
Aplicación para niños con problemas de
accesibilidad y de aprendizaje



TRABAJO DE FIN DE GRADO

Grado en Ingeniería Software

Amador Moreno Manzanaro Ranchal

Dirigido por el Doctor

Manuel Montenegro Montes

**Facultad de informática
Universidad Complutense de Madrid**

Septiembre 2019

Este documento está preparado para ser impreso a doble cara.

Aplicación para niños con problemas de accesibilidad y de aprendizaje

*Memoria que presenta para optar al
Graduado en Ingeniería del Software*

Amador Moreno Manzanaro Ranchal

Dirigido por el Doctor
Manuel Montenegro Montes

**Facultad de informática
Universidad Complutense de Madrid**

Septiembre 2019

Índice

Agradecimientos	8
Resumen.....	9
Palabras clave.....	10
Abstract.....	11
Keywords	12
1. Introducción.....	13
1.1. Antecedentes	13
1.2. Objetivos.....	14
1.2.1. API de la Universidad Complutense de Madrid.....	14
1.2.2. Gestor de contenido para profesores	16
1.2.3. Aplicación móvil para alumnos	16
1.3. Plan de trabajo	16
1.3.1. Formación y selección de herramienta	17
1.3.2. Desarrollo de la arquitectura web.....	17
1.3.3. Desarrollo de la arquitectura aplicación móvil.....	17
2. Introduction	17
2.1. Background.....	17
2.2. Objectives	19
2.2.1. API of the Complutense University of Madrid	19
2.2.2. Content manager for teacher	19
2.2.3. Mobile application for student.....	19
2.3. Work plan	20
2.3.1. Training and tool selection	20
2.3.2. Web architecture development	20
2.3.3. Mobile application architecture development	20
3. Selección de tecnologías, lenguajes y herramienta	20
3.1. Del lado del servidor de la web	21
3.2. Del lado del cliente de la web	22
3.3. Del lado de la aplicación móvil.....	23
3.4. Herramientas de apoyo	23
4. Diseño de la base de datos	25
4.1. Modelo ER.....	25

4.2.	Modelo relacional	27
5.	Arquitectura de la aplicación del servidor.....	32
5.1.	Funcionalidades de los controladores.....	32
5.1.1.	HomeController	32
5.1.2.	TeacherController	33
5.1.3.	GrupoAlumnosController.....	33
5.1.4.	AlumnoController.....	33
5.1.5.	EjercicioController	33
5.1.6.	PictogramaController	33
5.1.7.	FrasePictogramaController	34
5.1.8.	DatosEjerciciosController	34
5.2.	Modelo ProfesorModel	34
6.	Arquitectura de la aplicación móvil.....	36
6.1.	Qué es la arquitectura multicapa.....	36
6.2.	Capa de Presentación	37
6.3.	Capa de Negocio	37
6.4.	Capa de Integración	38
7.	Conclusión y trabajo futuro.....	38
7.1.	Objetivos alcanzados.....	38
7.2.	Dificultades encontradas.....	39
7.3.	Trabajo futuro	39
8.	Conclusions and future work	40
8.1.	Achieved goals	40
8.2.	Difficulties encountered	40
8.3.	Future work.....	41
	Bibliografía.....	42

Agradecimientos

Quiero dar las gracias a mi familia, mis padres y mis hermanos, porque cada uno de ellos a su manera me han dado la fuerza para que lo lleve a cabo.

Agradecer a mis amigos que con sus ideas e ingenio se ha conseguido que el proyecto sea un trabajo más ameno y divertido.

Y, en especial, quiero agradecer a Manuel Montenegro, tutor y director de este proyecto, por haberme apoyado, ayudado y confiado a pesar de mis idas y venidas.

Resumen

El proceso de aprendizaje y comprensión lectora de un niño puede ser, a veces, una tarea difícil y larga según sus discapacidades.

En la actualidad existen gran variedad de actividades que, a modo de juego, ayudan a los niños en este proceso, sobre todo haciendo uso de novedades tecnológicas. La tecnología ha influido en la creación de actividades interactivas que ayudan a los niños en la etapa de aprendizaje. Estos juegos tienen que estar elaborados por profesionales en este sector de la educación especializada, para que sean didácticos y efectivos.

Por este motivo hemos creado una plataforma web para que el docente pueda generar y organizar ciertos tipos de ejercicios para sus alumnos, y así tener un control sobre los ejercicios realizados por sus alumnos y también obtener datos sobre la realización de esos ejercicios, en particular, saber cuáles han tenido más dificultades para los alumnos, saber cuáles son los que mejoran más el proceso de aprendizaje, y realizar un seguimiento del progreso de los alumnos, ya sea a nivel global, a nivel de clase o a nivel individual.

Todos los ejercicios y actividades generadas por los alumnos se podrán usar en cualquier dispositivo Android, aunque estarán más enfocadas para su uso en tabletas, ya que tienen una pantalla más grande, mejorando considerablemente la visualización de las imágenes y la interacción entre alumno y dispositivo.

Palabras clave

Dificultad de aprendizaje, aplicación Android, gestión de ejercicios, modelo vista controlador, ASP.net, Android Studio, juego pictograma, arquitectura multicapa.

Abstract

The learning process and reading comprehension of a child can sometimes be a difficult and long task depending on their disabilities.

Currently there are a variety of activities, conceived as games, that help children in this process, most of them making use of technological advances. Technology has influenced the creation of interactive activities that help children in the learning stage. These games have to be made by professionals in the sector of specialized education so that they are didactic and effective.

For this reason we have created a web platform so that the teacher can generate and organize certain types of exercises for their students, and thus have control over what is going to be shown to the students. A teacher may also obtain data on the completion of those exercises, on which of them the students had more problems and what are the exercises which greatly improve the learning process. Teachers can also track the progress of the students, either globally or at the class or individual level.

All the exercises and activities generated by the students can be used on any Android device although the project will be more oriented towards tablets, since they have a larger screen and the images and interaction between student and device is better.

Keywords

L Learning difficulty, Android app, exercise management, model view controller, ASP.net, Android Studio, pictogram game, multilayer architecture.

1. Introducción

En este capítulo se explicará el proyecto a desarrollar, los objetivos, cuáles fueron las motivaciones para realizar este proyecto, qué problemas resuelve y se hará un estudio de otros sistemas con funcionalidad similar.

1.1. Antecedentes

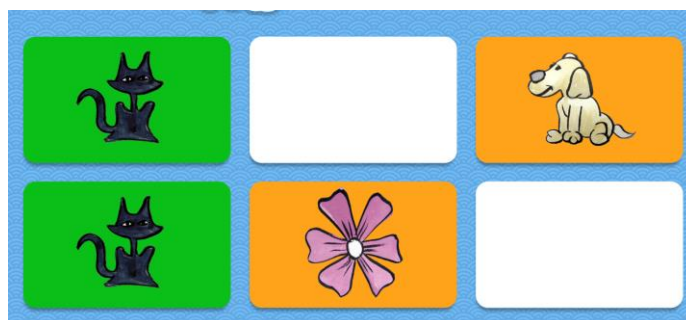
Revisando información sobre los problemas en el proceso aprendizaje de los niños, uno de los problemas más frecuentes es la dislexia. Más del 10% de los niños suelen tener este tipo de problemas, que solo se suelen detectar después de ir a la escuela.

En el proceso de aprender a leer se observa cómo el niño pasa de una lectura palabra por palabra, con una monótona expresión y aparentemente sin atención a los signos de puntuación (Young y Bowers, 1995), hacia una lectura más automática, precisa y expresiva. En otras palabras, puede decirse que los niños que aprenden a leer se van familiarizando cada vez más con la palabra escrita, hasta alcanzar una lectura fluida (Kuhn y Stahl, 2003).

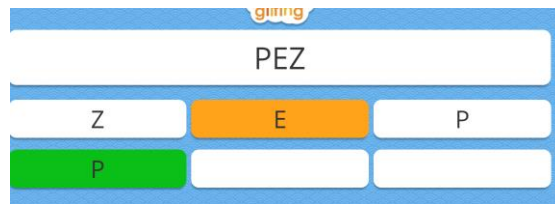
En los últimos años se ha ido comprobando y estudiando que la tecnología es el método más efectivo para facilitar el aprendizaje de los niños, ya sea a través de tabletas, móviles, portátiles o incluso ordenadores de sobremesa, aunque estos últimos más en desuso.

En la actualidad existen muchas aplicaciones orientadas a ayudar al aprendizaje. Entre estas aplicaciones tenemos, por ejemplo, Glifing[1]. Glifing es un método de entrenamiento para la lectura, utiliza un formato lúdico para que los niños no lo vean como un ejercicio sino como un juego.

