



Universidad
Internacional
de Valencia

Trabajo Fin de Máster:
Título del TFT

Titulación:

XXXX

Curso académico
202X – 202X

Alumno/a: Apellido Apellido.

Nombre

D.N.I: XXXX

Director/a de TFM: XXXX

Convocatoria:

Primera o Segunda

Índice

1. Introducción	2
2. Historia del arte	3
3. Especificación del Proyecto	5
4. Planificación.....	6
5. Desarrollo	9
Base de datos	9
Autómata	16
Creación del entorno.....	18

1. Introducción

Hoy en día, vivimos en una sociedad en la cual, el uso de internet es tan común como comer. 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.

Diferentes estudios han demostrado que la memoria de las personas funciona de forma similar, independientemente de su CI (Coeficiente Intelectual). Ya que mediante patrones de repetición los humanos somos capaz de aprender diferentes tipos de recursos.

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.

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.

2. Historia del arte

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

Está basado en un sistema de secciones y a medida que vas acertando, vas pasando de sección.

Luego tenemos a Memrise 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.

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

Para la parte front 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.

El problema de estas, es que para dominarlas hay que hacer algún curso que te permita tener conocimientos. Para la gente que no tiene tiempo o ganas, está la opción de hacer con lo más básico, que sería una combinación de HTML + CSS + JavaScript.

De esta forma funcionalmente quedara bien, pero hay que hacer mucho más esfuerzo, y requiere de más tiempo que si se usa alguno de los frameworks mencionados anteriormente.

Para el back debería de utilizarse un lenguaje interpretado, esto nos va a permitir que nuestro programa no tenga que compilarse cada vez que queramos probarlo. Por ello nos ahorrará mucho tiempo a la hora de ir probando y viendo cómo queda nuestra aplicación. Uno de los más potentes para web es PHP, el cual tiene a su vez frameworks que facilitan la vida al programador.

En cuanto a la base de datos, se utiliza una de tipo relacional, ya que se necesita guardar relación de que está estudiando cada usuario, además de toda su configuración personalizada. A parte de la relación y traducción que puede haber entre palabras.

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. 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 he decidido usar PHP junto con su framework Laravel. Me he decantado por esta opción, porque aparte de ser el lenguaje que hemos visto durante el curso en diversas asignaturas, es con el que llevo yo trabajando varios años ya, y con el que me siento más cómodo a la hora de desarrollar un proyecto desde 0.

Para la parte del frontend, me he decantado por usar HTML + JavaScript + JQuery. Esta elección se debe a que aparte de que tengo soltura con estas herramientas, para poder utilizar Angular o React tendría que hacer algún curso para mejorar mis capacidades ya que solo tengo alguna pincelada básica de cómo usarlos, y ya que no dispongo de mucho tiempo para poder invertir en poder mirar estos cursos, ha sido por lo que me he decidido por lo comentado arriba.

En cuanto a la base de datos, he decidido utilizar una base de datos relacional, ya que necesito tener muchas entidades relacionadas entre sí. En este caso usare MariaDB, ya que tengo bastante practica con ella.

Como control de versiones utilizaré la plataforma Github, que me permitirá llevar un control sobre mi proyecto de forma sencilla.

Para el despliegue de la aplicación, se utilizará una infraestructura PaaS proporcionada por AWS que se llama Elastic Beanstalk. He decidido usar este servicio ya que los PaaS nos proporciona toda la infraestructura, como desarrollador solo tengo que encargarme de subir el código.

4. Planificación

A continuación, se va a explicar de forma detallada la planificación que se va a seguir a lo largo de todo el proyecto. Cabe destacar la 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.

