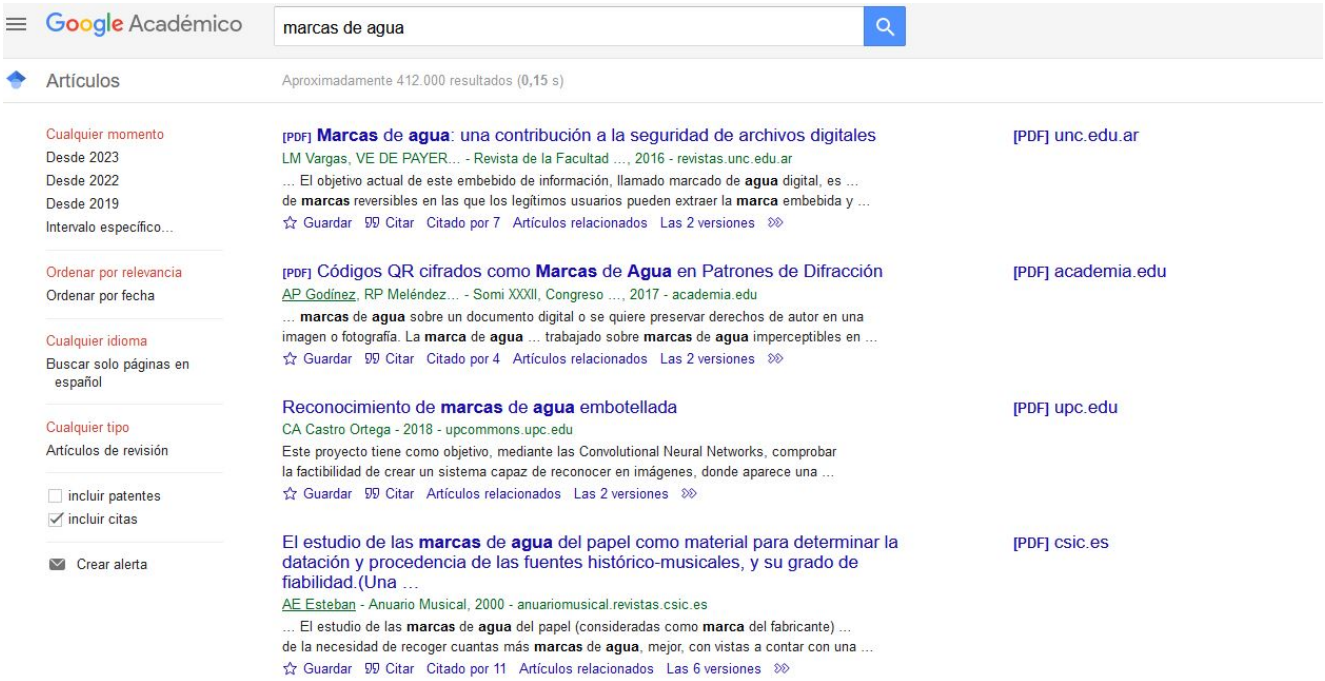

Figura 4.3: Google Scholar o Académico

4.4.2. Fase de desarrollo local

Tras haber realizado una investigación previa y haber asimilado los conocimientos necesarios para la implementación en Python de los primeros módulos, se empezó con el desarrollo y pruebas en local(Ordenador de sobremesa).

Marcado de documento

El primer paso fue generar un módulo o aplicación que al recibir una dirección de correo electrónico, la transformara en una imagen utilizable y con el fondo transparente, además de prefijar el tamaño y fuente del texto(correo electrónico). Tras la generación de dicha imagen, se genera un segundo módulo que transforma esa imagen en un documento PDF para su uso posterior como marca de agua en los documentos.

Con estos dos módulos finalizados, se soluciona el problema de crear una marca de agua de forma fácil y rápida, para poder ser utilizada en el tercer módulo. Este último módulo se encarga de agregar la marca de agua a las hojas de un PDF en la posición prefijada que se prefiera.

Al finalizar esta parte de la fase de desarrollo local, se consiguió solucionar el problema de generación de una marca de agua a partir de un texto dado (en este caso un correo electrónico) y de adición en todas las hojas del documento en

la posición previamente determinada.

