Trabajo Fin de Máster
Máster en Ingeniería de Telecomunicación

# Entrenamiento y despliegue de un modelo de clasificación de audio

Autor: Antonio José Aragón Molina

Tutor: María del Mar Elena Pérez

**Dpto. Ingeniería Electrónica**
**Escuela Técnica Superior de Ingeniería**
**Universidad de Sevilla**

Sevilla, 5 de noviembre de 2023

Departamento de Teoría de la Señal y Comunicaciones

Trabajo Fin de Máster
Máster en Ingeniería de Telecomunicación

# Entrenamiento y despliegue de un modelo de clasificación de audio


Autor:
Antonio José Aragón Molina

Tutor:
María del Mar Elena Pérez
Profesor Titular

Dpto. Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla



Sevilla, 5 de noviembre de 2023

Trabajo Fin de Máster: Entrenamiento y despliegue de un modelo de clasificación de audio

Autor: Antonio José Aragón Molina
Tutor: María del Mar Elena Pérez

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

# Agradecimientos

El diseño de una hoja de estilo en LaTeX para un texto no es en absoluto trivial. Por un lado hay que conocer bien los usos, costumbres y reglas que se emplean a la hora de establecer márgenes, tipos de letras, tamaños de las mismas, títulos, estilos de tablas, y un sinfín de otros aspectos. Por otro, la programación en LaTeX de esta hoja de estilo es muy tediosa, incluida la selección de los mejores paquetes para ello. La hoja de estilo adoptada por nuestra Escuela y utilizada en este texto es una versión de la que el profesor Payán realizó para un libro que desde hace tiempo viene escribiendo para su asignatura. Además, el prof. Payán ha participado de forma decisiva en la adaptación de dicha plantilla a los tres tipos de documentos que se han tenido en cuenta: libro, tesis y proyectos final de carrera, grado o máster. Y también en la redacción de este texto, que sirve de manual para la utilización de estos estilos. Por todo ello, y por hacerlo de forma totalmente desinteresada, la Escuela le está enormemente agradecida.

A esta hoja de estilos se le incluyó unos nuevos diseños de portada. El diseño gráfico de las portadas para proyectos fin de grado, carrera y máster, está basado en el que el prof. Fernando García García, de la Facultad de Bellas Artes de nuestra Universidad, hiciera para los libros, o tesis, de la sección de publicación de nuestra Escuela. Nuestra Escuela le agradece que pusiera su arte y su trabajo, de forma gratuita, a nuestra disposición.

*Juan José Murillo Fuentes*
*Subdirección de Comunicaciones y Recursos Comunes*

*Sevilla, 2013*

# Resumen

En nuestra Escuela se producen un número considerable de documentos, tantos docentes como investigadores. Nuestros alumnos también contribuyen a esta producción a través de sus trabajos de fin de grado, máster y tesis. El objetivo de este material es facilitar la edición de todos estos documentos y a la vez fomentar nuestra imagen corporativa, facilitando la visibilidad y el reconocimiento de nuestro Centro.

# Abstract

In our school there are a considerable number of documents, many teachers and researchers. Our students also contribute to this production through its work in order of degree, master's theses. The aim of this material is easier to edit these documents at the same time promote our corporate image, providing visibility and recognition of our Center.

*... -translation by google-*

# Índice Abreviado

# Índice

# 1 Introducción

*I am going into an unknown future, but I'm still all here, and still while there's life, there's hope.*

John Lennon, '70s

Durante los últimos años, la sociedad ha experimentado un auge en el empleo de la Inteligencia Artificial (IA) en diferentes ámbitos. La gran capacidad de escpecialización en un problema que concreto que presentan estos sistemas, junto con la gran cantidad de datos que se generan en la actualidad, han hecho que la IA se haya convertido en una herramienta muy útil en la resolución de problemas complejos.

Estas características han fomentado el empleo de soluciones basadas en IA, obtiendo resultados aceptables para problemas que de otro modo serían irresolubles.

De forma paralela, se observa un crecimiento en el interés por el uso de estas tecnologías. Muchos sistemas empiezan a aparecer en el día a día de usuario promedio, como por ejemplo: [1]

- Sistemas de búsqueda y recomendación
- IA generativa de texto, imágenes o audio
- Sistemas de predicción de eventos
- Asistentes de voz
- Vehículos autónomos
- Compras personalizadas

Sin embargo, la investigación sobre la aceptación de tecnologías que incluyen IA está aún en curso. Algunos estudios sugieren que en ciertos escenarios culturales, la necesidad de contacto humano no puede ser replicada o reemplazada. [2]

Este trabajo busca adentrarse en los límities sobre qué puede hacer un modelo de IA, en concreto en el campo de classificación de audios. El objetivo es crear un modelo que sea capaz de clasificar audios en diferentes categorías, identificadas por emociones humanas.