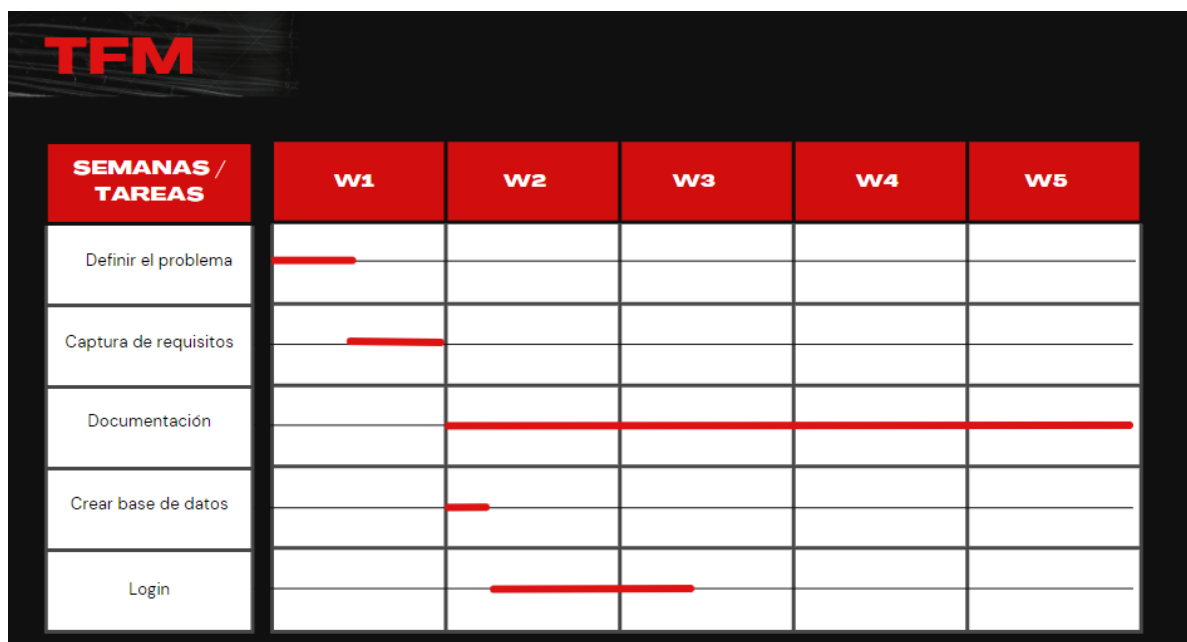


Ilustración 1

La primera semana del proyecto se va dedicar entera a el 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, ya que a medida que vamos avanzando programáticamente iremos documentado.

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.

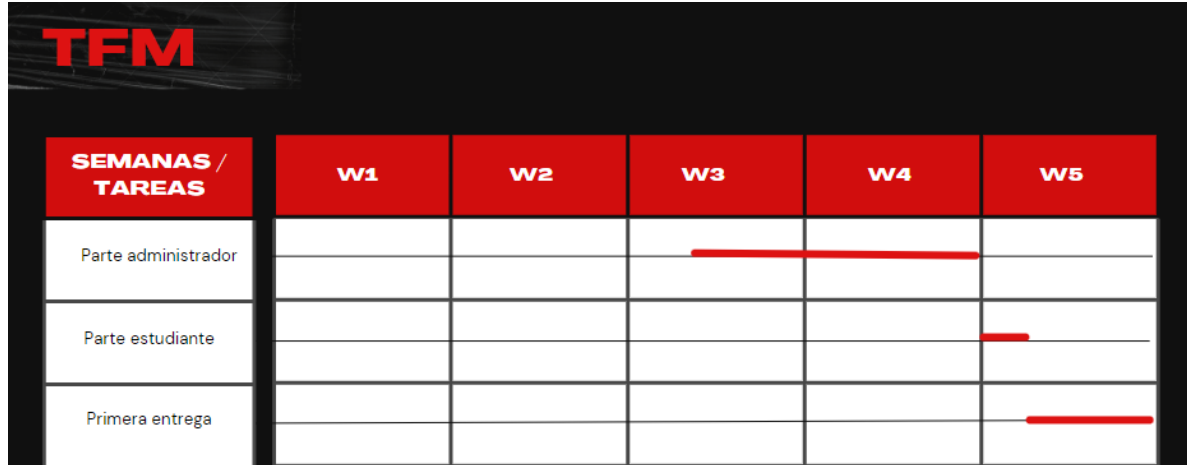


Ilustración 2

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.