Seguridad del Documento

Anteriormente se ha marcado el documento. La marca de agua generada es débil y poco robusta, por lo que podía ser eliminada por aplicaciones para eliminación de marcado de documentos e imágenes.

Es por esto que se decidió fortalecer la marca de agua generada y además obtener el texto o correo electrónico, que se usará para la marca de agua del documento y así evitar errores de escritura por parte del autor.

El siguiente módulo creado es el encargado de leer un texto y extraer el correo que se utiliza al generar la marca de agua. A partir de este módulo, se añadió la funcionalidad de obtener el día y hora de lectura del texto y encriptarlo (codificarlo de forma segura), para luego concatenarlo al correo y generar la marca de agua segura, además de un documento de texto con la clave de encriptación para el autor.

Tras tener la marca de agua segura, se procedió a generar un módulo adicional para la modificación de los metadatos del nuevo documento marcado, para añadir un nivel de seguridad extra al documento. También se modificó la marca de agua para ser imperceptible ante cualquiera que quisiera eliminarla, sin ser el propio autor.

Al finalizar las pruebas, se obtuvo único programa modular que consiguió solventar el problema de la seguridad, la automatización de la creación de la marca de agua y el marcado en el documento.

4.4.3. Fase de Desarrollo en el Servicio Web

En este punto, se tiene una aplicación funcional capaz de ejecutarse en cualquier ordenador, el problema que aún no se había solucionado era el coste en tiempo, dinero y la cantidad de documentos posibles a procesar.

En ese momento, se decidió utilizar Amazon Web Service(AWS)[3] como servicio web por su capacidad de procesar una gran cantidad de datos e hilos de ejecución(Programas que puede procesar) y el bajo coste de su uso.

Tras un estudio y aprendizaje intensivo de dos semanas y media sobre Amazon Web Service, registro, familiarización con la plataforma, pruebas de uso y análisis de costes, se procedió a la migración del código en Python, añadiendo

[3]Dirección web:`https://aws.amazon.com/es/what-is-aws/`

las instrucciones Lambda de Amazon para su correcto funcionamiento.

Se decidió ir probando los diferentes módulos en Amazon Web Service y finalmente, se generó una versión totalmente operativa que superaba las expectativas en tiempo, coste y capacidad de ejecución, Se generó todo lo esperado de la arquitectura tras solucionar varios problemas de integración con Python.

Capítulo 5. Arquitectura e implementación

5.1. Arquitectura de Proyecto en Amazon Web Service

En este apartado se va a mostrar y explicar la arquitectura desarrollada en Amazon Web Service. Se mostrará su funcionamiento paso a paso.

Primero se mostrará el esquema de la arquitectura y luego se desglosará de la siguiente forma:

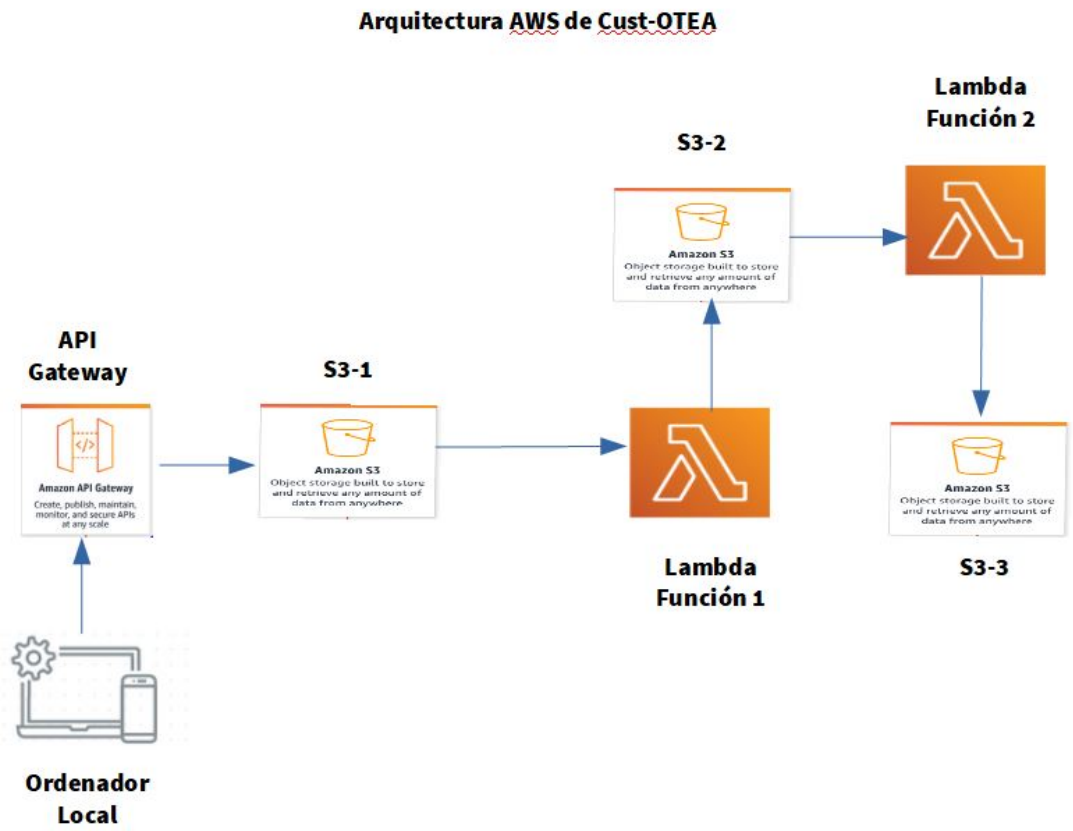

Figura 5.1: Arquitectura del proyecto en Amazon Web Service

5.1.1. Dockerización del proyecto

Durante el desarrollo se generaron problemas debido a las librerías asociadas a las dos funciones Lambda. Para solucionarlos se empaquetó el proyecto en un Docker (librerías incluidas) y se subió a un ordenador local mediante un API Gateway. De esta forma se solucionaron los problemas de incongruencia y pérdida de librerías (asociadas con el tratamiento de imágenes) al ejecutar estas funciones.

Esto ha provocado una penalización en el tiempo de ejecución total de la arquitectura, ya que se tiene que desempaquetar y utilizar los paquetes en vez de tenerlos ya instalados por defecto en los Layers de cada una de las dos funciones Lambda, pero no supone un gran incremento en el coste total de tiempo y dinero de la arquitectura.

5.1.2. API Gateway

Una vez dockerizado el proyecto se subirá a la API Gateway previamente creada y concediéndole una política de permisos de invocación que se basa en los recursos de función. Pudiendo enviar la información o las APIs(Librerías necesarias para el buen funcionamiento del código) a las Funciones Lambda 1 y 2 o enviando los datos de entrada como el documento a marcar y el documento que contiene el correo electrónico al S3-1.

En nuestro caso, la API Gateway invoca una función síncronamente que contiene una solicitud JSON de evento y queda a la espera de respuesta por cualquiera de los módulos que necesiten las APIs o la información que contiene. S3-1 mandaría una respuesta a API Gateway solicitando los documentos de entrada, que sería el PDF a marcar y el documento de texto que contiene el correo electrónico. La API Gateway responde a S3-1 y envía los documentos a S3-1, para su uso en el lambda función 1 a la vez que este también solicitará las APIs o librerías a API Gateway para la correcta ejecución del código que contienen.

5.1.3. S3

En la arquitectura se usaron tres contenedores S3, que servirán como almacenamiento temporal de los datos de entrada, salida e intermedios que necesitaran utilizar nuestras dos funciones Lambda y sobre los que trabajaran en su ejecución.

El S3-1 contendrá los documentos de entrada que utilizará la Lambda Función 1 y que han sido requeridos a la API Gateway mediante la función de evento-respuesta del mismo, como se ha explicado en el punto anterior. La S3-2 que

contendrá el resultado de la ejecución de Lambda Función 1 y lo almacenará, esperando el requerimiento de los mismos para su utilización por Lambda Función 2. El S3-3 que recibirá los datos finales de toda la ejecución de la arquitectura, devolviendo el documento pdf marcado y el documento de texto con la clave-valor.

