
**AdaptaMaterialEscolar2.0: Herramienta para la
adaptación de asignaturas a necesidades educativas
especiales**

**AdaptaMaterialEscolar2.0: Tool for adapting subjects to
special educational needs**



**Trabajo de Fin de Grado
Curso 2021–2022**

Autores

**Alberto Alejandro Rivas Fernández, Álvaro Gómez Sittima, Dunia
Namour Doughani, Johan Sebastian Salvatierra Gutierrez**

Directoras

**Virginia Francisco Gilmartín
Raquel Hervás Ballesteros**

**Trabajo de Fin de Grado en Ingeniería del Software
Facultad de Informática
Universidad Complutense de Madrid**

AdaptaMaterialEscolar2.0: Herramienta para la adaptación
de asignaturas a necesidades educativas especiales
AdaptaMaterialEscolar2.0: Tool for adapting subjects to
special educational needs

Trabajo de Fin de Grado en Ingeniería del Software

Autores

**Alberto Alejandro Rivas Fernández, Álvaro Gómez Sittima,
Dunia Namour Doughani, Johan Sebastian Salvatierra
Gutierrez**

Directoras

**Virginia Francisco Gilmartín
Raquel Hervás Ballesteros**

Dirigida por el Doctor

**Virginia Francisco Gilmartín
Raquel Hervás Ballesteros**

**Trabajo de Fin de Grado en Ingeniería del Software
Facultad de Informática
Universidad Complutense de Madrid**

17 de octubre de 2022

Autorización de difusión

Los abajo firmante, matriculados en el Grado en Ingeniería del Software de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: AdaptaMaterialEscolar2.0, realizado durante el cuarto curso académico bajo la dirección de Virginia Francisco Gilmartín y Raquel Hervás Ballesteros en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Alberto Alejandro Rivas Fernández, Álvaro Gómez Sittima, Dunia
Namour Doughani, Johan Sebastian Salvatierra Gutierrez

17 de octubre de 2022

Dedicatoria

Texto de la dedicatoria...

Agradecimientos

Texto de los agradecimientos

Resumen

Resumen en español del trabajo

Palabras clave

Máximo 10 palabras clave separadas por comas

Abstract

Abstract in English.

Keywords

10 keywords max., separated by commas.

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
2. Introduction	3
3. Estado del arte	5
3.1. Adaptación curricular	5
3.1.1. Concepto de la adaptación curricular	5
3.1.2. Adaptaciones de acceso	6
3.1.3. Adaptaciones del currículo	7
3.1.4. Ejemplificación de algunas adaptaciones curriculares asociadas a diferentes necesidades	9
3.1.5. Conclusión	10
3.2. Herramientas existentes para adaptaciones curriculares	11
3.3. AdaptaMaterialEscolar1.0	11
3.3.1. Captura de requisitos	11
3.3.2. Diseño de la aplicación	12
3.3.3. Implementación	12
3.3.4. Evaluación	13
4. Chicha	15
4.1. Metodología de desarrollo	15
4.1.1. Políticas explícitas	15
4.1.2. Clases de servicios	16
4.2. Plan de Pruebas	17
4.2.1. Tipos de pruebas	17
4.2.2. Pruebas relacionadas con la memoria	17
4.2.3. Pruebas de implementación	17

4.2.4. Integración continua	19
4.3. AdaptaMaterialEscolar2.0	19
4.3.1. Requisitos	19
5. Conclusiones y Trabajo Futuro	23
5. Conclusions and Future Work	25
Bibliografía	27

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Motivación

La educación escolar tiene como objetivo promover el desarrollo de ciertas habilidades, y el aprendizaje de ciertos contenidos necesarios para que los estudiantes se conviertan en miembros activos de la sociedad. Para ello, la escuela debe dar respuestas educativas que eviten la discriminación y promuevan la igualdad de oportunidades.

