



Universidad
Internacional
de Valencia

Trabajo Fin de Máster:
IdioGrabber, aprende idiomas de forma
divertida

Titulación: Máster U. en
Desarrollo de Aplicaciones y
Servicios Web

Curso académico
2022-2023

Alumno/a: Gutiérrez Nebreda
Josu
D.N.I.: 78996415V

Director/a de TFM: Héctor
Martínez Cebrián

Convocatoria:
Segunda

Índice

1. Introducción	5
Motivación.....	6
Objetivos.....	6
2. Marco tecnológico.....	7
3. Metodología	13
Especificación del proyecto	13
Planificación.....	15
4. Desarrollo	18
Base de datos	18
Autómata	25
Creación del entorno.....	27
Registro	29
Login	30
Administrador.....	31
Usuario	34
5. Servicios en la nube.....	39



AWS Elastic Beanstalk	39
AWS RDS	41
AWS S3	42
6. Resultados y discusión	44
7. Conclusiones	46
8. Trabajo futuro.....	48
9. Anexo I: Código automata	50
10. Bibliografía.....	53

Lista de ilustraciones

Ilustración 1. Modelo - Vista - Controlador	13
Ilustración 2. Gestión de servicios en la nube	14
Ilustración 3. Planificación de la primera entrega.	15
Ilustración 4. Planificación de la entrega final.....	17
Ilustración 5. Relación rol – usuario.....	19
Ilustración 6. Relación usuario – configuración.	20
Ilustración 7. Relación usuario – recurso.....	20
Ilustración 8. Relación configuración – idioma.	21
Ilustración 9. Relación configuración – dificultad.	21
Ilustración 10. Relación dificultad – recurso.	22
Ilustración 11. Relación recurso – idioma.....	22
Ilustración 12. Relación recurso – vocabulario.	23
Ilustración 13. Relación vocabulario – familia recursos.....	23
Ilustración 14. Relación usuario – redacción.....	24
Ilustración 15. Relación idioma – redacción.	24
Ilustración 16. Diagrama de flujo del funcionamiento del autómata.	25
Ilustración 17. Configuración del entorno para conectarse a la base de datos.	27
Ilustración 18. Pantalla de registro de nuevos usuarios.	29
Ilustración 19. Pantalla de login.....	30
Ilustración 20. Cambio de idioma.	30
Ilustración 21. Pantalla de administrador para la gestión de los recursos del sistema.	31
Ilustración 22. Pantalla de administrador para la gestión de las configuraciones del usuario.....	32
Ilustración 23. Pantalla de administrador para la gestión de las redacciones.	33
Ilustración 24. Pantalla para la gestión de los datos personales del usuario.....	34
Ilustración 25. Pantalla de elección de estudios.	35
Ilustración 26. Recurso de tipo palabra acertado y fallado.	36
Ilustración 27. Recurso de tipo frase acertado y fallado	36
Ilustración 28. Recurso de tipo audio acertado y fallado.	37
Ilustración 29. Redacción a la espera de ser corregida.....	37
Ilustración 30. Redacción corregida.	37
Ilustración 31. Pantalla para escribir una nueva redacción.	38

Ilustración 32. Resultado de del CI/CD.....	40
Ilustración 33. Creación de la base de datos mediante Terraform.	41
Ilustración 34. Creación del Bucket de S3.....	43
Ilustración 35. Selección de recurso 1	50
Ilustración 36. Selección de recurso 2.....	51
Ilustración 37. Selección de recurso 3.....	51
Ilustración 39. Comprobar respuesta 2.....	52
Ilustración 38. Comprobar respuesta 1.....	52

1. Introducción

Hoy en día, vivimos en una sociedad en la cual, el uso de internet es vital para nuestra vida cotidiana. Todos nuestros sistemas están conectados a la red, y son ya miles de personas las que aprovechan para estudiar vía online.

Por todo ello, cada vez hay más plataformas para el acceso a estudios online, ya sea de universidades, empresas particulares, etc...

Teniendo en cuenta esto, se pretende realizar una aplicación web para el aprendizaje, en este caso de idiomas, basado en un principio de la neurociencia: repetición espaciada.

La repetición espaciada podría ayudar a las personas que tienen ciertos problemas a la hora de aprender, además de poder mejorar los resultados de los exámenes escolares. (Domotizar.com, 2018)

La repetición espaciada se basa en aprender palabras, sonidos, frases, repitiéndolas en un periodo de tiempo establecido por el algoritmo y en base a la dificultad que tiene el usuario para aprenderla, es decir, si el usuario va acertando el recurso a medida que se le va mostrando en la aplicación, este aparecerá de forma menos casual. En el caso contrario, el recurso se repetirá más veces, de forma que el usuario acabe por memorizarlo. (Alex Figueroba, 2017)

Cada usuario podrá adaptar su nivel a sus necesidades, y el plazo de tiempo en el que quiere aprender los recursos. De esta forma, dependiendo del nivel que tengas podrás personalizar tus estudios a tu gusto.

Como resumen, esta aplicación nos va a permitir aprender un idioma de manera sencilla gracias a la neurociencia.

Motivación

Durante el desarrollo del master se propusieron diversas propuestas para realizar, leyendo todas e investigando un poco el tema del que trataban, se optó por la realización de este trabajo, ya que la idea era muy interesante, y da pie a mucho juego, ya que da la posibilidad de permitir aprender varias temáticas. También desarrollar una aplicación web, que permita a la gente ayudar a aprender de una forma entretenida y con una metodología poco conocida a día de hoy.

Objetivos

El objetivo principal es la **realización de la aplicación IdioGrabber**, para todos los navegadores web disponibles, tanto en ordenadores como en su versión móvil. A continuación, se explican los objetivos con más detalle:

- **Desarrollo de interfaces graficas:** Para que se pueda ver en todos los navegadores, es necesario que todas las interfaces sean adaptables a ellos, en cualquier tamaño que pueda existir.
- **Desarrollo de un autómata:** El autómata será el cerebro, parte encargada de que la aplicación funcione correctamente.
- **Desarrollo del almacenamiento de datos:** La aplicación necesitara una estructura donde poder guardar cantidades de datos para poder luego gestionarlos de forma correcta.

- **Creación de una infraestructura:** Es necesario crear una infraestructura para que todo este proyecto pueda ser escalable.

2. Marco tecnológico

En la actualidad podemos encontrar aplicaciones como Duolingo la cual nos permite aprender diferentes idiomas mediante diferentes recursos, ya sea con palabras sueltas o escuchando sonidos. (Duolingo,2023).

Está basado en un sistema de secciones y a medida que vas acertando, vas pasando de sección. alguna de las desventajas que presenta esta aplicación es que tiene muy poca variedad de ejercicios, a medida que vas haciendo se repiten con bastante facilidad. No tiene un soporte que permita a los usuarios pedir ayuda para solucionar dudas a medida que se hace el curso, además que es una web que está más centrada en el estudio de nuevo vocabulario que de gramática o fonética, unido a esto las voces que utiliza para reproducir los audios son robóticas. Un aspecto visual que tiene molesto es que saltan demasiadas notificaciones constantemente, además de que si comienzas un nivel y no llegas a finalizar un apartado tendrías que empezar de nuevo.

Luego tenemos a Memrise (Memrise ,2023) que es una aplicación que tú eliges que quieres ir estudiando, te dan una serie de recomendaciones y luego puedes elegir que aprender, no sigue un orden como si hace Duolingo. Es una web que está basada en aprender vocabulario nuevo, no tiene ningún tipo más por lo que sería una aplicación más orientada a complementar unos estudios.

Para el desarrollo de una aplicación web hay dos partes fundamentales, el backend y el frontend.

Para la parte frontend se podría usar frameworks como Angular o React, los cuales son bastante potentes y dan mucha posibilidad, gracias a librerías que tienen para hacer cosas visuales. O simplemente trabajar con JavaScript nativo, junto con HTML y JQuery.

Angular es un framework desarrollado por Google, que nos permite desarrollar aplicaciones web. Su programación se basa en el uso de TypeScript o JavaScript.

Alguna de las características más importantes que proporciona Angular es el Data binding que nos permite mostrar elemento en nuestra web de manera dinámica, pasando información entre las diferentes vistas de nuestra aplicación.

Tiene una sintaxis muy sencilla, aparte de proporcionar comandos que facilitan la creación e indexación de las vistas de la web.

Uno de los principales problemas que tiene angular es que genera aplicaciones demasiado grandes, ya que instala muchos paquetes de JavaScript necesarios para que funcione, esto puede suponer un problema de tamaño, porque nuestra estructura será grande. Además, de que la carga en tiempo de ejecución puede verse lastrada por este motivo. Otra desventaja de Angular es que tiene una curva de aprendizaje más difícil que otros frameworks, aparte de que su documentación oficial no es muy completa, puesto que muchas veces falta ejemplos y no siempre está actualizada, aunque por internet hay muchos tutoriales que pueden ayudarnos con el problema que tengamos.

La ventaja que tiene esto es que Angular nos cubre todo, es decir, al tener que instalar tantos paquetes, nuestra aplicación estará bien cubierta. Cada seis meses se lanza una nueva versión de Angular, por lo que tendremos mejoras cada poco tiempo, además de que las versiones nuevas son compatibles con las anteriores, por lo que no haría falta actualizar todo nuestro proyecto. (Juan Maria Hernandez (2013), Nicolas Molina (2023))

React es una biblioteca de JavaScript desarrollada por Facebook, cada página web está compuesta por varios componentes que son reutilizables, además de que es una única página, es decir, cada vez que hagamos una petición a un servidor, en vez cargar una página web nueva, se cargaran los componentes que ya tenemos creados.

Nos proporciona una carga reactiva, lo que hará que cuando hagamos un cambio en nuestra aplicación automáticamente se muestre el contenido. También proporciona varios componentes ya preparados para que los programadores puedan hacer sus aplicaciones más rápido. Se puede enlazar con React Native que nos va a permitir tener nuestra aplicación para móvil. (Reactjs vs React Native,2023)

Al ser un framework relativamente nuevo, evoluciona a un ritmo muy alto, por lo que estar al día de este framework es un poco tedioso. Aparte, no sigue ningún tipo de patrón de diseño, por lo que todo queda en manos del programador y en su buen hacer.

JavaScript es el lenguaje más usado en todo el mundo, nos permite generar sitios web interactivos con contenido dinámico. Alguna de las ventajas que tiene es que nos permite integrarlo con otro tipo de lenguajes de programación, como puede ser PHP, es decir, si se crea un archivo con extensión .php dentro podremos meter código JavaScript.