Una vez creado el modelo, se aborda el problema de despliegue del modelo en un entorno de producción, para que pueda ser utilizado por usuarios finales. En un caso real de uso, podría diseñarse una interfaz o una API por ejemplo, de modo que la interacción con el modelo se adapte a las requerimientos del proyecto.

Debido a esta clara diferenciación, la memoria ha sido dividia en dos partes, que darán nombre a los capítulos principales: **Modelado** y **Despliegue**. En el capítulo 2 se aborda el problema de modelado, mientras que en el capítulo 3 se aborda el problema de despliegue.

## 1.1 Motivación

La idea de realizar un modelo capaz de diferenciar emociones humanas a partir de audios surge de la participación en un proyecto anterior. Este proyecto, a grandes rasgos, busaba crear un modelo que sirviese de ayuda a personas a detectar emociones muy concretas para el caso de aplicación.

Por diversas circunstancias, el proyecto no pudo avanzar correctamente y acabó siendo abandonado. Sin embargo, un año más tarde y tras pensar en abordar de nuevo el problema, con un enfoque mucho más abierto, las sensaciones fueron muy positivas

En primer lugar, el avance de la tecnología a lo largo del tiempo ha permitido resolver problemas que antes parecían imposibles. En concreto, ha sido notable la diferencia en cuanto a todo lo relacionado con el modelo, dataset, plataforma de desarrollo, hosting, etc.

En segundo lugar, durante este tiempo, gracias al aprendizaje adquirido en el Máster y a la experiencia laboral, la problemática ha podido ser abordada con más madurez y conocimiento de las tecnologías existentes. Contar con algo de experiencia junto con una visión más amplia de las herramientas disponibles, permite dislumbrar opciones que antes eran invisibles. Además, enfocar un problema conociendo la pila completa ayuda a dividir en bloques la solución y a buscar alternativas para cada uno de ellos en caso de que sea necesario.

Por otro lado, al ser un trabajo con fines de aprendizaje, se eliminaron varias restricciones impuestas durante la realización del proyecto. Para ilustrar cómo limitaban estas restricciones y el resultado obtenido al haberlas modificado, se dedica un apartado para cada una de ellas.

#### Dataset

En el anterior proyecto, la sensibilidad de los datos era especialmente elevada. Esto generaba diversos problemas, ya que en ningún caso podrían salir de las instalaciones de la empresa.

Por otro lado, otro obstáculo fue la creación del dataset, ya que los datos debían ser recogidos y etiquetados por la propia empresa. Esto puede parecer una liberación de carga de trabajo en un principio. Sin emargo, al no disponer de datos de entrada, es difícil avanzar en el desarrollo del modelo.

En este proyecto, al no estar diseñando la solución para un dataset concreto, se puede realizar una búsqueda más general en bancos de datos públicos y elegir alguno similar al objetivo. De este modo se podría haber avanzado en el desarrollo del modelo, mientras se recogían los datos por parte de la empresa. El aprendizaje obtenido en este asunto es que es posible avanzar en el desarrollo del modelo, aprendiendo las técnicas y requisitos necesarios para el problema, con un dataset semejante, sin necesidad de esperar a tener los datos finales.

#### Enfoque

Durante el proyecto, fue difícil encontrar un método para abordar el problema. La búsqueda de proyectos que resolviesen problemas similares no fue muy fructífera, debido a la particular naturaleza del problema.

De forma similar a lo comentado en el párrafo anterior, un mejor acercamiento hubiese sido encontrar un problema similar y adaptarlo poco a poco al objetivo. La mejora en el estado del arte y una mayor madurez en el conocimiento de la materia han permitido realizar búsquedas de soluciones que pueden ser adaptadas al problema.

Además, la aparición de herramientas de mayor alto nivel permiten un acercamiento más paulatino al problema, pudiendo introducirnos en detalles más concretos más adelante, también con los datos finales. El enfoque de lo general a lo particular, junto con la división por bloques, permite avanzar en el desarrollo de cualquier proyecto, pero de nuevo, la experiencia ayuda en estas tareas.

#### Hardware

Uno de los prinicpiales problemas que presenta el desarrollo de modelos de IA es la necesidad de una alta capacidad de cómputo. Otros inconvenientes como la necesidad de una gran cantidad de datos pueden ser apaciguados empleando técnicas de Data Augmentation, pero para entrenar un modelo complejo es necesaria una gran capacidad de cómputo.