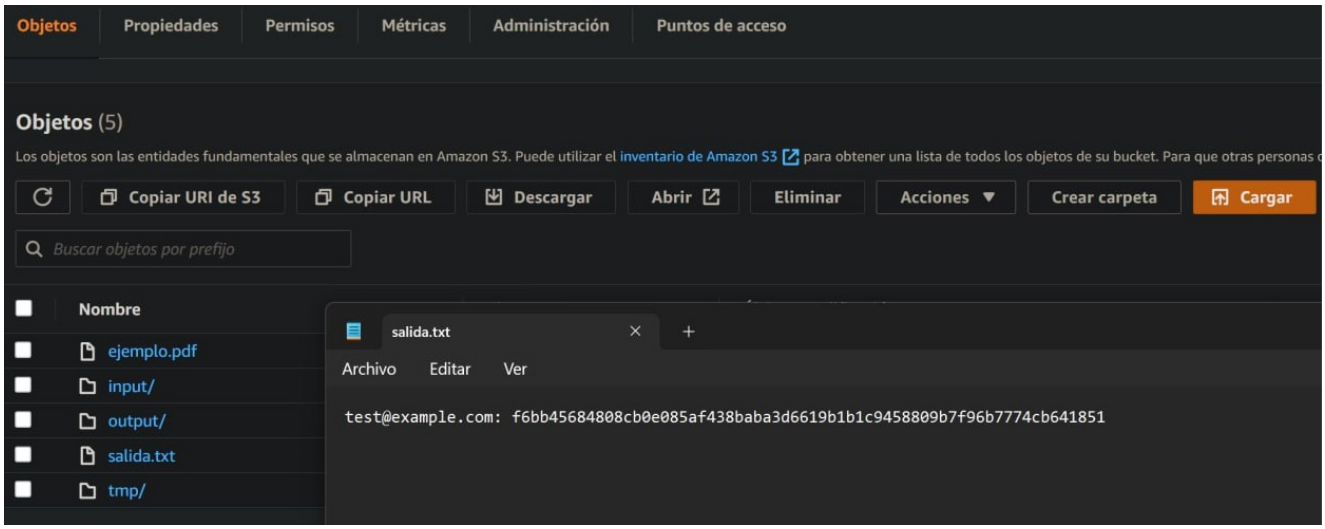

Figura 5.2: S3-3 con los archivos de salida al final de la ejecución de la arquitectura

5.1.4. Lambda Función 1

Esta función Lambda se activa al recibir el documento que hay que marcar por parte de S3-1. Una vez se activa, empieza a ejecutar y a descargar los archivos necesarios. Luego lee el archivo de texto para obtener el correo electrónico para fabricar la marca de agua y lo guardará como una variable de texto.

A continuación se obtendrá una cadena de texto con la fecha y la hora actual del sistema, creando un diccionario clave-valor con el correo electrónico y la variable de texto que contiene la cadena de fecha y hora con hash codificado en formato sha-256, para finalmente concatenarlo con el correo electrónico para ser usado por la marca de agua.

Se generará la imagen que servirá como base para la marca de agua y se agregará como texto la concatenación anterior de correo+diccionario y se volverá transparente, para luego finalmente transformarla en un PDF de una sola hoja que se usará como marca de agua.

El siguiente paso sería proceder a extraer las hojas del documento a marcar y se fusionan con la marca de agua, generando un nuevo documento PDF con la marca de agua invisible en todas sus hojas y totalmente seguro. Al finalizar la función lambda, se enviará a S3-2 el documento de texto con el diccionario y el

nuevo PDF marcado de forma segura para su utilización por parte de Lambda Función 2.

5.1.5. Lambda Función 2

La función Lambda recibe el evento de ejecución y extrae los documentos de S3-2, tomando el documento PDF seguro que se ha creado para la modificación de los campos que se quieran de sus metadatos como segundo método de asegurar el documento y su autoría frente a posibles intentos de manipulación.