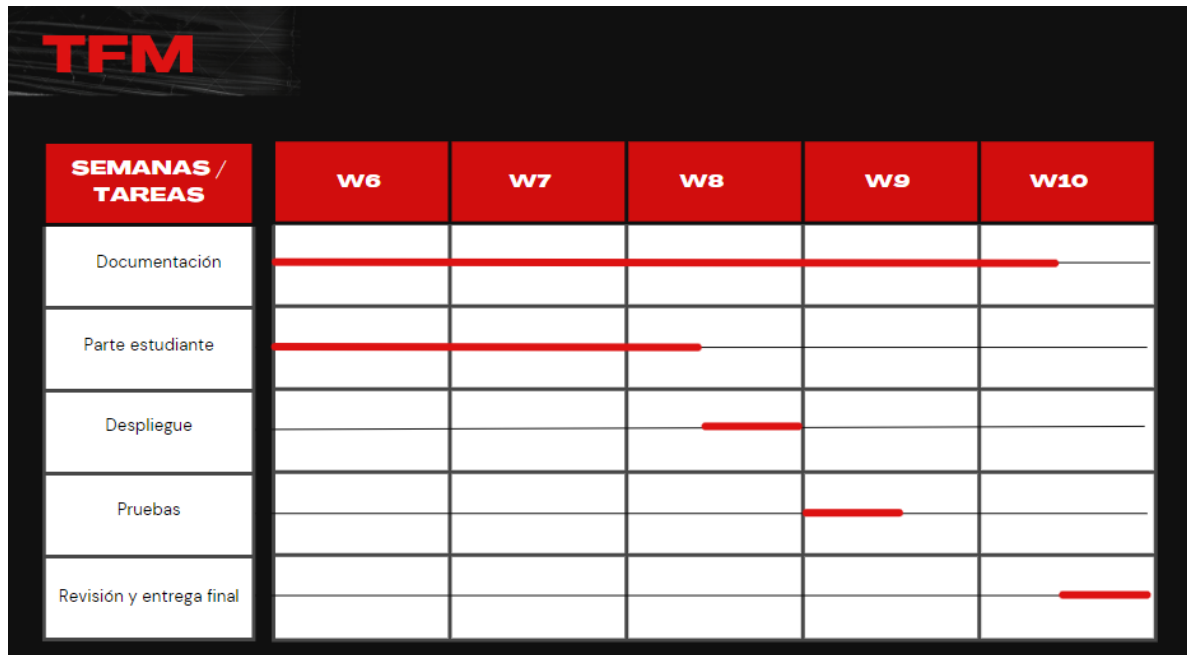


Ilustración 3

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.

Y por último sería la entrega final del proyecto.

5. 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.

A continuación, se muestra el modelo entidad relación de la base de datos de la aplicación. Como es muy grande se va a exponer en dos imágenes, para su correcta visualización.