No todos los problemas requieren de soluciones extremadamente complejas, de hecho en el mundo del Machine Learning (ML) a veces las soluciones más simples ofrecen mejores resultados. Sin embargo, como se verá más adelante, la complejidad del problema requiere de un modelo complejo, lo que nos obliga a disponer de hardware con gran capacidad de computación en paralelo si queremos obtener resultados en un tiempo razonable.

Hoy en día existen diversas alternativas para entrenar grandes modelos aunque no dispongas de un gran equipo (el crecimiento en el número de alternativas viene motivado a su vez por el crecimiento en el interés en la IA). Continuando con las reflexiones anteriores, en este caso la solución ha sido emplear una plataforma de entrenamiento en la nube, brindando la posibilidad de entrenar modelos complejos sin necesidad de disponer de un equpo dedicado o desgastar en exceso un equipo personal, todo ello por un coste reudcido.

Volviendo al proyecto original, es necesario recordar que esta solución no podría ser aplicada debido a que los datos no podrían salir de las instalaciones de la empresa. De todos modos, puede servir para iniciar en el proceso de investigación y creación de modelos alternativos, con datasets alternativos.

#### Entorno de despliegue

Otro punto problemático residía en torno al entorno de despliegue. Ante un gran problema, difícil de dividir a simple vista, el no disponer de una visualización del resultado final genera malestar y dudas que no favorecen al desarrollo del proyecto. Para que no ocurriera esto, se ha optado por buscar como resultado gráfico final lo mínimo necesario por una persona para poder utilizar el modelo.

Más allá de la interfaz, el paradigma de despliegue de cualquier software requiere de un entorno robusto que asegure su funcionamiento. El aprendizaje recibido durante el máster sobre contenedores, ha permitido utilizarlos para encapsular el modelo como una aplicación web y desplegarlo en una plataforma de hosting, asegurando su funcionamiento.

La conclusión en cuanto a este apartado es que, para poder pensar en un producto final, con una interfaz gráfica perfilada y plena funcionalidad, es necesario primero probar una versión más simple, con funcionalidad reducida, pero que permita validar el modelo y el despliegue. Teniendo esto en mente, es posible avanzar en el desarrollo del modelo, a la vez que se imaginan las distintas posibilidades para el caso de uso concreto.

## 1.2 Planificación

Una vez definido el contexto en el que surge esta idea, vamos a definir cómo se va a desarrollar el proyecto. Debemos definir el objetivo y el alcance del proyecto, para no perder la visión del mismo y poder evaluar el resultado final.

Al mismo tiempo, se inlcuye un listado de requisitos, que ayudarán a dar soporte a la definición del objetivo.

Además, se incluye la planificación temporal inicial del proyecto, en la que descatan varios hitos importantes.

### 1.2.1 Objetivo y alcance

El **objetivo** principal de este proyecto consiste en la creación y el despliegue de un modelo clasificador de audios según emociones humanas.

Al no ser un proyecto destinado para un caso de uso concreto (además de muchas otras limitaciones que presenta), el alcance del proyecto es un poco más abierto. Definimos el **alcance** del proyecto como la creación de una primera versión de una aplicación que implemente la mínima funcionalidad. Debe poder permitir utilizar el modelo de un modo sencillo, sin necesidad de tener conocimientos técnicos, debido a que en un caso real de uso, el usuario final no tendría por qué tener conocimientos técnicos.

### 1.2.2 Requisitos

Para acompañar al objetivo y el alcance, se ha diseñado una tabla de requisitos, separados por categorías:

#### Requisitos funcionales

- **F.1:** El sistema debe ser capaz de clasificar audios en diferentes categorías.
- **F.2:** El sistema debe ser capaz de recibir audios en formato WAV.
- **F.3:** Debe ser accesible desde múltiples dispositivos.
- **F.4:** El modelo permitirá ser actualizado periódicamente con nuevos datos de entrenamiento.

#### Requisitos operacionales

- **O.1:** La respuesta del sistema debe ser inferior a 5 segundos.
- **O.2:** El sistema debe ser robusto en entornos de grabación de alto ruido.
- **O.3:** El sistema debe ser accesible desde cualquier lugar y en cualquier instante.
- **O.4:** El coste computacional de la inferencia del sistema debe ser reducido.

#### Requisitos de diseño

- **D.1:** La implementación del sistema se realizará en Python.
- **D.2:** El despliegue del sistema será desplegado mediante contenedores.
- **D.3:** El equipo de desarrollo será un portátil personal.
- **D.4:** El sistema se ejecutará en un equipo con 2vcpu y 4GB de RAM.