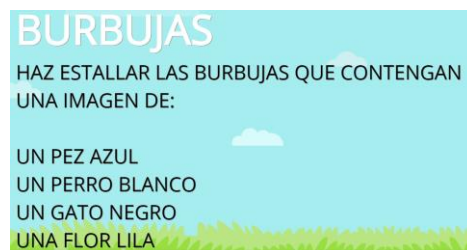
En la siguiente imagen se puede ver un típico juego para desarrollar la memoria. Entre sus características se destaca su estilo sencillo y amigable.



En la imagen mostrada a continuación, el minijuego, centrado en la escritura, consiste en ordenar las letras de una palabra. Se puede observar que empieza con palabras muy sencillas, adaptando la dificultad según la franja de edad seleccionada.



Otro juego más interactivo es el juego de las burbujas, que abarca en sí varios aspectos a tratar: la lectura, la comprensión lectora y la memoria, ya que el usuario tiene que recordar las frases y asociar cada frase a la imagen correspondiente.



Esta aplicación es bastante completa, pero delega demasiado sus contenidos en el desarrollador de la aplicación y no hay un mayor control del profesor docente para poder personalizar los ejercicios y adaptarlos a las circunstancias del alumno.

1.2. Objetivos

El objetivo principal de este trabajo es el desarrollo de una serie de aplicaciones de apoyo para el aprendizaje de niños con problemas de accesibilidad. Estas aplicaciones se apoyarán en una API desarrollada por el Grupo de Procesamiento de Lenguaje Natural de la UCM. A continuación, se detallan los objetivos específicos de este trabajo.

1.2.1. API de la Universidad Complutense de Madrid

Para la elaboración de este proyecto se especificó que se utilizaría la API de la Universidad Complutense de Madrid. Esta aplicación, o *API REST*, consta de una serie de servicios que dan recursos para elaborar ejercicios enfocados a accesibilidad web.



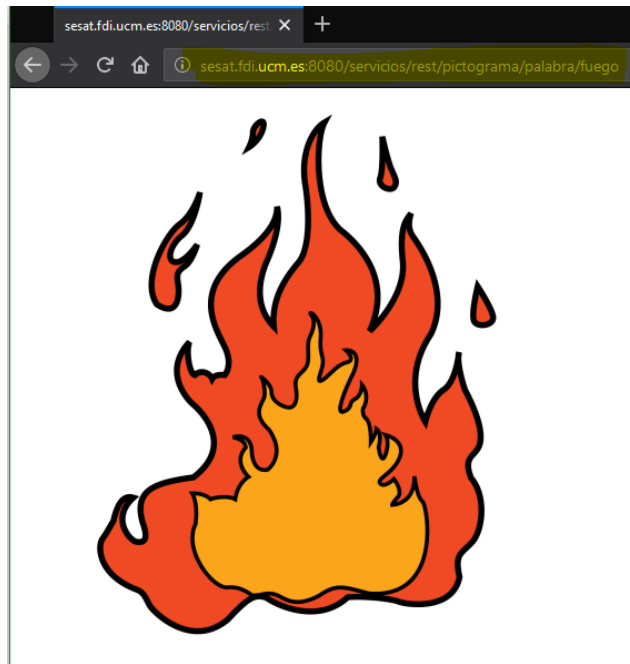
Este servicio se encuentra en el siguiente enlace:

<http://sesat.fdi.ucm.es:8080/Web/>

Como se puede observar en la anterior imagen, el sistema ofrece las siguientes funcionalidades: obtener sinónimos de palabras, obtener antónimos, traducir una palabra al inglés, obtener un pictograma dada una palabra, convertir una palabra en otra de comprensión más sencilla, obtener la definición de una palabra e indicar si una determinada palabra es difícil o no. El uso de esta aplicación mediante su aplicación web es sencillo; basta con poner la URL del servicio que queremos utilizar, seguida de la palabra a buscar. Por ejemplo, si queremos obtener el pictograma de la palabra *fuego* tendremos que hacer lo siguiente:

- Hacemos clic en el enlace Pictogramas y seleccionamos la URL que nos indica en la página que será la siguiente:
<http://sesat.fdi.ucm.es:8080/servicios/rest/pictograma/palabra/>
- Copiamos esta URL y la pegamos en nuestro navegador seguido de la palabra a buscar (que nuestro caso es *fuego*) quedando la URL así:
<http://sesat.fdi.ucm.es:8080/servicios/rest/pictograma/palabra/fuego>
- Pulsamos y nos devolverá la siguiente imagen.

Aunque estos pasos describen el uso de la aplicación desde un navegador web, en realidad la API está pensada para su uso desde otra aplicación mediante llamadas REST.



En conclusión, la API de accesibilidad es una herramienta bastante útil para poder obtener imágenes y datos sin tener que importar gran cantidad de datos en nuestra aplicación.

1.2.2. Gestor de contenido para profesores

Otro de los objetivos de la aplicación es el desarrollo de una aplicación web en la que el profesor se encargue de gestionar y controlar el contenido a utilizar por los alumnos. El profesor, gracias a la aplicación, podrá generar grupos de alumnos según sus necesidades y crear distintos tipos de ejercicios. Podrá gestionar qué palabras elegirá para cada ejercicio, para qué alumnos esté dicho ejercicio destinado, y podrá visualizar el resultado de los ejercicios realizados por los alumnos, ya sea a nivel de alumno, a nivel de grupo, o a nivel global de la aplicación.

1.2.3. Aplicación móvil para alumnos

La aplicación móvil será el método por el cual los alumnos realizarán los ejercicios elaborados por el profesor. Los ejercicios solo podrán jugarse a través de una aplicación Android accediendo a través del nick del alumno y el identificador del ejercicio creado por profesor.

1.3. Plan de trabajo

A continuación, se explicarán brevemente las fases por la que se ha ido desarrollando el proyecto y en que sección de esta memoria se desarrollan.

1.3.1. Formación y selección de herramienta

Para esta fase inicial del proyecto se seleccionaron las herramientas a utilizar según el contexto de la aplicación. Por un lado, se decidió qué tipo de aplicación móvil se iba a desarrollar. El acceso a la información para la creación de las aplicaciones Android llevó alrededor de 3 semanas con la realización de un curso *online* ya que no se tenían conocimientos previos de la herramienta. Por otro lado, se eligió para la aplicación web el uso de *ASP.Net MVC 5*, ya que se tenía conocimiento previo del lenguaje y las herramientas. Y finalmente se escogió como gestor de base de datos *SQL SERVER*, ya que es bastante compatible con *ASP.Net* y *Entity Framework* y Android tiene una gran variedad de librerías que permiten la conexión con este gestor de base de datos. Estas herramientas se explicarán más detalladamente en el capítulo 3.

1.3.2. Desarrollo de la arquitectura web

El desarrollo de la arquitectura web se explica más detenidamente en el capítulo 4, aunque gracias a la herramienta de Visual Studio de Microsoft, éste se ha visto facilitado por el uso del patrón MVC (Modelo Vista Controlador). Para el desarrollo de esta fase se han necesitado alrededor de 3 meses y medio.

1.3.3. Desarrollo de la arquitectura aplicación móvil

Para la fase de desarrollo de la aplicación móvil se necesitaron alrededor de 6 meses, ya que no había suficiente conocimiento de la herramienta y se tuvo que dedicar bastante tiempo. Además, ha sido la que más se necesitó. Se utilizó la arquitectura multicapa porque facilita en gran medida el desarrollo de la aplicación. Todo el desarrollo se explica en el capítulo 5.

2. Introduction

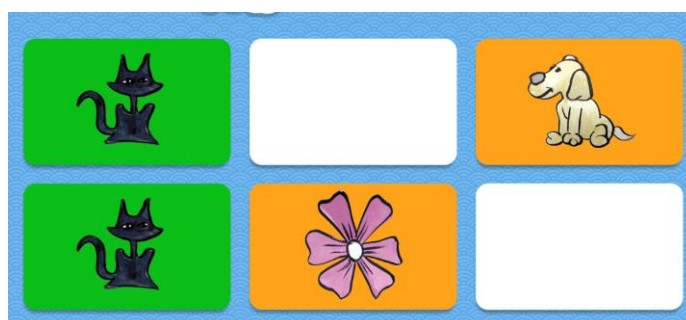
2.1. Background

Reviewing information about problems in the children's learning process, one of the most frequent problems is dyslexia. More than 10% of children have this kind of problem that is usually only detected after going to school.

In the process of learning to read, we observe how the child goes from a word-by-word reading, with a monotonous expression and seemingly without attention to the punctuation marks (Young and Bowers, 1995), towards a more automatic, accurate and expressive reading. In other words, you can say that children who learn to read are becoming more and more familiar with the written word, until they reach a fluent reading (Kuhn and Stahl, 2003).

In recent years it has been verified that technology is the most effective method to facilitate the learning of children, whether through tablets, mobile phones, laptops or even desktop computers, although the latter more in disuse.