Al ejecutarse en el lado del cliente, va a hacer que la carga sea mucho más rápida, aparte de que es multiplataforma y proporciona ayudas para detectar en qué tipo de dispositivo nos encontramos. (Keepcoding,2022)

Lo malo que tiene este lenguaje es que es al ser un código de lado de cliente, cualquier persona puede leerlo y tener acceso a él, aparte si un usuario desactiva el JavaScript del navegador, el código dejará de funcionar, por lo que puede causar algunos problemas en la página web. También necesita ser complementado por otro lenguaje de backend que le permita hacer gestiones con una base de datos u otras operaciones. (Carlos Herrera ,2020)

Si se combina JavaScript con HTML y JQuery se puede desarrollar una página web bastante potente, ya que HTML es el lenguaje de marcado estándar para el diseño de páginas web, con el que se puede diseñar cualquier elemento en una web. Aparte tiene librerías externas como puede ser bootstrap que nos ayuda mucho con el tema de estilos, puesto que proporciona algunas cosas predeterminadas para ayudar al programador. JQuery es un acompañante perfecto para JavaScript, ya que nos va a permitir hacer muchas cosas, que hacemos con este último de manera más sencilla. Utiliza una sintaxis muy parecida a CSS, y tiene varias librerías externas que nos permiten hacer cosas visuales muy interesantes.

Para el backend que es la parte donde vamos a hacer toda la lógica de nuestra aplicación, también hay varias opciones donde elegir, como por ejemplo ASP.NET que es un lenguaje de programación desarrollado por Microsoft, además es compilado, esto implica que cada vez que realicemos un cambio y queramos verlo tendremos que compilar toda la aplicación. Este lenguaje nos permite combinarlo con otros lenguajes de programación para crear webs interactivas, proporciona un patrón MVC (Modelo, Vista, Controlador) con el que programador podrá tener mejor estructurada la aplicación. (Asler Cedeño ,2020)

Lo malo que tiene ASP.NET es que, al ser un lenguaje compilado, tiene tiempos muy elevados a la hora de desarrollar, además de que necesita muchos recursos para funcionar correctamente. Por otra parte, depende de un entorno de desarrollo específico como es Visual Studio, y de un sistema operativo que es Windows. (Estrada web group ,2023)

Por otro lado, tenemos Laravel que es framework de PHP el cual facilita mucho el desarrollo de aplicaciones en este lenguaje, ya que proporciona al programador herramientas para facilitar programación. PHP es un lenguaje interpretado del lado backend, esto significa que no hace falta compilar la aplicación cada vez que queramos ver un cambio, además hace uso del MVC.

Laravel tiene un sistema de plantillas llamados Blades que nos permiten hacer unas interfaces muy potentes. También proporciona Eloquent ORM, que lo que hace es facilitar al usuario las conexiones con la base de datos, al igual que el acceso a las relaciones entre diferentes entidades. En cuanto a seguridad, en cada formulario de tipo POST (que es para hacer alguna inserción en base de datos) nos da mayor seguridad gracias a una directiva llamada CSRF. Tiene una gran documentación y comunidad detrás, por lo que está bastante bien cubierto.

Como parte mala, es necesario utilizar la consola para crear ciertos archivos, y la curva de aprendizaje puede ser un poco difícil si nunca se ha trabajado con PHP.

En cuanto a la base de datos, hay que diferenciar entre dos tipos. Las bases de datos relacionales, las cuales como su nombre indica, relacionan tablas entre sí, por lo que todo está conectado. Cumple con las propiedades ACID (Atomicidad, Consistencia, Integridad y Durabilidad). Como parte negativa es que tienen un escalado muy malo, y suele tener problema con el guardado de datos multimedia.

Luego tenemos las bases de datos no relacionales o NoSQL son un tipo de base de datos que se caracterizan porque no se puede usar lenguaje SQL, además de que sus datos no están almacenados en tablas con atributos, sino en documentos. Esto nos proporciona mucha más flexibilidad a la hora de guardar nuestros datos, ya que al no depender de una estructura podemos guardar siempre lo que queramos sin tener que preocuparnos tanto de que contenga ciertos atributos. También tiene mayor escalabilidad y un alto rendimiento, ya que están preparadas para trabajar con altos volúmenes de datos. (Ayudaley, 2023)

El problema de usar una base de datos no relacional es que no cumple con las propiedades ACID que, si tiene las bases de datos relacionales.

También se necesita tener un buen conocimiento de base de datos, porque si no se monta bien una estructura previa nos costará acceder a los datos que vayamos guardando, incluso no saber cómo acceder a ciertos datos.

Y, por último, la aplicación al tener bastante carga de usuario estará alojada en un servidor tanto local, en la nube o híbrido, al igual que la base de datos.

Si la aplicación contiene muchos datos sensibles, lo más recomendable es tener la base de datos en un almacenamiento local, así los datos estarán en tu propiedad y se evita que los tenga un tercero.

3. Metodología

Especificación del proyecto

A continuación, se explica las tecnologías elegidas para el desarrollo del proyecto, teniendo en cuenta lo mencionado en el apartado anterior.

El primero de ellos es que lenguaje de programación se va a usar para la parte de backend. El cual se ha decidido utilizar PHP junto con su framework Laravel. Se ha elegido esta opción, porque aparte de ser el lenguaje que se ha visto durante el curso en diversas asignaturas, proporciona gran ayuda al desarrollo de aplicaciones web.

Para la parte del frontend, se ha decidido usar HTML + JavaScript + JQuery. Esta elección se debe a que, investigando un poco, se ha llegado a la conclusión de que son las mejores herramientas para trabajar junto al framework de Laravel.

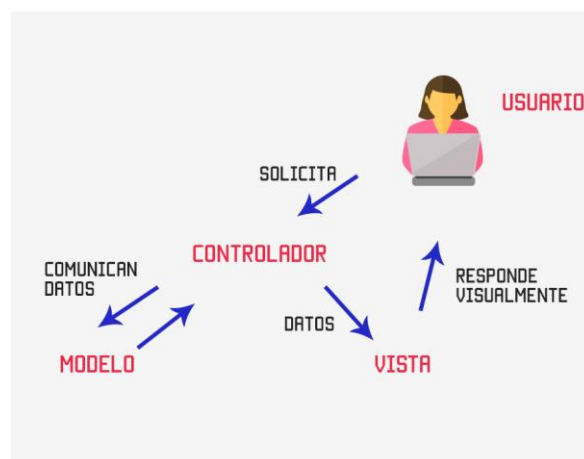


Ilustración 1. Modelo - Vista - Controlador

En cuanto a la base de datos, se ha elegido utilizar una base de datos relacional, ya que se necesita tener muchas entidades relacionadas entre sí. En este caso la opción elegida es MariaDB.

Como control de versiones se usará la plataforma Github, que permitirá llevar un control sobre mi proyecto de forma sencilla, para el desarrollo en local. Una vez finalizado se llevara todo el proyecto a la plataforma GitLab, la cual proporciona unas herramientas de integración y despliegue continua muy sencillas de utilizar, para poder subir nuestra aplicación a la nube.

Para el despliegue de la aplicación, se utilizará una infraestructura PaaS proporcionada por AWS que se llama Elastic Beanstalk. Se ha decidido usar este servicio, ya que los PaaS nos proporciona toda la infraestructura, como desarrollador solo hay que encargarse de subir el código. En cuanto a la base de datos en la nube se utilizará una instancia RDS, la cual crearemos mediante Terraform, en la que podremos migrar nuestra base de datos de local. Y por último para el almacenaje de los ficheros de audio, utilizaremos S3 que nos permitirá tener gran volumen de archivos almacenados en la nube sin colapsar nuestra máquina de Elastic Beanstalk.

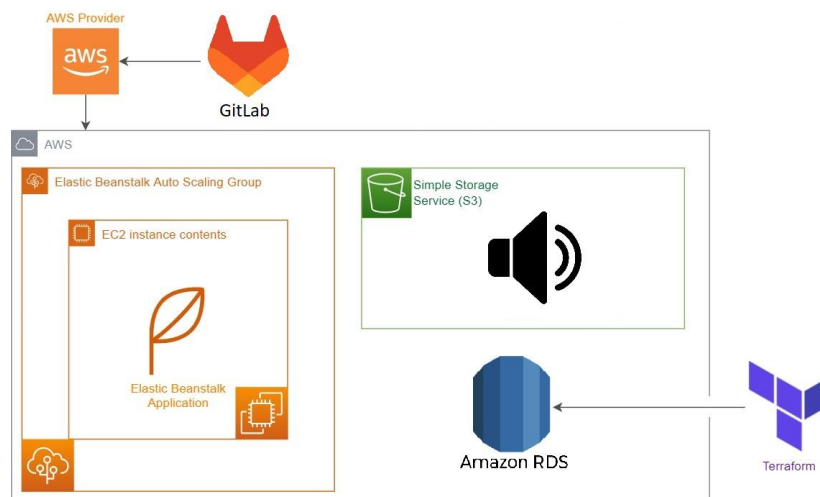


Ilustración 2. Gestión de servicios en la nube

Planificación

A continuación, se va a explicar de forma detallada la planificación que se va a seguir a lo largo de todo el proyecto. Se puede apreciar que hay dos entregas en la planificación, eso es porque hay que entregar la mitad del proyecto antes de cierta fecha y la otra mitad ya al final.