#### Requisitos de seguridad

- **S.1:** Los audios no podrán abandonar nunca las instalaciones.
- **S.2:** La implementación web debe realizarse sobre un servidor seguro.
- **S.3:** El sistema debe ser robusto ante ataques de denegación de servicio.

Muchos de estos requisitos son utilizados a modo de muestra, pero no tienen mayor relevancia en el desarrollo del proyecto. Sin embargo, han servido para imaginar cómo podría ser el sistema en un caso de uso real, además de problemas que podrían surgir en caso de que el sistema se desplegase en un entorno real. Estos requisitos nos ayudan a dar forma a la solución final, entendiendo cómo podríamos mejorar su funcionalidad, además de su robustez.

Es interesante comentar que, aunque no es alcance para este proyecto, los requisitos de seguridad pueden llegar a ser los más críticos en un caso de uso real. En este caso, se ha optado por no profundizar en este aspecto, pero desde luego no es un detalle que deba pasarse por alto.

## 1.3 Diseño de la solución

Previamente, se ha explicado de qué trata el problema y de dónde surge la idea. En esta sección, intentamos responder a la pregunta de cómo se ha abordado el problema.

Siguiendo el famoso dicho, divide y venverás, se ha optado por dividir el problema en dos bloques principales: **Creación del modelo** y **Despliegue del modelo**. De este modo, podemos centrarnos en cada uno de los bloques por separado, sin perder la visión del proyecto.

Los dos bloques son lo suficientemente importantes y están separados entre sí lo suficiente como para que puedan ser abordados por separado y no interfieran entre sí.

En un proyecto real, esta división sería apropiada para separar dos grupos de trabajo. Se podría designar cada bloque a un grupo de trabajo o departamento interno, de modo que cada uno de ellos se encargue únicamente de su parte. Conseguiríamos así una mayor especialización en cada uno de los bloques, además de una mayor eficiencia en el desarrollo del proyecto.

Como en este caso solo hay una persona trabajando en el desarrollo de la solución, la paralelización no influye directamente en el tiempo de desarrollo. Sin embargo, pensarlo de este modo ayuda a:

- Dividir la complejidad del problema a la mitad en primera instancia.
- Focalizar el objetivo de cada bloque.
- Independizar los bloques en caso de que uno de ellos no pueda ser completado.

### 1.3.1 Creación del modelo

El primer bloque, **Creación del modelo**, se centra en la creación de un modelo capaz de clasificar audios según emociones humanas. El objetivo de este bloque será crear un modelo que pueda clasificar los audios con cierta precisión entre un conjunto de emociones previamente definidas.

Algunos aspectos que se tendrán en cuenta en este bloque son:

- Elección del dataset de entrenamiento
- Estudio del estado del arte
- Elección de la arquitectura del modelo
- Entrenamiento del modelo
- Evaluación del modelo
- Exportación del modelo

### 1.3.2 Despliegue del modelo

El segundo bloque, **Despliegue del modelo**, se centra en la creación de una aplicación que permita utilizar el modelo creado en el bloque anterior. El objetivo de este bloque será utilizar el modelo de un modo determinado, ponerlo a disposición del usuario final, y que pueda ser desplegado en cualquier entorno.

Algunos aspectos relevantes en cuanto a este bloque son:

- Diseño de la interfaz
- Despliegue de la aplicación
- Hosting de la aplicación
- Pruebas de funcionamiento

# 2 Diseño del modelo

*I don't know where I'm going from here, but I promise it won't be boring.*

David Bowie

## 2.1 Introducción

Aquí voy a explicar algo muy bonito y divertido adsf

# 3 Despliegue del modelo

*Less is more.*

Ludwig Mies van der Rohe

## 3.1 Problemática y solución

Nos encontramos en esta situación: el modelo ha sido entrenado o está siendo desarrollado por otro equipo. Nuestro trabajo consiste en implementarlo en un entorno de producción para que pueda ser utilizado por los usuarios finales.

Este problema puede ser abordado de varias formas, y la solución estará altamente condicionada por los requisitos del proyecto. Al no tener ningún requisito impuesto para este trabajo, surgen muchos interrogantes cuya respuesta no es trivial:

- **Aspecto final de la solución**: ¿Cómo se va a utilizar el modelo? ¿Qué tipo de interfaz se va a utilizar? ¿Qué tipo de dispositivo se va a utilizar?
- **Requisitos de la solución**: ¿Qué requisitos de rendimiento tiene la solución? ¿Qué requisitos de seguridad tiene la solución? ¿Qué requisitos de escalabilidad tiene la solución?
- **Arquitectura de la solución**: ¿En qué lenguaje se va a implementar la solución? ¿Qué tipo de arquitectura se va a utilizar? ¿Qué tipo de servidores se van a utilizar?