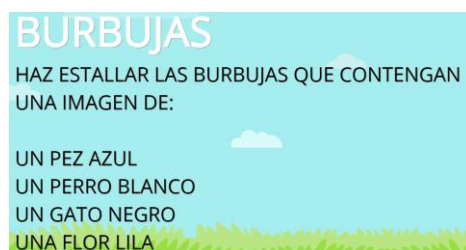
Currently there are many applications aimed at supporting the learning process. Among these applications we have, for example, Glifing [1]. Glifing is a training method for reading. It uses a playful interface so that children do not see it as an exercise but as a game. In the following image you can see a typical game to develop memory. Among its features, its simple and friendly style stands out.



In the image shown below, the minigame, centered on writing, consists in ordering the letters of a word. It can be seen that it begins with very simple words, adapting the difficulty according to the age range selected.



Another more interactive game is the bubble game, which covers several aspects to be treated: reading, reading comprehension and memory, since the user has to remember the sentences and associate each sentence with the corresponding image.





This application is quite complete, but it delegates its contents too much to the developer of the application and the teacher has no greater control in order to be able to customize the exercises and adapt them to the circumstances of each student.

2.2. Objectives

The main objective of this work is the development of a series of support applications for the learning of children with accessibility problems. These applications will be supported by an API developed by the UCM Natural Language Processing Group. The specific objectives of this work are detailed below.

2.2.1. API of the Complutense University of Madrid

For the elaboration of this project it was specified that the API of the Complutense University of Madrid would be used. This application, or REST API, consists of a series of services that provide resources to develop exercises focused on web accessibility.

2.2.2. Content manager for teacher

Another of the objectives of the application is the development of a web application in which the teacher is responsible for managing and controlling the content to be used by students. The teacher, thanks to the application, can generate groups of students according to their needs and create different types of exercises. Teachers will be able to manage what words they will choose for each exercise, so that students are assigned that exercise, and they will be able to visualize the result of the exercises performed by the students, either at the student level, at the group level, or at the global level of the application.

2.2.3. Mobile application for student

The mobile application will be the method by which students will perform the exercises developed by the teacher. The exercises can only be played through an Android application by accessing through the student's nickname and the identifier of the exercise created by the teacher.

2.3. Work plan

Next, the phases by which the project has been developed and in which section of this report are developed will be briefly explained.

2.3.1. Training and tool selection

For this initial phase of the project, the tools to be used were selected according to the context of the application. On the one hand, it was decided what type of mobile application was to be developed. The access to the information for the creation of the Android applications took about 3 weeks with the realization of an online course since there was no previous knowledge of the tool. On the other hand, the framework *ASP.Net MVC 5* was chosen for the web application, since there was prior knowledge of the language and tools. And finally, we chose the *SQL SERVER* database manager, since it is quite compatible with *ASP.Net* and *Entity Framework* and Android has a large variety of libraries that allow connection to this database manager. These tools will be explained in more detail in Chapter 3.

2.3.2. Web architecture development

The development of the web architecture is explained in more detail in Chapter 4, although thanks to Microsoft's Visual Studio tool, it has been facilitated by the use of the MVC pattern (Model Controller View). For the development of this phase about 3 and a half months have been needed.

2.3.3. Mobile application architecture development

It took about 6 months for the mobile application development phase, since there was not enough knowledge of the tool and enough time had to be spent. In addition, it has been the most demanding requirement. The multilayer architecture was used because it greatly facilitates the development of the application. All development is explained in chapter 5.

3. Selección de tecnologías, lenguajes y

herramientas

En esta sección se comentará qué tecnologías, lenguajes y herramientas se han utilizado para la realización del proyecto, tanto en la parte de la web, como en la base de datos y en el desarrollo de la aplicación móvil.

3.1. Del lado del servidor de la web

CSharp (C#)

Es un lenguaje orientado a objetos creado por Microsoft que forma parte de la plataforma .Net. Su sintaxis básica deriva de C y C++ y utiliza un modelo de objetos similar al de Java, a pesar de que incluye otras adaptaciones y ventajas de otros lenguajes. Se seleccionó este lenguaje porque es el que forma parte del Framework *ASP.NET MVC* para la creación de aplicaciones web. Además, ya había trabajado con él anteriormente, y tenía un conocimiento previo que me ayudaría a desarrollar con más rapidez la aplicación.

ASP.NET MVC 5

Es un *framework* para aplicaciones web que utiliza el patrón MVC (modelo-vista-controlador). Basándose en este patrón, es posible permite crear aplicaciones web a través del IDE *Visual Studio* (descrito a continuación) y nos permite tener una separación visible entre la presentación de la web, correspondiente a la parte de las vistas, y su parte lógica, parte *back-end*, modelo y controladores.

También se seleccionó este *framework* porque junto con el IDE Visual Studio es posible generar fácilmente el código de controladores y sus vistas en sencillos pasos, sin tener que escribirlo en su totalidad.

IDE Visual Studio 2017 / 2019 Community Edition

Es un entorno de desarrollo integrado preparado para la creación y desarrollo de aplicaciones de escritorio, webs, *plugins*, aplicaciones móviles, etc. Este entorno es compatible con la plataforma .Net. La ventaja de este entorno es que permite el uso de una amplia variedad de lenguajes de programación como Python, Ruby, PHP, etc. y, por supuesto, C, C++, C#, Visual Basic.Net.

Se escogió esta herramienta, ya es que necesaria para la elaboración de la aplicación web en *ASP.NET MVC 5*, por su robustez a la hora de crear la aplicación, por los conocimientos y experiencias previas con la herramienta y por las ventajas a la hora de realizar conexión con base de datos *SQL SERVER* y el uso de *Entity Framework*.

A lo largo del desarrollo de la aplicación empecé usando la versión Visual Studio 2017 *Community*. Cuando se lanzó la versión 2019 *Community* probé esta última versión y, al tener una experiencia de usuario más eficaz, decidí actualizarme a la misma.

ADO.Net Entity Framework 6 (EF6)

Entity Framework es una herramienta de mapeo objeto-relacional (O/RM) que permite a los desarrolladores de .NET trabajar con una base de datos mediante objetos .NET. Elimina la necesidad de la mayor parte del código de acceso a datos que los desarrolladores normalmente deben escribir [2].

La principal ventaja de este *framework* es la opción de crear en primer lugar el modelo de base de datos mediante código (metodología comúnmente llamada “*Code First*”) y el framework genera el modelo relacional de base de datos automáticamente. Como segunda

opción, existe la posibilidad llamada “*Data Base First*”, que genera un modelo a través de una base de datos existente.

Para el desarrollo se utilizó “*Data Base First*” porque primeramente se generó el modelo de base de datos. *ADO.Net* junto con SQL SERVER, a partir del cual el IDE Visual Studio permite generar rápidamente modelos de entidad. Por este motivo se eligió este *framework*.

SQL SERVER

Para la gestión de la base de datos se eligió SQL SERVER, que es sistema de base de datos de tipo relacional desarrollado por la empresa Microsoft.

La principal característica y ventaja que hizo que se seleccionara este gestor es su compatibilidad con *Entity Framework* y *ADO.Net*.

3.2. Del lado del cliente de la web

HTML 5

HTML [2] es el acrónimo de *HiperText Markup Language*, es decir, lenguaje de marcas de hipertexto. Mediante HTML podemos definir la estructura de las páginas web. HTML 5 no es un estándar estático, dado que fue diseñado para ser ampliado a medida que surjan las necesidades. Esto hará que no se quede retrasado con respecto a la evolución del resto de tecnologías Web.

CSS 3

CSS [4] son las siglas de *Cascade Style Sheets*, es decir, hojas de estilo en cascada, y es un lenguaje que permite asignar estilo a los documentos HTML. Se utilizó la versión 3, ya que trae nuevas propiedades aplicables al estilo del documento como, por ejemplo, las sombras, esquinas redondeadas, etc.

JAVASCRIPT y JQUERY

La programación de *JavaScript* [2] se basa en la edición de scripts que se integran en el código HTML de la página y cuyo código se puede visualizar en cualquier navegador mediante la opción Ver código fuente del mismo. *JavaScript* es un lenguaje interpretado (a diferencia de los lenguajes compilados), por lo que el código se analiza y se ejecuta en el navegador del cliente. No existe una etapa de compilación del código como sucede en otros lenguajes de programación, por ejemplo, C++.

jQuery es una librería de *JavaScript* de código abierto para la programación de tareas comunes con menos código, optimizando la interacción con los documentos HTML. La base de esta librería es la función de sobrecarga que permite distintas funcionalidades dependiendo de los parámetros introducidos.