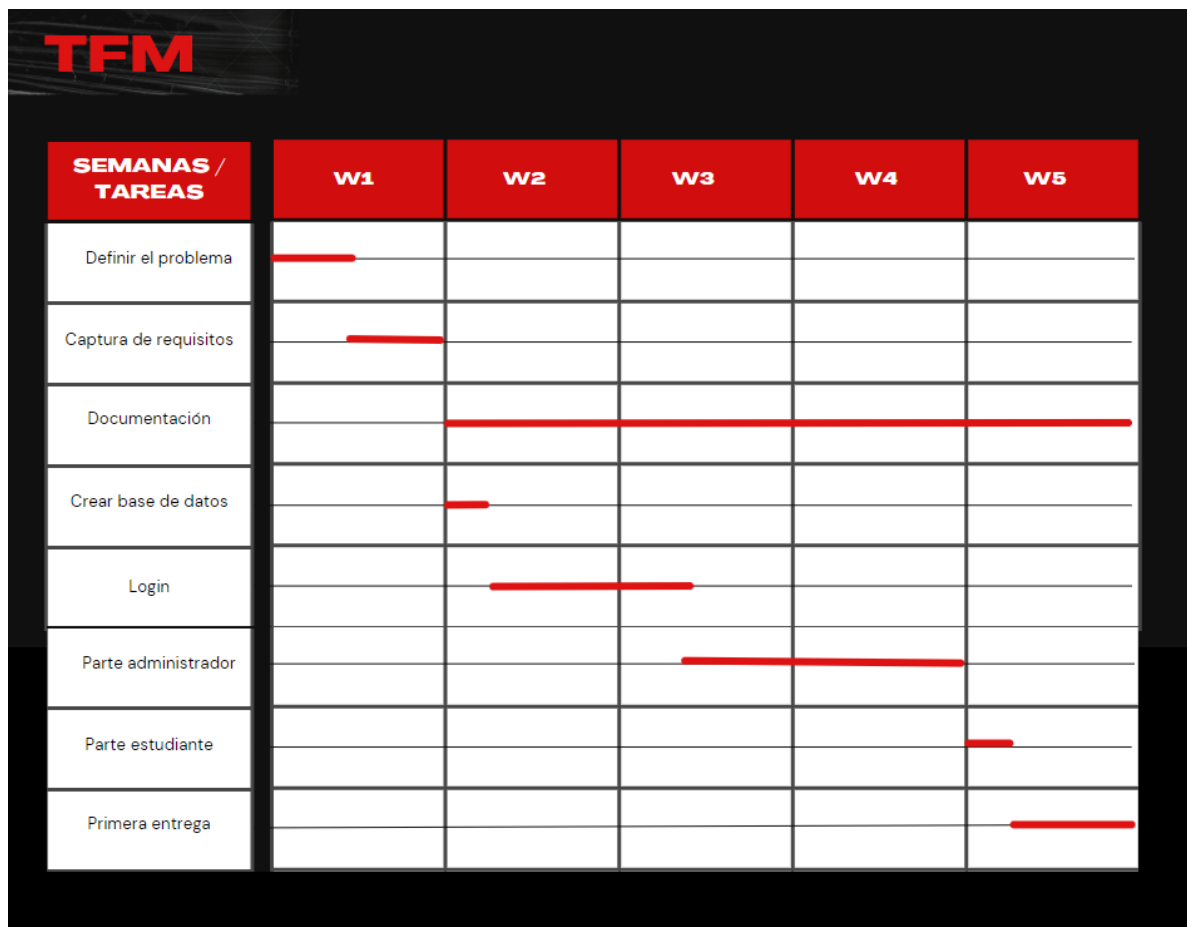


Ilustración 3. Planificación de la primera entrega.

- La primera semana del proyecto se va a dedicar entera al comienzo del proyecto, recogiendo sus requisitos y planteando todo que conllevara su desarrollo.
- A partir de la segunda semana, se empezará con la documentación, ya que esta va a durar durante todo el proyecto como se puede apreciar, por el hecho de que a medida que vamos avanzando programáticamente iremos documentando.
- También en esta semana haremos el modelo entidad relación de la base de datos, y además empezaremos a programar el login y registro de nuestra aplicación.
- La tercera semana de proyecto acabaremos con el apartado anterior y dedicaremos una buena parte al administrador, el cual tiene que gestionar la plataforma, por lo que debemos de hacer cuidadosamente su parte.
- La semana cinco del proyecto se intentará empezar un poco con la parte del estudiante, y además sería la primera entrega comentada anteriormente.

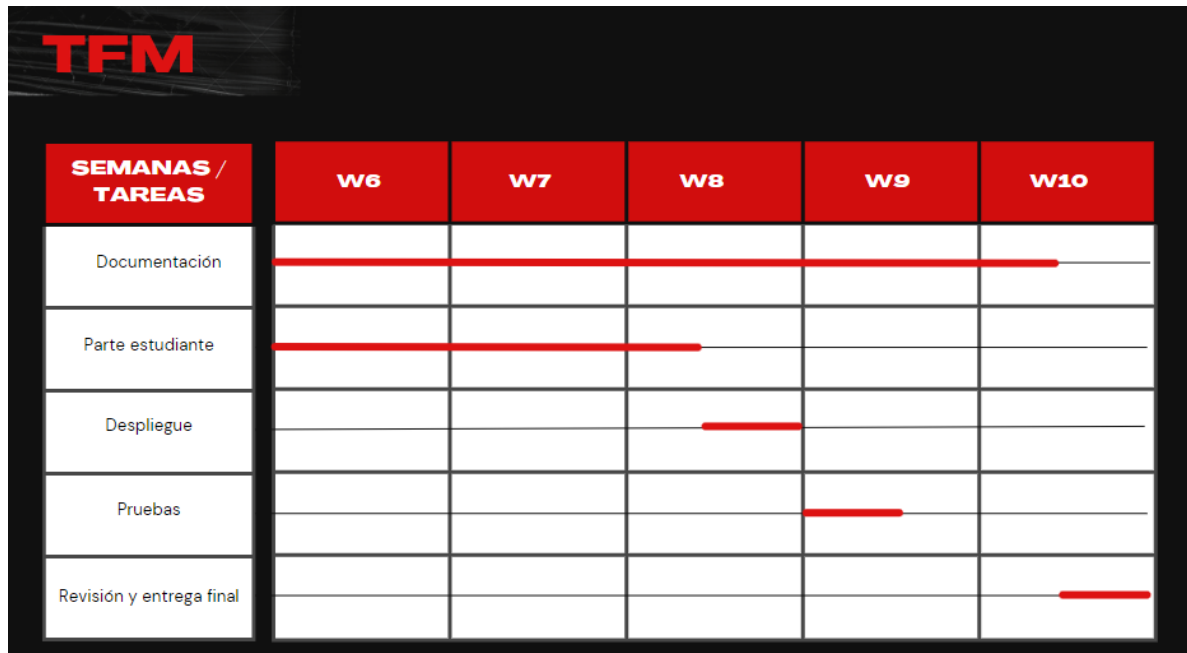


Ilustración 4. Planificación de la entrega final.

- En las últimas etapas del proyecto, podemos ver como se vuelve a mencionar la documentación presente en todo el proyecto. También cabe destacar que la parte del estudiante en total nos va a llevar tres semanas de desarrollo, ya que tenemos que hacer la implementación del autómata que es el corazón de la aplicación.
- Las últimas semanas, se dedicará a desplegar la aplicación en el servidor y a realizar pruebas. Se ha dejado un hueco de tiempo en blanco entre la semana nueve y diez, ya que si por algún casual alguna tarea anterior, no se acabará en el tiempo establecido, tendríamos un margen para retrasar el resto de tareas.

4. Desarrollo

Base de datos

Para empezar el desarrollo del proyecto, se ha decidido comenzar por el diseño entidad relación de la base de datos. La decisión de empezar por esta parte, es que, necesitamos tener bien montada la estructura donde vamos a almacenar los datos de la aplicación, ya que una base de datos mal diseñada, podría acarrear consecuencias negativas a la larga. Al igual que si empezamos a programar sin hacer la base de datos, iríamos a ciegas ya que no tendríamos donde guardar nuestros datos.

Ahora se procederá a explicar brevemente las diferentes entidades que componen la base de datos.

Rol: En esta entidad se van a definir los diferentes roles que pueden tener los usuarios, de momento solo habrá un par de roles. El administrador, el cual, es el que va a gestionar las palabras de la aplicación y modificar alguna configuración de los estudiantes.

El otro rol será el del estudiante, el cual, podrá gestionar su propia configuración y estudiar.

Usuario: Son todas aquellas personas que están registradas en la plataforma, y, por lo tanto, pueden acceder a ella.

Configuración: Es el registro de que está estudiando cada usuario.

Recurso: Son todas las palabras, frase, sonidos que se le mostraran al usuario para su aprendizaje en diferentes idiomas.

Dificultad: El nivel de exigencia que tiene el usuario a la hora de aprender.

Idioma: Los idiomas disponibles para estudiar en la aplicación.

Familia recurso: Son todas las agrupaciones de palabras que hay en la plataforma.

Vocabulario: Son todas las palabras disponibles en el sistema.

Redacciones: En esta tabla se almacenarán todas las redacciones escritas por los usuarios.

Una vez explicadas todas las entidades, se va a proceder a explicar cada relación, para ello se va a desglosar el modelo entidad relación, y se ira explicando una por una.

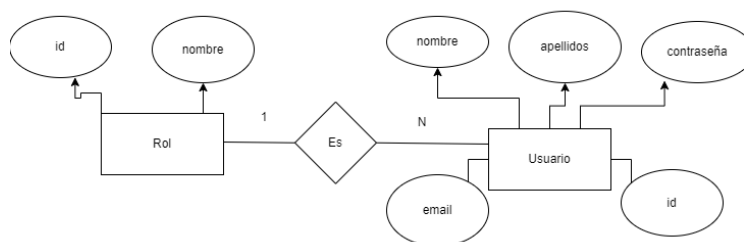
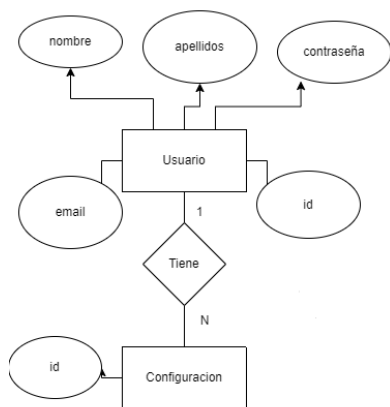


Ilustración 5. Relación rol – usuario.

Cada usuario va a tener un rol asignado y dependiendo de él podrá hacer unas cosas u otras. Ya que al tener una parte de administración no todo el mundo puede acceder a ella, y por ello hay que limitar el acceso a los usuarios mediante roles.



Los usuarios de la plataforma van a poder aprender varios idiomas, por lo que podrán tener varias configuraciones, y ya dentro de la plataforma elegir con cual quieren continuar.

Ilustración 6. Relación usuario – configuración.

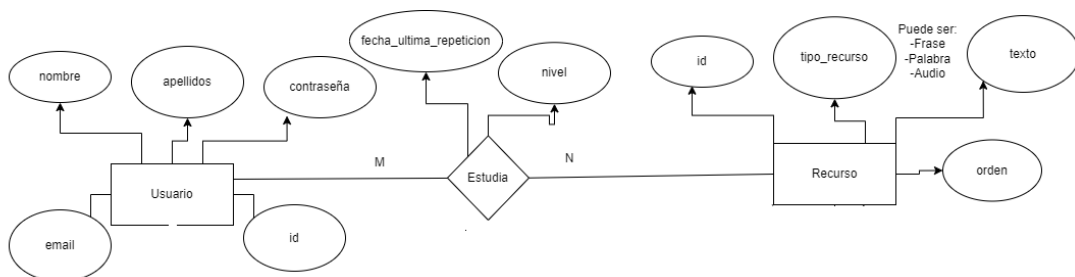


Ilustración 7. Relación usuario – recurso.

Al ser una plataforma de aprendizaje, tendremos un registro de que usuario ha estudiado que recurso. Cabe destacar que vamos a guardar cuando ha sido la última vez que se le ha preguntado el recurso al usuario y el nivel que tiene de él, ya que nuestro algoritmo se basara en esos datos.