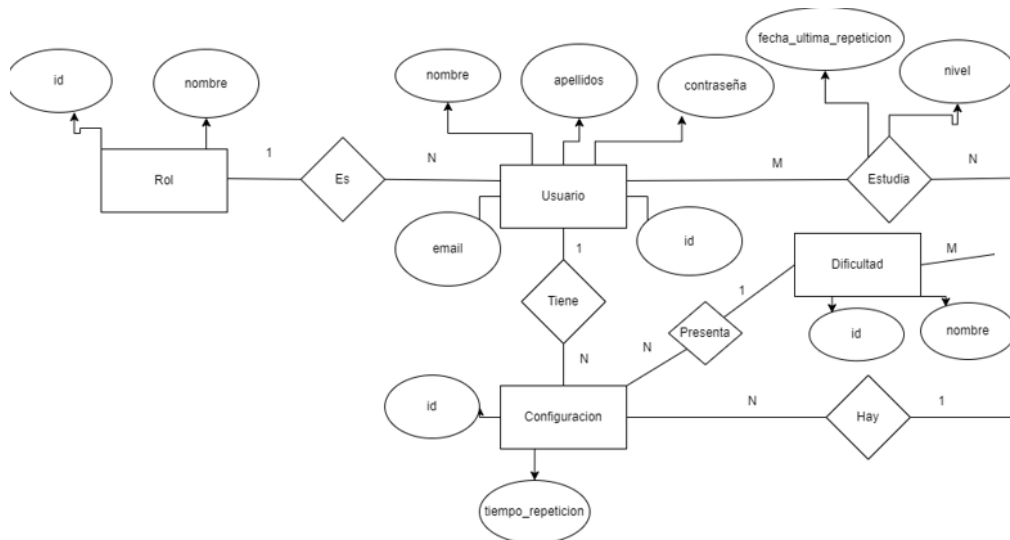


Ilustración 4

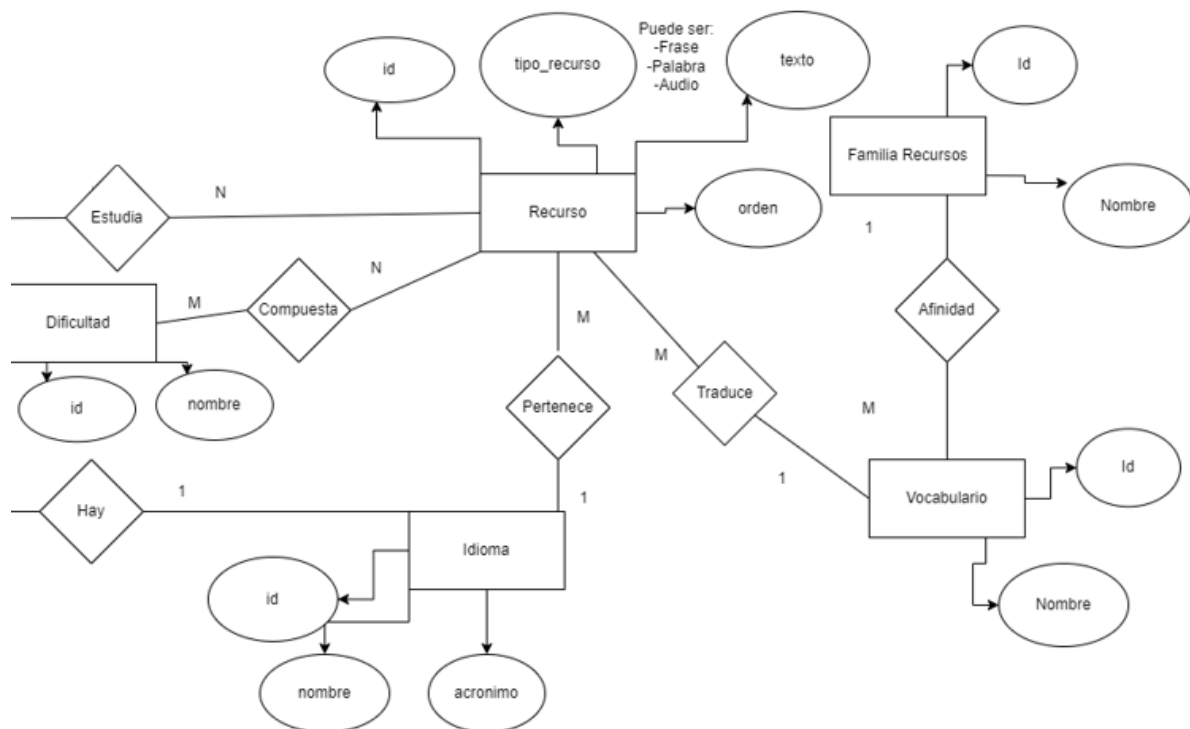


Ilustración 5

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 a el 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.

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.

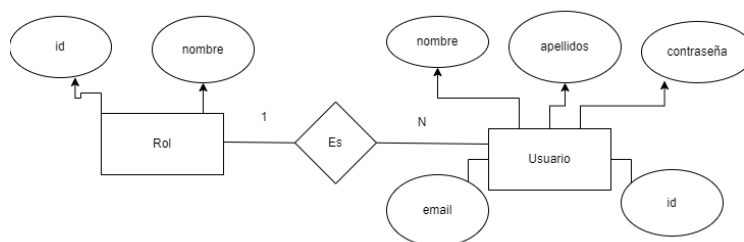
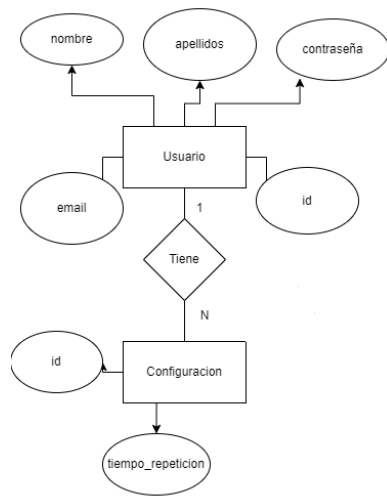


Ilustración 6

Cada usuario va a tener un rol asignado y dependiendo de él podrá hacer unas cosas u otras.