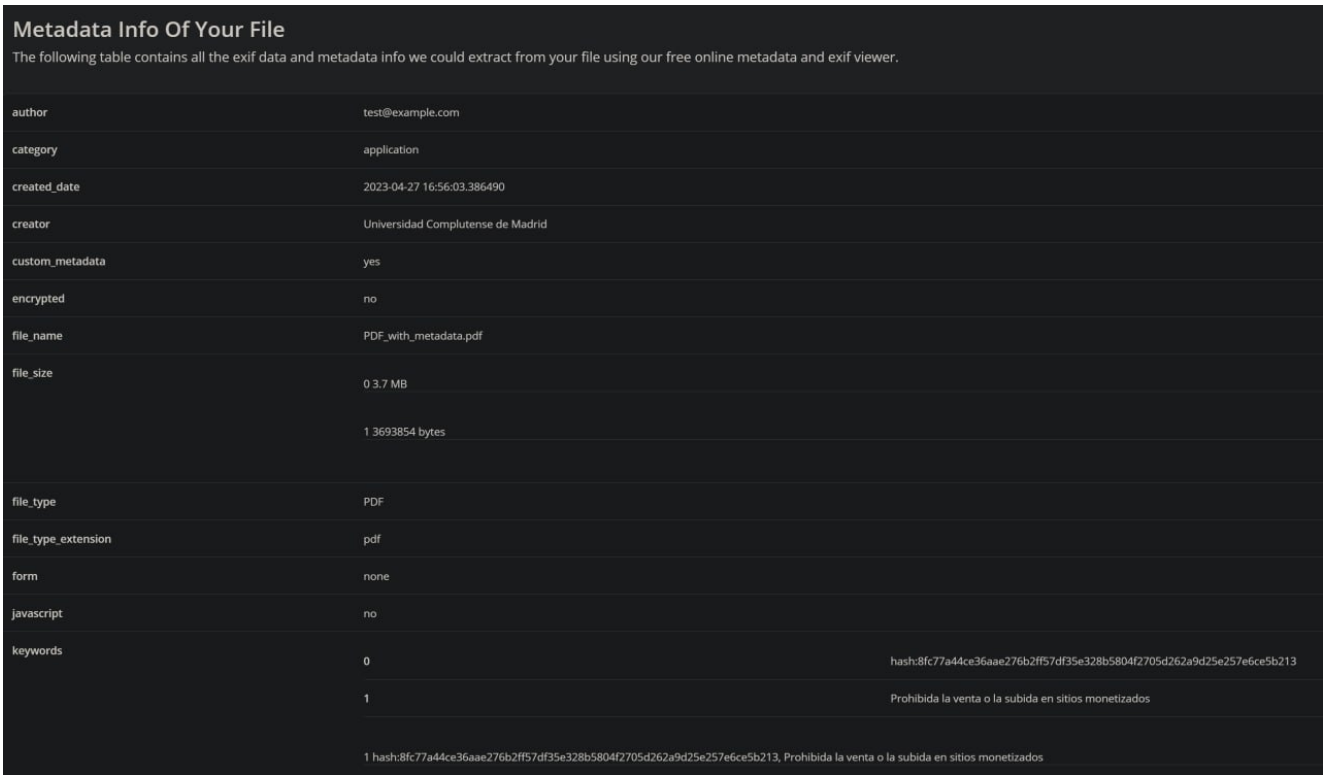

Figura 5.3: Metadatos modificados del documento marcado de forma segura

Al finalizar el proceso de modificación de los metadatos, se envía a S3-3 el documento ya marcado y con los metadatos modificados, además del archivo de texto con el correo y el diccionario cifrado listos para descargar al ordenador local.

Capítulo 6. Mediciones y resultados

6.1. Pruebas de costes y tiempos

Se ha procedido a realizar las pruebas de costes y de tiempo en diferentes entornos, para hacer una comparativa de cual sería opción más viable para una empresa o institución. Los entornos de prueba son los siguientes:

- Ordenador Local.
- Google Cloud.
- Amazon Web Service.

Los cálculos han sido realizados utilizando la calculadora de costes de Google cloud y AWS Lambda Cost Caculator, en comparativa con el coste total de un ordenador local estándar de 1000€ y cuanto tiempo tardaría en amortizar las operaciones.

6.1.1. Google Cloud vs Ordenador local

Los cálculos sobre un total de 1000 operaciones son los siguientes:

Parámetros Comparativos	**Google Cloud**	**Ordenador Local**
Número de iteracciones	1000	1000
Espacio de almacenamiento	2GB	500GB
Tiempo en realizar una iteracción	4.8seg	4.8seg
Tamaño de la memoria RAM	1GB	16GB
Coste total/mes	0.61USD/mes	83.33€ (Amortización: 1000/12)

Cuadro 6.1: Comparativa entre Google Cloud y el ordenador local

En el Cuadro 6.1 se puede observar que Google Cloud tiene un menor coste mensual para analizar 1000 documentos con una capacidad menor y mismo tiempo de ejecución. Para ejecutar el mismo número de documentos en local se necesitarían 136 años.