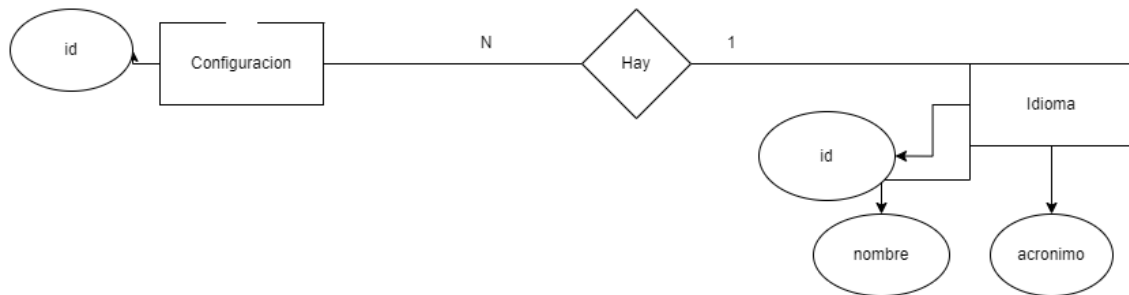
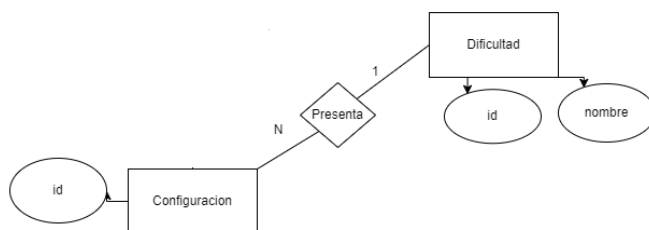


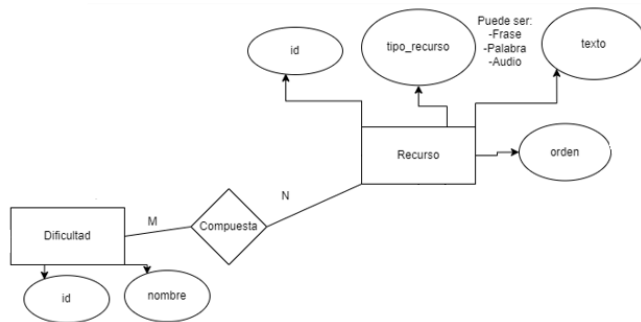
Ilustración 8. Relación configuración – idioma.

Como se ha comentado anteriormente las configuraciones nos determinan que idioma estamos estudiando.



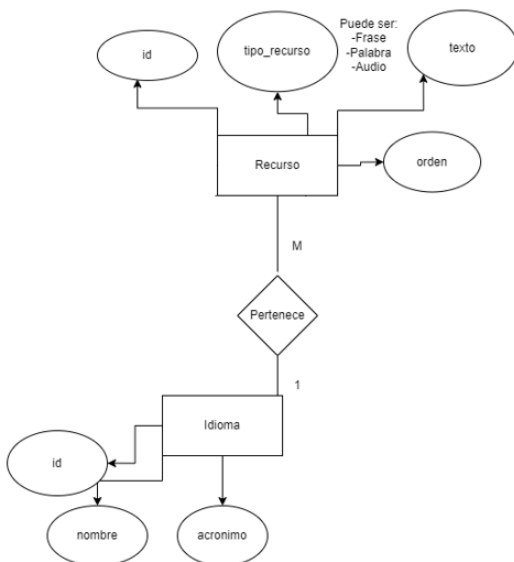
Cada configuración tendrá una dificultad asociada, eligiendo el usuario cual quiere.

Ilustración 9. Relación configuración – dificultad.



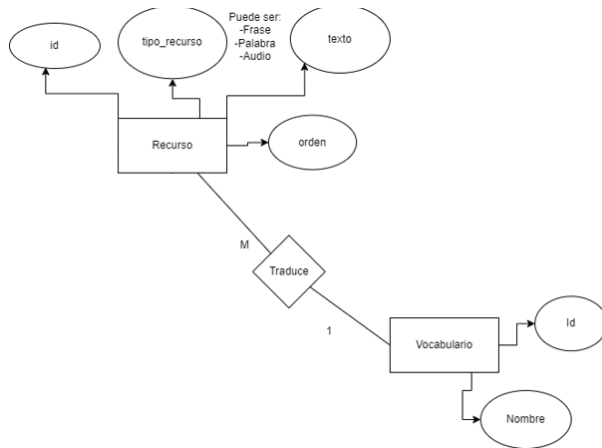
Cada dificultad tendrá unos recursos asociados, no será lo mismo lo que se estudiará en un nivel básico que un nivel avanzado.

Ilustración 10. Relación dificultad – recurso.



Cada recurso estará en un idioma diferente, por lo que tendremos que saber a cuál pertenece.

Ilustración 11. Relación recurso – idioma.



Como hemos descrito en la entidad vocabulario, esta estará compuesta por todas las palabras del sistema, y en recurso tendremos todas las traducciones de esta palabra. Por ejemplo, si en vocabulario tenemos la palabra coche, en recurso tendremos coche y car por lo que esta relación nos dice cuáles son las traducciones de la palabra en vocabulario.

Ilustración 12. Relación recurso – vocabulario.

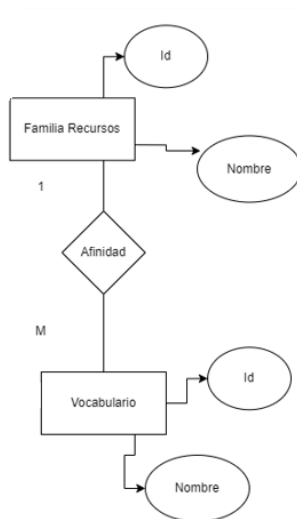


Ilustración 13. Relación vocabulario – familia recursos.

Para que la experiencia del usuario sea mejor, vamos a agrupar las palabras de nuestro vocabulario. Por ejemplo, en familia recursos tendremos guardado transportes y de nuestro vocabulario todas las palabras que tengan que ver con transporte estarán asociadas a esta familia, así cuando a el estudiante le salga para traducir coche, le saldrán opciones parecidas a coche y no algo aleatorio como café, ducha, manzana.

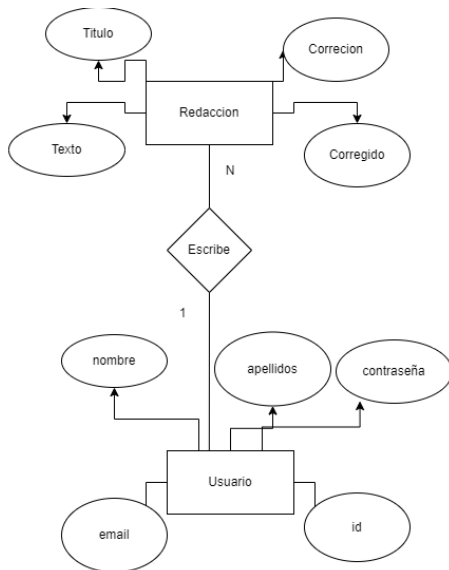


Ilustración 14. Relación usuario – redacción.

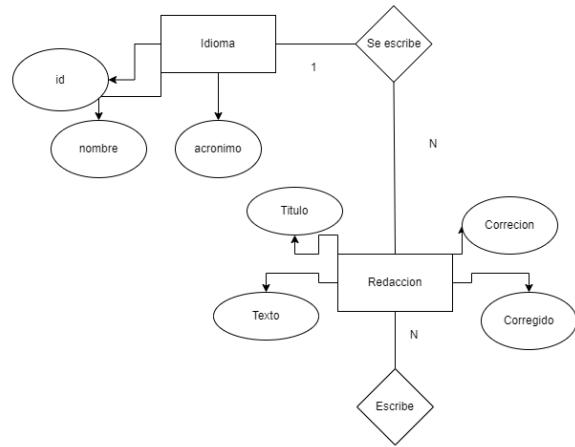


Ilustración 15. Relación idioma – redacción.

Por último, tenemos las relaciones con la tabla redacciones, por un lado, se necesitará saber quién ha escrito cada redacción, y por otro lado saber en qué idioma se ha redactado.

Autómata

El autómata es la parte más importante de la aplicación, ya que es quien va a decidir que estudia el usuario basándonos en unas reglas predefinidas. Como se explicó en la introducción, se pretende usar la repetición espaciada para así mejorar el aprendizaje del usuario, y esto mismo es lo que queremos trasladar a nuestro algoritmo.

En la ilustración de abajo se puede observar el diagrama de flujo del funcionamiento del autómata, para cuando un usuario está estudiando.

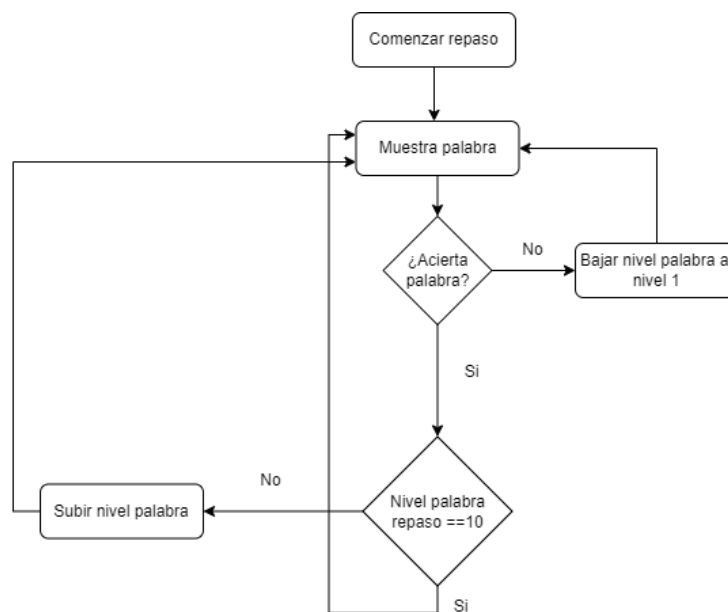


Ilustración 16. Diagrama de flujo del funcionamiento del autómata.

A continuación, se procede a explicar paso a paso su funcionamiento

El usuario le dará a comenzar el repaso para el idioma que tenga en alguna de sus configuraciones.

A continuación, es donde entra en juego el autómata, ya que este deberá seleccionar el recurso idóneo para el usuario.

Para la elección del recurso, primero hay que ver qué nivel de dificultad tiene el usuario, con lo que haremos un primer barrido dejando solo aquellos recursos que sean de ese nivel y ese idioma.