BOOSTRAP

Es un *framework* o conjunto de herramientas de código abierto para establecer el diseño de aplicaciones web. Dentro de esta biblioteca se encuentran plantillas de diseño de tipografía, menús de navegación, botones, visualización de tablas, animaciones, etc. [3]. Se seleccionó esta librería para agilizar y mejorar la visualización de la aplicación y que sea *responsive*, es decir, que se pueda visualizar en todas las plataformas posibles (incluyendo dispositivos móviles).

3.3. Del lado de la aplicación móvil

Java 8

Java es un lenguaje de programación orientado a objetos de alto nivel. Las ventajas de Java radican en que es orientado a objetos, es muy flexible y está preparado para la reutilización de código. También es un lenguaje multiplataforma por lo que funciona en una gran variedad de sistemas operativos, a diferencia de herramientas de, por ejemplo, *Microsoft* que generalmente solo funcionan en entornos *Windows*.

Este es el lenguaje nativo que utiliza Android para desarrollar aplicaciones móviles. Existen otros tipos de lenguaje para el desarrollo de aplicaciones móviles, pero se escogió Java porque ya se tenían conocimientos previos de este lenguaje.

IDE Android Studio

Es un entorno de desarrollo integrado para la creación de aplicaciones *Android*. Está basado en el software *IntelliJ IDEA* de la compañía *JetBrains*, el cual proporciona varias mejoras con respecto al plugin *ADT (Android Developer Tools)* para *Eclipse*. Además de ser un potente editor de código y disponer de herramientas para desarrolladores, ofrece más funciones como un sistema de compilación flexible basado en *Gradle*, un emulador rápido con una amplia funcionalidad, y un entorno unificado donde poder desarrollar para cualquier dispositivo Android.

Esta herramienta se utilizó para la creación de la aplicación móvil. Dado que al principio del desarrollo del proyecto no se tenía conocimiento en este ámbito, se realizó un curso online de formación que utilizaba esta herramienta.

3.4. Herramientas de apoyo

DBeaver

Es una herramienta de administración de base de datos relacionales que permite trabajar con distintos gestores de bases de datos, poder modificar datos como si de una hoja de cálculo se tratara, exportar datos y muchas funcionalidades más [6].

Esta herramienta ha sido seleccionada por su versatilidad y facilidad a la hora de visualizar y chequear las modificaciones en la base de datos que se iban realizando a la hora de usar la aplicación web y la aplicación móvil.

GitHub

Es un servicio en la nube que permite a los desarrolladores poder almacenar y administrar el código generado, llevar un control y registro sobre los cambios realizados y su autoría y también poder revertir el estado del proyecto a un estado anterior.

Sublime Text 3

Es un potente y ligero editor de texto multiplataforma. Esta herramienta admite una amplia cantidad de lenguajes de programación que junto con su estilo y coloreado de palabras reservadas ayuda al desarrollo de código.

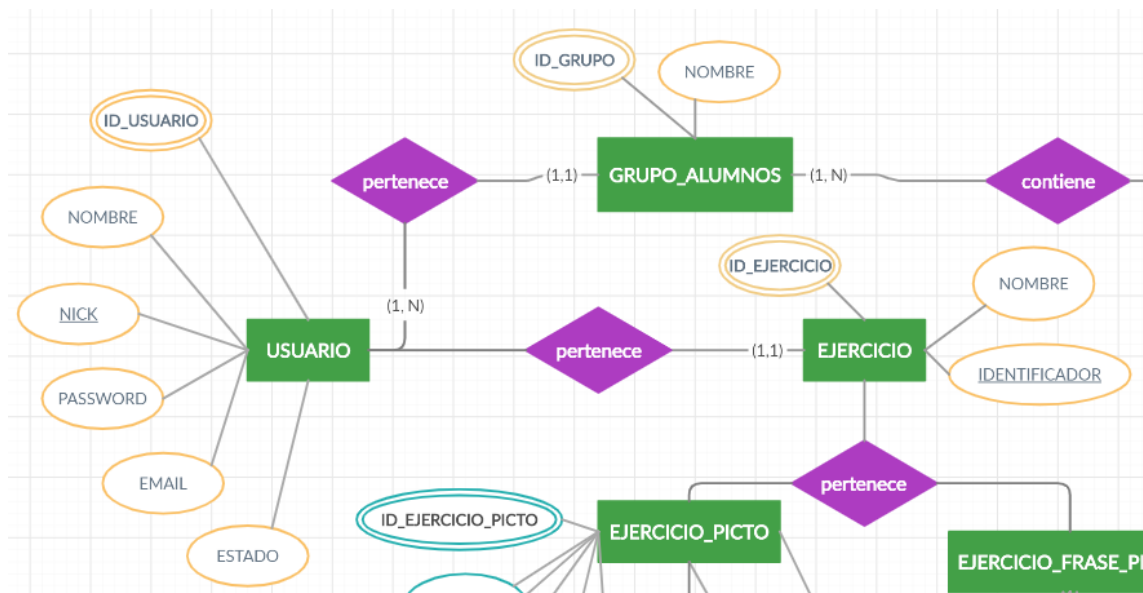
Esta herramienta fue muy útil para comprobar y chequear los cambios realizados en variables globales de controladores de la aplicación web y elaboración de un *backup* de la base datos.

4. Diseño de la base de datos

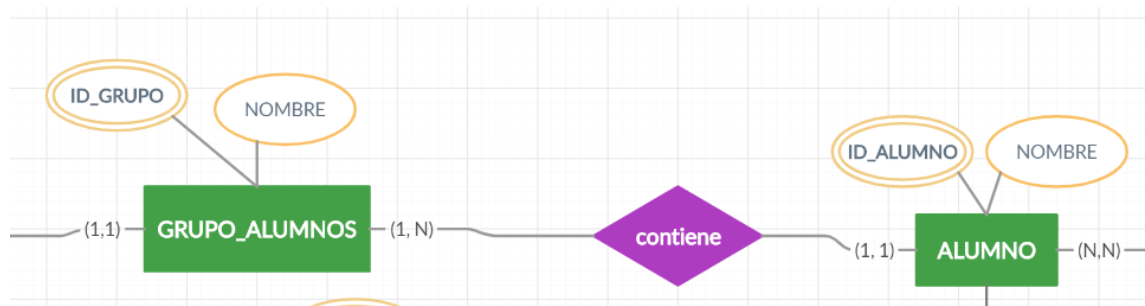
En este apartado se explicará el diseño que se hizo para la creación de la base de datos

4.1. Modelo ER

En las siguientes imágenes se muestra el modelo entidad relación de la base de datos. Cabe destacar la relación entre USUARIO y las entidades GRUPO_ALUMNOS y EJERCICIOS, ya que por cada usuario puede existir una cantidad N de grupos de alumnos y ejercicios.

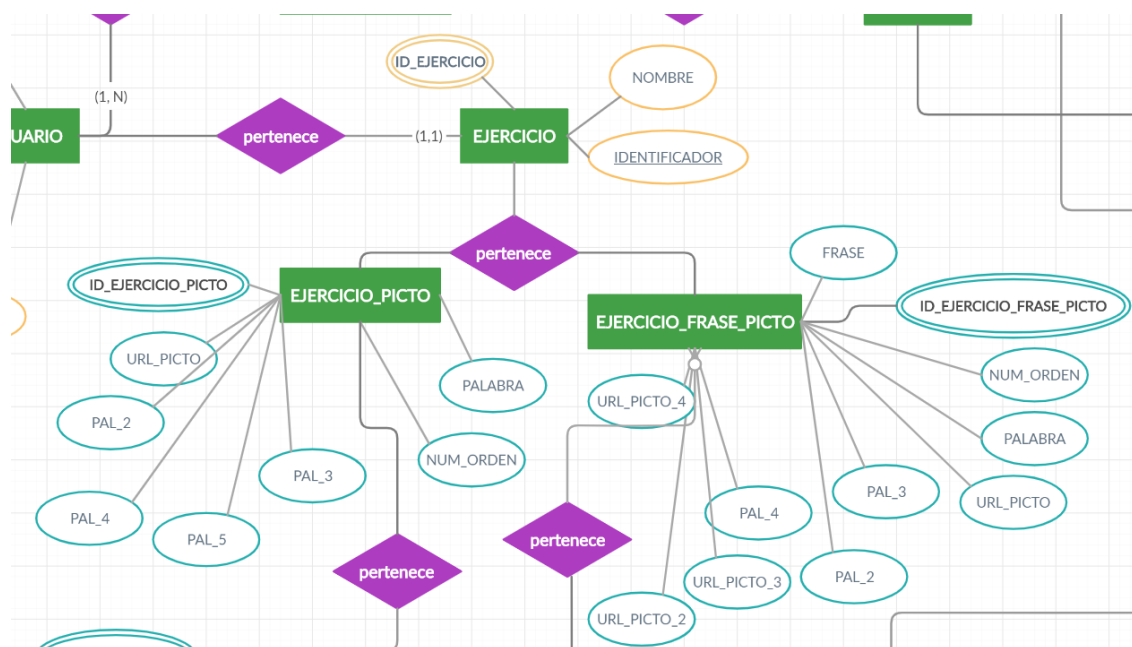


En la siguiente imagen se muestra la relación 1 a N que existe con alumnos y Grupo de alumnos. Por cada grupo de alumnos puede haber N alumnos.

