Facultad de Informática

Trabajo de Fin de Máster

Black Bolt-Otea: Plataforma Libre para la Transcripción de Relatos para Personas con Discapacidad Visual/Black Bolt-Otea: Free Platform for Transcribing Stories for People with Visual Impairments

Por

Samuel Antonio Eugercios Nevado

UNIVERSIDAD COMPLUTENSE
MADRID

Dirigido por

José Luis Vázquez Poletti
David Pacios Izquierdo

Nota Convocatoria junio-Julio: 9

Abstract

This work aims to create an audio-relay generation system for visually impaired people. For this purpose, Python 3.8 has been used as development language to generate the audio-relays in natural voice. Futhermore, LaTeX was used to develop this document. In addition, comparisons of processing time of different amounts of audio-relay generation on a local computer versus cloud computing and serverless have been made, as well as the cost of each of these services. It was found that the serverless architecture can generate a greater amount of audio-relays at a better cost and time than the other two technologies. Therefore, we have proceeded to create a distributed architecture to generate audio-relays. As future work, the application will send the audio-relays by email and will connect with Moodle.

Keywords: Lambda, serverless, audio-stories, visual impairment, cloud computing.

Resumen

Este trabajo pretende crear un sistema de generación de audio-relatos para personas con problemas visuales. Para ello, se ha utilizado como lenguaje de desarrollo Python 3.8 para generar los audio-relatos en voz natural. Por otro lado, se ha utilizado LaTeX para desarrollar el documento. Además, se ha realizado comparaciones de tiempo de procesado de diferentes cantidades de generación de audio-relatos en un ordenador local frente *cloud computing* y*serverless*, además del coste de cada uno de estos servicios. Se ha comprobado que la arquitectura *serverless*, puede generar una mayor cantidad de audio-relatos con un coste y tiempo menores que las otras dos tecnologías. Por ello, se ha procedido a crear una arquitectura distribuida para generar audio-relatos. Como trabajo futuro, la aplicación enviará los audio-relatos por email y se conectará con Moodle.

Palabras Clave: Lambda, serverless, audio-relatos, problemas visuales y cloud computing.

Índice general

Abstract IV

Capítulo 1 **Introduction** 1

Capítulo 1 **Introducción** 1

Capítulo 2 **Estado del Arte** 3

Capítulo 3 **Tecnologías** 14

Capítulo 4 **Diseño de solución** 22

Capítulo 5 **Arquitectura e implementación** 29

Capítulo 6 **Mediciones, resultados, conclusiones y trabajo futuro** 35

Capítulo 6 **Measurements, conclusions, results and future work** 46

Capítulo 7 **Enlace a los archivos del TFM** 57

Bibliografía 59

Capítulo 1. Introduction

1.1. Motivation

The human being perceives the world around us with his senses, receiving a sea of stimuli and sensations through them, allowing us to know our environment with ease. The problem is that the human body is fragile, causing these senses to be affected by factors such as illness and age, making everyday and simple activities impossible to perform normally.

The vast majority do not give importance to these limitations, either because their state of health is good or their age does not produce any impediment. But the world and life is not fair, at any time an illness, age or even an accident can take away any of your senses. It is not unreasonable, as it has happened even globally recently with the Covid pandemic, leaving as sequels that many of the affected patients lost their senses of taste and smell for days, months and in rarer cases years.

Two of our most important senses are sight and hearing, which allow us to see and hear the world around us clearly. Losing either of these senses, or having their capacity be less than usual, can leave us in darkness in the case of sight and in absolute silence in the case of hearing.

However, new technologies open up a range of possibilities to combat this type of sensory impairment. From the loudspeakers at traffic lights in the city of Madrid, which indicate with a high-pitched beep to blind people when they can cross the street, to cochlear implants, which replace the damaged parts of the inner ear (cochlea) and directly stimulate the auditory nerve, allowing the brain to interpret sounds and a person to hear again.

In conclusion, there will always be factors that interfere with our senses throughout our lives, complicating tasks that until a few years or months ago we could perform without a problem and that now may be difficult or impossible to do.

But, our will to move forward makes us find solutions to be able to continue with our day to day life, despite the problems of losing or having diminished capabilities of our senses.

1.2. Objective to be achieved

It is intended to create an architecture of transcription of a text from a web of stories, to translate it into a desired language and convert it into an audio story to help people who have low visibility or have no vision in their eyes with the following

features.

- Implementation of PDF generation with the transcribed text.
- Implementation of translated text and audio to a desired language.
- Audio enhancement with intonation.

The operations will be developed:

- Generation of PDF document with the text.
- Modification of the original language of the document.
- Audio generation of the original document or of the modified document into another language.
- Improvement of the generated audio.

This tool is composed of two modules:

- Document generation and enhanced audio.
- Modification of the document and audio to another language.

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

El ser humano percibe el mundo que nos rodea con sus sentidos, recibiendo un mar de estímulos y sensaciones a través de ellos, permitiendo conocer nuestro entorno con facilidad. El problema es que el cuerpo humano es frágil, provocando que dichos sentidos se vean afectados por factores como enfermedades y la edad, haciendo de actividades cotidianas y sencillas algo imposible de realizar con normalidad.

La gran mayoría no da importancia a estas limitaciones, ya sea porque su estado de salud es bueno o su edad no le produce ningún impedimento. Pero el mundo y la vida no es justa, en cualquier momento una enfermedad, la edad o incluso un accidente puede arrebatar alguno de tus sentidos. No es algo descabellado, ya que ha sucedido a incluso a nivel global de forma reciente con la pandemia del Covid, dejando como secuelas que muchos de los pacientes afectados perdieran los sentidos del gusto y el olfato durante días, meses y en casos más raros años.

Dos de nuestros sentidos más importantes son la vista y el oído, que nos deja ver y escuchar el mundo que nos rodean con claridad. Perder alguno de estos sentidos o que su capacidad sea menor de lo habitual puede dejarnos en las tinieblas en caso de la vista y en el silencio absoluto en el caso de la audición.

Pese a ello, las nuevas tecnologías abren un abanico de posibilidades para luchar contra este tipo de afecciones sensoriales. Desde los altavoces de los semáforos de la ciudad de Madrid, que indican con un pitido agudo a las personas invidentes cuando pueden cruzar la calle, hasta los implantes cocleares, que sustituyen las parte dañadas del oído interno (cóclea) y estimulando directamente el nervio auditivo, lo que permite que el cerebro interprete los sonidos y una persona vuelva a escuchar.

En conclusión, siempre habrá factores que interfieran con nuestros sentidos a lo largo de nuestras vidas, complicando tareas que hasta hace unos pocos años o meses podíamos realizar sin problema y que ahora pueden ser difíciles o imposibles de hacer.

Pero, nuestra voluntad de seguir adelante hace que encontremos soluciones para poder seguir con nuestro día a día, pese a los problemas de perder o tener disminuidas las capacidades de nuestros sentidos.

1.2. Objetivo a realizar

Se pretende crear un arquitectura de transcripción de un texto de una web de relatos, para traducir a un idioma deseado y convertirlo en un audio-relato para la ayuda a las personas que tienen baja visión o no tienen visión con las siguientes características.

- Implementación de generación de PDF con el texto transcrito.
- Implementación de texto traducido y audio a un idioma deseado.
- Mejora del audio con entonación.

Las operaciones se desarrollarán:

- Generación de documento PDF con el texto.
- Modificación del idioma original del documento.
- Generación de audio del documento original o del modificado a otro idioma.
- Mejora del audio generado.

Esta herramienta está compuesta por dos módulos:

- Generación del documento y audio mejorado.
- Modificación del documento y del audio a otro idioma.

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 utilización de audiolibros para ayuda de personas con problemas visuales.

2.1.1. Audible

La aplicación de Audible[1] (mostrada en la Figura 2.1), es una *app* para iOS y Android que permite escuchar audiolibros y *podcasts*. Fue creada por Amazon y requiere una suscripción de pago. Cuenta con un gran catalogo de audiolibros, pero debemos estar pagando por la suscripción. La versión "gratuita" cuenta con unos servicios ofrecidos muy limitados, en cambio, la versión de pago te permite incluso descargar los audiolibros y *podcasts*, tener acceso ilimitado a todos los audiolibros disponibles o incluso poder cambiar algo tan sencillo como la velocidad de reproducción del los audios. Además hay varios estudios sobre Audible [1] entre ellos destaca el artículo de Neil Verma, Universidad Northwestern, EE. UU. denominado: "*From the narrator's lips to yours: Streaming, podcasting, and the risqué aesthetic of Amazon Channels*". En este estudio se muestra cómo la elección del audio-libro tiene en cuenta una serie de factores externos y el mismo acceso a ellos es un impedimento económico, ya que, el precio de esta suscripción ha ido variando a lo largo de los años.

2.1.2. Aplicación web móvil de audiolibros en la Universidad Regional Autónoma de los Andes – Uniandes sede Ibarra

En la Universidad Regional Autónoma de los Andes, se creó una aplicación web móvil de los audiolibros de la biblioteca de su universidad [2] de la mano de don Eduardo Luis Gudiño Llerena (mostrada en la Figura 2.2), debido a la carencia de material en audio en los repositorios de la universidad para los estudiantes. Con la puesta en marcha del desarrollo de esta aplicación[2] se buscaba la mejora de la accesibilidad a información educativa para alumnos y profesores en el ámbito académico, dentro y fuera del campus universitario, además de tener una gestión de la información educativa de la biblioteca y poder difundirla de manera adecuada bajo un control exhaustivo.

[1]Dirección web: `https://www.audible.es/ep/apps`
[2]Dirección web: `https://dspace.uniandes.edu.ec/handle/123456789/18229`

Figura 2.1: Imagen de la web de Audible

2.1.3. Una secuencia didáctica basada en audiolibros para la transformación de la habilidad de escucha del inglés en el grado octavo de la Institución Educativa Departamental General Santander de Sibaté – Cundinamarca

En el trabajo de fin de grado de José Alexander Cita Jiménez [3] de la Universidad Tecnológica de Pereira[3], se llevó a cabo esta investigación para tratar de comprender la mejora de pronunciación y comprensión del inglés en alumnos de octavo de bachillerato de Colombia con la utilización de audiolibros, lo que demuestra que los estudiantes que trabajan con audiolibros en inglés tienen una mejor comprensión auditiva. También al final de dicho estudio (mostrado en la Figura 2.3), se dan recomendaciones de la aplicación del uso de los audiolibros en el contexto educativo de Colombia, ya que aumenta los recursos y la facilidad para los estudiantes dentro del ámbito académico.

[3]Dirección web: `https://repositorio.utp.edu.co/entities/publication/22aaa157-b19a-4cef-9149-6f1827af722c`

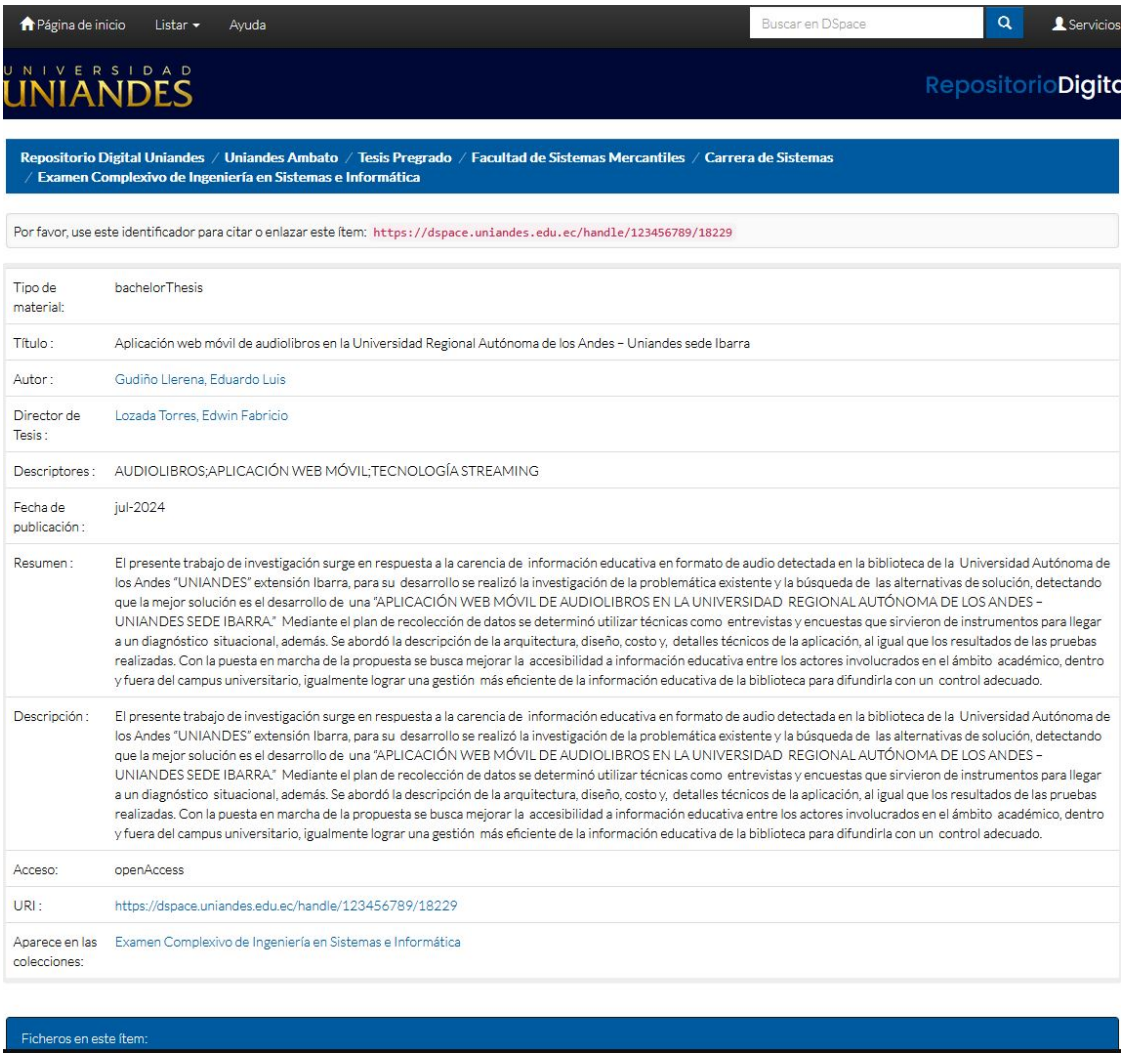


Figura 2.2: Imagen del repositorio web de UniAndes

2.1.4. Los audiolibros en la comprensión lectora de textos narrativos en los estudiantes del segundo grado de educación secundaria del colegio "Santa Rosa de Viterbo" de la ciudad de Huaraz – 2020

Esta investigación se realizo sobre treinta estudiantes de bachillerato del colegio de educación secundaria "Santa Rosa de Viterbo"[4] por parte de la Universidad Nacional Santiago Antunez de Mayolo(UNASAM), con el objetivo de mejorar la comprensión lectora de los alumnos.

Dichos alumnos del experimento carecían de la comprensión lectora del nivel de estudios educativos que cursaban, por lo que primero se les dieron textos narrativos para su lectura y comprensión (repositorio mostrado en la Figura 2.4), además de audiolibros para poder mejorar ambos aspectos. Los test realizando antes y después del experimento demuestran una mejora en la comprensión lectora y comunicación gracias a los Audiolibros. Además se les aplicó la prueba Wilcoxon [5], para compro-

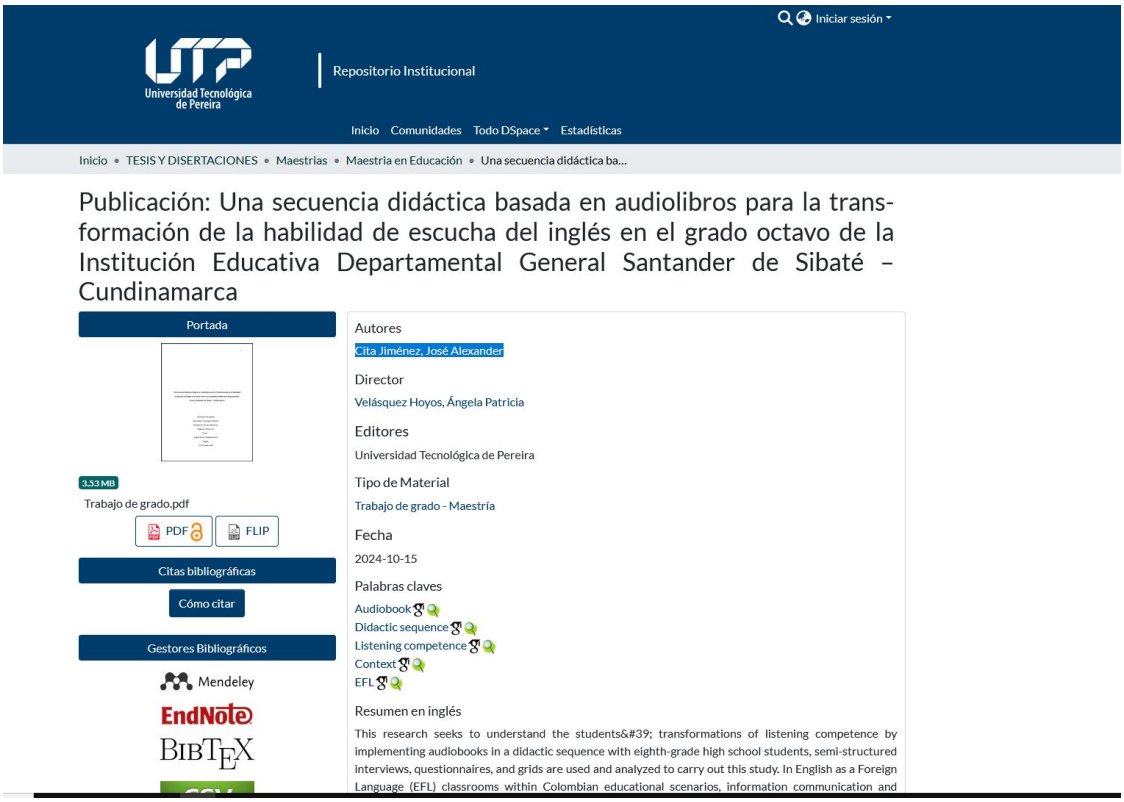


Figura 2.3: Imagen del repositorio web de la Universidad Tecnológica de Pereira

bar la mejora estilísticamente de los alumnos que se habían sometido a dicho experimento. Comprobando que era un éxito rotundo el uso de audiolibros en los alumnos para mejorar su capacidad de comprender textos y mejorar su comunicación.

2.1.5. ¿Leer con los oídos?: audiolibros y literatura infantil y juvenil

El informe del año 2019 realizado por Araceli García-Rodríguez; Raquel Gómez-Díaz de la Universidad de Salamanca [6] hace un estudio del aumento del uso de los audiolibros en la sociedad española y cómo se ha ido incrementando su uso.

Figura 2.4: Imagen del repositorio web de la Universidad Antunez de Mayolo

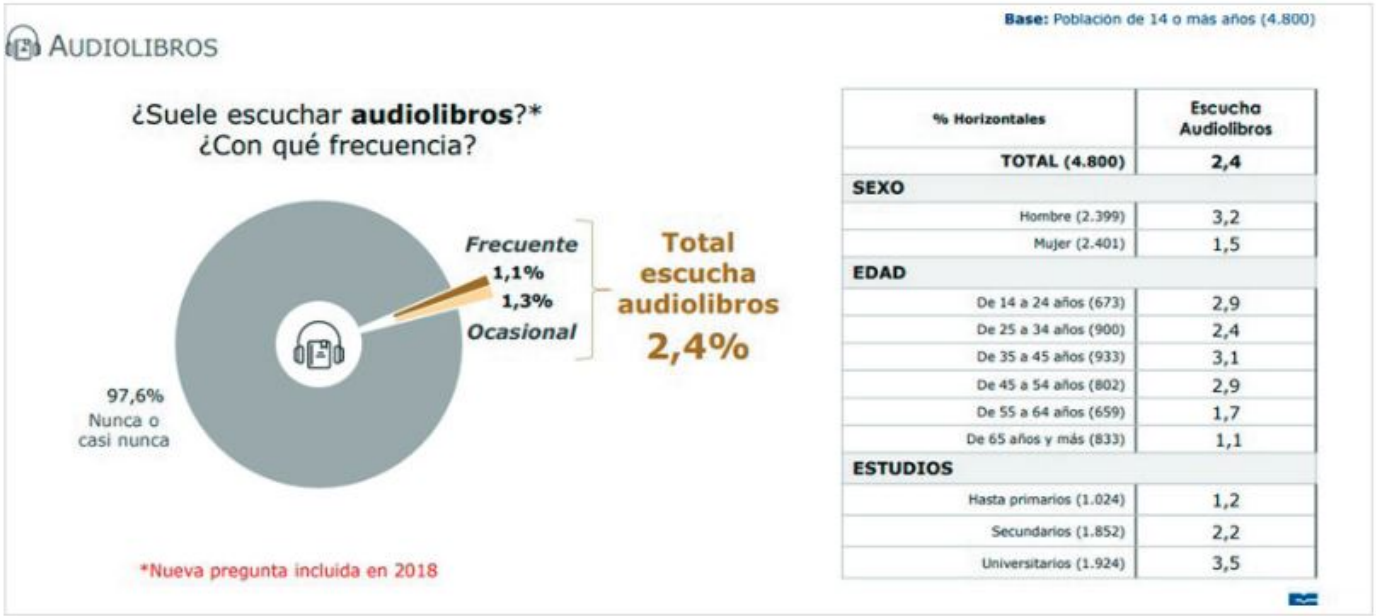


Figura 1. Utilización de los audiolibros
Fuente: (Ministerio de Cultura y Deporte, 2019)

Figura 2.5: Imagen del porcentaje de gente que escucha audiolibros en el 2018

Pese a que en el 2018 los números aportados son bajos (mostrados en la Figura 2.5), los usuarios han ido aumentando progresivamente y cuando más se escucha los audiolibros es cuando se está fuera de las viviendas normalmente más de la mitad de los usuarios y veinticinco de cada cien personas lo utilizan haciendo deporte. Pese

Cuadro 2.1: Colección de audiolibros infantiles en la plataforma E-Biblio disponible en las Bibliotecas Públicas del Estado. Fuente: eBiblio consultada el 22 de abril de 2019.