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 7

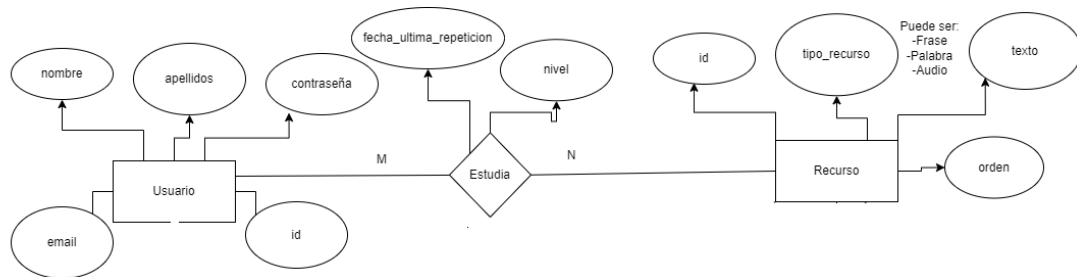


Ilustración 8

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.

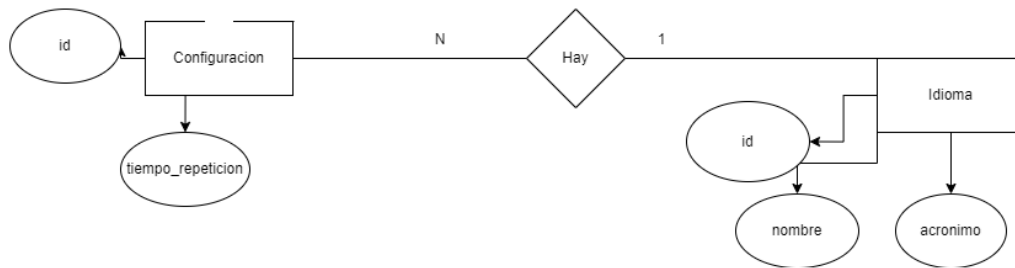
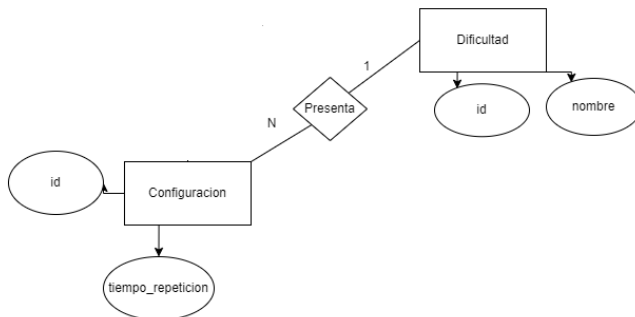