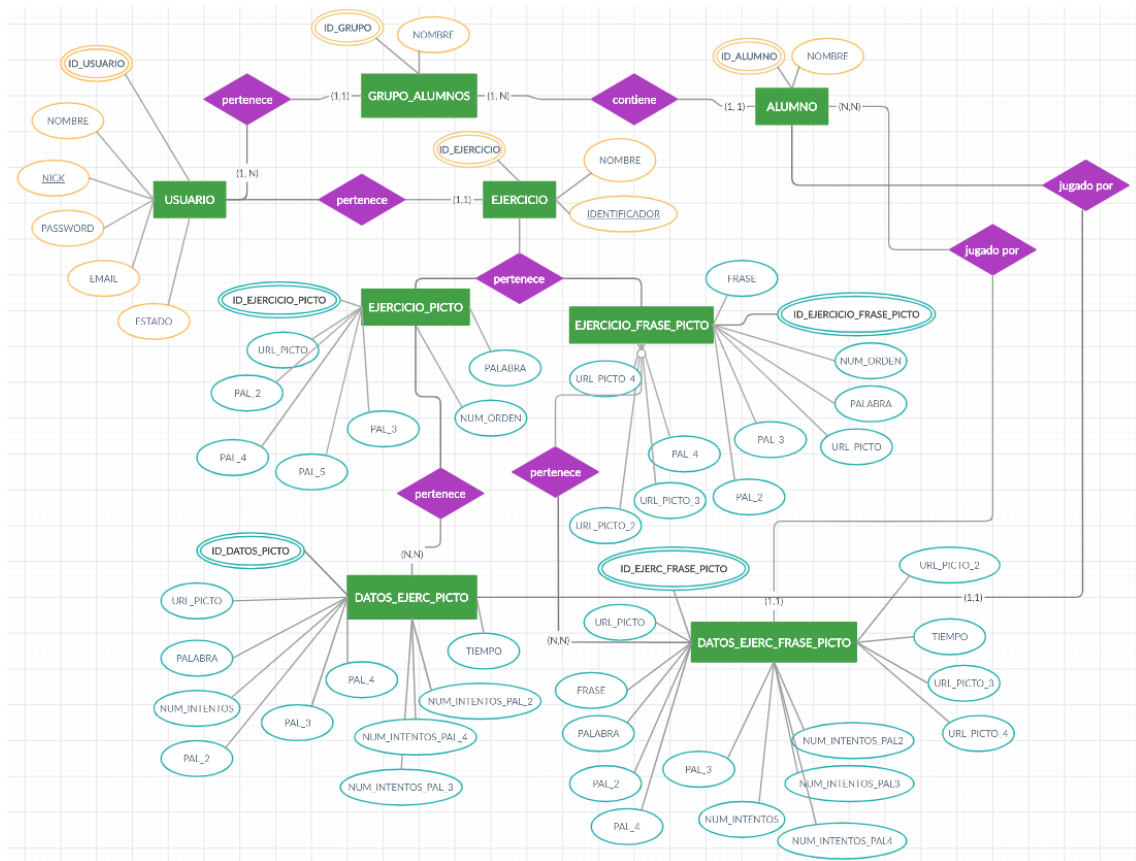


Los tipos de ejercicios tendrán la referencia del Id del ejercicio, tal y como se muestra en la siguiente imagen. Los dos tipos de ejercicios que se han creado han sido ejercicios de tipo

Pictograma, que contendrá una palabra a buscar, su respectivo url que contiene la imagen a mostrar y 4 palabras diferentes a elegir por el profesor. El otro ejercicio es de tipo Frase con pictogramas, en la que se compone de la frase, la palabra a buscar con la url del pictograma asociada y 3 palabras diferentes junto con su url de pictograma también. Todos estos campos son cadena de texto, excepto NUM_ORDEN y sus claves primarias que campos numéricos.

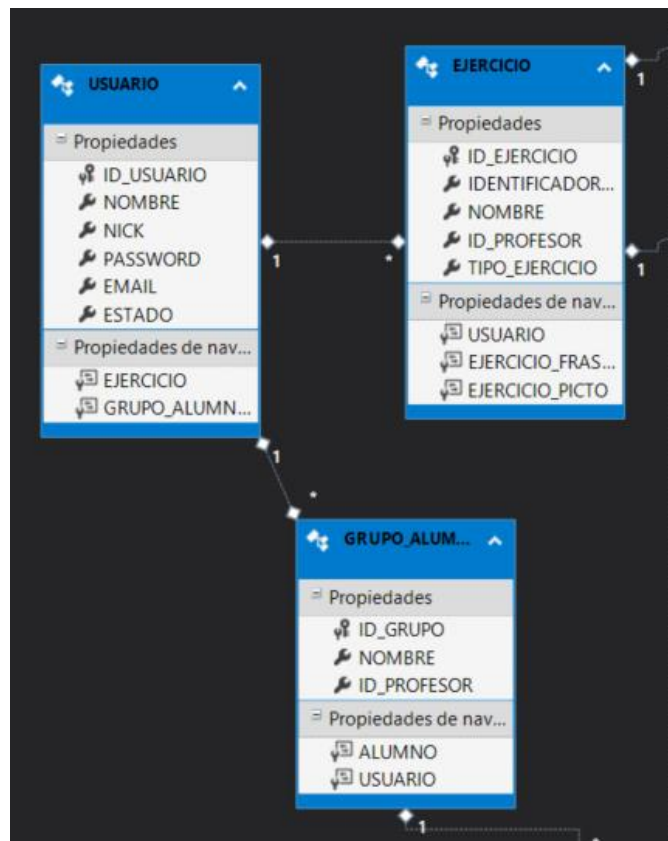


Con respecto a los resultados de los ejercicios de tipo pictograma y frase pictograma, estos se referenciarán a través de los identificadores de los ejercicios de las entidades pictograma y frase pictograma y también se referencia a través de un id del alumno que realiza el ejercicio.



4.2. Modelo relacional

En este apartado explicaremos cada una de las tablas y que contiene.



La primera tabla que se creó fue la tabla de USUARIO. En este contexto los usuarios, en realidad, son los profesores, los cuales gestionarán los grupos, alumnos, los ejercicios, y podrán evaluar el resultado de los ejercicios realizados por cada alumno. A continuación, se explica que función y uso tiene cada columna:

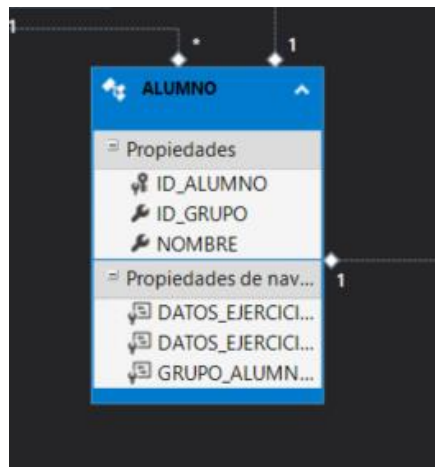
- La columna ID_USUARIO es el identificador del usuario que se genera automáticamente y es la clave primaria.
- La columna NOMBRE contiene el nombre con que se le mostrará en la aplicación web al usuario que la esté utilizando.
- El NICK será el identificador para acceder a la plataforma web y será único.
- El PASSWORD que junto con el NICK será la forma de acceso a la plataforma.
- El EMAIL se utilizará para que, en caso de olvido de contraseña, se pueda recuperar la contraseña vía email.
- El ESTADO, que por defecto será 1, indicará el estado lógico del usuario, es decir, si el usuario está dado de alta o de baja.

La siguiente tabla a generar es GRUPO_ALUMNOS para poder así agrupar los alumnos. A continuación, se explica el significado de las columnas de la tabla:

- La columna ID_GRUPO es el identificador del grupo que se genera automáticamente y es la clave primaria.
- La columna NOMBRE indica el nombre del grupo. La aplicación no podrá tener dos grupos con el mismo nombre para el mismo usuario.
- La clave referencia ID_PROFESOR muestra solo los grupos por el identificador del profesor, que es la columna ID_USUARIO de la tabla USUARIO.