Comunidad Autónoma	Audiolibros	Audiolibros infantiles	Audiolibros infantiles streaming	Audiolibros infantiles descarga
Andalucía	352	26	19	7
Aragón	173	5	0	5
Asturias	142	5	0	5
Baleares	150	5	0	5
Canarias	140	5	0	5
Cantabria	429	5	0	5
Castilla La Mancha	142	5	0	5
Castilla y León	142	5	0	5
Cataluña	607	5	0	5
Ceuta	139	5	0	5
Extremadura	250	23	23	0
Galicia	157	5	0	5
La Rioja	211	5	0	5
Madrid	393	25	20	5
Melilla	140	5	0	5
Murcia	304	5	0	5
Navarra	142	5	0	5
Valencia	142	5	0	5

a entrar de forma lenta y con retraso en el mercado español (como se muestra en la Tabla 2.1), su aumento es progresivo por parte de editoriales, que están utilizando como reclamo actores famosos en sus doblajes para atraer nuevos audio-lectores y dar a conocer este formato. Sin olvidarnos de las ventajas para los niños del uso de este medio, que fortalecen la capacidad del niño para escuchar, prestar atención y emocionarse, gracias a los matices que la voz y la narración dan al audio.

2.1.6. La plataforma e-Biblio

La plataforma e-Biblio del Ministerio de Cultura[4] para el préstamo de libros digitales empezó a funcionar en septiembre de 2014. En la actualidad tiene un total de 58.552 títulos distintos en castellano, otras lenguas cooficiales y en idiomas extranjeros. El libro electrónico, frente a otras tipologías como las revistas o los audiolibros, sigue siendo lo más prestado, representando casi la mitad de la totalidad de los préstamos.

Según datos de principios de 2024 (ilustrados en la Figura 2.6), ha habido un total de 4.113.814 préstamos, incrementándose estos en todos los tipo de documentación disponible, destacando en las revistas (18 %), aunque también en los libros electrónicos (13 %), audiolibros (16 %) y periódicos (16 %).

Los tipos de libros más prestado, en libros electrónicos se ve una clara preferencia por los autores nacionales contemporáneos como Juan Gómez Jurado, Irene Vallejo

[4]Dirección web: `https://ebiblio.es/en-cifras.html`

o Alana S. Portero, entre otros, aunque también es notoria en audiolibros la demanda que incluye títulos de autoría extranjera, entre ellos algunos clásicos como Cien años de soledad. En relación a la prensa, los tres periódicos más prestados son de ámbito nacional (El País, ABC y La Razón), y las revistas favoritas por los usuarios son las de crónica social y las de salud, como Lecturas o Computer Hoy.

Figura 2.6: Datos de E-Biblio en el año 2024

2.1.7. Producción de un audiolibro orientado a la educomunicación para niños no videntes de 5 a 7 años de edad

Este trabajo de Fin de Grado [7] (repositorio mostrado en la Figura 2.7) de Ana María Durán González, Doménica Natalia Fajardo Palacios y Andrew Stefano Romero Arévalo de la Universidad del Azuay de Ecuador[5] en el 2018, donde se desarrolló un audiolibro con cuentos infantiles para niños invidentes para introducirlos a la lectura de la escuela Claudio Neira Garzón de la Ciudad de Cuenca.

La metodología utilizada por este equipo de desarrollo y de investigación fue utilizar texto y escritos de expertos en comunicación educativa, técnicas de producción audio y expertos docentes en la formación y educación de personas invidentes. Como

[5]Dirección web: `https://dspace.uazuay.edu.ec/handle/datos/7666`

Figura 2.7: Repositorio del la Universidad de Azuay

resultado de esta investigación se llego a desarrollar un audiolibro con un conjunto de ocho cuentos para la ayuda a la educación de niños de entre 5 a 7 años.

2.1.8. Creación de audiolibros como una estrategia innovadora en el método de enseñanza universitaria

Es un artículo de la Conferencia Civinedu 2020(4th International Virtual Conference on Educational Research and Innovation cuyo cartel se muestra en la Figura 2.8)[8] de Jerónimo Aragón Vela, Sara Torres Rusillo y Francisco Javier del Rio Olvera en colaboración con la Universidad de Cádiz. Se plantea la creación de audiolibros para el material docente realizado por el profesorado. Estos audiolibros se pusieron en marcha en asignaturas pertenecientes al Grado de Psicología y al grado de Educación Primaria con la ayuda del responsable de Atención a la Discapacidad de la Universidad de Cádiz. Además se dio oportunidad a los alumnos de tercero y cuarto de ambos grados para participar en el proceso de selección de poner voz a los diferentes audiolibros, además de darles una formación específica para desarrollar dicha actividad.

Una vez aportado todo el material por parte del profesorado, desde la oficina de Atención a la Discapacidad de la Universidad de Cádiz empezaría la difusión de los audiolibros a todos los alumnos susceptibles de necesitar dicho material en sus estudios universitarios.

Figura 2.8: Cartel de la conferencia Civinedu 2020

2.1.9. Audiolibros en griego y en latín

En la Universidad Aútonoma de Madrid, Iván Andrés-Alba [9] presenta este artículo[6] sobre el uso de audiolibros para estudiantes que no conozcan latín y griego antiguo. Está enfocado en aquellos estudiantes que no están familiarizados en dichos idiomas en sus estudios de bachillerato o que su nivel no sea muy alto, para ayudarlos a comprender y mejorar con estas lenguas muertas o sin apenas utilización.

Dicha iniciativa tiene como propósito no solo el aprendizaje positivo de los alumnos, si no también la creación de un repositorio de textos clásicos en ambos idiomas para su uso pedagógico por parte del profesorado. Además se obtienen los siguientes resultados dentro de los estudios de lenguas clásicas:

- Mejorar la eficacia del trabajo con textos.
- Apoyar a los estudiantes en sus primeros años universitarios.
- Fomentar la práctica autónoma.
- Contribuir a la visibilidad de los Estudios Clásicos

[6]Dirección web: `https://revistas.uma.es/index.php/thamyris/article/view/18354/18373`

2.1.10. Audiolibros como herramientas para fomentar la lectura en niños con dificultades de aprendizaje

Este artículo de la revista on-line Polo del Conocimiento[7] (repositorio mostrado en la Figura 2.9), realizado por Mónica Maricela Franco González, Kassandra Marlene Baque Cedeño, Lucio González Livia Susana y Lexandra Estupiñán Casquete [10], pone foco en la utilización de los audiolibros para ayudar a fomentar la lectura en niños con dislexia, TDAH y otras dificultades del aprendizaje.

Figura 2.9: Repositorio de la Revista digital Polo del Conocimiento

En el mismo se expone que los audiolibros ayudan a mejorar la comprensión lectora al reducir la carga cognitiva de la descodificación del texto, fomenta la lectura y comprensión lectora de los niños, amplían su vocabulario, mejora su expresión oral y mejora la retención de la información escuchada en los audiolibros.

Por ello exponen que deben integrarse el uso de audiolibros en el circuito escolar, para ayudar a los niños con dichos problemas desde el primer momento para que la lectura y el aprendizaje no sea un obstáculo para ellos.

2.1.11. Audiolibros para el desarrollo del pensamiento creativo

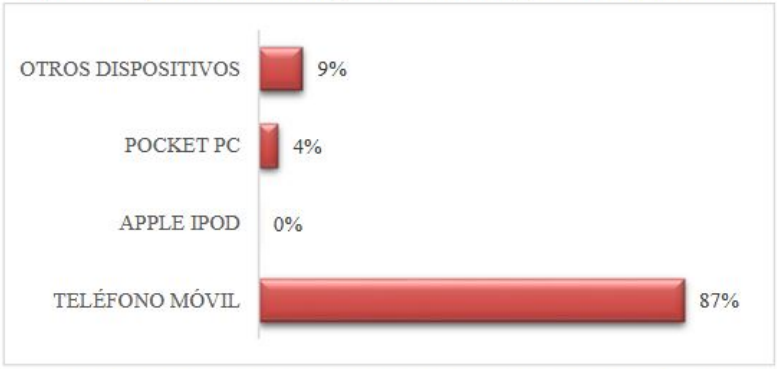
Este artículo de la revista de la Universidad de la Rioja y de la Fundación Dialnet[8], realizado por Lucía Atenaida Pin-García y Ángela María Cevallos-Cedeño [11], expone que un audiolibro ayuda a complementar la lectura escrita y no la sustituye

[7]Dirección web: `https://polodelconocimiento.com/ojs/index.php/es/article/view/9233`
[8]Dirección web: `https://dialnet.unirioja.es/servlet/artculo?codigo=8231701`

en ningún momento, además antes de realizarse este estudio los indicadores marcaban un retroceso en los hábitos de lectura de Ecuador.

Se procedió a un estudio científico-analítico en un centro educativo de la parroquia Eloy Alfaro del cantón Manta en Ecuador, sobre un grupo de 79 estudiantes de dicho centro que habían utilizado audiolibros y se les hizo responder varios test.

Figura 1: Dispositivos utilizados por los estudiantes para escuchar audiolibros.



Fuente: Encuesta web realizada a los estudiantes del décimo año básico de una Unidad Educativa del cantón Manta.

Figura 2.10: Resultado de medios utilizados para escuchar los audiolibros por parte de los alumnos.

La Figura 2.10 muestra cómo en la práctica la elección del dispositivo para realizar la audiolectura es esencial y es un vehículo a tener en cuenta a la hora de generar un vínculo que fortalezca y fomente la lectura.

De este artículo se obtuvo que esta práctica fomenta el pensamiento creativo, fortaleciendo el gusto por la lectura, la capacidad de imaginar y tener una mayor comprensión lectora en los alumnos con los que se ha realizado el estudio.

Capítulo 3. Tecnologías

Las tecnologías que vamos a utilizar a continuación se basan en la extrapolación de los estudios realizados sobre el Capítulo 2. En nuestro caso los relatos vamos a usarlos de una web de relatos en vez de un repositorio de libros o libros digitales (como por ejemplo e-biblo que dispone de audiolibros y libros digitales), se van a usar además librerías específicas para traducir los textos a otros idiomas específicos y de generación de audios (en vez de personas como en el articulo de Civenedu 2020 y el de los libros en griego y latín, que iban a ser generados y verbalizados por profesores y alumnos)y perfectamente audible para el usuario.

3.1. Blog del relatos: El rincón del escriba eterno

Para nuestro proyecto, hemos utilizado el blog de relatos: *el rincón del escriba eterno*[1] (mostrado en la Figura 3.1), que tiene un repositorio de más de 175 relatos escritos en español. Se ha utilizado este blog, ya que tanto los relatos como el mismo blog son de mi autoría y propiedad, evitando problemas de posibles acusaciones de Web scrapting[12] o de apropiación indebida.

3.2. Amazon Web Service (AWS)

AWS [13] es una colección de servicios en la nube, que es la más completa y adaptada del mundo. Al contar en su repositorio (interfaz mostrada en la Figura 3.2) con más de 200 servicios, es utilizado para el desarrollo de arquitecturas y software en el ámbito público y privado. Para usar esta herramienta se debe estar registrado. La herramienta ha sido utilizada para optimizar el rendimiento de los diferentes módulos de código, ya que puede gestionar centenares de hilos de ejecución a un bajo coste y con poca capacidad de memoria disponible.

3.3. Amazon Lambda

Amazon Lambda o simplemente *Lambda*[2], es un servicio sin servidor basado en la ejecución de eventos, por lo que permite ejecutar el código de cualquier aplicación y backend disponible, sin tener que hacer reservas de servidores o de memoria. La Figura 3.3 muestra a modo de tutorial cómo funciona Lambda en caso de que se suba una foto al almacenamiento S3 para ejecutar una función de renderizado de imagen.

[1]Dirección del blog y de la imagen`https://elrincondelescribaeterno.blogspot.com/`
[2]Dirección de lambda y de la imagen`https://aws.amazon.com/es/lambda/`

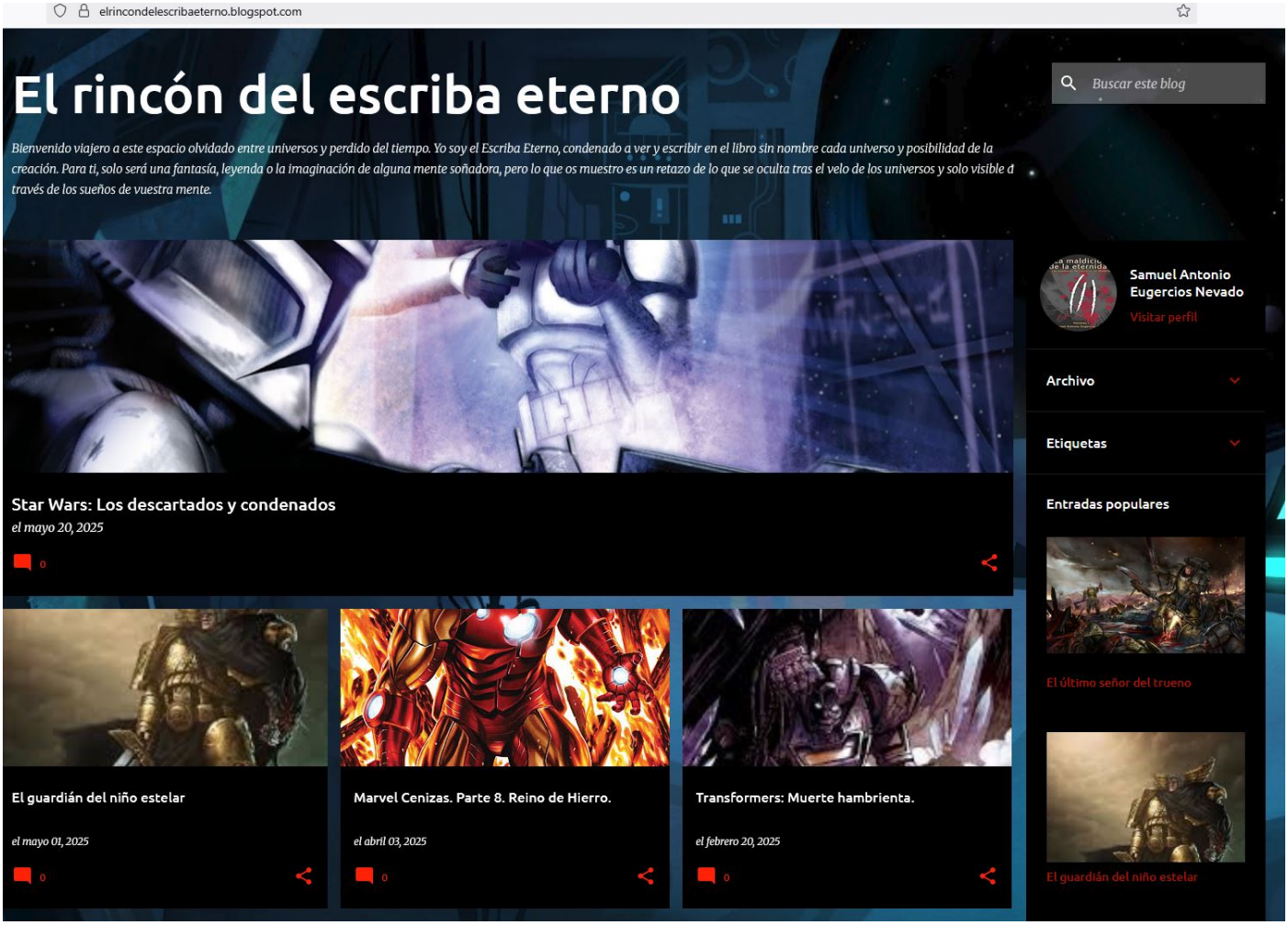


Figura 3.1: Imagen del Blog de el rincón del escriba eterno.

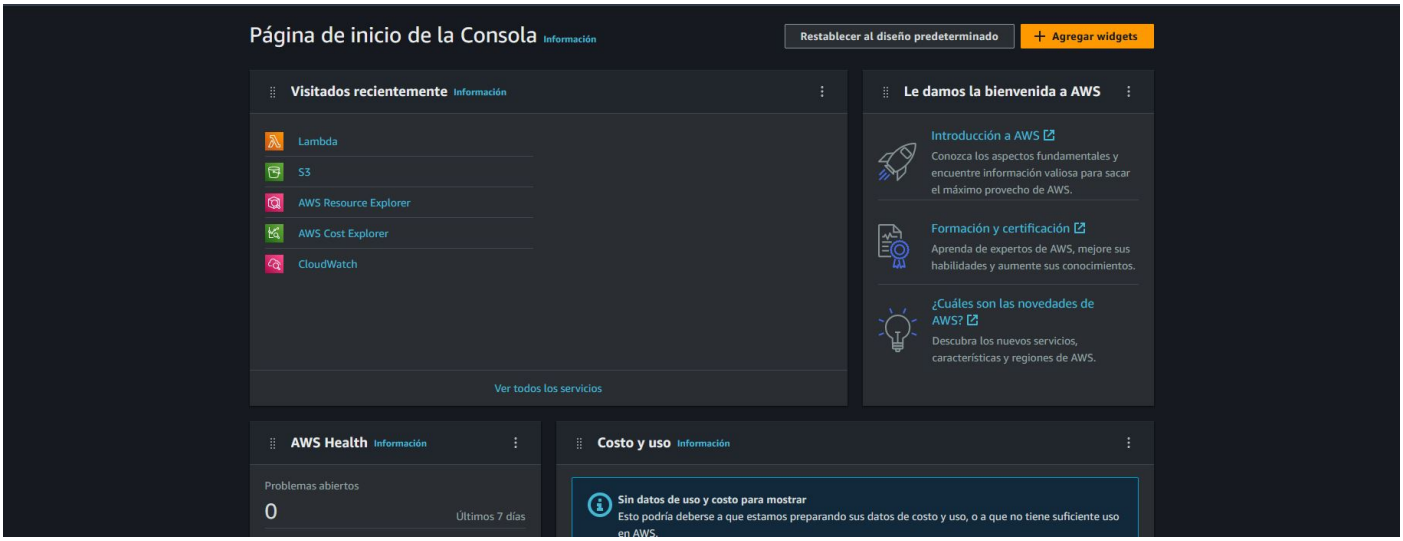


Figura 3.2: Imagen del entorno de bienvenidade AWS.

3.4. Amazon S3

Amazon S3 o también denominados contenedores S3[3], es un servicio que almacena cualquier objeto tanto de entrada como de salida(en nuestro caso va almacenar

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

Figura 3.3: Funcionamiento de Lambda para procesamiento de archivos.

texto, pdf y audio). Ofrece una alta escalabilidad, seguridad y los datos están disponibles para ser usados por todas las aplicaciones Lambda de nuestro diseño. La Figura 3.4 muestra los distintos tipos de archivos que se pueden subir al contendor S3 para ser almacenados. Este tipo de almacenamiento funciona como *Google Drive* salvo que con este tipo de almacenamiento se pueden ejecutar distintas funcionalidades con multitud de herramientas dentro de *Amazon Web Service*.

Figura 3.4: Esquema de funcionamiento de S3 de Amazon.