Seguido el algoritmo hará un segundo filtro ordenando los recursos por orden de tiempo más alejado (*Ilustración 5*) y además por un nivel ascendente, es decir, saldrán primero los que tienen un nivel más bajo.

Una vez elegido el recurso a estudiar, se obtiene su traducción para mostrársela al usuario (*Ilustración 10*).

Y, además, elegiremos uno o dos recursos más que tengan relación con él, (*Ilustración 11*). Con ello mostraremos al usuario una palabra a estudiar con varias opciones.

Para la gestión de las palabras, el autómata se basará en sí acierta o falla la respuesta, bajando o subiendo el nivel en cada caso. Y, en cualquier caso, siempre se volverá a la elección del recurso, para continuar con el estudio. En el anexo 1 se muestra fragmentos de código mostrando lo comentado arriba implementado.

Creación del entorno

Una vez está hecha la base de la aplicación, se va a comenzar con la creación del proyecto e instalar todas sus dependencias.

Para ello vamos a empezar creando un proyecto Laravel con el siguiente comando:

- **composer create-project laravel/laravel:^9.0 tfm**

Esto nos va a crear toda la estructura necesaria, además de instalar todas las librerías necesarias para el correcto funcionamiento.

Laravel es un framework de PHP y como ya hemos mencionado al ser un lenguaje interpretado, necesitamos tener la versión 8.0 de PHP instalado en nuestro ordenador.

Una vez tengamos todo esto con ejecutar “**php artisan serve**” en nuestro terminal podremos correr la aplicación en local. (Stackoverflow, 2022)