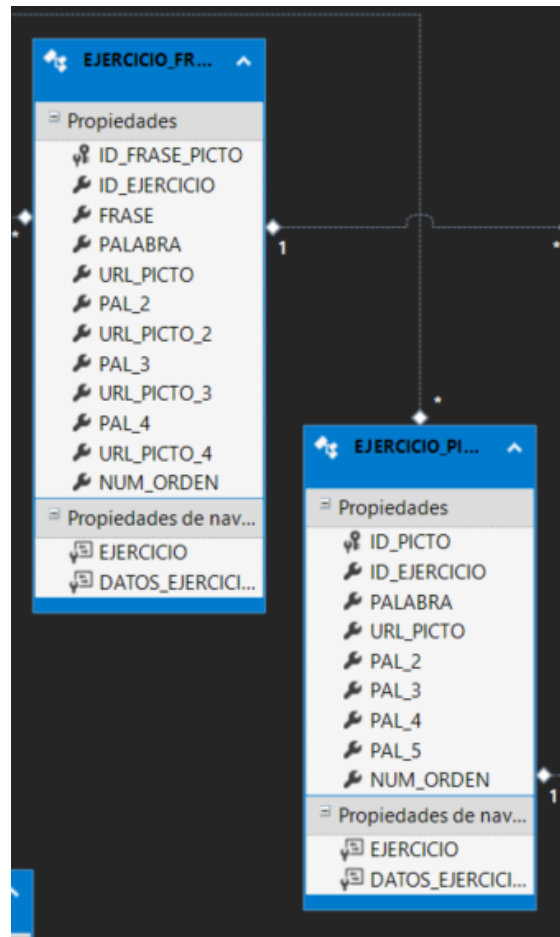
La tabla de EJERCICIOS contendrá los identificadores para poder acceder desde la aplicación móvil. A continuación, se explica con más detalles:

- La columna ID_EJERCICIO que es el identificador, clave primaria, que se generará para la creación de cada ejercicio.
- La columna NOMBRE indica el nombre del ejercicio que verá el profesor en la aplicación web cuando vaya añadiendo más ejercicios.
- La columna IDENTIFICADOR_PUBLICO será el nombre que utilizará para que los alumnos, junto con su nombre, puedan acceder a los ejercicios creados por el profesor.
- La clave referencia ID_PROFESOR sirve para mostrar solo los ejercicios que correspondan a un determinado profesor, al igual que en la tabla GRUPO_ALUMNOS.
- La columna TIPO_EJERCICIO sirve para identificar el tipo de ejercicio, ya sea, en nuestro caso, de pictogramas o frases pictogramas.



La tabla de ALUMNO contiene los alumnos que se crearan para acceder a los ejercicios. Hay que tener en cuenta que se puso la mínima información posible para que los datos no fueran sensibles a temas de protección de datos del menor. A continuación, se explica con más detalle:

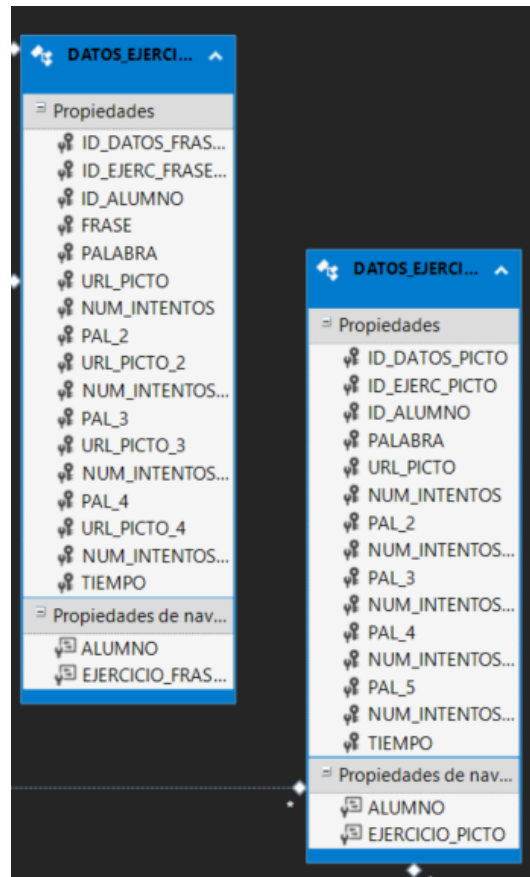
- La columna ID_ALUMNO que es el identificador, clave primaria, que se generará para la creación de cada alumno.
- La columna NOMBRE indica el nombre del alumno que verá el profesor en la aplicación web cuando vaya añadiendo más alumnos a cada grupo.
- La columna ID_GRUPO será el identificador referenciado para saber a qué grupo pertenece ese usuario.



Las siguientes tablas EJERCICIO_PICTO y EJERCICIO_FRASE_PICTO contienen la lista de cada paso del ejercicio que ha ido generando el profesor para que los alumnos puedan probar y usar en la aplicación móvil. A continuación, se explica con más detalle:

- Las columnas ID_FRASE_PICTO y ID_PICTO son los identificadores, claves primarias respectivamente, que se generarán para la creación de cada tipo de ejercicio.
- La columna ID_EJERCICIO indica a que ejercicio de la tabla EJERCICIO esta referenciando.
- La columna PALABRA será la palabra en el juego que se va a buscar, que es la opción correcta del ejercicio. Estará en ambas tablas.
- La columna FRASE solo para la tabla EJERCICIO_FRASE_PICTO y será sobre la que se mostrará omitiendo la palabra elegida.
- La columna URL_PICTO será la URL que contiene el pictograma de la palabra a mostrar en la aplicación web y móvil. Estará en ambas tablas.
- Las columnas PAL_2, PAL3, PAL_4 y PAL_5 son las otras palabras que se mostrar en el ejercicio para elegir la correcta. Estas también son para ambas tablas a excepción de la columna PAL_5 ya que solo se usa para ejercicios de pictograma.
- Las columnas URL_PICTO_2, URL_PICTO_3 y URL_PICTO_4 serán las otras URLs que contendrán la imagen para seleccionar la correcta entre las otras imágenes y la correcta. Estas solo están en la tabla EJERCICIO_FRASE_PICTO.

- La última columna NUM_ORDEN sirve para poder ordenar cada actividad a elección del profesor.



Las siguientes tablas contienen información sobre el tiempo y número de intentos para realizar cada actividad del ejercicio. Son las tablas DATOS_EJERCICIO_PICTO y DATOS_EJERCICIO_FRASE_PICTO. A continuación, se explica con más detalle cada una de sus columnas:

- Las columnas ID_DATOS_FRASE_PICTO y ID_DATOS_PICTO son los identificadores, claves primarias respectivamente, que se generarán para cada ejercicio jugado por el alumno.
- Las columnas ID_EJERC_PICTO y ID_EJERC_FRASE_PICTO indica a que ejercicio esta referenciado.
- La columna PALABRA será la palabra en el juego que se va a buscar y es la correcta. Estará en ambas tablas.
- La columna FRASE solo para la tabla EJERCICIO_FRASE_PICTO será sobre la que se mostrará omitiendo la palabra elegida.
- La columna URL_PICTO será la dirección URL que contiene el pictograma de la palabra a mostrar en la aplicación web y móvil. Estará en ambas tablas.
- Las columnas PAL_2, PAL3, PAL_4 y PAL_5 son las otras palabras que se mostrar en el ejercicio para elegir la correcta. Estas también se aplican a ambas tablas, a excepción de la columna PAL_5 ya que solo se usa para ejercicios de pictograma.

- Las columnas URL_PICTO_2, URL_PICTO_3 y URL_PICTO_4 serán las otras URL que contendrán las imágenes alternativas, para que el alumno escoja entre estas imágenes y la correcta. Estas solo están en la tabla EJERCICIO_FRASE_PICTO.
- La última columna NUM_ORDEN sirve para poder ordenar cada actividad a elección del profesor.

5. Arquitectura de la aplicación del servidor

En este apartado se explicará que arquitectura se utilizó para la aplicación web haciendo una explicación a nivel técnico del funcionamiento interno y algunos fragmentos de código de la aplicación.

5.1. Funcionalidades de los controladores

Los controladores en el patrón MVC (Modelo Vista Controlador) se encargan de responder a las solicitudes recibidas en la aplicación web. Dependiendo de la solicitud recibida, el framework web ejecutará un controlador específico.

Por ejemplo, si lo que queremos es acceder a nuestra aplicación, antes debemos loguearnos por lo que primeramente debemos ir a la página de acceso, o login, que tendrá esta forma de enrutamiento:

```
http://localhost/Home/Login
```

Donde **Home** representa el controlador **HomeController** y **Login** será el método encargado de realizar la acción dentro de ese controlador.

A continuación, se detallarán cada uno de los controladores utilizados para la aplicación.

5.1.1. HomeController

HomeController gestionará solo las peticiones de **Login** y **Registrarse**, las cuales tendrán su respectiva función POST que se encargará de hacer la validación de los datos recibidos y de realizar las peticiones correspondientes a la base de datos. Dentro de cada método del controlador podremos elegir que, por ejemplo, nos devuelva la vista de la acción o que nos redirija a otra acción de otro controlador o del mismo controlador según sea necesario.