3.5. Python

Se ha elegido este lenguaje[4] de desarrollo por su gran amplitud de librerías disponibles, su fácil manejabilidad y su compatibilidad con AWS. En la Figura 3.5 se puede observar cómo este lenguaje se coordina fácilmente con otras herramientas, en este caso Visual Studio Code, para realizar múltiples funcionalidades.

[4]Enlace a la plataforma para su descarga:`https://www.python.org/`

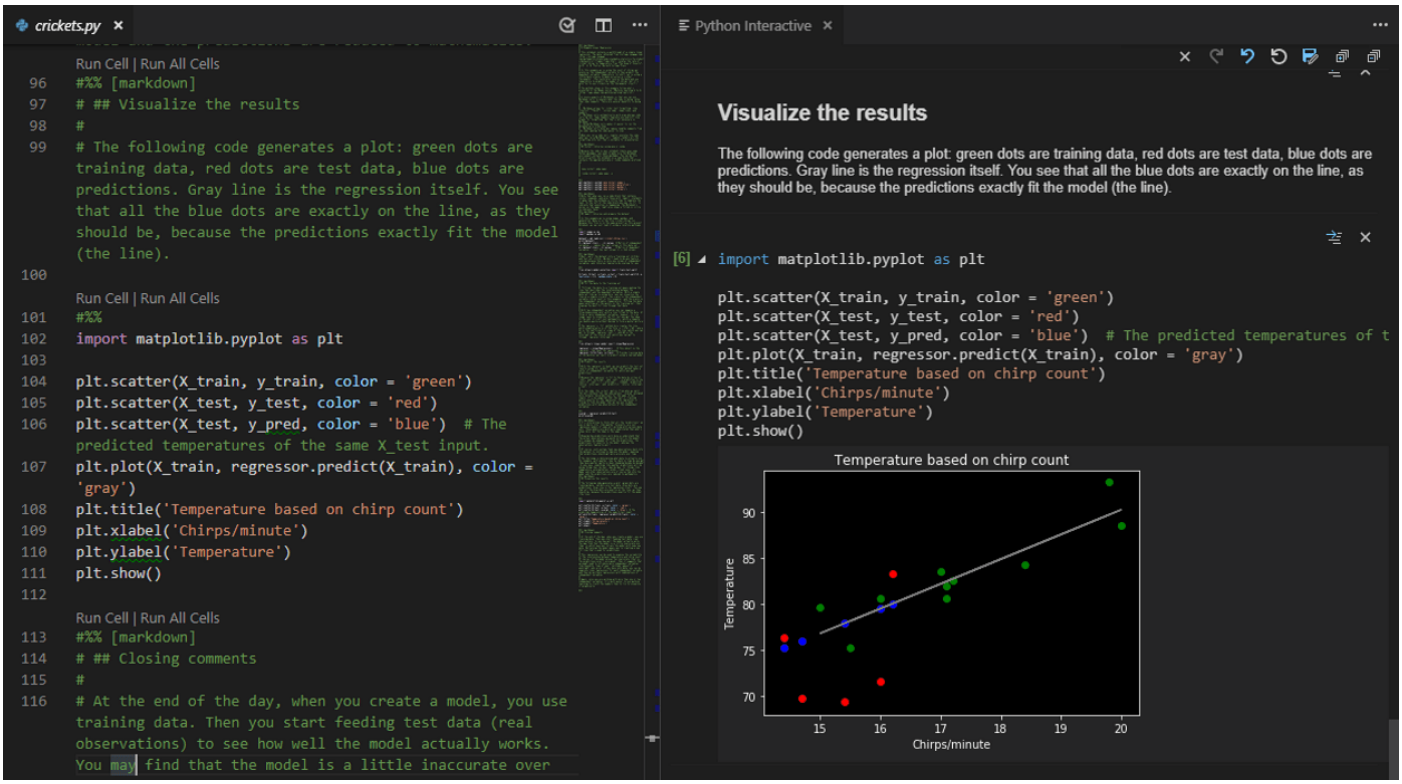


Figura 3.5: Interfaz del lenguaje Python en la herramienta Visual Studio Code.

3.5.1. GoogleTranslator

API gratuita e ilimitada de Google[5] para Python para traducir texto a diferentes idiomas, detectar automáticamente el idioma de un texto y también realizar traducciones en masa.

3.5.2. matplotlib

Es una librería de Python[6] que permite crear visualizaciones de datos. Un ejemplo de uso es el mostrado en la Figura 3.5 donde es utilizado para ilustrar una fórmula matemática.

3.5.3. BeautifulSoup

Esta API de Python[7] se usa para extraer datos de documentos HTML y XML, como en nuestro caso texto de una web. Como en caso de la Figura 3.6 donde se muestra cómo se puede extraer un texto de una página web.

[5]Enlace a la API:`https://cloud.google.com/translate/docs/reference/rest`
[6]Enlace a la librería:`https://matplotlib.org/`
[7]Enlace a la API:`https://pypi.org/project/beautifulsoup4/`

```
<div class="row" id="all_quotes">
  <div class="col-6 col-lg-3 text-center margin-30px-bottom sm-margin-30px-top">
    <a href="/inspirational-quotes/8052-never-mistake-knowledge-for-wisdom-one-helps">
      <img alt="Never mistake knowledge for wisdom. One helps you make a living; the other helps you make a life. #<Author:0x00007f1917948248>"
      class="margin-10px-bottom shadow" src="https://assets.passiton.com/quotes/quote_artwork/8052/medium/20200324_tuesday_quote.jpg?1584726475"
      width="310" height="310" data-no-retina>
    </a>
    <h5 class="value_on_red">
      <a href="/inspirational-quotes/8052-never-mistake-knowledge-for-wisdom-one-helps">WISDOM</a>
    </h5>
  </div>
```

Figura 3.6: Aplicación de la librería para realizar un scrapping de una página web.

3.5.4. FPDF

API de Python[8] para la creación rápida y sencilla de documentos PDF.

3.5.5. fitz o PyMuPDF

Esta librería de Python[9] nos permite el manejo de documentación PDF, ya sea para la lectura de documentos o crearlos o fusionarlos, extracción de imágenes e inserción en documentos.

3.5.6. PyPDF2

Esta librería de Python[10] nos permite el manejo de documentación PDF, ya sea para la lectura de documentos o crearlos o fusionarlos, además de la extracción de los metadatos.

3.5.7. Langdetect

Esta librería de Python[11] sirve para detectar hasta 55 idiomas diferentes de Google, dándonos una gran gama de detección del lenguaje de un documento.

3.5.8. gtts

Esta librería de Python[12] sirve para transformar archivos de texto en audios. Un ejemplo ilustrativo de su sintaxis es el mostrado en la Figura 3.7 donde se realiza una transformación de un audio en inglés a texto.

3.5.9. requests

Esta librería de Python[13] es una herramienta para realizar peticiones HTTP. Facilita la creación, envío y manejo de solicitudes a servidores web y otros servicios que

[8]Enlace a la librería:`https://pypi.org/project/fpdf/`
[9]Enlace a la librería:`https://products.documentprocessing.com/es/merger/python/pymupdf/`
[10]Enlace a la librería:`https://pypi.org/project/PyPDF2/`
[11]Enlace a la librería:`https://pypi.org/project/langdetect/`
[12]Enlace a la librería:`https://pypi.org/project/gTTS/`
[13]Enlace a la librería:`https://pypi.org/project/requests/`

```python
from gtts import gTTS
from io import BytesIO
import pygame

def speak(text):
    mp3_fp = BytesIO()
    tts = gTTS(text, lang='en')
    tts.write_to_fp(mp3_fp)
    # mp3_fp.seek(0)
    # return this to the client side.
    return mp3_fp

pygame.init()
pygame.mixer.init()
sound = speak("This was cool")
sound.seek(0)
pygame.mixer.music.load(sound, "mp3")
pygame.mixer.music.play()
```

Figura 3.7: Ejemplo de código en Python utilizando la librería *gtts* para transformar un audio en inglés en texto.

utilicen el protocolo HTTP.

3.6. ffmpeg

Es un programa gratuito[14], que puede trabajar con archivos de video y audio, para grabarlos, generarlos, mejorarlos, convertirlos en otro formato y hacer streaming.

3.7. Visual Studio Code

Visual Studio Code (VS Code) [14] es un editor de código fuente gratuito de *Microsoft*, disponibles para todos los sistemas operativos (*Windows*, *Linux* y *Apple*), utilizado para el desarrollo de aplicaciones debido a su capacidad para soportar una amplia gama de lenguajes de programación. La Figura 3.8 muestra cómo se puede trabajar en esta herramienta con múltiples lenguajes.

3.8. LaTeX

LaTeX[15] (logo mostrado en la Figura 3.9) es un sistema de composición de textos que esta orientado a la creación de documentos escritos de alta calidad tipográfica. Normalmente se usa de forma regular en la generación de TFG, TFM, composición de artículos de investigación, tesis doctorales y libros de carácter científico.

[14]Enlace a la web de ffmpeg:`https://ffmpeg.org/download.html`
[15]Enlace a la herramienta:`https://www.latex-project.org/`

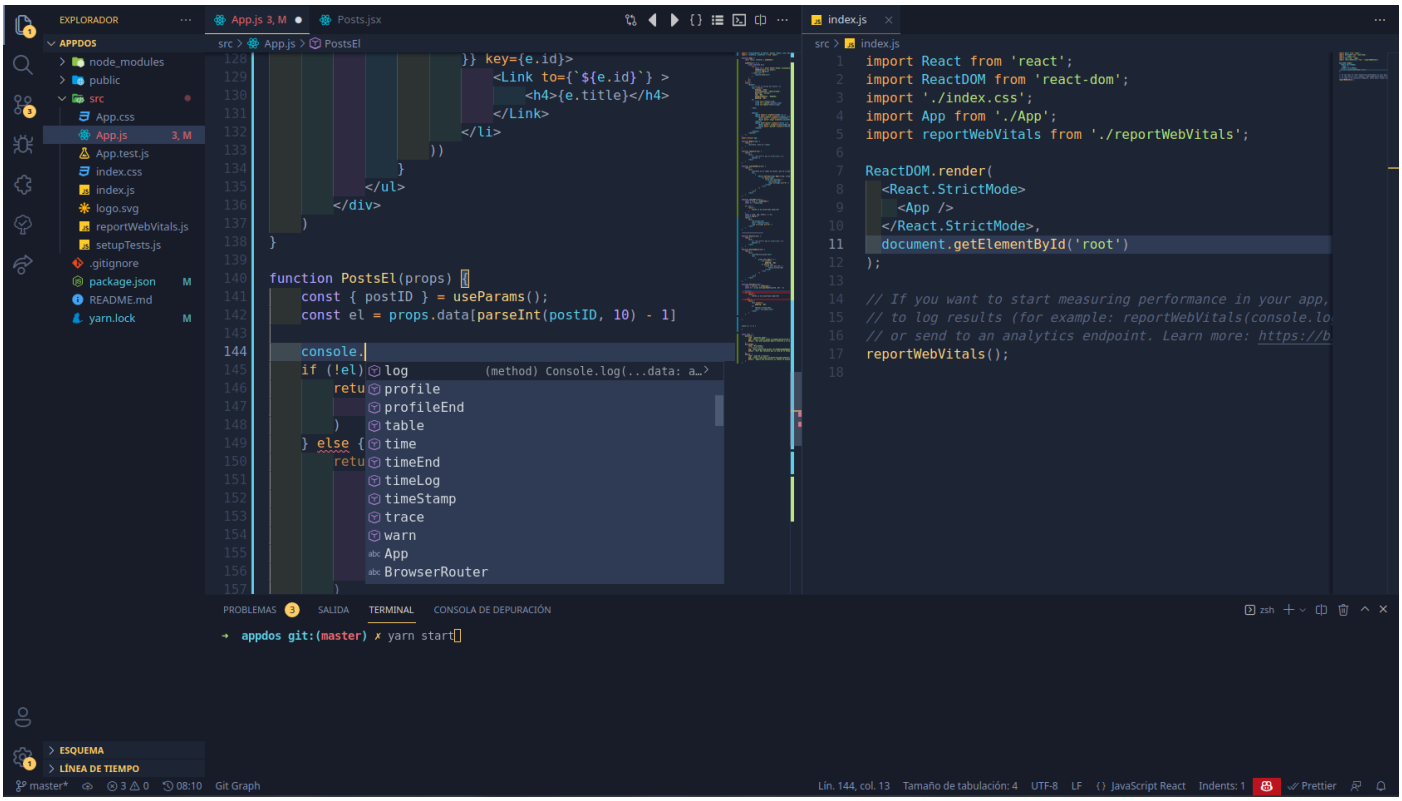


Figura 3.8: Pantalla de trabajo de Visal Studio Code.

Figura 3.9: Logo del proyecto LaTeX.

3.9. Overleaf

Overleaf[16] es una plataforma de desarrollo de documentación en LaTeX online, que permite además de compartir y modificar los documentos en tiempo real con varias personas, realizar control de versiones y exportar dichos documentos a diferentes formatos. Un ejemplo de este entorno de trabajo es el mostrado en la Figura 3.10 donde se muestra a la izquierda el editor de texto donde podemos escribir nuestro código y a la derecha cómo se ve ese código en el documento.

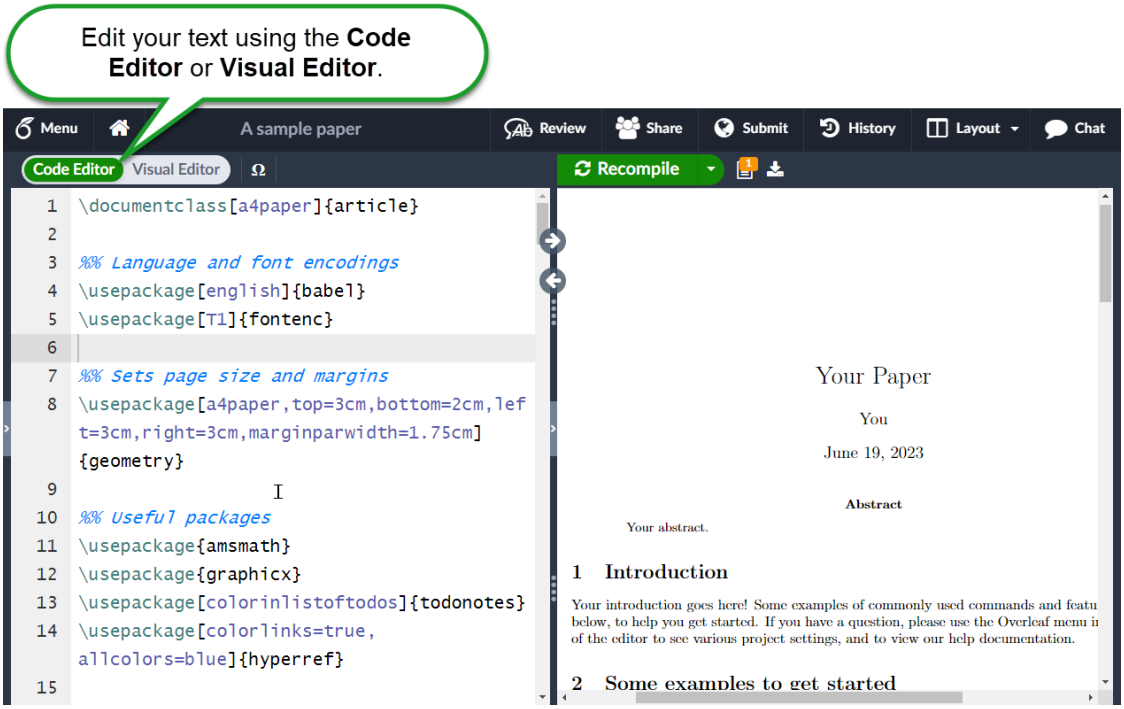


Figura 3.10: Entorno de trabajo de Overleaf.

[16]Enlace a la plataforma:`https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`

Capítulo 4. Diseño de solución

4.1. De relato a audio-relato

4.1.1. ¿Qué es un relato?

Un relato es una historia real o inventada, ya sea con fines religiosos, didácticos o para el disfrute del lector. Antes de la invención de la escritura estos relatos eran contados de forma oral por los distintos testigos del mismo o por sus generaciones siguientes.

La Biblia, la Ilíada (portada mostrada en la Figura 4.1) y la Odisea fueron de los primeros textos escritos de eventos que hasta relativamente poco eran narrados de forma oral y eran entendidos de forma literal. Estos documentos narran distintos contextos tanto ficticios como históricos que definen a la humanidad tal y como la conocemos ahora.

4.1.2. El poder de la palabra

Las palabras muchas veces tienen un poder y peso propio, pueden animar a las personas tristes, encender el fuego del amor o hacer volar nuestra imaginación, ya seas escritas o dichas de viva voz. Un relato sirve para enseñar, animar o transportarnos a un lugar totalmente diferente con nuestra imaginación, dejando atrás nuestros problemas, tristezas y la monotonía que pueden aprisionar la mente de las personas.

4.2. El problema de los relatos escritos

4.2.1. El problema de no poder ver

Según ha reportado[1] el Colegio de Ópticos Optometristas de Galicia alrededor un 2.14 % de la población en España sufre algún tipo de baja visión que le impide hacer su vida normal. Este dato es corroborado por el artículo [15] donde se indican las múltiples causas de la baja visión y cómo son de esenciales sus herramientas para poder detectarlas a tiempo. Un ejemplo de herramienta para su detección es la rejilla de Amsler donde se puede detectar la pérdida de la visión de un paciente sólo indicando cómo ve esos cuadros y diagnosticando DMAE(degeneración macular asociada a la edad). Este ejemplo está mostrado en la Figura 4.2.

Según el informe de OMS[2] se estima que aproximadamente 1300 millones de per-

[1]Enlace al boletín de 2024: `https://opticosoptometristasdegalicia.org/wp-content/uploads/2024/05/Boletin-de-Galicia.-Mayo-2024.pdf`

[2]Dirección del informe de la OMS del 2020:`https://iris.who.int/bitstream/handle/10665/331423/`

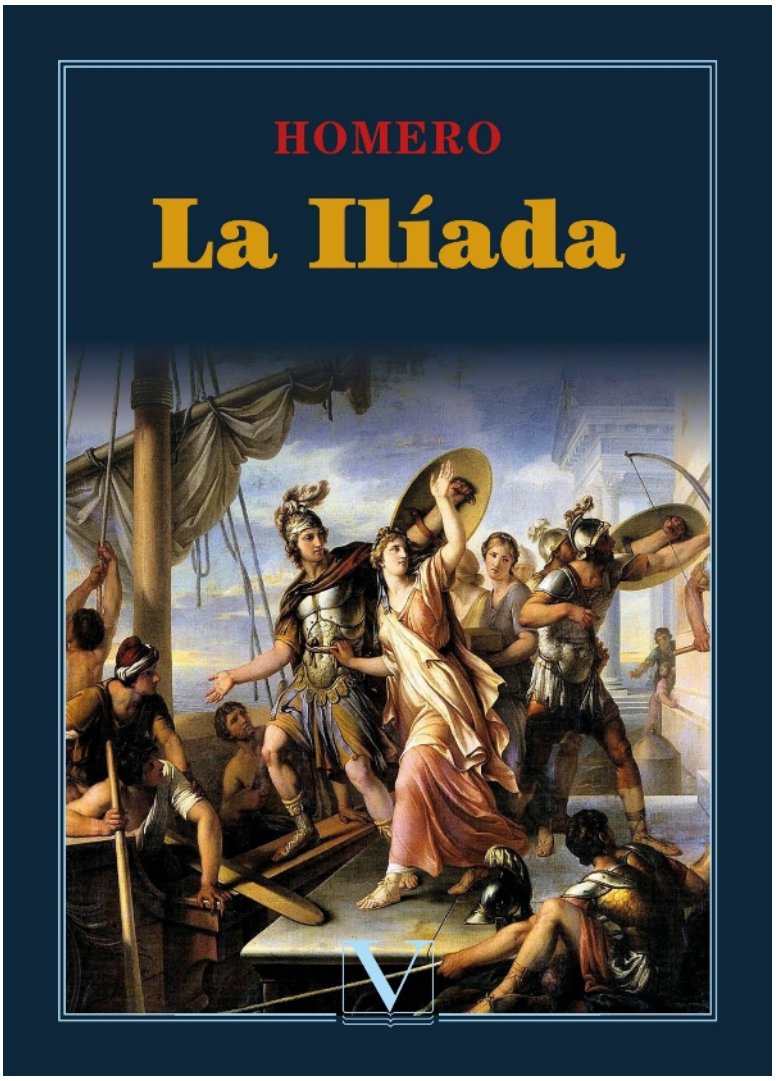


Figura 4.1: Portada de la Ilíada

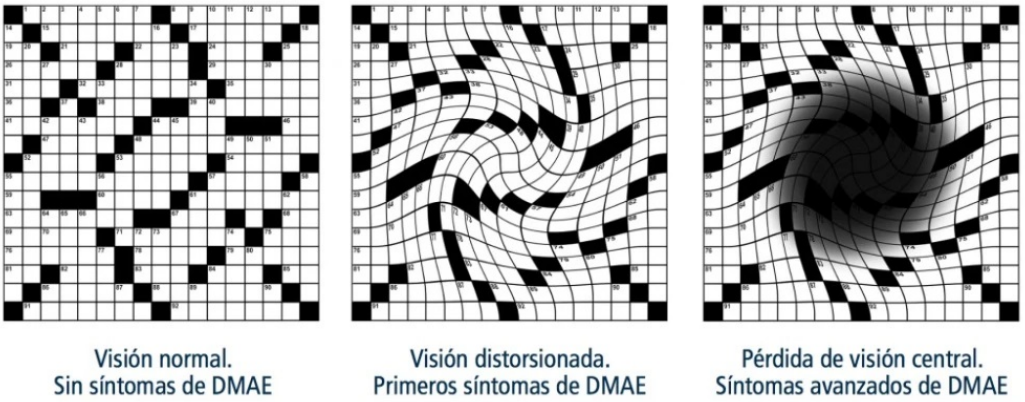


Figura 4.2: Test de perdida de visión

sonas viven con alguna forma de deficiencia visual. Esto hace que muchas de estas personas no tengan acceso a los libros o textos, ya que pese a que hay muchas obras en braille, muy pocas personas con deficiencias visual o ceguera saben dicho método

9789240000346-spa.pdf

de lectura.