En este caso, se ha optado por simplificar el aspecto final de la solución para centrarnos en la implementación del modelo en un entorno de producción. Se ha decidido crear una interfaz web que permita a los usuarios finales interactuar con el modelo, y una implementación mediante contenedores Docker para facilitar el despliegue en cualquier entorno.

## 3.2 Enfoque escogido

En un principio se pensó en realizar inferencia en tiempo real, es decir, que la aplicación, una vez iniciada, estuviera grabando continuamente y realizando predicciones. Esta opción sin embargo, no ha podido ser llevada a cabo de forma exitosa debido al tiempo de respuesta que ofrece el servidor en el que se ha desplegado la aplicación.

Además, para implementarlo de forma correcta, habría que añadir diversos mecanismos que ayuden a filtrar los audios y nos permitieran extraer muestras que pudieran ser utilizadas para realizar predicciones. Esto implicaría detección de inicio y fin de actividad vocal, filtrado de silencios, etc. La complicación de este proceso, unido al tiempo de respuesta del servidor, ha hecho que se descarte esta opción.

Finalmente, contamos con una aplicación web que permite a los usuarios finales grabar audios y obtener una predicción de la clase a la que pertenece el audio. La aplicación está pensada para ser utilizada en una sola dirección: el usuario debe iniciar la grabación, grabar un mensaje, parar la grabación y esperar el resultado. Una vez obtenido el resultado, puede volver a grabar otro mensaje.

La aplicación es accesible desde cualquier dispositivo que tenga un navegador web a través de la dirección *https://www.classifier-web.com/*.

## 3.3 Aplicación web

Una forma sencilla de crear una interfaz que sea accesible desde cualquier dispositivo es crear una aplicación web.

Aunque el modelo creado puede ser integrado utilizando cualquier lenguaje de programación, se ha optado por utilizar Python para la implementación de la aplicación web. Python cuenta con una gran cantidad de librerías que facilitan la implementación de aplicaciones web, como Flask o Django, quizás las más populares.

Se ha optado por utilizar Flask, debido a que es una librería más ligera que Django y a que es más sencilla de utilizar. Además, contamos con cierta experiencia previa en el uso de Flask, lo que nos permite acelerar el desarrollo de la aplicación.

### 3.3.1 Flask

Flask es un microframework para Python que permite crear aplicaciones web de forma sencilla.

Está diseñado para ser extensible, por lo que es posible añadirle funcionalidades mediante extensiones, aunque en este caso no vamos a utilizar ninguna. Sin embargo, estas extensinoes de alto nivel nos abren las puertas a posibles líneas futuras, como lecturas de bases de datos, autenticación de usuarios, etc.

Utilizar Python para la creación web no siempre es la solución idónea, ya que existen otros lenguajes de programación que están más orientados a la creación de aplicaciones web. Sin embargo, si no se tiene experiencia previa en estos lenguajes, o el objetivo es lanzar una aplicación web de forma rápida, Flask es idóneo. No estamos exentos de tener que crear plantillas en otros lenguajes propios de la web, como HTML, CSS o JavaScript, pero Flask nos permite crear una aplicación web funcional en muy poco tiempo.

### 3.3.2 Estructura de la aplicación

La estructura de la aplicación es muy sencilla, y se puede ver en la figura """poner figura"""

Primero el modelo es cargado en memoria, y se crea una instancia de Flask.

El modelo es cargado en memoria para evitar tener que cargarlo cada vez que se realiza una predicción, lo que se traduciría en un aumento del tiempo de respuesta de la aplicación. Esto puede ralentizar sin embargo el arranque de la aplicaión, pero es una operación que se realiza una única vez, por lo que no es un problema.

Posteriormente se crean las rutas de la aplicación, que son las direcciones a las que se puede acceder desde un navegador web.

Contamos con la ruta principal, que es la que se utiliza para cargar la página principal de la aplicación, y la ruta de predicción, que es la que se utiliza para realizar las predicciones.

La ruta principal simplemente carga un fichero HTML que contiene el código de la página principal.

La ruta de predicción es llamada internamente mediante una petición POST cuando un usuario termina una grabación. La grabación se guarda localmente en el servidor momentáneamente para que el modelo pueda realizar la predicción sobre ella, y posteriormente se borra, por cuestiones de espacio y privacidad.