Las peticiones a base datos se realizarán a través de llamadas a una entidad llamada **_repository** que devolverá los datos solicitados.

Este objeto se encarga de conectar el contexto de base datos, es decir, *ADO.Net*, con las entidades de la base datos para obtener los datos deseados.

Los controladores también tendrán un objeto privado llamado **_service** que se encargará de hacer las peticiones a la API ofrecidas por la aplicación para mostrar, por ejemplo, los pictogramas.

5.1.2. TeacherController

TeacherController se encarga de dar soporte a las acciones relativas al usuario. Estas acciones son **Logout**, para salir de la sesión, y **UsuarioNick** y **UsuarioPwd** que son para modificar el nick y la contraseña del usuario.

Este controlador tiene un método privado que se encargará de chequear si existe una sesión activa, obtener el identificador de usuario logueado y cargar sus datos para mostrar en la aplicación. En caso contrario cierra sesión y sale de la aplicación.

5.1.3. GrupoAlumnosController

GrupoAlumnosController se encarga de dar soporte a las acciones relativas a la creación, visualización, modificación y eliminación de grupos de alumnos. Estas acciones son: **MisGrupos** que carga los grupos creados por ese usuario, **Grupo** para añadir nuevos grupos, **EditGrupo** para editar cualquier grupo creado y **BorrarGrupo** que se encargará de borrar el grupo seleccionado por el usuario. En caso de que no haya una sesión abierta redirigirá al método **Logout** para borrar la sesión. Tendrá tres métodos privados que cargará según las necesidades de la acción, que son **LoadUsuario**, **LoadGrupoAlumnos** y **LoadInfoGruposYUsuario**.

5.1.4. AlumnoController

AlumnoController se encarga de dar soporte a las acciones relativas a la creación, visualización, modificación y eliminación de alumnos. Este controlador es similar al anterior pero enfocado en los Alumnos.

5.1.5. EjercicioController

EjercicioController se encarga de dar soporte a las acciones relativas a la creación, visualización, modificación y eliminación de ejercicios. Este controlador es similar al controlador de **GrupoAlumnosController** pero enfocado en los Ejercicios.

5.1.6. PictogramaController

PictogramaController se encarga de dar soporte a las acciones relativas a la creación, visualización, modificación y eliminación de ejercicios de tipo pictograma. Este controlador es similar al controlador de **GrupoAlumnosController** pero enfocado en los ejercicios de Pictogramas. Además, también contiene unos métodos privados como **IsValidsPicto**, **NoRepeatPicto**, **NoDuplicados** y **CleanPalabrasPicto** que se encargarán de chequear que los datos del pictograma sean correctos.

5.1.7. FrasePictogramaController

FrasePictogramaController se encarga de dar soporte a las acciones relativas a la creación, visualización, modificación y eliminación de ejercicios de tipo frase de pictograma. Este controlador es similar al controlador de **GrupoAlumnosController** pero enfocado en los ejercicios de Pictogramas. Además, también contiene unos métodos privados similares como **IsValidPicto**, **NoRepeatPicto**, **NoDuplicados** y **CleanPalabrasPicto** que se encargarán de chequear que los datos del pictograma sean correctos.

5.1.8. DatosEjerciciosController

FrasePictogramaController se encarga de dar soporte a las acciones relativas a la creación, visualización, modificación y eliminación de ejercicios de tipo frase de pictograma. Este controlador es similar al controlador de **GrupoAlumnosController** pero enfocado en los ejercicios de Pictogramas.

5.2. Modelo ProfesorModel

El modelo **ProfesorModel** se encargará de cargar los datos necesarios para mostrar en la vista, aunque no se visualicen todos ellos en cada momento, y consta de la siguiente información:

```

public class ProfesorModel
{
    #region Usuarios
    public USUARIO UsuarioSession { get; set; }
    #endregion
    #region Ejercicios
    public List<EJERCICIO> MisEjercicios { get; set; }
    public EJERCICIO NuevoEjercicio { get; set; }
    public int IdEjercicioSeleccionado { get; set; }
    #endregion
    #region Grupos
    public List<GRUPO_ALUMNOS> MisGrupos { get; set; }
    public GRUPO_ALUMNOS NuevoGrupo { get; set; }
    public int IdGrupoSeleccionado { get; set; }
    public List<InfoDatosGrupo> InfoGrupos { get; set; }
    #endregion
    #region Alumnos
    public ALUMNO NuevoAlumno { get; set; }
    public List<ALUMNO> MisAlumnos { get; set; }
    #endregion
    #region Pictogramas
    public EjercicioPictograma InfoEjercicioPicto { get; set; }
    public EJERCICIO_PICTO NuevoPictograma { get; set; }
    public bool NuevoPictogramaEsValido { get; set; }
    public string UrlPictoSeleccionado { get; set; }
    #endregion

    #region FrasesPictograma
    public EjercicioFrasesPictograma InfoEjercicioFrasesPicto { get; set; }
    public EJERCICIO_FRASE_PICTO NuevaFrasePictograma { get; set; }
    public bool NuevaFrasePictogramaEsValido { get; set; }
    public string UrlFrasePictoSeleccionado { get; set; }
    #endregion
    public TiposServicio Servicios { get; set; }
}

```

UsuarioSession que será el profesor registrado en la sesión.

MisEjercicios, tendrá una lista de ejercicios creador por el profesor.

NuevoEjercicio, para cuando se quiera crear un nuevo ejercicio o editarlo.

IdEjercicioSeleccionado, para indicar que ejercicio se ha seleccionado previamente.

MisGrupos, lista de grupos de alumnos creados por el profesor.

NuevoGrupo, entidad que sirve para crear nuevos grupos de alumno o editar grupos ya existentes.

IdGrupoSeleccionado, para indicar o seleccionar un grupo en la vista correspondiente.

InfoGrupos, listado de información de grupos y su listado de alumnos por grupos respectivamente.

NuevoAlumno, entidad para los alumnos para crear uno nuevo o editar uno existente.

MisAlumnos, para mostrar los alumnos existentes asociados al profesor.

InfoEjercicioPicto, contiene la información del ejercicio y el listado de ejercicios pictograma asociado a ese ejercicio.

NuevoPictograma, información de un ejercicio pictograma ya sea para crear uno nuevo o para editarlo.

NuevoPictogramaEsValido, booleano para indicar si los datos para el pictograma a tratar son válidos.

UrlPictoSeleccionado, URL del pictograma seleccionado.

InfoEjercicioFrasesPicto, contiene la información del ejercicio y el listado de ejercicios de frases pictograma asociado a ese ejercicio.

NuevaFrasePictograma, información de un ejercicio frase pictograma ya sea para crear uno nuevo o para editarlo.

NuevoFrasePictogramaEsValido, booleano para indicar si los datos para la frase pictograma a tratar son válidos.

UrlPictoFraseSeleccionado, URL del pictograma seleccionado en un ejercicio de tipo frase pictograma.

ResultadoPicto, listado de los resultados de los ejercicios de pictogramas.

ResultadosFrasePicto, listado de los resultados de los ejercicios de frase pictograma.

```
public ProfesorModel()
{
    Servicios = new TiposServicio();
}

#region Resultados
public List<DATOS_EJERCICIO_PICTO> ResultadosPicto { get; set; }
public List<DATOS_EJERCICIO_FRASE_PICTO> ResultadosFrasePicto { get; set; }
#endregion
}

public class EjercicioPictograma
{
    public List<EJERCICIO_PICTO> Palabras { get; set; }
    public EJERCICIO InfoEjercicio { get; set; }
}

public class EjercicioFrasesPictograma
{
    public List<EJERCICIO_FRASE_PICTO> Frases { get; set; }
    public EJERCICIO InfoEjercicio { get; set; }
}

public class InfoDatosGrupo
{
    public GRUPO_ALUMNOS DatosGrupo { get; set; }
    public List<ALUMNO> AlumnosGrupo { get; set; }
}
```

6. Arquitectura de la aplicación móvil

En este capítulo se explicará con un nivel técnico más detallado la arquitectura implementada en la aplicación móvil.

6.1. Qué es la arquitectura multicapa

La arquitectura multicapa es un patrón de arquitectura cliente-servidor en el que la carga del sistema se divide en tres partes, también llamado capas, con una diferenciación clara de funciones. La primera es la capa de presentación, que corresponde a la interfaz de usuario. La segunda capa corresponde a la capa de negocio y la tercera capa corresponde a la capa de integración, o persistencia, donde se almacenan y se obtienen los datos.

6.2. Capa de Presentación

Para la aplicación solo habrá tres interfaces por las que accederá el usuario, **VistaLogin**, **VistaJuegoPictograma** y **VistaJuegoFrasePictograma**. Estas interfaces son con las que interactuará el alumno.