4.2.2. El problema del dinero y del tiempo

Actualmente, cada vez hay más audiolibros creados por las editoriales para que personas con problemas de visuales y público en general puedan disfrutar de un libro en forma sonora. Pero esto conlleva un desembolso monetario elevado, debido a la aplicación de efectos sonoros, personas expertas en doblaje y maquetación del audiolibro.

Además de todos los procesos por los que pasa el audiolibro hasta que sale a la venta, también esta el problema de su alto precio a veces que es casi tan caro como la edición física impresa para el lector, pese a no tener gastos de edición, impresión y distribución, además de la dificultad de poder encontrarlo en algunas plataformas con streaming de audiolibros.

4.3. ¿Como enfrentarse al problema?

Lo primero que vemos son los problemas que se generan al elegir una página web desde la que se extrae el texto seleccionado, hasta transformarlo en audio y con una calidad optima para que parezca casi lenguaje natural, sin tener la sensación de que el relato es leído por una voz sintética.

Para ello, se ha creado una arquitectura o programa capaz de solucionar cada uno de los inconvenientes paso a paso, hasta poder generar un audio relato de la página que hemos usado de prueba de forma rápida y eficiente para los usuarios.

Las dificultades que hemos encontrado al crear esta solución fueron las siguientes:

- Extracción del Texto de la Web.
- Generar un documento PDF con el texto extraído.
- Traducir dicho texto a un idioma totalmente diferente.
- Generar el audio-relato.
- Mejorar la calidad de audio del audio-relato.

4.4. Fases de resolución

4.4.1. Fase de investigación o estudio

Durante esta fase se estudió tanto el problema como la solución. Además se valoraron diferentes herramientas y lenguajes para su uso en la viabilidad del proyecto a desarrollar.

Se decidió utilizar Python debido a su fácil manejo y la gran cantidad de librerías de uso especifico, además de ser un lenguaje muy versátil. Esto permitió un rápido aprendizaje del lenguaje y un uso óptimo de las diferentes aplicaciones del ámbito local como Jutipyter o Visual Studio Code en el desarrollo de la futura aplicación del proyecto.

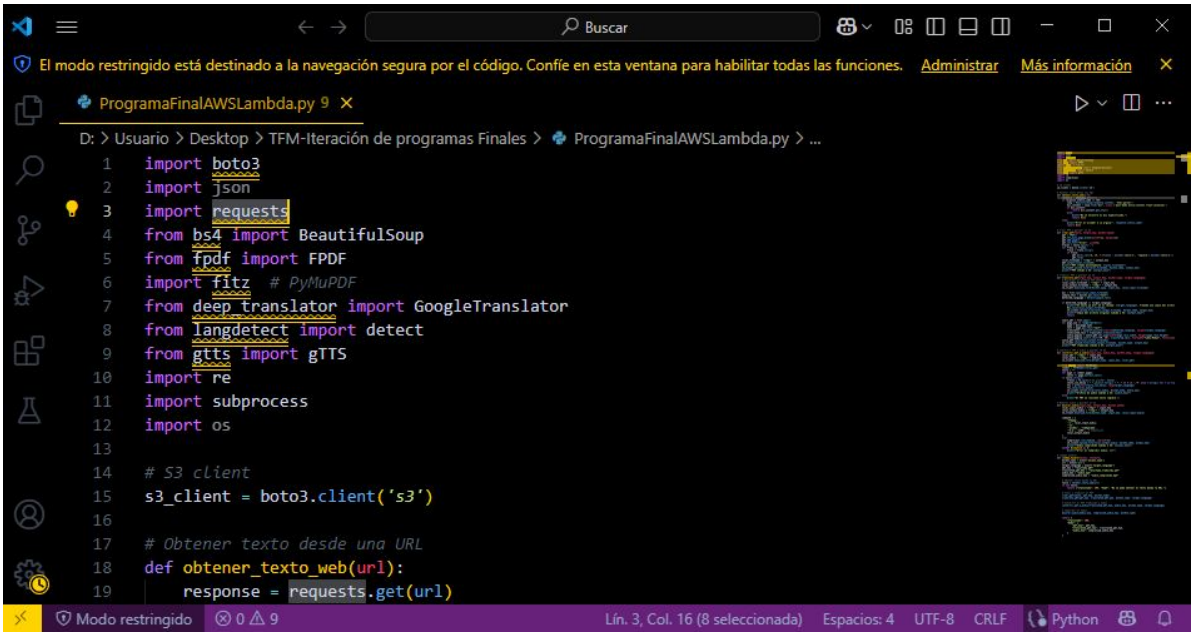


Figura 4.3: Pantalla de desarrollo de Visual Studio Code

A continuación se realizó una investigación exhaustiva en distintos artículos y congresos, comprobando las posibles soluciones previas y sus posibles usos en el ámbito del problema que intentamos resolver. Una muestra de esta búsqueda es la realizada en la Figura 4.4 donde se ve cómo se ha realizado la búsqueda de artículos y congresos relacionados con este trabajo en la plataforma Google Scholar.

4.4.2. Fase de desarrollo local

Tras haber realizado la investigación previa y asimilar todos los conocimientos necesarios para el desarrollo utilizando las diferentes librerías en Python. Se empezó con el desarrollo de los módulos y pruebas en local(ordenador de sobremesa).

Extracción del Texto y generación del PDF

El primer paso fue crear el módulo que recibía la url de donde se va a extraer el texto, tras acceder a la web utilizando la librería requests y a la posición exacta donde se encuentra el texto que queremos, se procede a la extracción de dicha información con la librería BeautifulSoup y se guarda para su utilización. Si la posición que hemos puesto es errónea o no existe, dará un aviso de error y también en caso de que no se pueda acceder desde requests a la web.

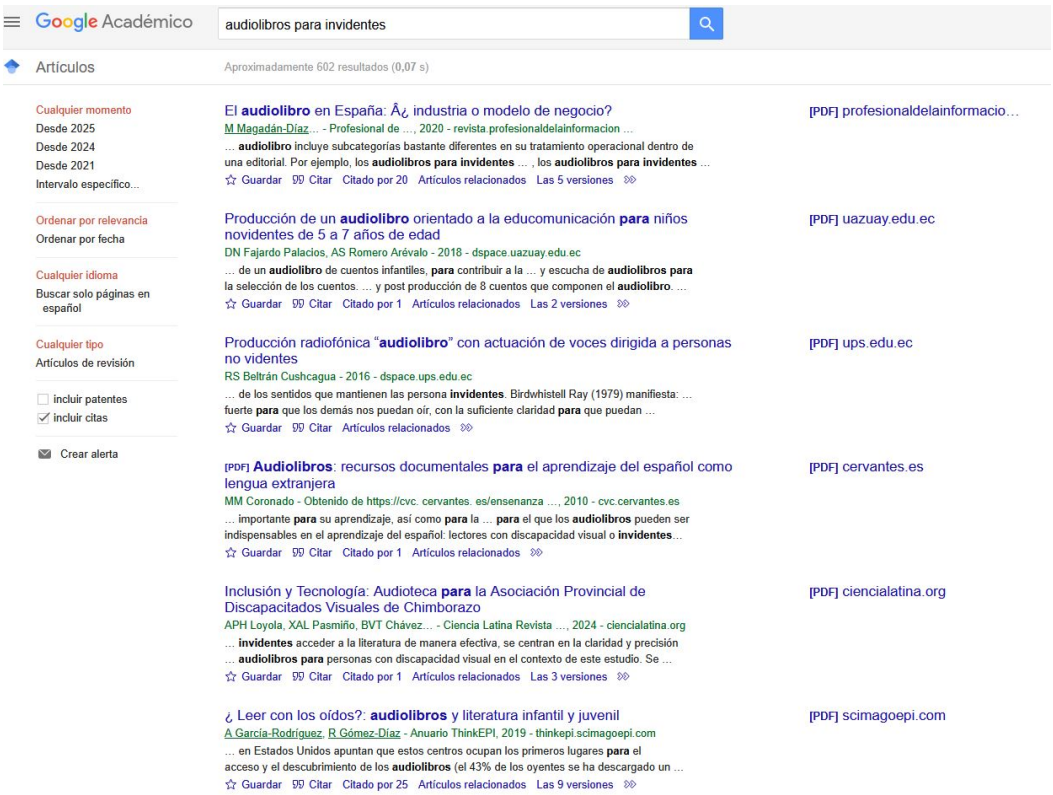


Figura 4.4: Imagen de la búsqueda realizada para el estado del arte.

A continuación se crea un documento vacío con la librería FPDF, se procede a elegir un tamaño de letra y un formato de texto, que hemos dejado predefinido, para volcar el texto extraído en el documento PDF vacío, listo para poder ser leído o utilizado por los siguientes módulos.

Con la finalización de este módulo, se consiguió tener un documento listo para su uso con el relato de la página web y así no tener que estar extrayendo dicho texto de forma constante cada vez que queramos iterar sobre él.

Traducción a un idioma especifico

El siguiente paso fue enviar el documento PDF y el idioma en el que queríamos por defecto el contenido del documento abriéndolo con la librería fitz para trabajar con él. Lo primero que decidimos hacer es que comprobara que no hiciera las iteraciones innecesarias, si el idioma del documento es el mismo del idioma deseado. Por ello se extrae el texto del documento y se detecta su idioma con la librearía Langdetect, pudiendo hacer así la comparación, terminará el proceso de forma directa devolviendo una copia del documento sin modificar e informando por pantalla.

Por otro lado, si el idioma es diferente el módulo comprobará los idiomas disponibles para su traducción para poder realizarla posteriormente en el texto.

Una vez comprobado que el idioma está entre los idiomas soportados, traducirá el

texto del documento usando la librería GoogleTranslator y lo plasmará en un nuevo documento PDF que irá uniendo las diferentes hojas con la librería fitz. Se decidió hacerlo de esta manera, ya que al intentar sobrescribir el texto del documento original, provocaba el problema de que no reconociera el texto traducido al pasarlo a audio, reproduciendo el texto original en vez del traducido.

Generación de audio

En este paso se utilizó el documento del primer módulo si no queríamos un audio en otro idioma diferente del original y también probamos utilizando un documento del segundo módulo leyéndolos con la libería PyPDF2, todo ello para probar la fiabilidad de la generación del audio en diferentes idiomas.

Para evitar que hubiera problemas con el audio (voz sin pausas o que fuera demasiado rápido) se usaron los signos de puntuación como los puntos, los signos de preguntas, las exclamaciones y las comas. De modo que se diera un tono más sosegado y natural a la narración del audio.

Como resultado se obtuvo un audio en el idioma del documento PDF y que hacía pausas naturales en la narración usando para crearlo la librería gtts, pero la calidad del audio aún no era la correcta.

Mejora de audio

Con el audio ya generado, en el siguiente módulo se pasó por filtros de audio y se mejoró la entonación del mismo usando ffmpeg en Python. Procesando el audio y generando un espectrograma con dicha mejora del audio, aumentando la calidad del audio significativamente, pero tardando un tiempo elevado en reprocesar el audio.

Envío de email

En este módulo final, se envía desde un email especifico que hemos creado para este módulo y dado una clave de validación de *Google* para poder hacer el envío de correos gracias a las librerías email. Incluiremos un email del destinatario y un mensaje, además de adjuntar el audio, además de el documento en español y el documento traducido en caso de encontrarse todo en la misma carpeta.

Al final este módulo se decidió dejarlo solo en el modo off-line o local y no se incluyó en la solución de AWS, por ser un módulo innecesario al tener contenedores S3 para la entrega y salida de datos

4.4.3. Fase de Desarrollo en el Servicio Web

En este punto, se tiene una aplicación operativa capaz de usarse en cualquier dispositivo local (ordenador de sobremesa o portátil), pero sin solucionar el coste de

tiempo y económico que conlleva la aplicación el local, además de la cantidad de generación de audios posibles de generar y procesar.

En ese mismo instante, se determinó utilizar *Amazon Web Service* como servicio web debido a su potencia, robustez y su capacidad de gestionar diferentes hilos de ejecución a muy bajo coste para los usuarios.

Tras un estudio intensivo y aprendizaje de cómo utilizar *Amazon Web Service*, registro de la plataforma, familiarización con los diferentes módulos, pruebas de uso y análisis de costes, se procedió a la migración de los módulos Python a AWS, añadiendo las instrucciones Lambda de Amazon para su correcto funcionamiento en el servicio web.

Pese a descartar el servicio de envío de emails en la migración, se obtuvo un resultado muy superior a la versión en local y al hacerlo de modo modular tanto en coste como en tiempo.

Además se encontró una nueva funcionalidad de cara al ámbito estudiantil de la Universidad Complutense, esta nueva funcionalidad se basaría en descartar o no usar el primer módulo y utilizar los restantes para la generación de documentos PDF traducidos a otros idioma y audios de los mismos para alumnos de extranjeros que tienen poco o nulo dominio del idioma español.

Capítulo 5. Arquitectura e implementación

5.1. Arquitectura de Proyecto en Amazon Web Service

En esta sección se va a mostrar y explicar la arquitectura desarrollada utilizando *Amazon Web Service*. Vamos a desglosar el funcionamiento paso a paso.

Los esquemas que vamos a desglosar son los mostrados en las Figuras 5.1 y 5.2.

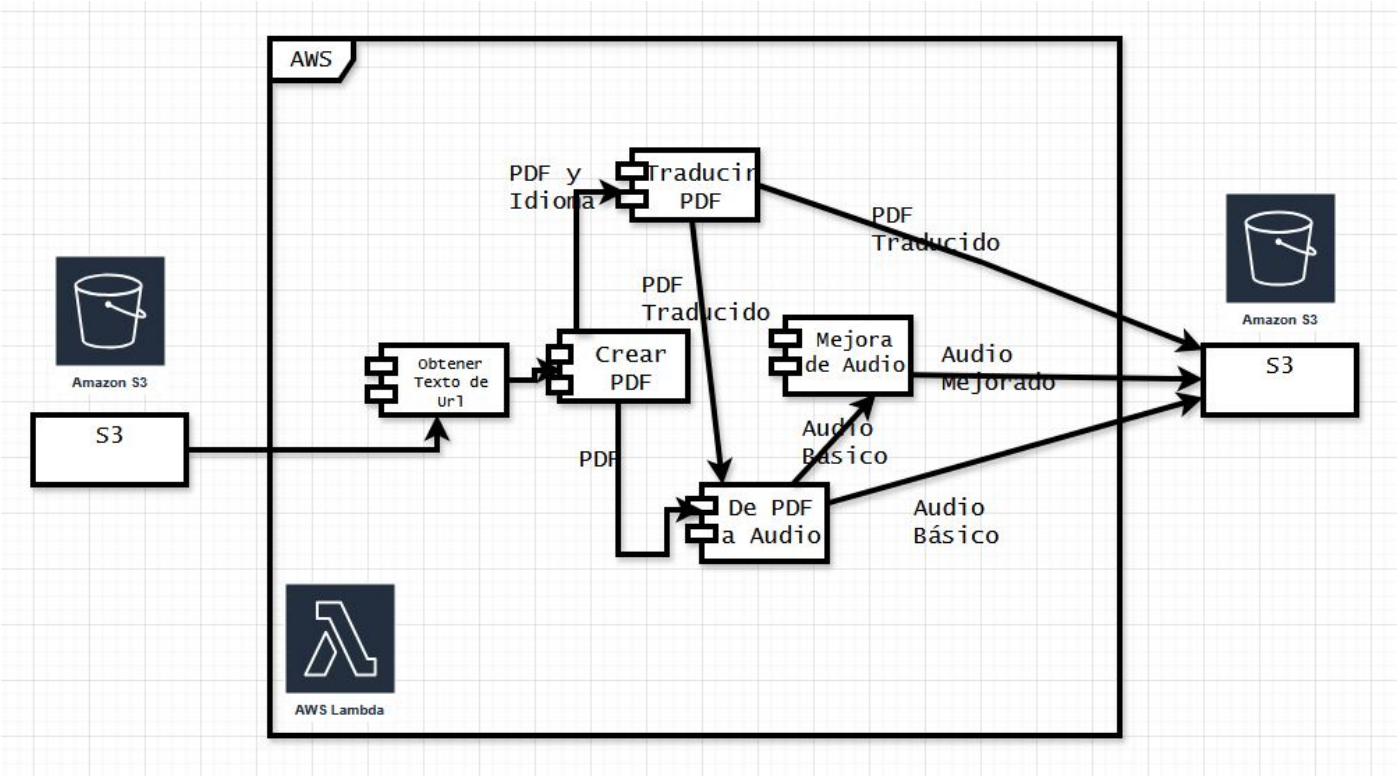


Figura 5.1: Arquitectura del proyecto en Amazon Web Service desde url

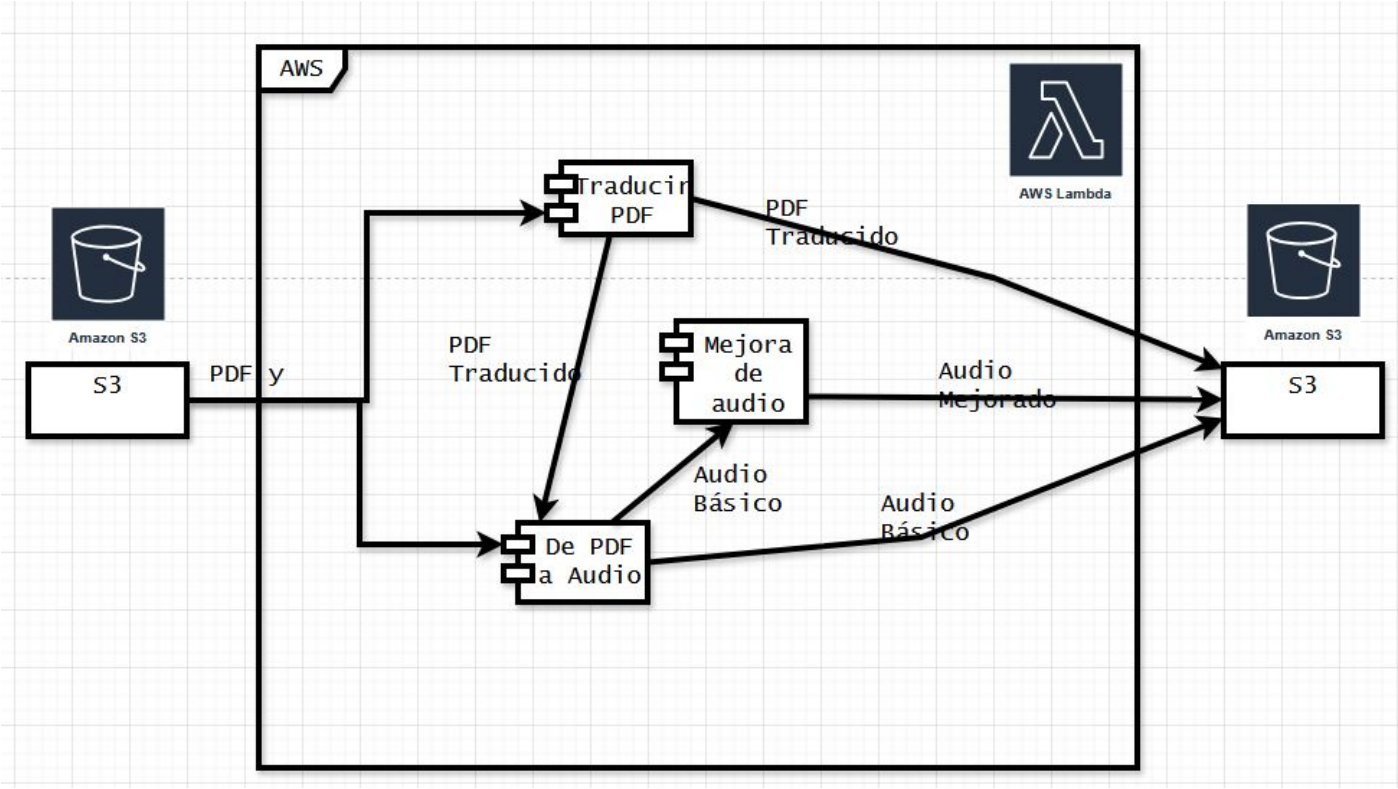


Figura 5.2: Arquitectura del proyecto en Amazon Web Service desde PDF

5.1.1. S3

Dependiendo de la estructura que usemos para crear nuestro audio, el número de contenedores S3 variará, ya que estos contenedores servirán como almacenamiento temporal de datos de entrada, salida e intermedios que necesitarán utilizar cada una de nuestras funciones Lambda y sobre los que se va a trabajar en su ejecución.