### 3.3.3 Interfaz web

El desarrollo de interfaces web es un mundo aparte, y no es el objetivo de este trabajo crear una interfaz especialmente atractiva, sino más bien que nos proporcione la funcionalidad básica. Existen desarrolladores especializados únicamente en el desarrollo de interfaces web, y es un campo que requiere de un conocimiento muy amplio, además de experiencia.

Debido a tratar esta parte como algo secundario, sumado a la falta de conocimiento acerca del manejo de audios en la web, se ha optado por basar la interfaz en trabajo previo realizado por otros desarrolladores. En concreto, este proyecto ha utilizado como base """insertar referencia a la interfaz web""".

La interfaz web es muy sencilla, y se puede ver en la figura """poner figura""". Contiene lo básico para que un usuario pueda realizar la grabación de un audio y obtener una predicción de la clase a la que pertenece el audio.

## 3.4 Aplicación Flask en producción

Flask integra un servidor web de desarrollo, que es el que se utiliza por defecto cuando se lanza la aplicación, llamado Werkzeug. Este servidor es muy sencillo de utilizar, pero no está pensado para ser utilizado en producción, ya que no está optimizado para ello.

### 3.4.1 Gunicorn

Para lanzar la aplicación en producción, se ha optado por utilizar Gunicorn, un servidor web HTTP WSGI para Python. Es uno de los servidores más utilizados para lanzar aplicaciones Flask en producción, y es el que se recomienda en la documentación oficial de Flask.??????????????

Gunicorn es un servidor web que se encarga de gestionar las peticiones HTTP que llegan a la aplicación, y de lanzar procesos de la aplicación para atender estas peticiones. Esto permite que la aplicación pueda atender varias peticiones simultáneamente, lo que se traduce en un aumento del rendimiento de la aplicación.

Para lanzar un servicio Flask con Gunicorn, simplemente hay que ejecutar el siguiente comando: """ Insertar comando """"

Este comando lanzará un servidor web en el puerto 8000, que es el puerto por defecto de Gunicorn.

El siguiente paso es configurar un servidor web que actúe como proxy inverso, para que las peticiones HTTP que lleguen al servidor web sean redirigidas al servidor Gunicorn.

### 3.4.2 Traefik

Para configurar el servidor web que actúe como proxy inverso, se ha optado por utilizar Traefik, un servidor web que permite realizar balanceo de carga y que actúa como proxy inverso.

Aunque Nginx es quizás el servidor web más utilizado para realizar esta tarea, se ha optado por utilizar Traefik prinipalmente por su facilidad de configuración. Es comentado que Nginx es más rápido que Traefik, a la vez que ofrece más funcionalidades, pero para este caso con una configuración básica es suficiente.

La mayor ventaja que nos ha brindado Traefik es la facilidad de generar certificados SSL para la aplicación, lo que nos permite utilizar HTTPS. Esto es importante, ya que si no se utiliza HTTPS, los navegadores web no permiten acceder al micrófono del dispositivo, lo que hace imposible la grabación de audios.

No es una tarea difícil de realizar correctamente para un desarrollador experimentado mediante un servidor web como Nginx, pero es mucho más sencillo de realizar con Traefik, y además, al contar con poca experiencia en este campo, nos ha permitido solventar este problema de forma rápida y sencilla. Además, Traefik aún está dando sus primeros pasos, y está ganando popularidad entre desarrolladores, por lo que quizás en un futuro sea una alternativa a Nginx también en entornos reales de producción.

""" insertar foto de *https://monitor.classifier-web.com* indicando admin:admin"""

### 3.4.3 Docker

Para facilitar el despliegue de la aplicación en cualquier entorno, se ha optado por utilizar contenedores Docker. Esta tecnología permite encapsular una aplicación y sus dependencias en un contenedor, que puede ser ejecutado en cualquier entorno que tenga instalado Docker. De este modo nos aseguramos que únicamente tenemos que preocuparnos de que el entorno tenga instalado Docker.

Esta tecnología ayuda a eliminar muchos problemas a la hora de desplegar servicios, pero incorpora otros de los que hay que ser conscientes. En particular, Docker presenta un problema de seguridad, ya que los contenedores son ejecutados por defecto con privilegios de root. Esto implicaría que si un atacante consigue acceder al contenedor, puede tener acceso a todo el sistema.