Antes de comenzar con la parte de implementar código, lo primero que se debe de hacer es cambiar un fichero que nos proporciona Laravel, el cual se llama “**.env**”. En este fichero debemos cambiar unas líneas que harán que nos permita conectarnos con nuestra base de datos.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=tfm
DB_USERNAME=root
DB_PASSWORD=root
```

Ilustración 17. Configuración del entorno para conectarse a la base de datos.

Seguido utilizaremos más comandos de Laravel que nos van a ayudar a crear unos ficheros llamados migraciones, que nos facilitaran la creación de la base de datos partiendo de nuestro modelo entidad relación.

- **php artisan make:model nombre_modelo --migration**

Una vez tengamos todas las entidades y relaciones creadas, procederemos a ejecutar “**php artisan migrate**”, para que todas las migraciones que hemos generado aparezcan en nuestra base de datos.

Aparte de esto vamos a usar una librería que tiene Laravel que nos va a ayudar con la parte de login y registro de nuestra plataforma, para ello hay que ejecutar este comando.

- **composer require laravel/ui:^2.4**

Una vez ejecutado e intentar acceder a la página del login, saltaba un error el cual no dejaba acceder a dicha sección de la web.

Una de las soluciones que se proponía en internet que ejecutar este par de comandos, que lo que nos van a hacer es instalar un par de librerías que necesita nuestro login.

- **npm install --save-dev vite laravel-vite-plugin**
- **npm install --save-dev @vitejs/plugin-vue**

Una vez ejecutados, funcionaba todo correctamente.

Registro



Registro de usuario

* NOMBRE	* APELLIDO
<input type="text"/>	<input type="text"/>
* CORREO ELECTRÓNICO	
<input type="text"/>	
* CONTRASEÑA	* REPITA SU CONTRASEÑA
<input type="text"/>	<input type="text"/>
* IDIOMA	* DIFICULTAD
<input type="text" value="Español"/>	<input type="text" value="Facil"/>

[Registrarse](#)

Ilustración 18. Pantalla de registro de nuevos usuarios.

En la parte del registro, es donde el usuario creará su cuenta y elegirá su primer idioma para estudiar, aunque dentro de la aplicación podrá crear más configuraciones.

Login

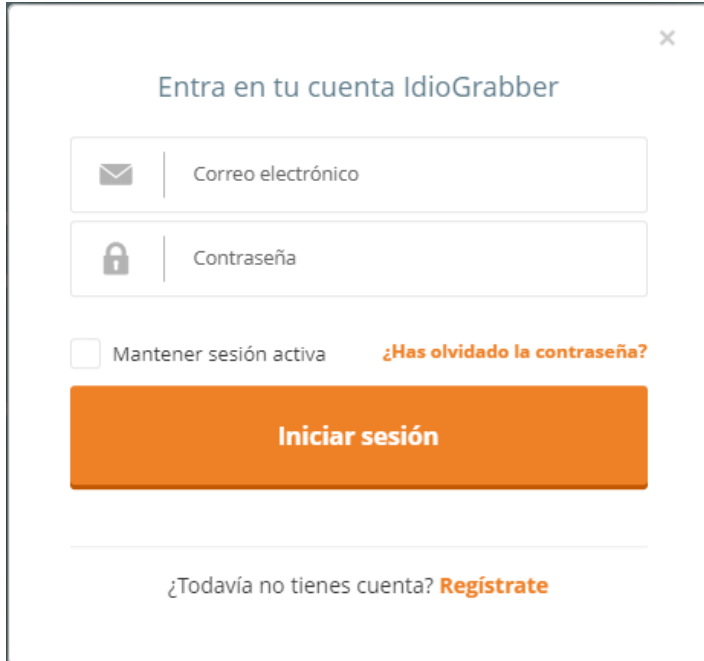


Ilustración 19. Pantalla de login.

Esta es la parte en la cual el usuario podrá entrar en la aplicación o si ha olvidado su contraseña, le llegara un correo a su email. Previamente podrá elegir el idioma en el que quiere ver la aplicación, ya que una vez identificado no podrá hacerlo. (NigmaCode, 2021)



Ilustración 20. Cambio de idioma.

Administrador

Gestión de palabras

	Vocabulario	Idioma	Traducción	Familia
<input type="checkbox"/>	Coche	Español	Coche	Transportes
<input type="checkbox"/>	Coche	Inglés	Car	Transportes
<input type="checkbox"/>	Coche	Francés	Voiture	Transportes
<input type="checkbox"/>	Avion	Español	Avion	Transportes
<input type="checkbox"/>	Avion	Inglés	Plane	Transportes
<input type="checkbox"/>	Avion	Francés	Avion	Transportes
<input type="checkbox"/>	Tren	Español	Tren	Transportes
<input type="checkbox"/>	Tren	Inglés	Train	Transportes
<input type="checkbox"/>	Tren	Francés	Former	Transportes
<input type="checkbox"/>	Autobus	Español	Autobus	Transportes

Ilustración 21. Pantalla de administrador para la gestión de los recursos del sistema.

Esta es la parte en donde los administradores del sistema podrán crear las palabras que luego le saldrán al usuario en la web, además de poder editarlas (pinchando en cualquiera de los atributos de las columnas). O eliminando las que ya no sirvan o estén mal.

Aparte, hay una opción de importar palabras, para la cual primero se habrá tenido que descargar la plantilla Excel. Una vez rellenada la plantilla se subirá, facilitando así la tarea del administrador de no ir creando las palabras una por una.

Los administradores deben de tener en cuenta unos criterios para que las diferentes relaciones entre recursos, ya que no se va a tratar de igual forma una palabra, una frase o un audio.

Para las palabras hay que relacionarlas por temática en su familia, por ejemplo, coche, avión, bus, moto... irán en la familia de transportes, para que cuando busquemos nos salga una palabra relacionada con ella. Y en su vocabulario ira la traducción general de cada palabra.

Las frases pertenecerán todas a la misma familia llamada Frases, ya que para este tipo de recurso nos interesa mostrar varias frases fragmentadas para que el usuario pueda formarlas, por lo que nos da igual que frases se cojan.

Por último, para los audios será parecido a las palabras, solo que, en este caso en vez de agruparse por temática, se agruparan por fonética, es decir, que suenan parecido.

Gestión usuarios

	Nombre	Email	Idioma	Dificultad
<input type="checkbox"/>	Josu	josu_gutierrez@hotmail.com	Inglés	Intermedia
<input type="checkbox"/>	Josu	josu_gutierrez@hotmail.com	Inglés	Facil
<input type="checkbox"/>	Josu	josu_gutierrez@hotmail.com	Español	Intermedia
<input type="checkbox"/>	Josu	josu_gutierrez@hotmail.com	Español	Difícil
<input type="checkbox"/>	Josu	josu_gutierrez@hotmail.com	Francés	Facil
<input type="checkbox"/>	Josu	josu_gutierrez@hotmail.com	Español	Facil

Ilustración 22. Pantalla de administrador para la gestión de las configuraciones del usuario.

En esta parte, se podrá no solo editar el correo del usuario por si se ha equivocado u olvidado, además de editar su configuración por si se ha equivocado de idioma o dificultad.

Gestión de redacciones

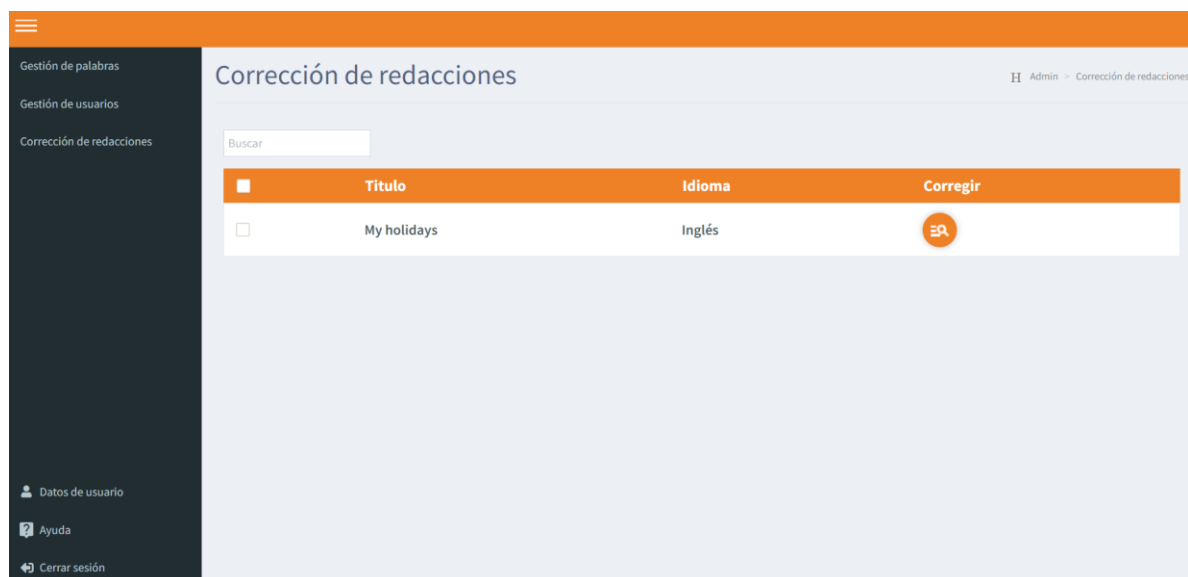


Ilustración 23. Pantalla de administrador para la gestión de las redacciones.

En esta sección los administradores, tendrán un listado con todas las redacciones que han escrito los usuarios y están sin corregir, cada administrador es libre de corregir la redacción que vea oportuna, basándose en el conocimiento que tiene sobre el idioma en el que está redactada.

Usuario

Gestión de datos

Estudios
Redacción

Datos de usuario

Ayuda

Cerrar sesión

Datos de usuario

Cuenta > Datos de usuario

PERFIL

NOMBRE
Josu

APELLIDO
Gutierrez

CORREO ELECTRÓNICO
josu_gutierrez@hotmail.com

Guardar

CONTRASEÑA

CONTRASEÑA ACTUAL

NUEVA CONTRASEÑA

CONFIRMAR NUEVA CONTRASEÑA

Cambiar contraseña

Ilustración 24. Pantalla para la gestión de los datos personales del usuario.

Esta es la parte donde el usuario podrá editar sus datos personales al igual que cambiar la contraseña, cabe destacar que se ha puesto un medidor de contraseña segura, el cual nos dirá como de Segura es la contraseña nueva.

Estudios



Ilustración 25. Pantalla de elección de estudios.

En esta pantalla el usuario podrá continuar con alguno de los estudios que está cursando, además de poder crear un nuevo.

Juego

Esta es la parte principal de la aplicación donde el usuario estará estudiando el idioma que ha elegido previamente en su configuración. Como se comenta previamente hay tres tipos de recursos, que dependiendo de uno u otro se mostrará e interactuará de una forma diferente. (Animate.css, 2023).

Palabras

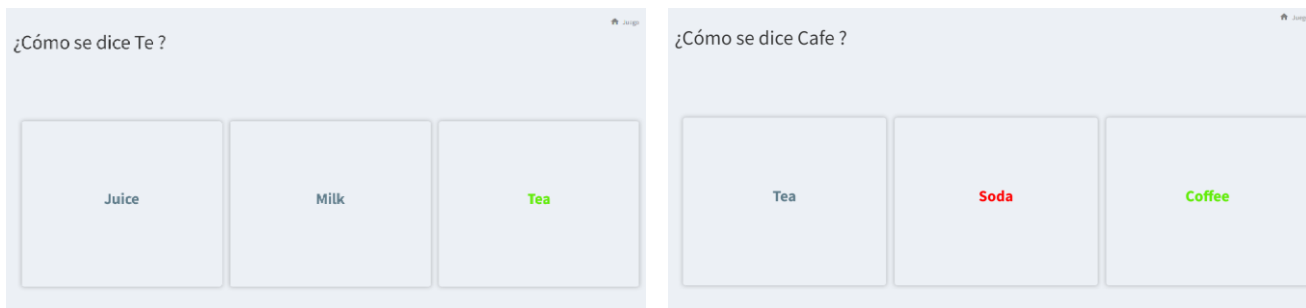


Ilustración 26. Recurso de tipo palabra acertado y fallado.

Nos saldrá un mensaje preguntándonos cual es la traducción de la palabra en cuestión, si acertamos nos saldrá el en verde y si fallamos aparecerá en rojo el recurso que hemos elegido y en verde el correcto, para que el usuario sepa su fallo.

Frases

En este tipo de recurso saldrán varias palabras sueltas las cuales el usuario deberá arrastrar en el orden que crea oportuno, cuando crea que está bien le dará al botón de mandar y hará algo parecido a las palabras, mostrándonos en verde si está bien y en rojo en caso de equivocarnos. (Ivan Salas, 2023) (Stackoverflow, 2020)

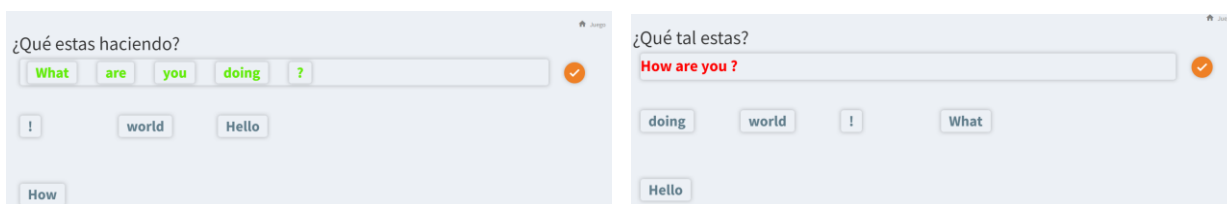


Ilustración 27. Recurso de tipo frase acertado y fallado

Audios



Ilustración 28. Recurso de tipo audio acertado y fallado.

Para los audios, será igual que las palabras, la única diferencia es que en vez de un mensaje aparecerá un audio para reproducir, el cual podrá escucharse tantas veces como quiera el usuario.

Redacciones

Listado

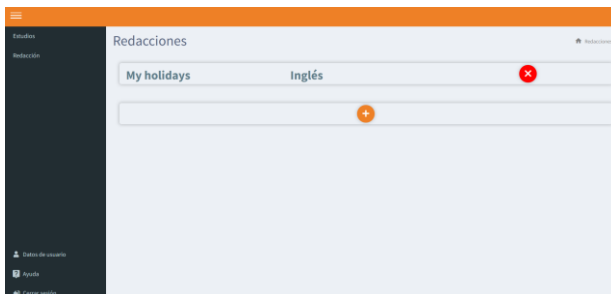


Ilustración 29. Redacción a la espera de ser corregida.

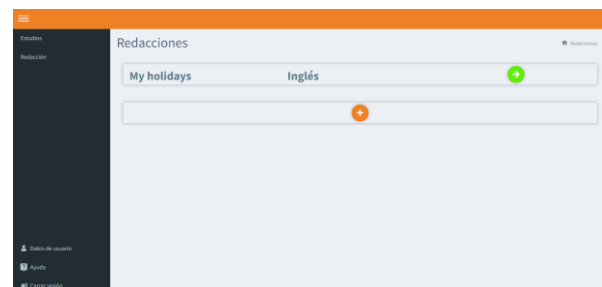


Ilustración 30. Redacción corregida.

En esta parte el usuario podrá ver comprar si alguna de sus redacciones está corregida, además de poder redactar una nueva.

Nueva redacción

El usuario tendrá la posibilidad de escribir una redacción la cual será corregida posteriormente por un administrador. Una vez corregida el usuario visualizará la redacción que ha escrito y debajo su correspondiente corrección.

The screenshot shows a web interface for creating a new essay. On the left is a dark sidebar with a menu containing 'Estudios' and 'Redacción', and at the bottom, links for 'Datos de usuario', 'Ayuda', and 'Cerrar sesión'. The main content area has an orange header bar. Below it, the title 'Nueva redacción' is displayed. To the right of the title is a home icon and the text 'Nueva redacción'. Below the title, there is a 'TITULO' label and a text input field. To the right of the input field is an 'IDIOMA' label and a dropdown menu currently set to 'Español'. Below these fields is a large text area labeled 'Redacción'. Above this text area is a rich text editor toolbar with various icons for text formatting (bold, italic, underline, strikethrough, text color, background color), alignment, indentation, bulleted and numbered lists, link, unlink, and a 'Fuente HTML' button. Below the toolbar are two dropdown menus labeled 'Estilo' and 'Formato', followed by a help icon. At the bottom of the main content area is a large orange button with a white plus sign.

Ilustración 31. Pantalla para escribir una nueva redacción.

5. Servicios en la nube

A continuación, se explican los diferentes servicios web, que harían falta para tener la aplicación funcionando en la nube, el problema que ha surgido es que para la realización de estos sub apartados hace falta disponer de cierto presupuesto en la máquina de Amazon, pero por diversos motivos el presupuesto de esta es en cero euros, por lo que se explicará cómo se hubieran utilizado los diversos recursos.

AWS Elastic Beanstalk

Este servicio permite desplegar aplicaciones web sin tener que el desarrollador tenga que preocuparse por la gestión de la máquina, es decir, crear el Sistema operativo, decidir cuánto va a ocupar, crear e instalar las diversas configuraciones para que la aplicación pueda correr, etc ...

Para complementar este servicio utilizaremos Gitlab CI/CD que es una herramienta de integración continua y despliegue continuo, esto permite al desarrollador subir al servidor cambios sin tener que estar pendiente de entrar a la maquina y tener que bajarlos el mismo, además de que se puede configurar para antes de hacer el despliegue se realicen ciertas comprobaciones para ver si no hay ningún tipo de fallo en el código subido.

Para realizar esta parte primero hace falta tener el código subido en un repositorio de Gitlab, y tener configurada una clave para que se coordine con el servicio de Amazon.

Se creara un fichero con extensión .yaml para disparar las acciones de CI/CD de Gitlab.

Una vez se tiene todo se ejecutará dos comandos:

- **git commit -am "ci file"**
- **git push**

Los cuales dispararán el pipeline de CI/CD y procederá al desligue de la aplicación, si todo ha ido bien en un rato veremos nuestra aplicación funcionando, en cambio si tenemos algún error en la configuración nos avisara de que está mal. (Amazon, 2023)




Status	Pipeline	Triggerer	Stages
<div>passed</div> <div>00:03:55</div> <div>just now</div>	<div>"</div> <div>#767508019</div> <div>main</div> <div>6251671e</div> <div>latest</div>		<div>✓✓</div> <div>Download</div>
<div>failed</div> <div>7 minutes ago</div>	<div>fix</div> <div>#767507043</div> <div>main</div> <div>6d2aa945</div> <div>yaml invalid error</div>		<div>Download</div>
<div>failed</div> <div>9 minutes ago</div>	<div>"</div> <div>#767506917</div> <div>main</div> <div>91141ed4</div> <div>yaml invalid error</div>		<div>Download</div>

Ilustración 32. Resultado de del CI/CD.

AWS RDS

Ya que nuestra aplicación va a estar alojada en Elastic Beanstalk, necesitamos un servicio que se ocupe de nuestra base de datos. Gracias a esto evitaremos las tareas como gestionar la base de datos, los backups o estar pendiente de su mantenimiento. (Amazon, 2023).

Para crear la base de datos tenemos tres opciones diferentes, la primera de ellas es ir al panel de control de Amazon y hacer manualmente, la segunda sería desde la consola de comandos de Amazon y por último sería utilizando Terraform, que es un lenguaje de configuración de alto nivel para definir infraestructura. Esta última opción es la más cómoda ya que introduciendo unas cuantas líneas sencillas de código nos va a permitir configurar nuestra base de datos con los parámetros que queramos de forma sencilla, otra ventaja que nos ofrece es que, para modificar nuestra configuración de la base de datos o su eliminación, en caso de que haga falta Terraform facilita mucho estas tareas sin tener que estar entrando en la consola de Amazon. (Terraform, 2023).

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.16"
    }
  }
  required_version = ">= 1.2.0"
}

provider "aws" {
  region = "us-east-1"
}

resource "aws_db_instance" "idioGrabber" {
  identifier      = "idioGrabber"
  engine          = "mysql"
  engine_version = "5.7.40"
  instance_class  = "db.t3.micro"
  db_name         = "idioGrabber"
  username        = "admIdio"
  password        = "IdioGrabber5674"
  backup_window   = "03:00-04:30"
  maintenance_window = "Mon:05:00-Mon:07:30"
  storage_type     = "gp2"
  allocated_storage = 15
  backup_retention_period = 15
  storage_encrypted = true
}
```

Ilustración 33. Creación de la base de datos mediante Terraform.