Los dos S3 fuera de la funciones AWS Lambda, servirán como como contenedores de entrada y de salida respectivamente al principio y final del proceso de ejecución de la funciones Lambda. Enviados el contenedor inicial de desde la arquitectura Url, la web y el idioma, mientras que en la arquitectura desde PDF contendrá el PDF y el idioma.

Igualmente el S3 final, que envía el resultado de todas las operaciones de AWS Lambda su contenido puede variar dependiendo de los módulos ejecutados, ya que al ser la aplicación modular no se tiene por que ejecutar cada módulo Lambda y el resultado puede variar.

5.1.2. Lambda Obtener Texto de url

Esta función Lambda (mostrada en la Figura 5.3) se activa al recibir la url (web del blog del relato especificado) del contenedor S3, una vez activada accederá a la url indicada con la librería request y la posición donde esta el texto que queremos

extraer con la librería BeautifulSoap.

El sistema comprueba si la url del blog esta operativa con la librería resquests y si la posición de código html de la url tiene texto que extraer, si no lo tiene dará un mensaje por pantalla indicando que no ha encontrado esa posición y también comprobará si la url es accesible o no.

Si todo es correcto, Beautifulsoap nos devolverá el texto, que se enviará a un S3 intermedio para el uso por el siguiente módulo.

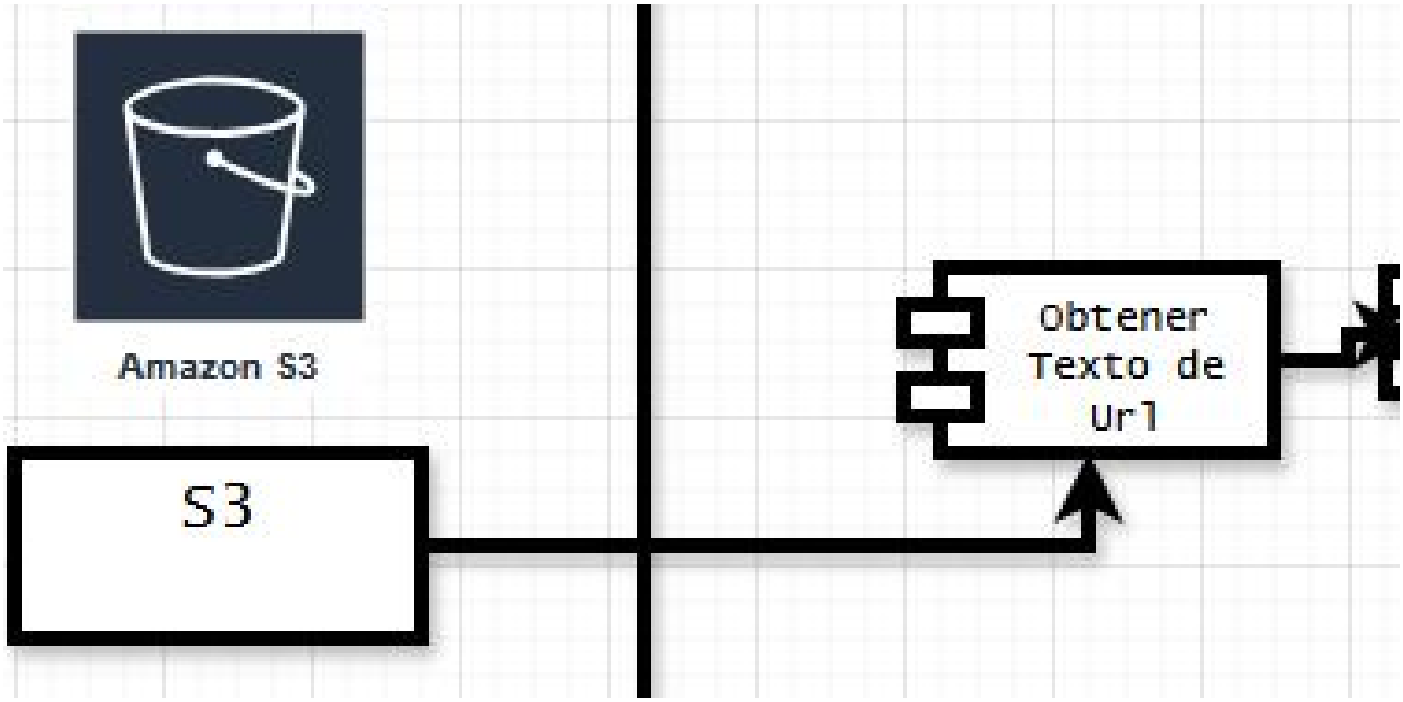


Figura 5.3: La arquitectura AWS desde URL activa el módulo Lambda Obtener Texto de url

5.1.3. Crear PDF

Esta función (mostrada en la Figura 5.4) recogerá el texto que la función Lambda ha dejado en el S3 intermedio, para extraerla y generar un PDF nuevo usando la librería FPDF donde volcar el texto.

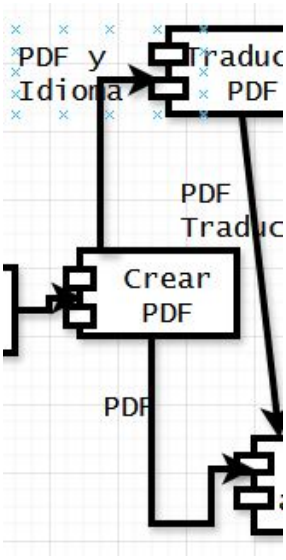


Figura 5.4: La arquitectura AWS desde URL activa el módulo Lambda Crear PDF

Configurará los márgenes del documento, el tamaño de la letra y la tipografía, para que quede correcto el documento PDF una vez volcado el texto y generará el documento PDF que será enviado al siguiente contenedor S3.

5.1.4. Traducir PDF

Esta función (mostrada en la Figura 5.5) recibe una PDF y un idioma, independiente que venga de un S3 intermedio como el la Arquitectura desde Url o desde la Arquitectura desde PDF.

El sistema comprobará el idioma introducido y si es diferente del documento PDF, para ello lo abrirá usando la liberería fitz y detectará su idioma con la librería detect. Si todo es correcto crar un nuevo documento PDF con fitz y empezará a generar una traducción, que la irá añadiendo al nuevo documento traducido en el idioma deseado.

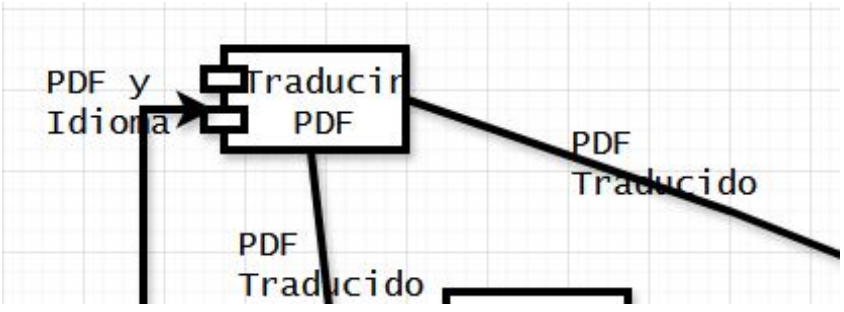


Figura 5.5: La arquitectura AWS desde URL activa el módulo Lambda Traducir PDF

Tras finalizar el proceso, enviará el nuevo documento a otro S3, ya sea a uno intermedio para ser usado por otros módulos o para devolverlo al usuario a través del S3 final. Si el idioma del documento origen es igual al deseado envía directamente

una copia al S3 y no realiza la operación de traducción.

5.1.5. De PDF a Audio

Esta función Lambda (mostrada en la Figura 5.6) va a recibir de un S3 un documento en PDF, que va a ser leído usando la librería PyPDF para ser transformado en un archivo de audio con la librería gtts.

Al formar el audio se irá generando con un delay para que el sonido de voz no salga a una velocidad excesiva y haciendo pequeñas pausas al encontrar diferentes símbolos de puntuación como comas, finales de exclamaciones y de preguntas, dando una sensación más natural y humana al audio.

Figura 5.6: La arquitectura AWS desde URL/PDF activa el módulo Lambda crear el audio

Una vez terminado el proceso de generación del audio, nos dará dos resultados diferentes, el primero que se genere un audio correcto y se envíe al S3 para la siguiente iteración o como solución final. Segundo que el documento PDF este en blanco o con un texto corrupto, por lo que nos informará el módulo que el texto es ilegible y no se podrá generar el audio.

5.1.6. Mejora de Audio

Esta función (mostrada en la Figura 5.7) recibirá del S3 intermedio un audio, para ello se va a utilizar ffmpeg que mejora enormemente tanto en la calidad y nitidez del audio.

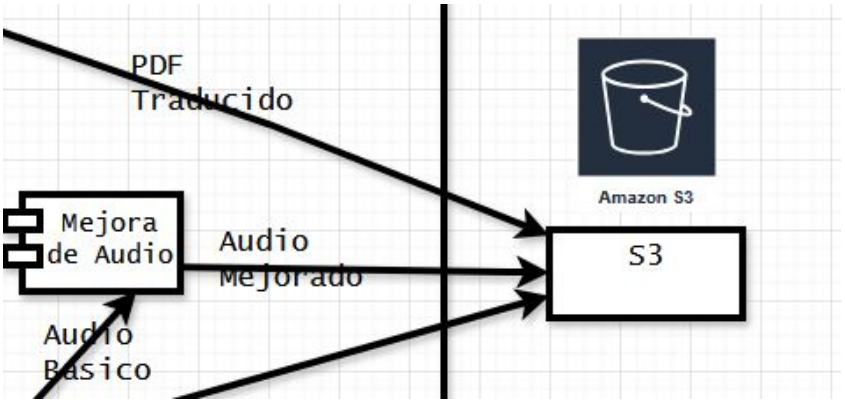


Figura 5.7: La arquitectura AWS desde URL/PDF activa el módulo Lambda Mejorar el audio

En caso que el audio llegue dañado o se produzca algún error al mejorarlo, saltará el control de errores e informará que el audio no se ha podido comprimir para generar el archivo sonoro mejorado. Si todo está correcto devolverá un audio con una calidad muy superior y nitidez al audio original que creamos a través del S3 final del sistema.

Capítulo 6. Mediciones, resultados, conclusiones y trabajo futuro

6.1. Pruebas de costes y tiempo

Tras el desarrollo, se ha procedido a realizar una serie de pruebas de precios y tiempo en diferentes entornos de pruebas, para comprobar cual sería opción más recomendable de uso por parte de una empresa o una institución. Los entornos de prueba utilizados:

- Ordenador Local.
- Google Cloud.
- Amazon Web Service.
- Microsoft Azure.
- Digital Ocean.

Para realizar los cálculos hemos utilizando la calculadora de costes de *Google Cloud*, *AWS Lambda Cost Caculator*, *Microsoft Azure* y textitDigital Ocean., en comparativa con el coste total de un ordenador local estándar de 1000 €y cuanto tiempo se tardaría en amortizarse las operaciones.

6.1.1. Google Cloud frente Ordenador local

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

Parámetros Comparativos	**Google Cloud**	**Ordenador Local**
Número de iteracciones mes	1000	1000
Espacio de almacenamiento	2GB	500GB
Tiempo en realizar una iteracción	60000ms	60000ms
Tamaño de la memoria RAM	4GB	16GB
Coste total/mes	4.35€/mes	83.33€(Amortización: 1000/12)

Cuadro 6.1: Comparativa entre Google Cloud y el ordenador local

En el Cuadro 6.1 se muestra la comparativa de coste mensual para generar 1000 audios en *Google Cloud* y en un Ordenador Local, vemos que para el una máquina de capacidad muy inferior el coste es relativamente muy bajo usando *Google Cloud* frente a un Ordenador Local. También se debe tener en cuenta que un Ordenador Local tendrá otros uso por parte de los usuarios, por lo que el desgaste y rendimiento del mismo no es igual a los servidores de Google.

6.1.2. AWS vs Ordenador local

Vamos a tomar como referencia el Ordenador Local del apartado anterior y vamos a compararlo con varios tipos de ejecuciones de AWS de Amazon, en esta comparativa hemos puesto tiempos elevados debido al volumen de módulos a ejecutar y por las transacciones intermedias de información de entrada y salida delos S3:

- En este primer apartado vamos a comprobar el coste de 100 solicitudes por hora al mes, un total de 73000 solicitudes al mes con un tiempo de ejecución de 60000 ms aproximadamente. La Figura 6.1 va a mostrar todas las cantidades calculadas.

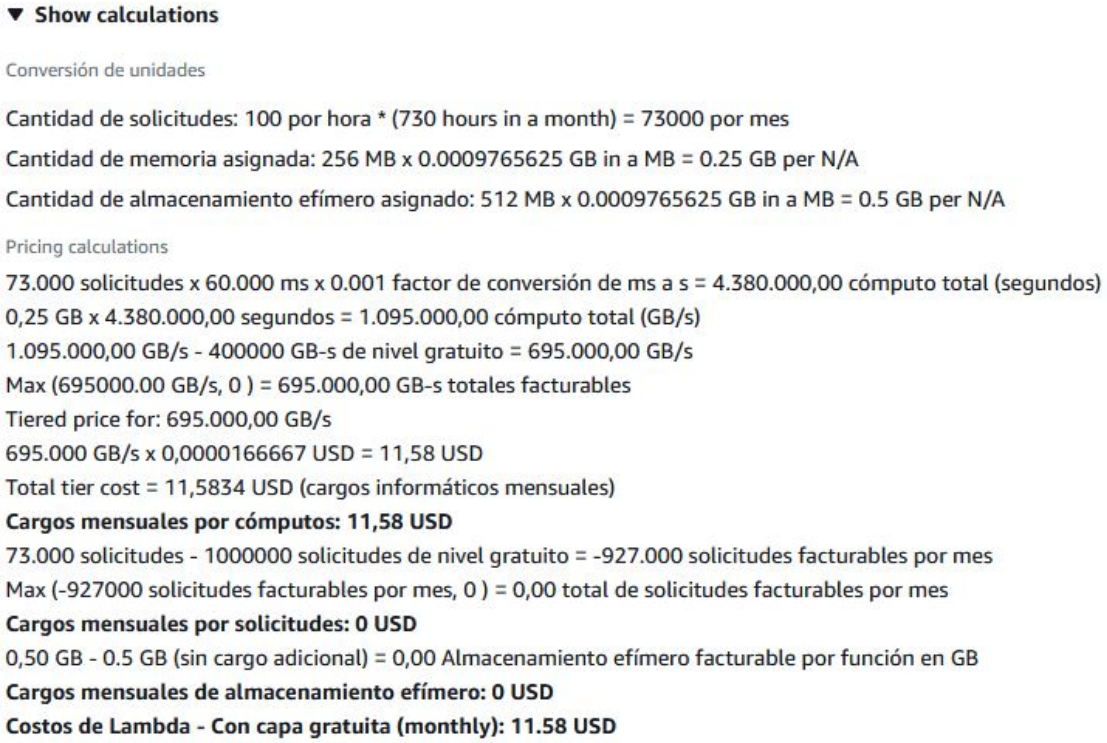


Figura 6.1: Cálculo de coste para 100 solicitudes por hora al mes.

- En este segundo apartado vamos a comprobar el coste de 100 solicitudes por hora al mes, un total de 73000 solicitudes al mes con un tiempo de ejecución de 180000 ms aproximadamente. La Figura 6.2 muestra los cálculos realizados para estas solicitudes.

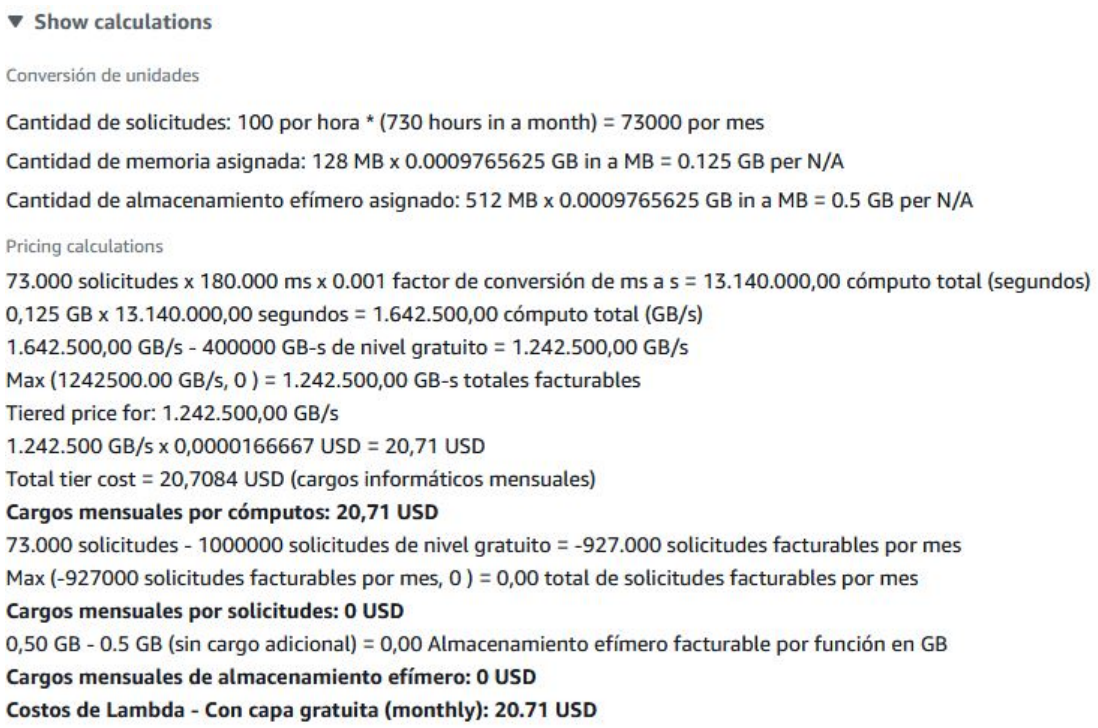


Figura 6.2: Cálculo de coste para 100 solicitudes por hora al mes.

- En este tercer apartado vamos a comprobar el coste de 1000 solicitudes por día al mes, un total de 30416.67 solicitudes al mes con un tiempo de ejecución de 60000 ms aproximadamente. La Figura 6.3 va mostrar los cálculos realizados para estas solicitudes.