Este problema se ha solventado creando un usuario no privilegiado dentro del contenedor al construir la imagen de la aplicación, y ejecutando la aplicación con este usuario. Sin embargo, las implicaciones de seguridad de Docker son un tema muy amplio y precisamente pueden llegar a ser determinantes para no utilizar esta tecnología en entornos de producción con requisitos de seguridad muy estrictos. No es el caso de este trabajo, pero es un tema que hay que tener en cuenta y debería ser estudiado en profundidad antes de utilizar Docker en entornos de producción.

A pesar de ello, las ventajas que ofrece Docker son muy interesantes, y es una tecnología que ha ganando mucha popularidad en los últimos años. Las principales ventajas que ofrece son las siguientes:

- **Portabilidad**: Docker permite encapsular una aplicación y sus dependencias en un contenedor, que puede ser ejecutado en cualquier entorno que tenga instalado Docker.

- **Escalabilidad**: Docker permite crear múltiples contenedores de una misma aplicación, lo que permite escalar la aplicación de forma horizontal.
- **Aislamiento**: Docker permite aislar una aplicación y sus dependencias en un contenedor, lo que permite que la aplicación no se vea afectada por otras aplicaciones que se estén ejecutando en el mismo entorno.
- **Rapidez**: Docker permite crear imágenes de aplicaciones de forma rápida, lo que permite desplegar aplicaciones en muy poco tiempo.

### 3.4.4 Docker Compose

Docker Compose es una herramienta que permite definir y ejectutar aplicaciones Docker de forma sencilla. Permite definir las imágenes de los contenedores, las redes, los volúmenes, etc., en un fichero YAML, y ejecutarlos con un único comando.

Es especialmente útil cuando se tienen varias aplicaciones que dependen unas de otras, ya que permite definir todas las aplicaciones en un único fichero. En nuestro caso contamos solo con dos contenedores, pero crear un fichero Docker Compose nos permite definirlos de forma sencilla, construir las imágenes con las dependencias que nosotros definamos y levantar el despliegue con un único comando.

La sintaxis general de un fichero Docker Compose consiste en definir los servicios que se van a utilizar, las imágenes que se van a utilizar para cada servicio, los volúmenes que deben ser creados, las redes, variables de entorno, etc.

En este caso, algunos aspectos a destacar de la definición del fichero son los siguientes:

- **Servicios**: Han sido definidos dos servicios, uno para la aplicación Flask y otro para el servidor Traefik.
- **Imagen**: Se ha utilizado la imagen oficial para el servidor Traefik, y una imagen de Python personalizada con las dependencias necesarias para la aplicación Flask.
- **Volúmenes**: Para el servicio de Traefik se han definido varios volúmenes para almacenar los certificados SSL y la configuración de Traefik.
- **Redes**: No ha sido necesario definir ninguna red, ya que por defecto Docker Compose crea una red interna que es suficiente para que los contenedores se comuniquen entre sí.
- **Variables de entorno**: Han sido definidas varias variables principalmente para el servicio de Traefik, de modo que pueda servir la aplicación Flask con HTTPS.

En el apéndice **??** se puede ver el fichero Docker Compose completo. Además, el proyecto completo está disponible en GitHub, en la siguiente dirección: *https://github.com/antaramol/classifier-web.git*.

## 3.5 Despliegue

Una vez que la aplicación está lista para ser desplegada, es necesario elegir un entorno de producción.

Varias opciones han sido probadas para este trabajo, ya que no se tenía ninguna restricción en cuanto al entorno de producción. Al no tener un servidor propio, se ha optado por utilizar servicios de terceros, que ofrecen servidores virtuales a un precio muy asequible.

De nuevo, igual que comentábamos en el apartado anterior, no es posible utilizar un ordenador personal para este servicio, ya que debería estar conectado todo el tiempo. Además, existen cada vez más opciones de hosting y es muy interesante utilizarlas, ya que nos permiten centrarnos en el desarrollo de la aplicación y no en la gestión del servidor.

Esta es la principal ventaja, nos olvidamos de gestionar la infraestructura. Además muchos servicios permiten gran facilidad para escalar horizontalmente, lo que nos permite aumentar la capacidad de la aplicación de forma sencilla.

Sin embargo, el principal problema es el coste, ya que estos servicios no son gratuitos. Es un inconveniente que, sobre todo para iniciados en este mundo, puede ser determinante para no utilizar estos servicios.

Varias opciones han sido probadas para este trabajo, ya que cada una de ellas ofrece diferentes opciones y precios.

### 3.5.1 Amazon Web Services

Amazon Web Services (AWS) es una plataforma de servicios en la nube que ofrece servicios de computación, almacenamiento, bases de datos, etc. Esta ha sido la primera opción probada debido a su gran popularidad y que se contaba con cierto conocimiento previo.