6.1.2. AWS vs Ordenador local

Los cálculos sobre un total de 1000 operaciones son los siguientes:

Parámetros Comparativos	**AWS**	**Ordenador Local**
Número de iteracciones	1000	1000
Espacio de almacenamiento	512MB	500GB
Tiempo en realizar una iteracción	4.8seg	4.8seg
Tamaño de la memoria RAM	128MB	16GB
Coste total/mes	0.01USD	83.33€ (Amortización: 1000/12)

Cuadro 6.2: Comparativa entre AWS y el ordenador local

Como se observa en el Cuadro 6.2 con una capacidad de almacenamiento menor y RAM, su coste es ínfimo al mes por la paralelización. Un ordenador local tardaría 69444 años en procesar estos 1000 documentos.

Capítulo 7. Conclusiones y trabajo a futuro

7.1. Conclusiones

Este proyecto tenía como objetivo crear una marca de agua totalmente segura y fortalecer los documentos contra su eliminación, además de proteger los derechos del autor y su uso fraudulento por parte de terceras personas o empresas sin consentimiento explícito. Para ello se han revisado las distintas soluciones creadas en el apartado del estado del arte y se ha buscado una solución efectiva para proteger documentos mediante marcas de agua.

La metodología utilizada para el desarrollo de esta arquitectura ha sido desplegada por fases progresivas, ampliando las funcionalidades de la misma de forma paulatina. Para la implementación de la arquitectura, se eligió como lenguaje de desarrollo Python en su versión 3.8 para ámbito local y AWS de Amazon para generar la versión final, por su potencia y capacidad de manejos de altas cantidades de datos en tiempos rápidos y a bajo coste.

El desarrollo empezó con la creación de los primeros módulos, donde a partir de un correo electrónico se generaba una marca de agua y marcaba documentos de una forma rápida. En los siguientes pasos se consiguió crear un diccionario de datos clave-valor encriptado y concatenado al correo electrónico, generando con ellos una marca de agua que se volvió segura haciéndola invisible y finalmente, se modificaron los metadatos necesarios del documento previamente creado.

Finalmente, tras el éxito en la versión local, se empezó con la migración o implementación en AWS. Al hacer las primeras pruebas sobre el entorno de Amazon, se generaron problemas en los Layers de los módulos o funciones Lambda al intentar acceder a las bibliotecas de Python previamente definidas, haciendo que el sistema fallara. Tras varias pruebas, se tomó la decisión de englobar todas las librerías y los módulos en un Docker, que se subió a una API Gateway como gestor de la librería para los diferentes módulos de AWS.

Al utilizar este método, la arquitectura funcionó de forma correcta, pasando las pruebas de forma correcta y siendo presentado a los responsables del proyecto para su validación.

7.2. Trabajo a futuro

7.2.1. Envíos por e-mail

Debido a que se facilita un correo electrónico al empezar el proceso, en un futuro se podría implementar un módulo adicional que envíe los documentos marcados y el documento con el diccionario clave-valor al usuario, recibiéndolos en su correo personal gracias a las librería email de Python generando un nuevo modulo Lambda adicional.

7.2.2. Uso a través de Moodle

Gracias a API Gateway se puede incorporar la posibilidad a la solución diseñada a la plataforma Moodle, facilitando la subida de la documentación ya marcada al cuerpo docente de cualquier facultad que utiliza esta plataforma para gestionar sus campus virtuales y donde es subida la documentación para el uso por parte del alumnado.

7.2.3. Documentos protegidos con contraseña

A través de una nueva función Lambda adicional se puede generar una contraseña para el documento marcado, generando un nuevo nivel más de seguridad a los documentos y evitando que no puedan ser directamente leídos por usuarios que no tengas la contraseña generada y evitando así poder ser reproducido sin consentimiento.

Chapter 7. Conclusions and future work

7.1. Conclusions

The aim of this project was to create a fully secure watermark and to strengthen documents against deletion, as well as to protect the author's rights and their fraudulent use by third parties or companies without explicit consent. For this purpose, the different solutions created in the state of the art section were reviewed and an effective solution was sought to protect documents by means of watermarking.

The methodology used for the development of this architecture has been deployed in progressive phases, gradually extending its functionalities. For the implementation of the architecture, Python in its 3.8 version was chosen as the development language for the local environment and Amazon's AWS to generate the final version, due to its power and capacity to handle large amounts of data quickly and at low cost.