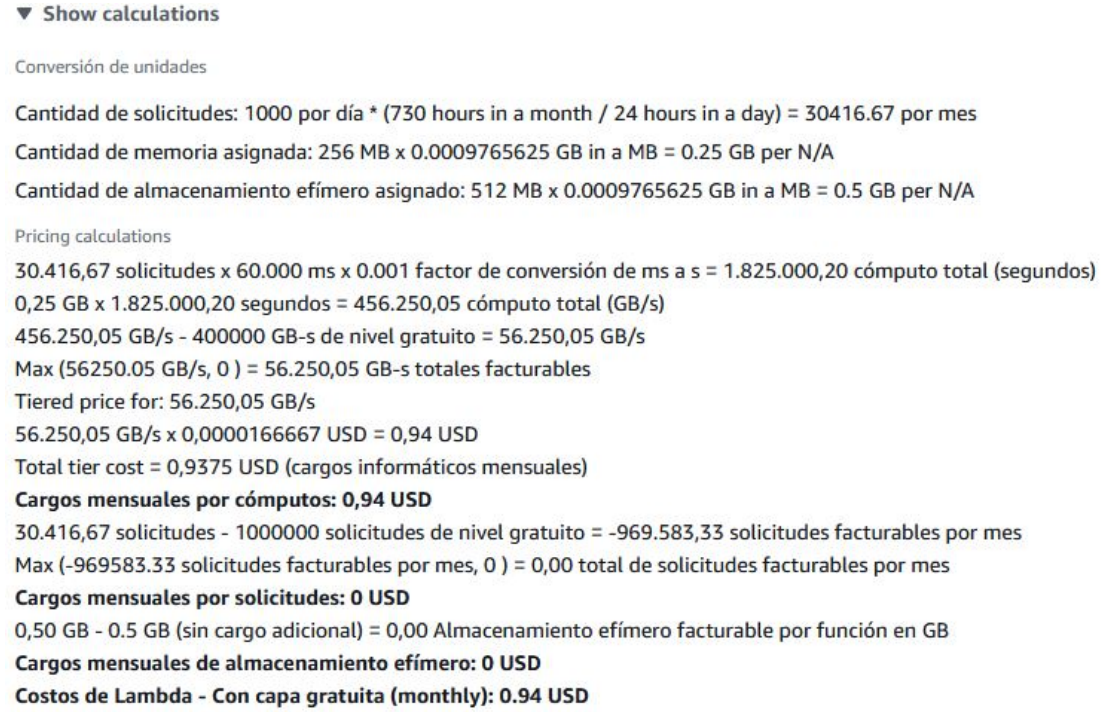


Figura 6.3: Cálculo de coste para 1000 solicitudes por día al mes.

- En este cuarto apartado vamos a comprobar el coste de 1000 solicitudes por día al mes, un total de 30416.67 solicitudes al mes con un tiempo de ejecución de 180000 ms aproximadamente. La Figura 6.4 va a mostrar los cálculos realizados para estas solicitudes.

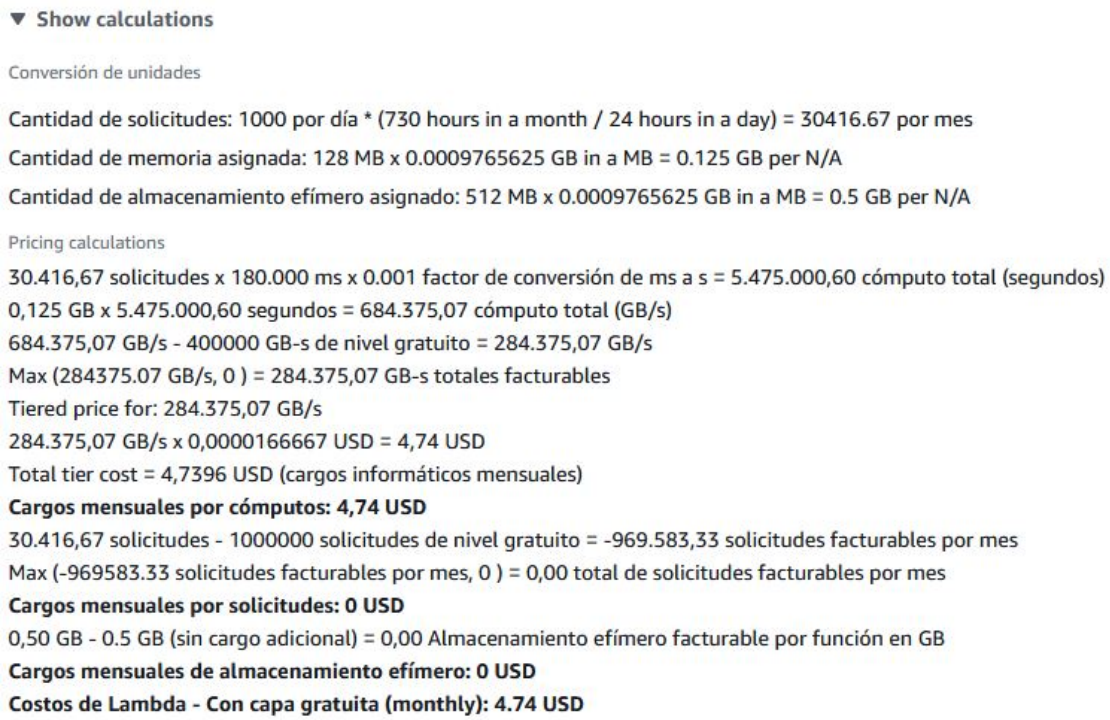


Figura 6.4: Cálculo de coste para 1000 solicitudes por día al mes.

- En este quinto apartado vamos a comprobar el coste de 25000 solicitudes al mes con un tiempo de ejecución de 60000 ms aproximadamente. La Figura 6.5 va a mostrar los cálculos realizados para estas solicitudes.

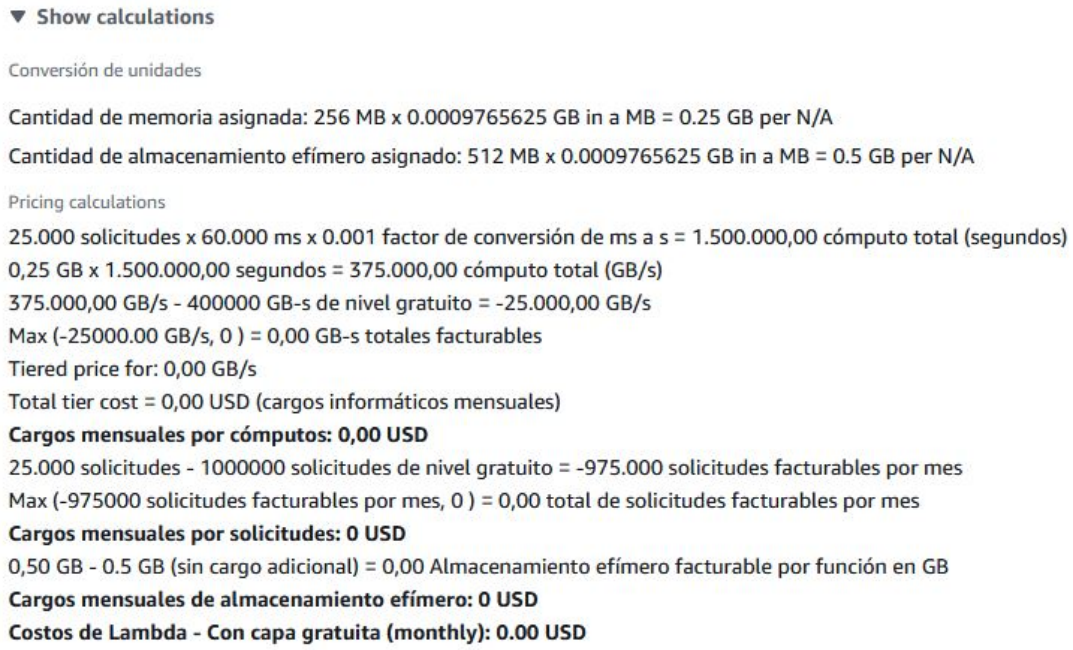


Figura 6.5: Cálculo de coste para 25000 solicitudes por mes.

- En este sexto apartado vamos a comprobar el coste de 25000 solicitudes al mes con un tiempo de ejecución de 180000 ms aproximadamente. La Figura 6.6 muestra los cálculos realizados para estas solicitudes.

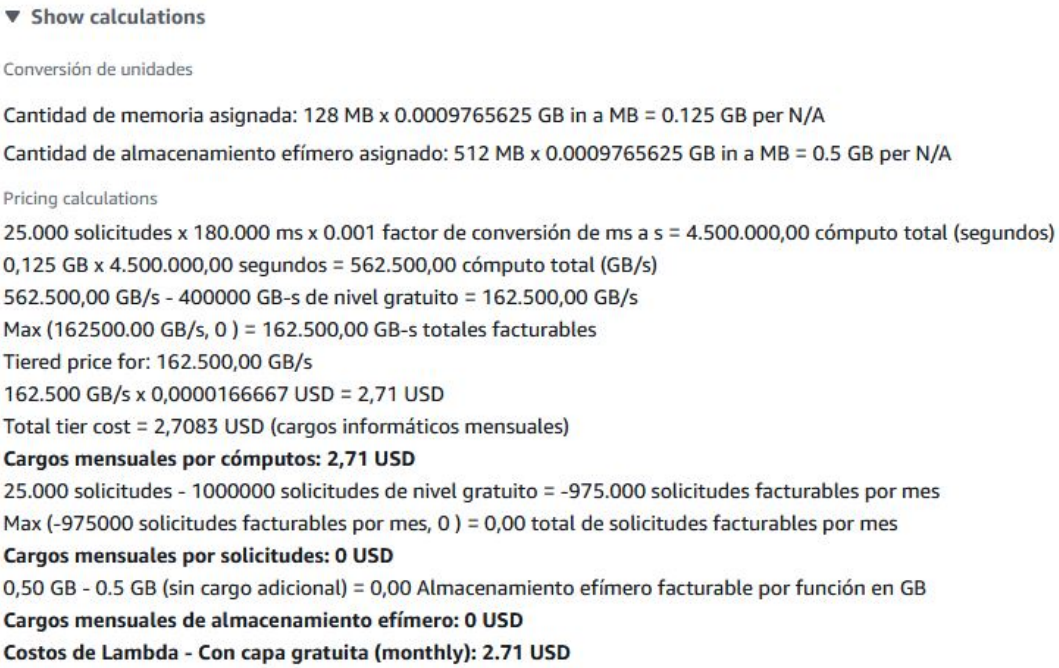


Figura 6.6: Cálculo de coste para 25000 solicitudes por mes.

6.1.3. Microsoft Azure

Vamos a tomar como referencia el Ordenador Local del apartado 6.1.1 y vamos a compararlo con varios tipos de ejecuciones de Microsoft Azure, en esta comparativa hemos puesto tiempos elevados debido al volumen de módulos a ejecutar y con diferentes cantidades de ejecuciones mensuales.

- En este primer apartado vamos a comprobar el coste de 73000 solicitudes al mes con un tiempo de ejecución de 60000 ms aproximadamente. La Figura 6.7 va a mostrar todas las cantidades calculadas.

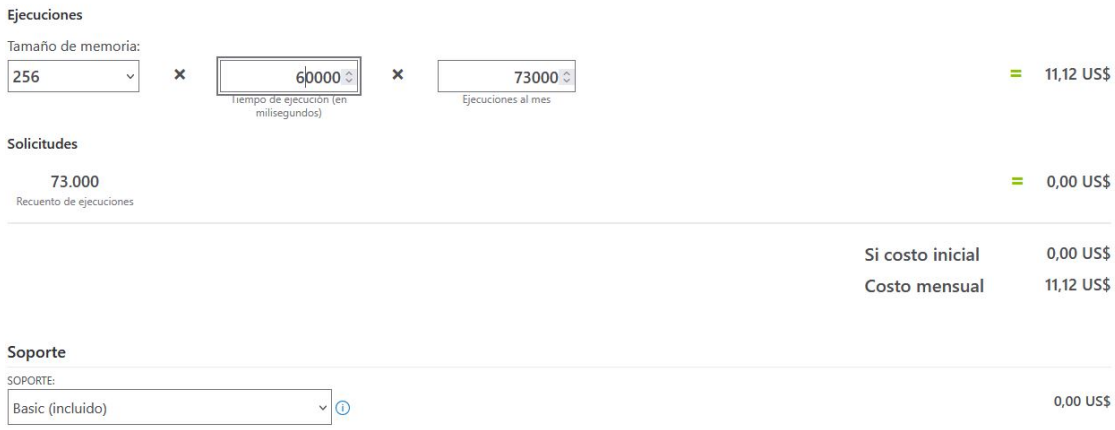


Figura 6.7: Cálculo de coste para 73000 solicitudes al mes con un tiempo de 60000ms.

- En este segundo apartado vamos a comprobar el coste de 73000 solicitudes al mes con un tiempo de ejecución de 180000 ms aproximadamente. La Figura 6.8 muestra los cálculos realizados para estas solicitudes.

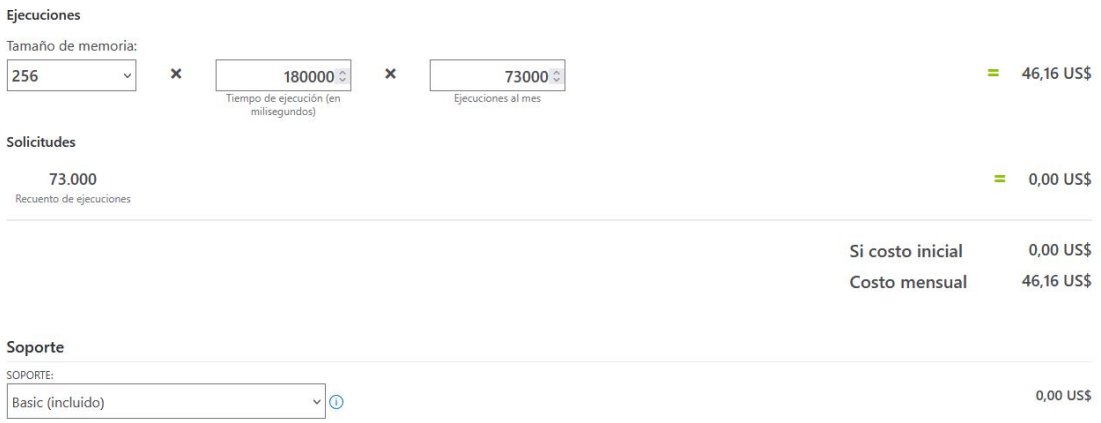


Figura 6.8: Cálculo de coste para 73000 solicitudes al mes con un tiempo de 180000ms.

- En este tercer apartado vamos a comprobar el coste de 30417 solicitudes al mes con un tiempo de ejecución de 60000 ms aproximadamente. La Figura 6.9 va mostrar los cálculos realizados para estas solicitudes.

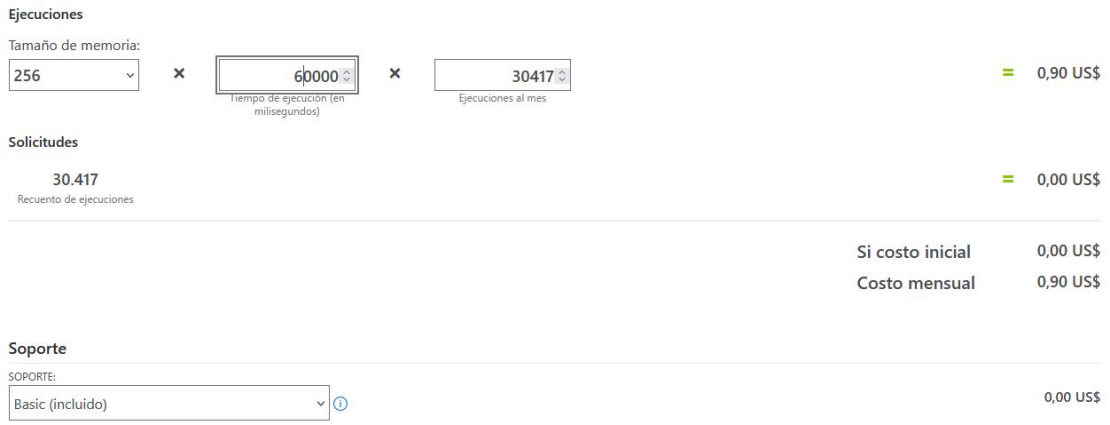


Figura 6.9: Cálculo de coste para 30417 solicitudes al mes con un tiempo de 60000ms.

- En este cuarto apartado vamos a comprobar el coste de 30417 solicitudes al mes con un tiempo de ejecución de 180000 ms aproximadamente. La Figura 6.10 va a mostrar los cálculos realizados para estas solicitudes.

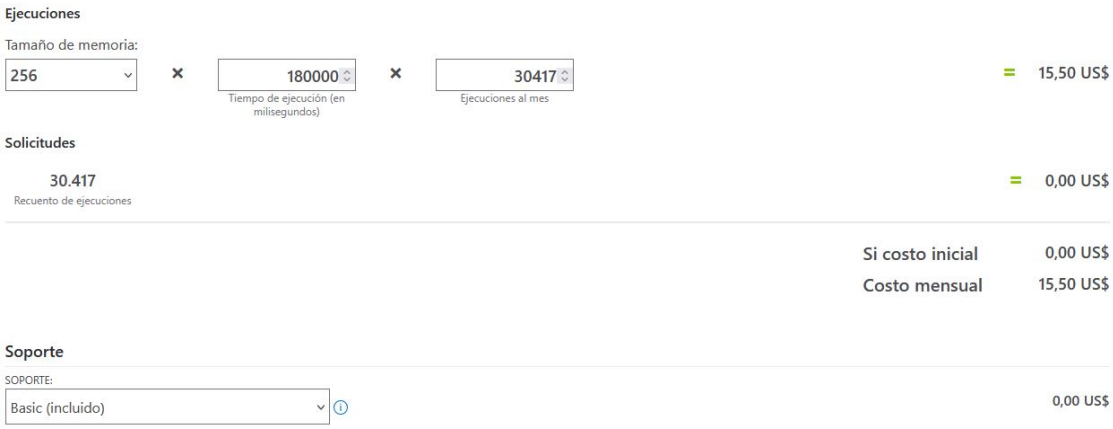


Figura 6.10: Cálculo de coste para 30417 solicitudes al mes con un tiempo de 180000ms.

- En este quinto apartado vamos a comprobar el coste de 25000 solicitudes al mes con un tiempo de ejecución de 60000 ms aproximadamente. La Figura 6.11 va a mostrar los cálculos realizados para estas solicitudes.

Figura 6.11: Cálculo de coste para 25000 solicitudes al mes con un tiempo de 60000ms.

- En este sexto apartado vamos a comprobar el coste de 25000 solicitudes al mes con un tiempo de ejecución de 180000 ms aproximadamente. La Figura 6.12 muestra los cálculos realizados para estas solicitudes.

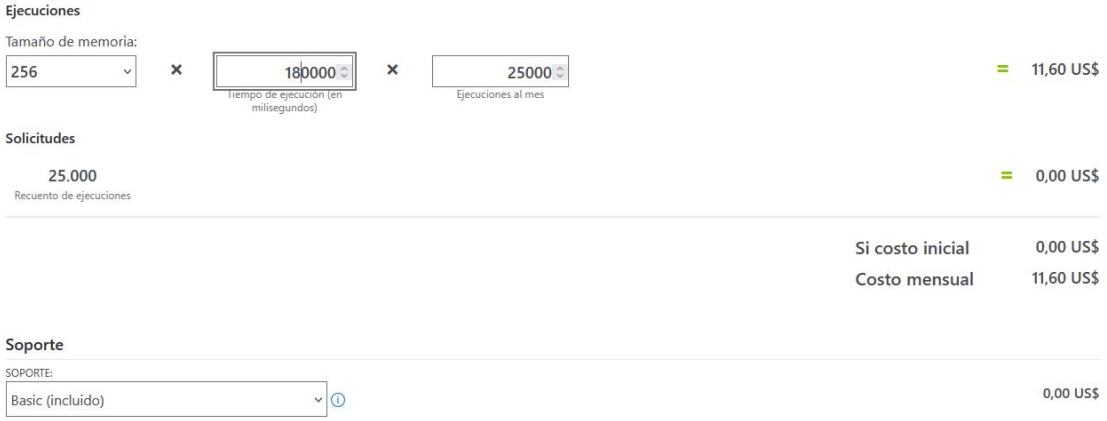


Figura 6.12: Calculo de coste para 25000 solicitudes al con un tiempo de 180000ms.

6.1.4. Digital Ocean

Vamos a tomar como referencia el Ordenador Local del apartado 6.1.1 y vamos a compararlo con varios tipos de ejecuciones de Digital Ocean, en esta comparativa hemos puesto tiempos elevados debido al volumen de módulos a ejecutar y diferentes cantidades de operaciones de ejecución al mes.

- En este primer apartado vamos a comprobar el coste de 73000 solicitudes al mes. La Figura 6.13 va a mostrar todas las cantidades calculadas.

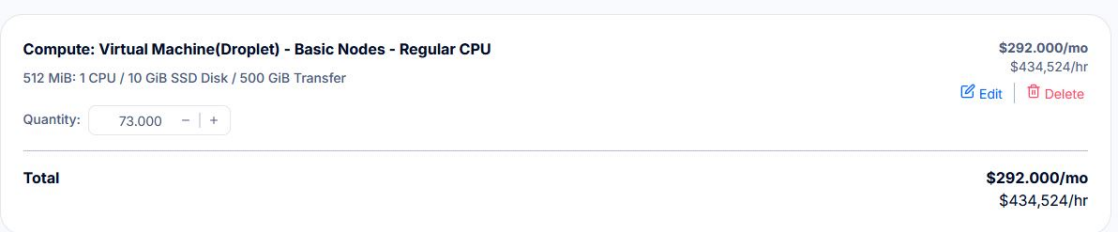


Figura 6.13: Cálculo de coste para 73000 solicitudes al mes.