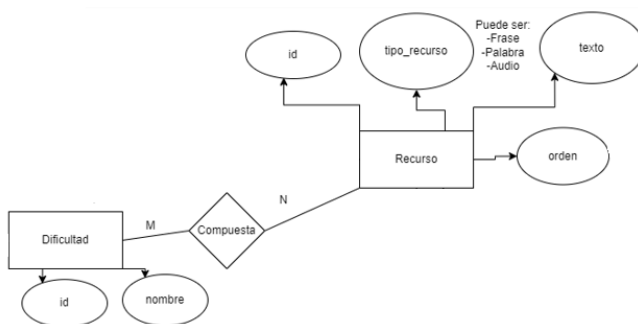
Ilustración 9

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 10



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

Ilustración 11

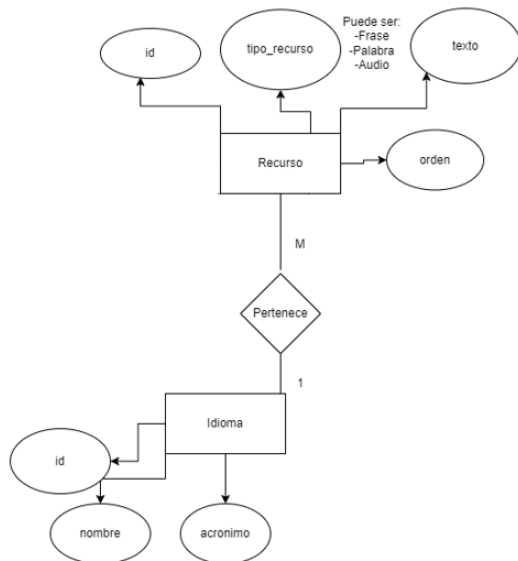


Ilustración 12

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

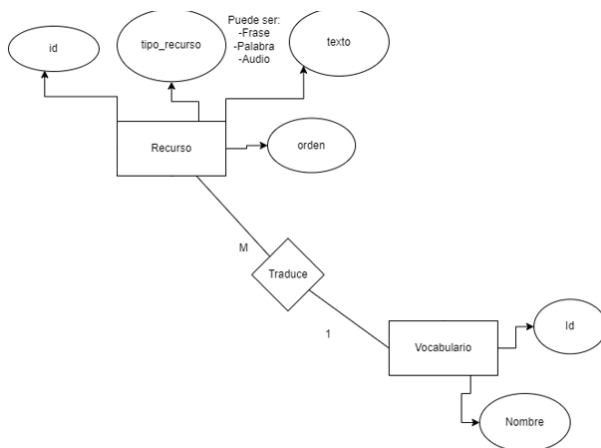


Ilustración 13

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.

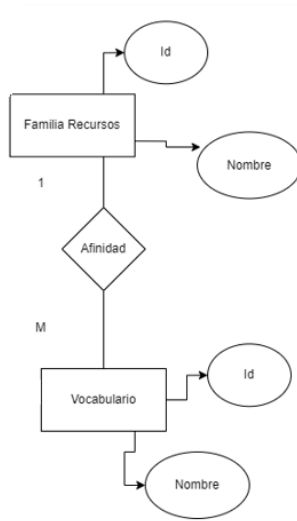


Ilustración 14

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.

Autómata

El autómata es la parte más importante de la aplicación, ya que es quien va a decidir que estudia el usuario en base a 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 imagen de abajo se puede observar el diagrama de flujo del funcionamiento del autómata, para cuando un usuario está estudiando.

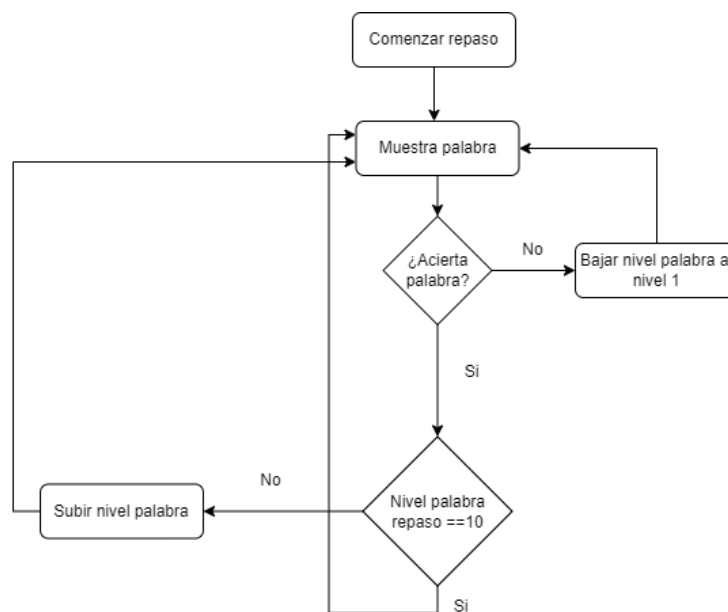


Ilustración 15

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

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

El segundo paso es donde entra en juego el autómatas, 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 8*) y además por un nivel ascendente, es decir, saldrán primero los que tienen un nivel más bajo.

Una vez elegido el recurso a estudiar, miraremos cuál es su traducción para mostrársela a el usuario (*Ilustración 13*).

Y, además elegiremos uno o dos recursos más que tengan relación con él, (*Ilustración 14*). Con ello mostraremos a el usuario una palabra a estudiar con varias opciones.

Si el usuario no ha acertado la respuesta, le bajamos el nivel a 1 y volvemos al paso dos.

Si acierta la palabra, hay que mirar que nivel tiene en esa palabra

Si es un nivel inferior a 10 aumentamos su nivel

Si no se mantiene el nivel 10

Y volvemos a el paso dos.

Creación del entorno

Una vez esta echa 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.

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 16

Seguido utilizaremos mas 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 creado 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.