AWS S3

Este servicio que proporciona Amazon, permite guardar una gran cantidad de objetos en la nube y accesibles en cualquier momento y lugar.

Alguna de las características más importantes que nos ofrece es que tiene una gran escalabilidad y una gran seguridad.

La idea de usar este servicio sería para no saturar nuestro servicio de Elastic Beanstalk, ya que, para los recursos de audios, aparte de que el administrador cree el recurso, este deberá de crear un archivo de audio el cual se llamará igual que el recurso. Estos archivos serán de tipo .mp4, los cuales individualmente no tienen un gran tamaño, por lo que si solo se usarían cien audios en toda la aplicación, merecería la pena guardarlo en el propio servicio de Elastic Beanstalk. Pero en este caso vamos a tener miles de audios, por lo que necesitamos un servicio donde poder alojarlos.

El sitio donde se almacenan estos ficheros se denomina 'Bucket', para la creación de uno de estos es muy sencillo, basta con ir a nuestra consola o panel de control de Amazon Web Service y buscar la opción de S3. Una vez dentro le daremos a crear nuevo Bucket con la configuración que nosotros deseemos.

Crear bucket Info

Los buckets son contenedores de datos almacenados en S3. [Más información](#)

Configuración general

Nombre del bucket

El nombre del bucket debe ser único en todo el mundo y no debe contener espacios ni letras mayúsculas. Consulte las reglas para la denominación de los buckets.

Región de AWS

EE. UU. Este (Norte de Virginia) us-east-1

Copiar la configuración del bucket existente: opcional

Solo se copia la configuración del bucket en los siguientes casos:

Propiedad de objetos Info

Controla la propiedad de los objetos escritos en este bucket desde otras cuentas de AWS y el uso de listas de control de acceso (ACL). La propiedad de los objetos determina quién puede especificar el acceso a los objetos.

☒ **ACL deshabilitadas (recomendado)**
Todos los objetos de este bucket son propiedad de esta cuenta. El acceso a este bucket y sus objetos se especifica solo mediante políticas.

☐ **ACL habilitadas**
Los objetos de este bucket pueden ser propiedad de otras cuentas de AWS. El acceso a este bucket y sus objetos se puede especificar mediante ACL.

Propiedad del objeto

Aplicada al propietario del bucket

Configuración de bloqueo de acceso público para este bucket

Se concede acceso público a los buckets y objetos a través de listas de control de acceso (ACL), políticas de bucket, políticas de puntos de acceso o todas las anteriores. A fin de garantizar que se bloquee el acceso público a todos sus buckets y objetos, active Bloquear todo el acceso público. Esta configuración se aplica exclusivamente a este bucket y a sus puntos de acceso. AWS recomienda activar Bloquear todo el acceso público, pero, antes de aplicar cualquiera de estos ajustes, asegúrese de que las aplicaciones funcionarán correctamente sin acceso público. Si necesita cierto nivel de acceso público a los buckets u objetos, puede personalizar la configuración individual a continuación para adaptarla a sus casos de uso de almacenamiento específicos. [Más información](#)

Cifrado predeterminado Info

El cifrado del lado del servidor se aplica automáticamente a los nuevos objetos almacenados en este bucket.

Tipo de clave de cifrado Info

☒ **Claves administradas por Amazon S3 (SSE-S3)**
☐ Clave de AWS Key Management Service (SSE-KMS)

Clave de bucket

Cuando se utiliza el cifrado de KMS para cifrar nuevos objetos en este bucket, la clave del bucket reduce los costos de cifrado al reducir las llamadas a AWS KMS. [Más información](#)

☐ Desactivar
 ☒ **Habilitar**

Ilustración 34. Creación del Bucket de S3.

Para subir los archivos tenemos dos formas, una sería manualmente mediante el panel de control del usuario de Amazon, otra sería mediante la consola de comandos de Amazon y por último se podría programar en el proyecto una funcionalidad para que el administrados elija varios archivos y se suban al momento. (Amazon, 2023)

6. Resultados y discusión

Una vez finalizado todo el desarrollo, la aplicación ha quedado bastante completa en lo que se refiere a usabilidad, a continuación, se hará un pequeño resumen de toda la funcionalidad disponible.

Por un lado, tenemos la parte del administrador, que serán las personas encargadas de llevar la aplicación. Para ello tienen a su disposición diferentes herramientas como es la creación de recursos, tanto de forma manual o mediante la importación de un archivo Excel, para que así no sea tan tedioso. Además de poder editar los recursos por si se han equivocado en alguno, o lo desean eliminar. También tienen la opción de poder gestionar las configuraciones de los usuarios, que son sus estudios, es decir, el idioma y la dificultad que tiene. Los administradores podrán tanto eliminar una configuración o editarla, ya que un usuario en su propia interfaz no podrá. Por último, dispondrán de una opción para corregir redacciones, en esta parte los administradores tendrán una tabla con las diferentes redacciones que han escrito los usuarios y en base a la experiencia que tiene cada uno en será libre de corregir o no.

Por otro lado, los usuarios tendrán la opción de elegir en que idioma van a querer ver todo el contenido de la página, una vez elegido podrán registrarse en la web, creando una primera configuración para empezar a estudiar.

Una vez identificados les saldrá la página de los estudios, que será una lista de todas las configuraciones que tiene creadas para estudiar, y dándole al botón de jugar en cada una de ellas, comenzara a estudiar.

El juego está dividido en tres tipos de recursos, tenemos las palabras en las que se mostrará un texto con una palabra a traducir en el idioma que se esté jugando, una vez elegido la

opción, el sistema nos dirá si es correcta marcándola en verde o incorrecta en rojo. También tenemos las frases que nos saldrá un cuadro grande donde tendremos que formar la frase que nos indica el sistema, para ellos tendremos que ir arrastrando palabras sueltas que tenemos en la pantalla, para así formar la respuesta final.

Y por último tenemos los audios, que funcionan de forma similar a las palabras, la única diferencia es que en vez de un texto a traducir nos reproducirá un audio y dependiendo lo que escuchemos, elegiremos una opción u otra.

Al igual que el administrador puede corregir redacciones, el usuario podrá redactar una, seleccionando el idioma en el que va a ser escrita y proporcionando un título. Una vez que un administrador haya corregido alguna de sus redacciones le saldrá la opción de poder ver dicha corrección.

Para finalizar el usuario dispondrá de una opción para cambiar sus datos personales y su contraseña, además de un apartado de ayuda donde se le facilitara un teléfono de contacto por si tienen algún problema.

7. Conclusiones

Una vez finalizado el proyecto, llega el momento de echar un vistazo a todo lo realizado. No ha sido sencillo llegar hasta aquí, ya que algunas veces se ha tenido que retroceder para poder seguir para adelante sin ningún tipo de problema, esta es alguna de las conclusiones que se pueden sacar de realizar este trabajo aparte de la capacidad de poder desarrollar un proyecto de este nivel partiendo desde cero.

Se han conseguido realizar casi todos los objetivos que se plantearon en un inicio, ya que como se menciona en el apartado de servicios en la nube, esa parte no se ha realizado por problemas con el presupuesto.

Uno de los posibles riesgos era una planificación temporal incorrecta, pero por suerte se ha podido llegar bien a todos los plazos, sin ningún tipo de contratiempo, incluso en la parte de administrador se ha podido acabar antes de tiempo, por lo que ha permitido adelantar el resto de fases. Esto ha ayudado a que en mitad del desarrollo surgió la idea de crear el apartado de las redacciones, ya que se vio que era algo que ninguna de las otras páginas webs en el mercado tenía. Para ello se tuvo que revisar la base de datos y añadir los modelos correspondientes, esto no supuso un gran problema ya que, la base de datos estaba desde un principio bien diseñada, por lo que permitió añadir esta funcionalidad sin ningún tipo de problema.

En cuanto al desarrollo del proyecto, lo que más ha podido costar ha sido pensar y diseñar la parte frontend, ya que se quería una estética limpia y, pero a la vez vistosa, sin que tuviera mucha animación ni mucha imagen para no dar la sensación de ser una aplicación solo para niños. Se dedicó tiempo a la investigación de diferentes tipos de librerías, para que ayudaran a las animaciones que se muestran mientras se está jugando, para darle un toque más

vistoso y que no se vuelva tan monótono el estudio. También la gestión de las tablas en la parte del administrador ha sido un poco tediosa, ya que la librería que se ha usado ha habido que tocar mediante Javascript para poder tener hacer todas las funcionalidades que se querían. Al igual que la validación del Excel en la parte frontend, es algo que ha costado bastante, ya que había muy poca documentación por internet y al final mediante prueba y error y comprendiendo como se cargaba el Excel se ha podido realizar.

En cuanto a la parte del backend no ha habido tanto problema, ya que la elección de Laravel ha sido la correcta para este trabajo, sus numerosas librerías y su gestión con la base de datos a ayudado mucho a la realización del proyecto. La parte que más ha costado ha sido la implementación del autómata, ya que se tenía que tener en cuenta varios factores a la hora de la elección del recurso idónea para el usuario.

Como valoración general del proyecto, debo de decir que estoy satisfecho con el resultado. La idea de la repetición espaciada me parece que es bastante desconocida y podría ayudar no solo a estudiar idiomas si no que ha diferentes materias.

8. Trabajo futuro

En cuanto al futuro de la aplicación no se descarta seguir con ella, para ello se necesitaría contar con un grupo de personas expertas en diversos idiomas, para la parte de administración, ya que serían los encargados de gestionar todos los tipos de recursos.

En caso de conseguir un grupo que estaría dispuesto a trabajar, lo siguiente sería la realización del apartado de servicios en la nube, ya que para que la gente pueda registrarse es necesario que este en la nube.