Dentro de esta capa también se encuentra el **Dispatcher** que se encarga de redirigir las vistas y **VistaFactoria** para gestionar todas las vistas de la aplicación que se vayan agregando.

6.3. Capa de Negocio

Esta capa consta de los siguientes servicios:

- **SAAlumno**: contiene los servicios que llaman al **DAOAlumno** para obtener datos encapsulado de los alumnos. Para ello tiene una clase llamada **TransferAlumno** que sirve para representar los datos de la entidad Alumno.
- **SADatosFrasePictograma**: contiene los servicios para insertar y actualizar los datos generados por los usuarios que haya utilizado la aplicación. Tendrá una clase **TransferDatosFrasePictograma** que se encarga de encapsular la información.
- **SADatosPictograma**: este servicio es similar al anterior salvo que en vez de referirse a frases pictogramas está asociado a pictogramas.
- **SAEjercicio**: este servicio se encargará de obtener información sobre los ejercicios. También tiene una clase llamada **TransferEjercicio** que encapsula la información recibida de la tabla ejercicio.
- **SAFactoria**: se encarga de instanciar todos los servicios de aplicación que se quieran llamar.
- **SAFrasePictograma**: aparte de su clase para encapsular la información del ejercicio, llamado **TransferFrasePictograma**, tendrá métodos para obtener la información del ejercicio requerido.
- **SAPictograma**: este servicio es similar al anterior salvo que hace referencia a otro tipo de ejercicios: los pictogramas. Tendrá también su propia clase para la encapsulación de datos llamada **TransferPictograma**.
- **SAServicioRest**: este servicio está más enfocado a la utilización de los servicios de la API que se utiliza para obtener los pictogramas u otros datos ofrecidos por la API. En este caso solo nos servirán para obtener la URL de pictograma que queremos mostrar en la aplicación.
- **SAUtilidades**: este servicio se encarga encapsular funciones comunes que pueden usarse con asiduidad, como parsear un objeto **HTML**, asignar una imagen a una vista, hacer una petición web, etc.

6.4. Capa de Integración

Esta parte de la aplicación se encarga de realizar la persistencia de datos. La principal característica de esta capa es el uso de los *DAO* por entidades y el uso de las transacciones para la concurrencia de los datos: Consta de las siguientes funcionalidades:

- **DAOAlumno**: esta interfaz se encarga de obtener información de los alumnos.
- **DAODatosFrasePictograma** y **DAODatosPictograma**: esta interfaz se encarga de insertar y actualizar información de los datos generados tras utilizar la aplicación.
- **DAOEjercicio**: su función es la obtención de información de la base de datos sobre los ejercicios.
- **DAOFactoria**: esta interfaz se encarga de instanciar todos los datos *DAOs* generados.
- **DAOFrasePictograma** y **DAOPictograma**: se encargan de obtener información sobre los ejercicios del tipo que sea, ya sea de tipo pictograma o de tipo frases de pictograma.
- **TransactionFactoria** y **TransactionManager**: se utilizan para la conexión, gestión y concurrencia de la base de datos. Se encarga de que toda conexión o gestión se abra y se cierre correctamente.

7. Conclusión y trabajo futuro

En este capítulo se explica detalladamente los objetivos que se han conseguido alcanzar, qué dificultades se han encontrado a lo largo del desarrollo del proyecto y también qué cosas se podrían añadir y mejorar a la aplicación web y móvil.

7.1. Objetivos alcanzados

Tras haber revisado los objetivos para este proyecto, se puede afirmar que se han llevado a cabo casi en su totalidad.

En un principio se pensó únicamente una aplicación con varios minijuegos, pero en la que el profesor o tutor no tendría opción de influir en el desarrollo de esos juegos, sino más bien en una colección de juegos predefinida. Esta idea no terminó de convencer, por lo que se modificó y se pensó que el profesor tuviese más impacto en la creación y contenido de los minijuegos, para que estuviesen más enfocados a las circunstancias particulares de cada grupo de alumnos.

Esto llevó a que se haya creado una aplicación que gestiona los alumnos y los ejercicios, por lo que el profesor podrá tener una implicación más directa con los alumnos y enfocarse más en las necesidades del alumno.

Se alcanzó el objetivo de que la aplicación web y móvil fuesen modulares, para que a esta aplicación se puedan añadir nuevos juegos, siendo posible su gestión por el profesor.

7.2. Dificultades encontradas

Según se iba desarrollando el proyecto surgieron distintas dificultades a resolver:

- Una de las principales dificultades fue el desconocimiento de la tecnología de desarrollo de aplicaciones *Android*, lo que provocó una gran inversión de tiempo y esfuerzo en el conocimiento de la herramienta y el lenguaje.
- Otra de las dificultades en la aplicación móvil fue que existían pocas librerías para establecer la conexión con bases de datos, ya que era preferible utilizar la misma base de datos para el profesor que para el alumno.
- Elegir la dinámica, como mostrar las imágenes en la aplicación móvil y crear el diseño fue bastante complicado, ya que se realizaron varios diseños hasta elegir el más adecuado.

7.3. Trabajo futuro

Una de las grandes fortalezas de esta aplicación es que se hizo de una forma muy modular para que se puedan ir añadiendo muchos tipos de ejercicios, obteniendo todos los datos de la API. Por ello, de cara un futuro, se podrían añadir más juegos y mejorar el gestor del contenido del profesor:

- En lo que respecta a la web, se podría implementar un servicio que facilite consultas a la API de una manera más dinámica para el profesor. Como, por ejemplo, si el profesor quiere buscar un pictograma de una palabra, es posible que según escriba en el cuadro de texto correspondiente se indique si existe esa palabra y, en caso afirmativo, mostrarla, o bien se indique la definición de una palabra.
- Otra funcionalidad interesante sería tener un recuento de las palabras más buscadas por el profesor para saber cuáles son las más relevantes de cara a un futuro de mejora y actualización de la API.
- En la aplicación móvil se podrían desarrollar nuevos minijuegos como, por ejemplo, a través de una definición identificar que palabra es la correcta, o incorporar otros juegos de memoria entre palabras y pictogramas.

8. Conclusions and future work

This chapter explains in detail the objectives that have been achieved, what difficulties have been encountered throughout the development of the project and also what things could be added and improved to the web and mobile application.

8.1. Achieved goals

After reviewing the objectives for this project, it turns out that they have been carried out almost entirely.

At first, only one application with several mini-games was conceived, but in which the teacher or tutor would have no option to influence the development of these games, instead it would be a more predefined collection of games. This idea did not seem quite convincing, so it was modified and it was thought that the teacher should have more impact on the creation and content of the mini-games, so that they were more specific to the students' needs.

This led to the creation of an application that manages the students and the exercises, so the teacher can have a more direct involvement with the students and focus more on the needs of the student.

The goal of modularity in the web and mobile application were modular was reached, so that this application can be extended with new games and can be managed by the teacher.

8.2. Difficulties encountered

As the project was developed, different difficulties arose:

- One of the main difficulties was the lack of knowledge of Android application development technology, which caused a great investment of time and effort in acquiring the necessary knowledge of the tool and the language.
- Another difficulty in developing the mobile application was that there were too few libraries to establish the connection to databases, since it was preferable to use the same database for the teacher as for the student.
- Choosing the dynamics, such as displaying the images in the mobile application and creating the design was quite complicated, since several designs were made until the most appropriate one was chosen.

8.3. Future work

One of the strengths of this application is that it was done in a very modular way so that many types of exercises can be added by obtaining all the corresponding API data. Therefore, for the future, more games could be added and the teacher's content manager could be improved:

- Regarding the web, a service that facilitates API queries could be implemented in a more dynamic way for the teacher. As, for example, if you want to look for a pictogram of a word, it might be possible that, while the word is being writing in a text field, it is indicated whether that word exists and, in this case, show it or, for example, show the definition of a word.
- Another interesting functionality would be to have a count of the most searched words by the teacher to know which are the most relevant for the future of improvement and updating of the API.
- In the mobile application, new mini-games could be developed, such as identifying a word through a definition, or including other memory games between words and pictograms

Bibliografía

- [1]<https://www.glifing.com/metodo-glifing/>, Que es el método Glifing, 2016
- [2]<https://docs.microsoft.com/es-es/ef/#pivot=entityfmwk&panel=entityfmwk1>,
Documentación Entity Framework, 2018
- [3] <https://getbootstrap.com/docs/4.3/about/overview/>, About Bootstrap, 2018
- [4] Scott McCracken. Curso de programación Web con HTML 5, CSS, JavaScript, PHP 5/6, MySQL. InforBook's
- [5] <https://github.com/dbeaver/dbeaver/wiki>, About DBeaver, 2018