- En este segundo apartado vamos a comprobar el coste de 30417 solicitudes al mes. La Figura 6.8 muestra los cálculos realizados para estas solicitudes.

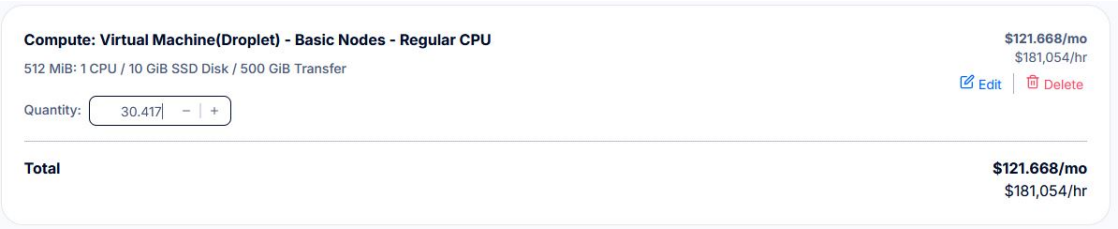


Figura 6.14: Cálculo de coste para 30417 solicitudes al mes.

- En este tercer apartado vamos a comprobar el coste de 25000 solicitudes al mes. La Figura 6.15 va mostrar los cálculos realizados para estas solicitudes.

Figura 6.15: Cálculo de coste para 25000 solicitudes al mes.

6.2. Comparativa de resultados final

Tras la diferentes mediciones tanto en local, como con las diferentes plataformas de ejecución en la nube, comprobamos que los precios de funcionamiento en Local, Google Cloud y Digital Ocean son bastante elevados, mientras que AWS de Amazon y Microsoft Azure están en unos margenes de coste y ejecución similares o idénticos en ejecuciones de bajo tiempo.

Pero se ve que en la mediciones con un tiempo elevado de ejecución, el claro ganador es AWS de Amazon, frente Microsoft Azure, debido que en este último los coste se disparan en ejecuciones de tiempos elevados.

6.3. Conclusiones

Este proyecto tuvo como objetivo el generar un audio de alta calidad de cualquier relato extraído de mi blog de relatos[1] para personas con problemas de visibilidad o totalmente ciegas.

Para ello se han revisado trabajos sobre el uso de audiolibros, tanto como herramienta de ayuda para personas invidentes o con baja visibilidad, como también como uso pedagógico en personas en formación académica. Demostrando que los audiolibros son un gran aliado a la hora de romper las barreras que impone una discapacidad visual o problemas del aprendizaje como dislexia o TDAH.

La metodología desplegada ha sido por iteraciones de desarrollo de forma paulatina, empezando con los módulos de forma local en Pyhton 3.8 y los módulos en AWS de forma posterior con la que generar tanto una versión final unificada, como versiones independientes de cada módulo Lambda.

Se utilizó Pyhton por su gran variedad de librerías y su fácil aprendizaje, también se optó por AWS de Amazon sobre Google Cloud, por su potencia de computación, rapidez y bajos costes frente al sistema de Google.

El desarrollo comenzó con los módulos iniciales de sacar un texto de la dirección de uno de los relatos y transformarlo en un PDF legible. A contaminación se decidió generar el módulo de traducción, pese a que la web con la que trabajamos sus relatos están en español, se sopesó poder traducir el relato a varios idiomas entre ellos francés, ingles, alemán e italiano. Una vez ya desarrollado este módulo se optó por crear el generador de audio, leyendo el texto independientemente del idioma y haciendo una pausa de lectura en ciertos signos de puntuación o expresión, para darle un carácter más humano y menos artificial.

Pese a las mejoras dispuestas en la generación del audio, se vio que la calidad obtenida distaba mucho de la deseada, por lo que se decidió crear un nuevo módulo de mejora de audio utilizando la tecnología ffmpeg, limpiando de estática residual y mejorando el sonido de la voz del audio. Posteriormente se implementó también un sistema de envío de email del documento, pero solo para la versión local en Python que fue descartado por no ser necesario en AWS al tener contenedores AWS.

Tras el éxito en funcionamiento de la versión local, se procedió al desarrollo en AWS de Amazon. Se migraron los módulos y se generó tanto una versión de un solo programa AWS Lambda y otra versión en módulos totalmente independientes que envían y recogen la información de módulos S3.

Al final, ambos modelos de diseños AWS se presentaron a los responsables del

[1]Dirección del rincón del escriba eterno: `https://elrincondelescribaeterno.blogspot.com/`

proyecto tras hacer las pruebas pertinentes, pasando la validación y encontrando un propósito adicional como herramienta de traducción y generación de audios en otros idiomas a partir de un pdf utilizando los módulos independientes.

6.4. Trabajo futuro

6.4.1. Uso a través de Moodle

Debido a su diseño modular, se puede añadir un módulo adicional para subir el audio a Moodle, facilitando la subida de audios explicativos o traducidos en otros idiomas por el 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 que necesite ayuda auditiva para leer algún tema o que no entienda la lengua oficial.

6.4.2. Envíos por email

Aunque se descartaron para la versión final de AWS, en cualquier momento se puede extrapolar el módulo de Python de la versión local y generar un nuevo Lambda con un S3, para que se envíe el audio y los documentos en PDF a través de email.

Capítulo 6. Measurements, conclusions, results and future work

6.1. Cost and time tests

After development, we proceeded to perform a series of price and time tests in different test environments, to check which would be the most recommendable option for use by a company or an institution. The test environments used:

- Local Computer.
- Google Cloud.
- Amazon Web Service.
- Microsoft Azure.
- Digital Ocean.

To perform the calculations we have used the cost calculator of *Google Cloud*, *AWS Lambda Cost Caculator*, *Microsoft Azure* and *Digital Ocean*, in comparison to the total cost of a standard 1000 standard local computer and how long it would take to amortize the operations.

6.1.1. Google Cloud vs local computer

Calculations on a total of 1,000 operations are as follows:

Comparative Parameters	**Google Cloud**	**Local Computer**
Number of iterations per month	1000	1000
Storage space	2GB	500GB
ime to perform one iteration	60000msg	60000msg
RAM size	4GB	16GB
Total cost/month	4.35€/month	83.33€(Amortization: 1000/12)

Cuadro 6.1: Comparison between Google Cloud and local computer

The Table 6.1 shows the monthly cost comparison for generating 1000 audios in *Google Cloud* and in a Local Computer, we can see that for a machine with a much lower capacity the cost is relatively very low using *Google Cloud* versus a Local Computer.

6.1.2. AWS vs local Computer

We are going to take as a reference the Local of the previous section and we are going to compare it with several types of executions of Amazon AWS, in this comparison we have put high times due to the volume of modules to be executed and the intermediate transactions of information in and out of the S3:

- In this first section we are going to check the cost of 100 requests per hour per month, a total of 73000 requests per month with an execution time of approximately 60000 msg. The Figure 6.1 is going to show all the calculated amounts.

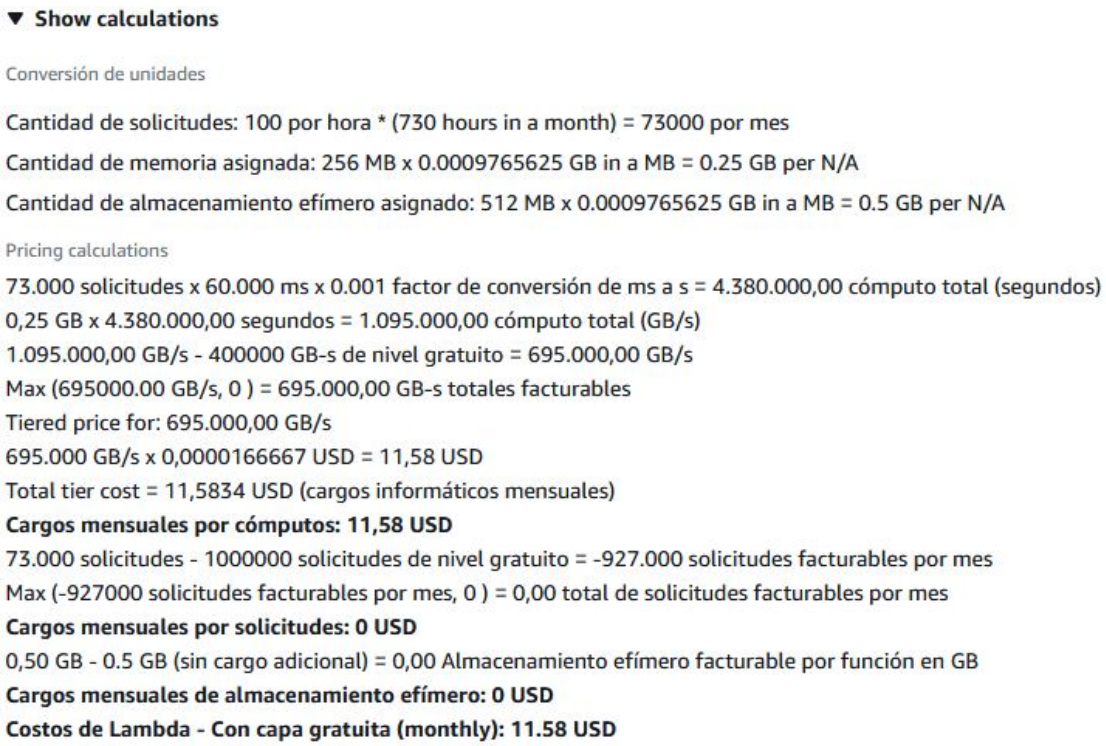


Figura 6.1: Cost estimate for 100 requests per hour per month.

- In this second section we are going to check the cost of 100 requests per hour per month, a total of 73000 requests per month with an execution time of approximately 180000 msg. Figure 6.2 shows the calculations performed for these requests.

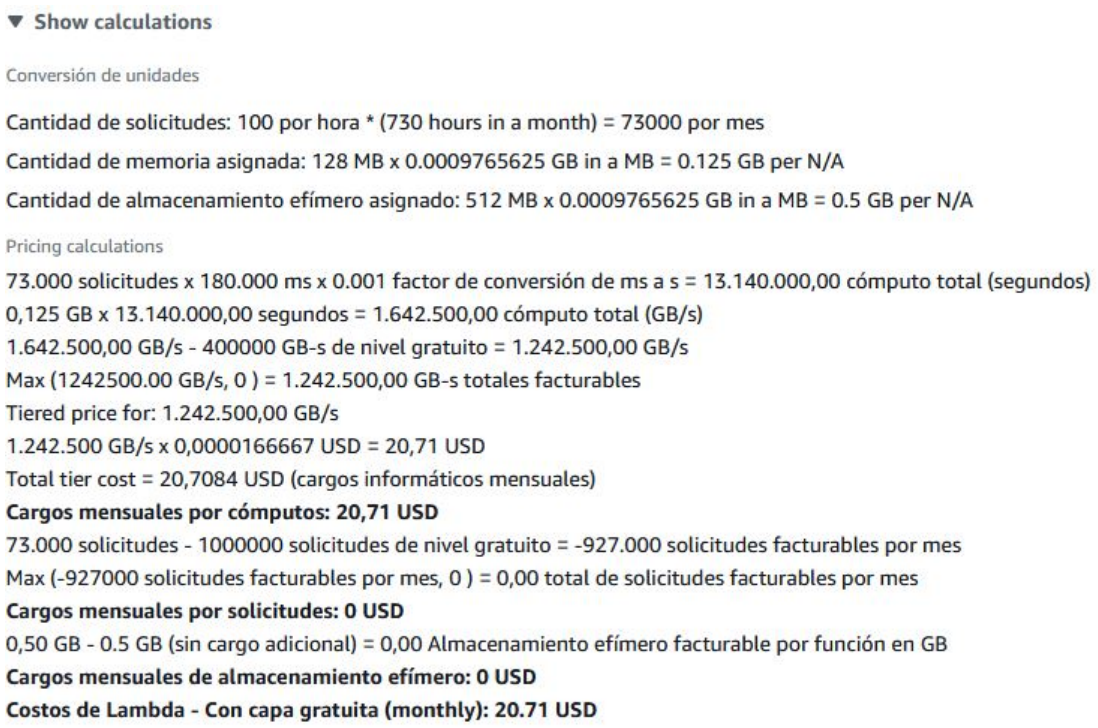


Figura 6.2: Cost estimate for 100 requests per hour per month.

- In this third section we are going to check the cost of 1000 requests per day per month, a total of 30416.67 requests per month with an execution time of approximately 60000 msg. Figure 6.3 will show the calculations performed for these requests.

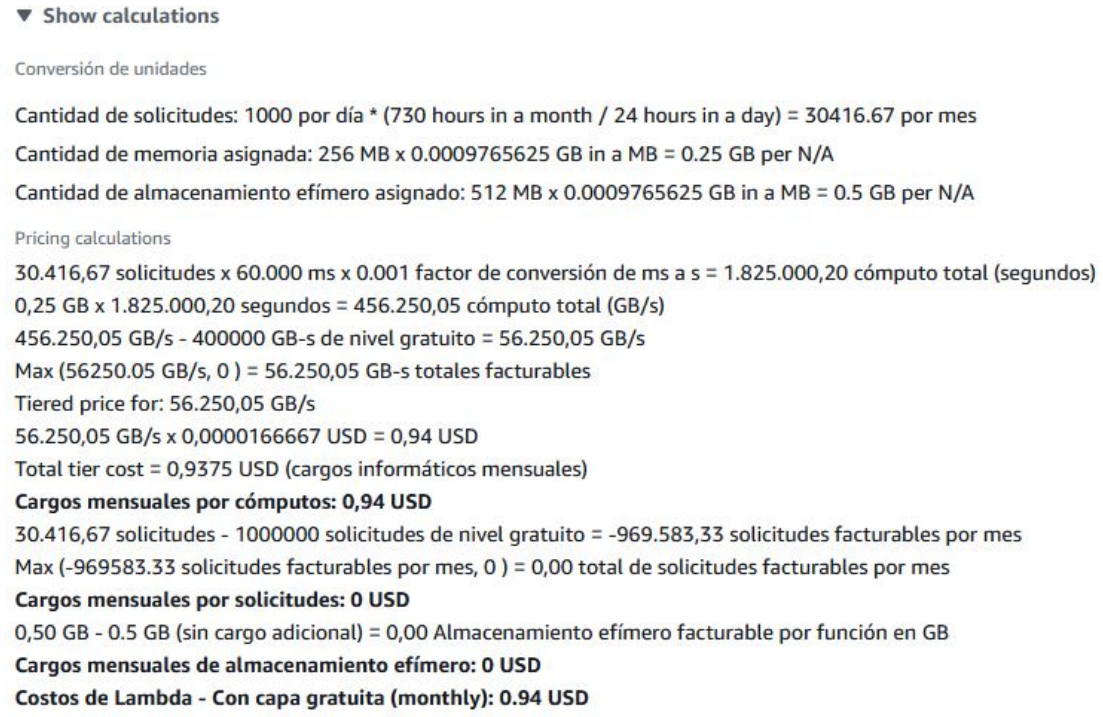


Figura 6.3: Cost estimate for 1000 requests per day per month.

- In this fourth section we are going to check the cost of 1000 requests per day per month, a total of 30416.67 requests per month with an execution time of approximately 180000 msg. The Figure 6.4 is going to show the calculations performed for these requests.

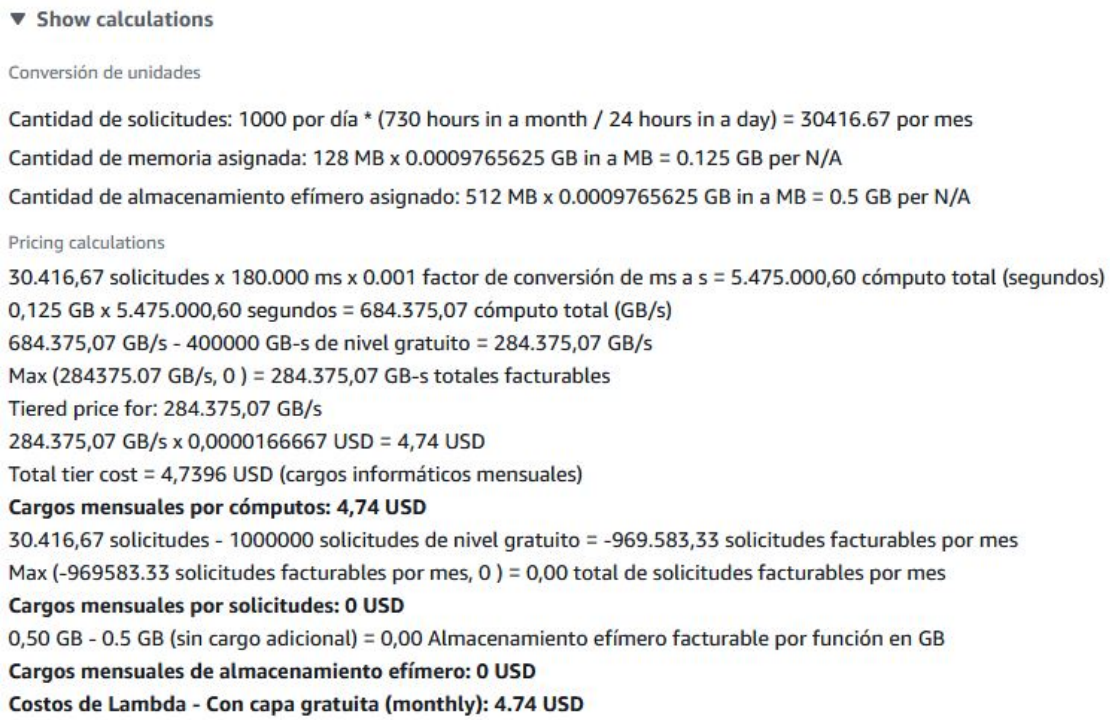


Figura 6.4: Cost estimate for 1000 requests per day per month

- In this fifth section we are going to check the cost of 25000 requests per month with an execution time of approximately 60000 msg. Figure 6.5 will show the calculations performed for these requests.

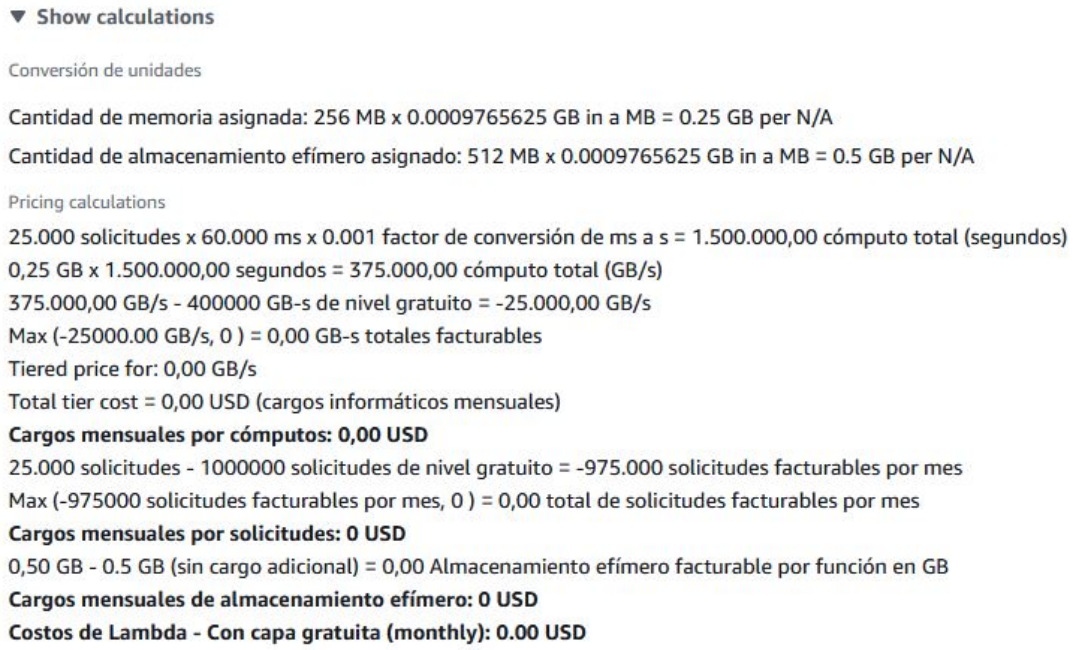


Figura 6.5: Cost estimate for 25000 requests per month.