""" Insertar gráfico de popularidad"""

AWS ofrece una gran cantidad de servicios, y es una de las plataformas más completas que existen. Este ha sido el principal problema, ya que está pensado para ser utilizado por empresas que necesitan una gran cantidad de servicios por su fácil integración entre ellos. Como este trabajo solo necesita un servidor virtual, quizás AWS no sea la mejor opción.

En concreto, ha sido probado el servicio EC2, que ofrece servidores virtuales en la nube. Cuenta con una capa gratuita, que permite utilizar un servidor virtual de forma gratuita durante un año, pero con ciertas limitaciones.

La capacidad de cómputo en la capa gratuita es muy limitada, pero es suficiente para probar la aplicación. Sin embargo, el número de horas de uso es limitado, y una vez que se agotan, hay que pagar por cada hora de uso. Esto hace que no sea una opción viable para este trabajo, ya que el coste sería muy elevado. Este segmento está quizás pensado para pruebas puntuales de aplicaciones, pero no para desplegar aplicaciones en producción.

Sin embargo, sirvió para realizar diversas pruebas de funcionamiento, pruebas de rendimiento, etc., y para familiarizarse con este tipo de servicios.

### 3.5.2 Google Cloud Platform

El resultado es similar al de AWS, ya que Google Cloud Platform (GCP) está enfocado igualmente a grandes proyectos.

Además, la documentación de estos servicios es tan extensa que puede llegar a ser abrumadora para un desarrollador que no esté familiarizado con este tipo de servicios. Una mejor opción para este trabajo es utilizar servicios más sencillos, que estén pensados para pequeños proyectos y que sean más fáciles de utilizar, suavizando así la curva de aprendizaje.

La atracción hacia Google Cloud Platform es que ofrece una gran cantidad de crédito gratuito para utilizar sus servicios, lo que permite utilizarlos de forma gratuita durante un tiempo. El resultado es muy similar a AWS, esta vez se ha utilizado el servicio Compute Engine, que ofrece servidores virtuales en la nube.

### 3.5.3 VPS

Una opción más sencilla y económica es utilizar un VPS (Virtual Private Server), que es un servidor virtual que se encuentra alojado en un servidor físico. Este tipo de servidores cuentan con una ventaja con respecto a los anteriores, y es que el precio es fijo, y no depende del uso que se haga del servidor.

En concreto, se ha utilizado el servicio de VPS de Contabo, recomendado por un compañero de la universidad. Este servicio ofrece servidores virtuales a un precio muy asequible, y con una gran cantidad de recursos.

Es muy sencillo desplegar una instancia de un servidor virtual, a la que podemos conectarnoso mediante SSH. Si a esto le sumamaos la facilidad que nos ofrece Docker Compose para desplegar la aplicación, junto con herramientas de edición como VSCode que nos permiten conectarnos y editar los ficheros de la aplicación de forma remota, el resultado es muy satisfactorio.

Contabo además nos ofrece muchas opciones que nos ayudan a perfilar el resultado final. En nuestro caso, se ha utilizado el área de manejo de DNS para configurar el dominio de la aplicación, de modo que podemos otorgarle un nombre más amigable a la aplicación. Contabo permite crear registros DNS de forma sencilla, y además ofrece un servicio de DNS dinámico, que permite asignar un nombre de dominio a una IP dinámica, que es la que nos ofrece el servidor virtual.

"""insertar imagen de dns zone de contabo"""

Para esta aplicación, se ha elegido un servidor de 4 núcleos, 8 GB de RAM y 50 GB de almacenamiento, suficiente para albergar el modelo. Si en un futuro se necesitara más capacidad, se podría escalar horizontalmente, creando más instancias de la aplicación y balanceando la carga entre ellas. Para esta implementación, quizás sería más conveniente estudiar en profundidad los servicios de AWS o GCP, ya que ofrecen más facilidades para escalar horizontalmente, pero para el propósito de este trabajo, es más que suficiente.

# 4 Conclusiones

*If we knew what it was we were doing, it would not be called research, would it?*

Albert Einstein

Aquí voy a explicar las conclusiones

# Índice de Figuras

# Índice de Tablas

# Índice de Códigos

# Bibliografía

[1] Universidad Internacional de Valencia, *¿cuáles son las aplicaciones de la ia actuales y futuras?*, April 2023.

[2] Sage Kelly, Sherrie-Anne Kaye, and Oscar Oviedo-Trespalacios, *What factors contribute to the acceptance of artificial intelligence? a systematic review*, Telematics and Informatics **77** (2023), 101925.