Se implementarían nuevas funcionalidades que por tiempo no se han podido desarrollar, y al igual que en las redacciones es algo que no está en las aplicaciones del mercado, como sería la parte de los “Listening”, la lectura de texto con sus preguntas para ver la comprensión lectora del usuario, todo ello siempre adecuado al nivel del usuario. Además, otra idea que se planteó fue la de describir una imagen, que funcionaría similar a las redacciones. Un usuario describiría una imagen ya sea una otorgada por el sistema o una que el decida subir y un administrador del sistema le corregiría. Podría añadirse alguna nueva funcionalidad al autómata, o desarrollar uno nuevo para que el usuario tenga dos opciones de elegir qué nivel quiere estudiar. Si sabe más o menos su nivel, tiene la opción de elegirlo, como está implementado ahora, pero también la opción de que se le haga un test y en base a esas respuestas, el autómata determine un nivel de estudio. Y como última mejora, sería implementar una inteligencia artificial que permita al usuario decir una palabra o una frase y la IA la compare con un audio previamente guardado en el sistema y le diga al usuario si su pronunciación es correcta.

Más adelante se podría contratar a más gente no solo para la parte de administración, si no para la de desarrollo y hacer que IdioGrabber no solo sea una aplicación de idiomas, si no

que expanda a otros sectores, como podrías ser historia, geografía, ya que como se ha mencionado varias veces la repetición espacia puede aplicarse a varios sectores.

9. Anexo I: Código automata

En este anexo se mostrará el código relacionado con el autómata, tanto la obtención del recurso, como la comprobación de la respuesta.

```
public function jugar(Request $request,$id){

    $usuario = Auth::user();
    $configuracion = configuracion::where('id','=',$id)->get()->first();

    $juego = [];
    //Query que obtiene el recurso mas antiguo y con menos nivel
    $recurso = recurso::select('recursos.*')
    ->join('dificultad_recursos','recursos.id','=',$dificultad_recursos.recurso_id')
    ->join('estudios','recursos.id','=',$estudios.recurso_id')
    ->where('dificultad_recursos.dificultad_id','=',$configuracion->dificultad_id)
    ->where('recursos.idioma_id','=',$configuracion->idioma_id)
    ->where('estudios.usuario_id','=',$usuario->id)
    ->where('fecha_ultima_repeticion','!=',Carbon::now()->format('Y-m-d'))
    ->orderBy('fecha_ultima_repeticion','ASC')
    ->orderBy('nivel','ASC')
    ->orderBy('orden','ASC')
    ->get()->first();

    if($recurso){
        if($recurso->tipo_recurso == 'Palabra' || $recurso->tipo_recurso == 'Audio'){
            //Si es una palabra o un audio sacaremos mas recursos aleatorios que esten ligados a su familia.
            $traduccion = $recurso->vocabulario->nombre;
            array_push($juego,$recurso->id);

            if($recurso->tipo_recurso == 'Palabra'){
                $vocabularios = vocabulario::where('familia_id','=',$recurso->vocabulario->familia_id)
                ->where('id','!=',$recurso->vocabulario->id)
                ->inRandomOrder()->limit(2)->get();
            }else{
                $vocabularios = vocabulario::where('familia_id','=',$recurso->vocabulario->familia_id)
                ->where('id','!=',$recurso->vocabulario->id)
                ->inRandomOrder()->limit(3)->get();
            }
        }
    }
}
```

Ilustración 35. Selección de recurso 1

```
$vocs = [];
foreach($vocabularios as $v){
    array_push($vocs,$v->id);
}

$recursosFalsos = recurso::whereIn('vocabulario_id',$vocs)
->where('idioma_id','=', $configuracion->idioma_id)
->get();

foreach($recursosFalsos as $r){
    array_push($juego,$r->id);
}

$recursosJuego = recurso::whereIn('id',$juego)->inRandomOrder()->get();

return view('business.home.envios.juego', ['traduccion' => $recurso,'recursos' => $recursosJuego]);
```

Ilustración 36. Selección de recurso 2

```
else if($recurso->tipo_recurso == 'Frase'){
    //Si es una frase obtendremos dos frases mas aleatorias
    $traduccion = $recurso->vocabulario->nombre;
    array_push($juego,$recurso->id);

    $vocabularios = vocabulario::where('familia_id','=', $recurso->vocabulario->familia_id)
->where('id','!=', $recurso->vocabulario->id)
->inRandomOrder()->limit(2)->get();

    $vocs = [];
    foreach($vocabularios as $v){
        array_push($vocs,$v->id);
    }

    $recursosFalsos = recurso::whereIn('vocabulario_id',$vocs)
->where('idioma_id','=', $configuracion->idioma_id)
->get();

    foreach($recursosFalsos as $r){
        array_push($juego,$r->id);
    }

    $recursosJuego = recurso::whereIn('id',$juego)->inRandomOrder()->get();

    $palabras = [];
    //Descomponemos las frases en palabras
    foreach($recursosJuego as $r){
        $texto = explode(' ', $r->texto);
        $palabras = array_merge($palabras,$texto);
    }
    $resultado = array_unique($palabras);
    //Barajamos las palabras
    shuffle($resultado);

    return view('business.home.envios.juego', ['traduccion' => $recurso,'recursos' => $resultado]);
```

Ilustración 37. Selección de recurso 3

En esta parte de código se puede ver la sentencia que elegirá nuestro recurso para luego mostrar a el usuario. Luego dependiendo el tipo de recurso que sea hará unas operaciones u otras.

```
public function comprobar(Request $request){
    $usuario = Auth::user();
    $correcto = $request->correcto;
    $respuesta = $request->recurso;
    $resultado = 'false';
    $recurso = recurso::where('id','=',$correcto)->get()->first();

    $estudio = estudio::where('usuario_id','=',$usuario->id)
    ->where('recurso_id','=',$correcto)
    ->get()->first();

    //Comprobamos que el recurso sea correcto
    if($recurso->tipo_recurso == 'Palabra' || $recurso->tipo_recurso == 'Audio'){
        if($correcto == $respuesta){
            if($estudio->nivel < 10){
                $estudio->nivel = $estudio->nivel + 1;
            }
            $resultado = 'true';
        }else{
            if($estudio->nivel > 1){
                $estudio->nivel = $estudio->nivel - 1;
            }
            $resultado = 'false';
        }
    }
}
```

Ilustración 39. Comprobar respuesta 1.

```
}else{
    if($recurso->texto == $respuesta){
        if($estudio->nivel < 10 ){
            $estudio->nivel = $estudio->nivel + 1;
        }
        $resultado = 'true';
    }else{
        if($estudio->nivel > 1){
            $estudio->nivel = $estudio->nivel - 1;
        }
        $resultado = $recurso->texto;
    }
}

$estudio->fecha_ultima_repeticion = Carbon::now();
$estudio->save();
return $resultado;
```

Ilustración 38. Comprobar respuesta 2.

En este código se muestra la forma en la que el autómata comprobará la solución introducida por el usuario, y en base a si es correcta o no hará unas acciones u otras.

10. Bibliografía

Domotizar.com. (2018). *Ventajas de utilizar la repetición espaciada*. Recuperado el 20 de Febrero de 2023 de <https://www.domotizar.com/ventajas-de-utilizar-la-repeticion-espaciada/>

Alex Figueroba. (2017). *Repetición espaciada qué es y como usarla*. Recuperado el 20 de Febrero de 2023 de <https://psicologiymente.com/inteligencia/repeticion-espaciada>

Memrise. (2023). Recuperado el 22 de Febrero de 2023 de <https://www.memrise.com/es/>

Duolingo. (2023). Recuperado el 22 de Febrero de 2023 de <https://www.duolingo.com/>

Nicolas Molina. (2023). *Que es Angular*. Recuperado el 25 de Febrero de 2023 de <https://platzi.com/clases/2478-angular/41035-que-es-angular-y-como-aprenderlo/>

Juan Maria Hernandez. (2013). *AngularJS: Lo bueno y lo malo*. Recuperado el 25 de Febrero de 2023 de <https://blog.koalite.com/2013/06/angularjs-lo-bueno-y-lo-malo/>

Logica7web. (2023). *Reactjs vs react native diferencias clave ventajas y desventajas*. Recuperado el 25 de Febrero de 2023 de <https://logica7web.es/apps/reactjs-vs-react-native-diferencias-clave-ventajas-y-desventajas/>

Keepcoding. (2022). *El porqué usar JavaScript*. Recuperado el 25 de Febrero de 2023 de https://keepcoding.io/blog/porque-usar-javascript/#Ventajas_de_JavaScript

Carlos Herrera. (2020). *Ventajas y desventajas de usar JavaScript en programacion web*. Recuperado el 25 de Febrero de 2023 de <https://bloqueropro.com/blog/ventajas-y-desventajas-de-usar-javascript-en-la-programacion-web>

Ayudaley. (2023). *Base de datos no relacional. ¿Qué es? Características y ejemplos*. Recuperado el 25 de Febrero de 2023 de https://ayudaleyprotecciondatos.es/bases-de-datos/no-relacional/#Que_es_una_base_de_datos_no_relacional_Definicion

Asler Cedeño. (2020). *Ventajas y desventajas de usar ASP.NET en la programación web*. Recuperado el 26 de Febrero de 2023 de <https://blogueroapro.com/blog/ventajas-y-desventajas-de-usar-asp-net-en-la-programaci%C3%B3n-web>

Estrada web group. (2023). *Ventajas y desventajas de ASP:NET Core*. Recuperado el 25 de Febrero de 2023 de <https://estradawebgroup.com/Post/Ventajas-y-desventajas-de-ASP-NET-Core/20611>

Stackoverflow. (2022). *Vite manifest not found at c users hp example app public build manifest-json*. Recuperado el 5 de Marzo de 2023 de <https://stackoverflow.com/questions/72798465/vite-manifest-not-found-at-c-users-hp-example-app-public-build-manifest-json>

Animate.css. (2023). Recuperado el 10 de Marzo de 2023 de <https://animate.style/>

Ivan Salas. (2023). *Puzzle Drag & Drop HTML5*. Recuperado el 10 de Marzo de 2023 de <https://codepen.io/isc7/pen/ARBpKv>

Stackoverflow. (2020). *Get span elements within div using javascript*. Recuperado el 10 de Marzo de 2023 de <https://stackoverflow.com/questions/60638334/get-span-elements-within-div-using-javascript>

NigmaCode. (2021). *Multilenguaje en Laravel*. Recuperado el 11 de Marzo de 2023 de <https://www.nigmacode.com/laravel/multilenguaje-en-laravel/>

Amazon. (2023). Recuperado el 15 de Marzo de 2023 de <https://aws.amazon.com/es/>

Terraform. (2023). Recuperado el 20 de Marzo de 2023 de <https://www.terraform.io/>