- In this sixth section we are going to check the cost of 25000 requests per month with an execution time of approximately 180000 msg. Figure 6.6 shows the calculations performed for these requests.

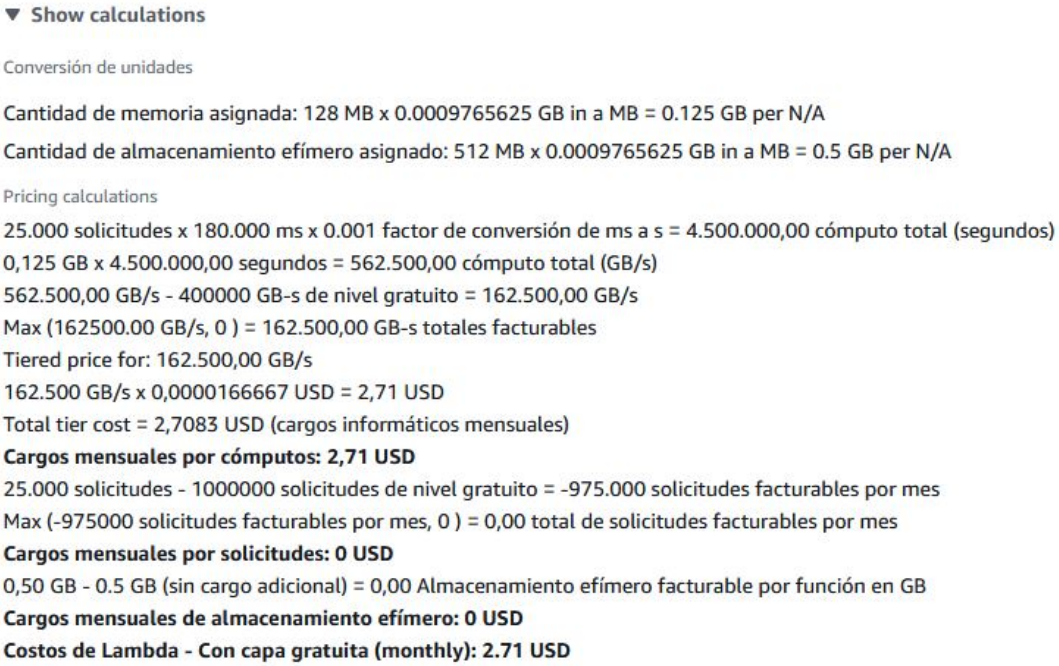


Figura 6.6: Cost estimate for 25000 requests per month.

6.1.3. Microsoft Azure

We are going to take as reference the Local Orderer of section 6.1.1 and we are going to compare it with several types of Microsoft Azure executions, in this comparison we have put high times due to the volume of modules to be executed and with different amounts of monthly executions.

- In this first section we are going to check the cost of 73000 requests per month with an execution time of approximately 60000 msg. Figure 6.7 is going to show all the calculated amounts.

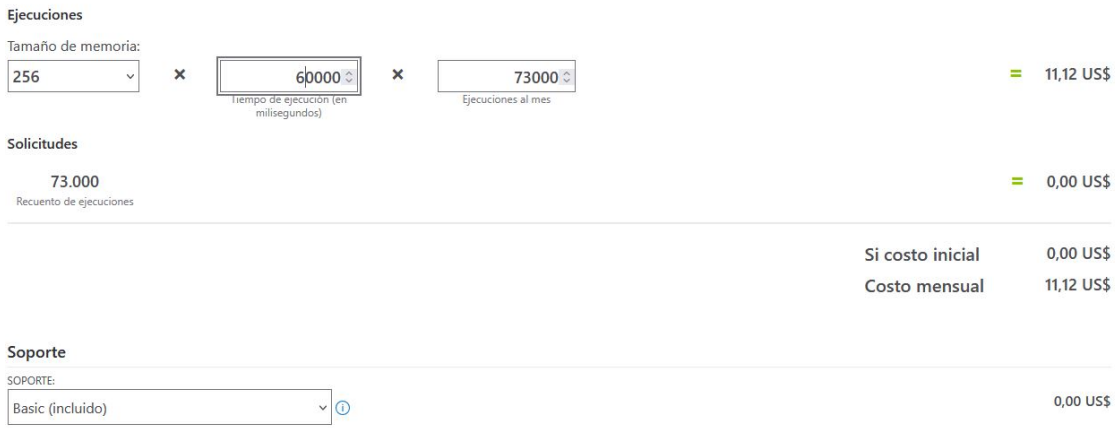


Figura 6.7: Cost estimate for 73000 requests per month with a time of 60000msg.

- In this second section we are going to check the cost of 73000 requests per month with an execution time of approximately 180000 msg. Figure 6.8 shows the calculations performed for these requests.

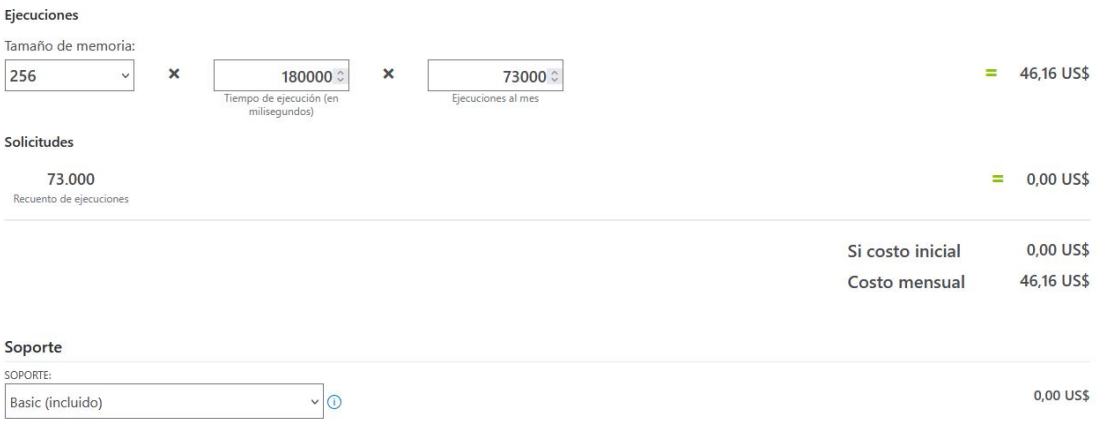


Figura 6.8: Cost estimate for 73000 requests per month with a time of 180000msg.

- In this third section we are going to check the cost of 30417 requests per month with an execution time of approximately 60000 msg. Figure 6.9 will show the calculations performed for these requests.

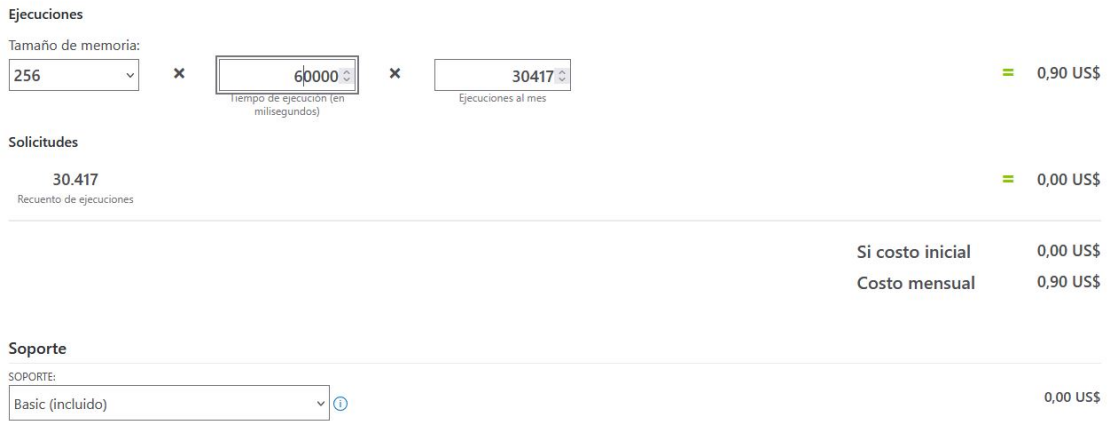


Figura 6.9: Cost estimate for 30417 requests per month with a time of 60000msg.

- In this fourth section we are going to check the cost of 30417 requests per month with an execution time of approximately 180000 msg. Figure 6.10 will show the calculations performed for these requests.

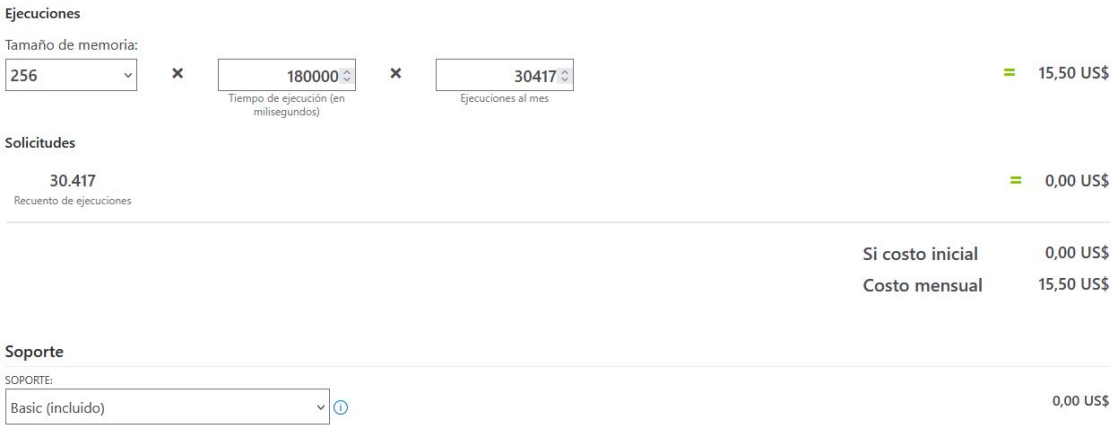


Figura 6.10: Cost estimate for 30417 requests per month with a time of 180000msg.

- In this fifth section we are going to check the cost of 25000 requests per month with an execution time of approximately 60000 msg. Figure 6.11 will show the calculations performed for these requests.

Figura 6.11: Cost estimate for 25000 requests per month with a time of 60000msg.

- In this sixth section we are going to check the cost of 25000 requests per month with an execution time of approximately 180000 msg. Figure 6.12 shows the calculations performed for these requests.

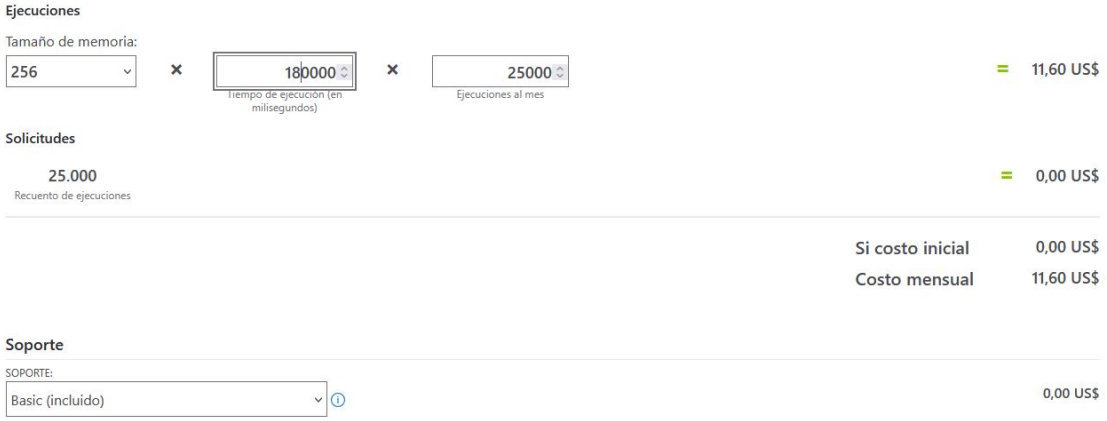


Figura 6.12: Cost estimate for 25000 requests per year with a time of 180000msg.

6.1.4. Digital Ocean

We are going to take as a reference the Local Ordering System of section 6.1.1 and we are going to compare it with several types of Digital Ocean executions, in this comparison we have put high times due to the volume of modules to be executed and different amounts of execution operations per month.

- In this first section we are going to check the cost of 73000 requests per month. Figure 6.13 will show all the calculated amounts.

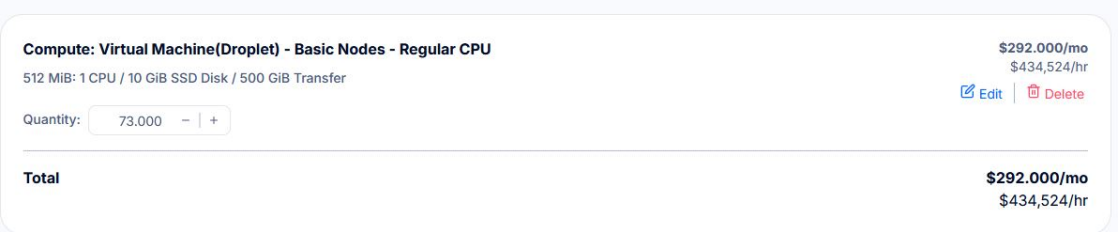


Figura 6.13: Cost estimate for 73000 requests per month.

- In this second section we will check the cost of 30417 requests per month. Figure 6.14 shows the calculations performed for these requests.

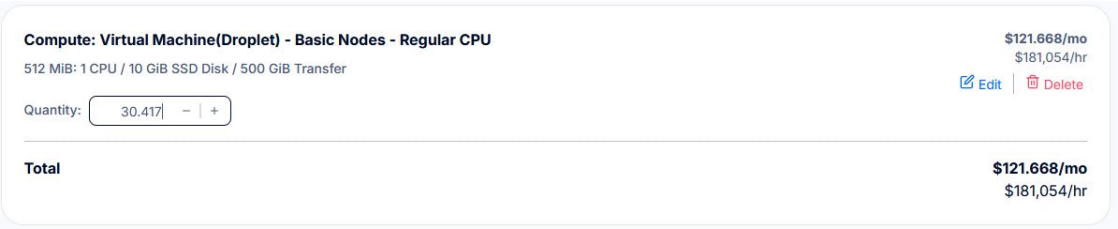


Figura 6.14: Cost estimate for 30417 requests per month.

- In this third section we are going to check the cost of 25000 requests per month. Figure 6.15 will show the calculations made for these requests.

Figura 6.15: Cost estimate for 25000 requests per month.

6.2. Comparison of final results

After the different measurements both locally and with the different execution platforms in the cloud, we found that the operating prices in Local, Google Cloud and Digital Ocean are quite high, while Amazon's AWS and Microsoft Azure are in a similar or identical cost and execution margins in low execution times.

But it is seen that in the measurements with a high execution time, the clear winner is Amazon's AWS, against Microsoft Azure, because in the latter the costs soar in high execution times.

6.3. Conclusions

This project aimed to generate a high quality audio of any story extracted from my blog of stories[1] for visually impaired or totally blind people.

For this purpose, we have reviewed works on the use of audiobooks, both as an aid tool for blind or low-visibility people, as well as for pedagogical use in people in academic training. Demonstrating that audiobooks are a great ally when it comes to breaking the barriers imposed by a visual impairment or learning problems such as dyslexia or ADHD.

The methodology deployed has been by iterations of development in a gradual way, starting with the modules locally in Pyhton 3.8 and the modules in AWS later to generate both a unified final version and independent versions of each Lambda module.

Pyhton was used for its wide range of libraries and easy learning, and Amazon's AWS was also chosen over Google Cloud for its computing power, speed and low costs compared to Google's system.

The development started with the initial modules of extracting a text from the address of one of the stories and transforming it into a readable PDF. After contamination it was decided to generate the translation module, although the web with which we work their stories are in Spanish, it was considered to be able to translate the story into several languages including French, English, German and Italian. Once this module was developed, it was decided to create the audio generator, reading the text independently of the language and pausing the reading at certain punctuation marks or expressions, to give it a more human and less artificial character.

Despite the improvements made in the audio generation, it was found that the expected quality was far from the desired one, so it was decided to create a new audio enhancement module using ffmpeg technology, cleaning residual static and improving the sound of the audio voice. Subsequently, a system to send email of the document was also implemented, but only for the local version in Python, which was discarded as it was not necessary in AWS as it had AWS containers.

After the successful operation of the local version, we proceeded to the development in Amazon AWS. The modules were migrated and both a single program version of AWS Lambda and another version in totally independent modules that send and collect information from S3 modules were generated.

In the end, both AWS design models were presented to the project managers after testing, passing validation and finding an additional purpose as a tool for translation

[1]Dirección del rincón del escriba eterno: `https://elrincondelescribaeterno.blogspot.com/`

and audio generation in other languages from a pdf using the independent modules.

6.4. Future work

6.4.1. Use via Moodle

Due to its modular design, an additional module can be added to upload the audio to Moodle, facilitating the uploading of explanatory or translated audios in other languages by the faculty of any faculty that uses this platform to manage their virtual campuses and where the documentation is uploaded for use by students who need audio help to read a subject or who do not understand the official language.

6.4.2. Shipments by email

Although they were discarded for the final version of AWS, at any time you can extrapolate the Python module from the local version and generate a new Lambda with an S3, so that audio and PDF documents are sent via email.

Capítulo 7. Enlace a los archivos del TFM

Aquí se adjunta enlace al Drive donde están los códigos finales:

`https://drive.google.com/drive/folders/1ilmmHpg3IzXK-e7sljKTejWgjLs0OnJ-?usp=drive_link`

Aquí se adjunta enlace al repositorio de Github con los códigos finales:

`https://github.com/SoySamu/TFM-Black-Bolt-Otea.git`

Bibliografía

[1] N. Verma, "From the narrator's lips to yours: Streaming, podcasting, and the risqué aesthetic of amazon channels," *Participations*, vol. 16, no. 2, pp. 273–97, 2019.

[2] E. L. Gudiño Llerena, "Aplicación web móvil de audiolibros en la universidad regional autónoma de los andes–uniandes sede ibarra," B.S. thesis, 2024.

[3] J. A. Cita Jiménez, "Una secuencia didáctica basada en audiolibros para la transformación de la habilidad de escucha del inglés en el grado octavo de la institución educativa departamental general santander de sibaté–cundinamarca," 2024.

[4] A. J. Brito Limas and M. S. Cordova Perez, "Los audiolibros en la comprensión lectora de textos narrativos en los estudiantes del segundo grado de educación secundaria del colegio "santa rosa de viterbo" de la ciudad de huaraz–2020."

[5] R. Turcios, "Prueba de wilcoxon-mann-whitney: mitos y realidades," *Rev Mex Endocrinol Metab Nutr*, vol. 2, pp. 18–21, 2015.

[6] A. García-Rodríguez and R. Gómez-Díaz, "¿ leer con los oídos?: audiolibros y literatura infantil y juvenil," *Anuario ThinkEPI*, vol. 13, 2019.

[7] D. N. Fajardo Palacios and A. S. Romero Arévalo, "Producción de un audiolibro orientado a la educomunicación para niños novidentes de 5 a 7 años de edad," B.S. thesis, Univesidad del Azuay, 2018.

[8] J. A. Vela, S. T. Rusillo, and F. J. del Rio Olvera, "Creación de audiolibros como una estrategia innovadora en el método de enseñanza universitaria," *CIVINEDU 2020*, p. 103, 2020.

[9] I. Andrés-Alba, "Audiolibros en griego y en latín," *Thamyris, nova series. Revista de Didáctica de Cultura Clásica, Griego y Latín*, vol. 14, pp. 185–194, 2023.

[10] M. M. F. González, K. M. B. Cedeño, L. G. L. Susana, and L. E. Casquete, "Audiolibros como herramientas para fomentar la lectura en niños con dificultades de aprendizaje," *Polo del Conocimiento*, vol. 10, no. 3, pp. 2846–2863, 2025.

[11] L. A. P. García and A. M. C. Cedeño, "Uso de audiolibros para el desarrollo del pensamiento creativo en los estudiantes de básica superior," *Dominio de las ciencias*, vol. 7, no. 1, p. 59, 2021.

[12] V. Krotov, L. Johnson, and L. Silva, "Tutorial: Legality and ethics of web scraping," 2020.

[13] Y. M. U. AWS and H. Singh, *Practical machine learning with AWS*. Springer, 2021.

[14] A. Del Sole and D. Sole, *Visual Studio Code Distilled*. Springer, 2019.

[15] L. Pablo, G. Garay-Aramburu, A. García Layana, A. Fernandez, I. Vázquez, X. Acebes, J. Zulueta, D. Balonga, L. Salinas-Ortega, Á. Muñoz *et al.*, "Assessing the economic burden of vision loss and irreversible legal blindness in spain (2021–2030): a societal perspective," *Health Economics Review*, vol. 14, no. 1, p. 70, 2024.