The development started with the creation of the first modules, where a watermark was generated from an email and documents were marked in a fast way. In the next steps, an encrypted key-value data dictionary was created and concatenated to the email, generating a watermark that was made secure by making it invisible and finally, the necessary metadata of the previously created document was modified.

Finally, after the success of the local version, the migration or implementation on AWS was started. When doing the first tests on the Amazon environment, problems were generated in the module Layers or Lambda functions when trying to access the previously defined Python libraries, causing the system to fail. After several tests, the decision was made to encompass all the libraries and modules in a Docker, which was uploaded to an API Gateway as a library manager for the different AWS modules.

By using this method, the architecture worked correctly, passed the tests correctly and was presented to the project managers for validation.

7.2. Future work

7.2.1. Shipments by email

Since an email is provided at the beginning of the process, in the future an additional module could be implemented to send the marked documents and the document with the key-value dictionary to the user, receiving them in his personal email thanks to the Python email library generating a new additional Lambda module.

7.2.2. Use via Moodle

Thanks to API Gateway it is possible to incorporate the possibility to the solution designed to the Moodle platform, facilitating the upload of the documentation already marked to the faculty of any faculty that uses this platform to manage their virtual campuses and where the documentation is uploaded for use by students.

7.2.3. Password protected documents

Through a new additional Lambda function it is possible to generate a password for the marked document, generating a new level of security to the documents and avoiding that they cannot be directly read by users who do not have the generated password and thus avoiding that they can be reproduced without consent.

Bibliografía

[1] A. P. Gehred, "Canva," *Journal of the Medical Library Association: JMLA*, vol. 108, no. 2, p. 338, 2020.

[2] N. de Acrobat Reader, "Ayuda de adobe acrobat reader 5.1."

[3] H. J. Vera Sánchez, "El uso del programa libreoffice writer como herramienta pedagógica en el desarrollo del aprendizaje colaborativo." B.S. thesis, Universidad de Guayaquil, Facultad de Filosofía, Letras y Ciencias de la . . . , 2018.

[4] R. D. C. Korgi, *El universo LATEX*. Univ. Nacional de Colombia, 2003.

[5] A. P. Godínez, R. P. Meléndez, and C. G. Treviño-Palacios, "Códigos qr cifrados como marcas de agua en patrones de difracción," in *Somi XXXII, Congreso De Instrumentacion*, 2017.

[6] G. de España, "Ley 3/2014, de 27 de marzo, por la que se modifica el texto refundido de la ley general para la defensa de los consumidores y usuarios y otras leyes complementarias, aprobado por el real decreto legislativo 1/2007, de 16 de noviembre," *Boletin Oficial del Estado (BOE): Madrid, España*, 2014.

[7] L. Ramos-Simón *et al.*, "Drm: Protección versus accesibilidad de la información digital," *Hipertext. net*, no. 2, 2004.

[8] C. A. Castro Ortega, "Reconocimiento de marcas de agua embotellada," B.S. thesis, Universitat Politècnica de Catalunya, 2018.

[9] N. R. Rodríguez, M. A. Murazzo, D. Medel, D. Arias Figueroa, L. Parra, A. L. Molina, F. Sánchez, A. E. Martín, H. Atencio, and M. Gómez, "Optimización en el desarrollo de aplicaciones serverless en entornos distribuidos," in *XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja)*, 2021.

[10] D. Pacios Izquierdo, "Arquitectura serverless para el procesado de datos y detección de anomalías en el instrumento marsis," 2022.

[11] P. Cinaglia, J. L. Vázquez-Poletti, and M. Cannataro, "Serverless computing for rna-seq data analysis," in *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2022, pp. 2175–2181.

[12] P. Jaimes, G. Hermosillo, and G. Roberto, "Una marca de agua inteligente aplicada al dinero electrónico," in *5th Ibero-American Congress on Information Security, CIBSI 09*, 2009, pp. 225–239.

[13] A. C. Sekhar and R. P. Sam, "A walk through of aws (amazon web services)," *International Research Journal of Engineering and Technology IRJET*, 2015.

[14] D. Phillips, *Python 3 object-oriented programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8.* Packt Publishing Ltd, 2018.