En el currículo escolar, todos los alumnos tienen necesidades educativas comunes. Sin embargo, no todos los estudiantes se enfrentan con las mismas capacidades de aprendizaje, sino que cada alumno tiene necesidades individuales. La mayoría de estas se abordan a través de acciones simples: dar a los alumnos más tiempo para aprender determinados contenidos, diseñar actividades, etc. Sin embargo, también existen necesidades individuales que no se pueden atacar por estos medios, lo que precisa una serie de medidas didácticas especiales, diferentes de las normalmente requeridas para la mayoría de los estudiantes. Dichas necesidades se pueden satisfacer con las adaptaciones curriculares. Estos ajustes curriculares no significativos para ACNEE (alumnos con necesidades educativas especiales) son realizadas por los profesores, pero no se les facilita una herramienta para ello, por esa razón nació `AdaptaMaterialEscolar1.0`.

1.2. Objetivos

La finalidad de este TFG es proporcionar una herramienta para el profesorado con el fin de adaptar los contenidos de las asignaturas de forma intuitiva, rápida y simple, con el objetivo de hacer unidades didácticas que se adapten a las necesidades de los alumnos.

Para poder ayudar al profesorado, partimos de la versión anterior, `Adap-`

taMaterialEscolar1.0 que incluye la creación de diferentes tipos de actividades para adaptar el material (por ejemplo, sopas de letras, rellenar espacios en blanco, etc). Después de estudiar dicha herramienta nos encontramos con que faltan algunas funcionalidades tales como un generador de resúmenes, un manual de usuario, una opción para poder cambiar el PDF, crear esquemas, realizar ejercicios de flechas, etc. Tras la evaluación por parte de los profesores, enunciaron una serie de propuestas de mejora como exportar el documento a formato Word para hacer modificaciones, añadir ejercicios con espacio para dibujar, enumerar ejercicios de forma automática, etc. Con relación a los problemas encontramos que el diseño de la aplicación no se adapta a diferentes tamaños de pantalla.

En conclusión, nos proponemos solventar los fallos, además de añadir las funcionalidades que faltan para proporcionar una herramienta más completa, con el fin de ayudar al profesorado pero también al alumno.

Chapter 2

Introduction

Capítulo 3

Estado del arte

3.1. Adaptación curricular

En nuestro sistema educativo, aceptar la diversidad del alumnado y la individualidad de cada uno de ellos, constituye la base del quehacer de los docentes. Los profesores deben modificar el currículum (*Regulación de los elementos que determinan los procesos de enseñanza y aprendizaje para cada una de las enseñanzas y etapas educativas.*) y el programa de aula con el fin de mejorar el desarrollo del aprendizaje. Para poder realizar esta actividad, el profesorado deberá detectar, evaluar y valorar al alumno y a los elementos curriculares y del entorno.

Una vez detectado esto, el profesorado se encuentra preparado para amoldarse a las circunstancias del alumno, gracias a esto el estudiante adquiere la atención educativa que requiere, logrando una mejora en su desarrollo personal y social.

Con esto, determinamos que la adaptación curricular es un mecanismo para particularizar el currículum, ayudando a efectuar las labores docentes con el fin de apoyar al alumno a conseguir el nivel exigido por el Currículum Oficial.

3.1.1. Concepto de la adaptación curricular

El BOE (2010) define la adaptación curricular como *la medida de modificación de los elementos del currículo a fin de dar respuesta a las necesidades del alumnado. En todo caso, la adaptación tendrá como referente los objetivos y las competencias básicas del currículo que corresponda.*

Es decir, la adaptación curricular es cualquier adaptación personal para estudiantes, cuyas necesidades no se encuentran cubiertas por el currículo ordinario y por tanto, no pueden acceder a él de la misma manera que sus compañeros.

En definitiva, son planes de acción y estrategias didácticas que incluye las modificaciones del currículo, asegurándose de que los estudiantes tengan éxito en el proceso de aprendizaje y alcancen los objetivos generales definidos.

Para poder aplicar la adaptación curricular a un alumno se debe precisar el tipo de adaptación en función de sus necesidades, para que pueda alcanzar los objetivos propuestos. Para ello, definimos los tipos de adaptaciones que más uso tienen, enfocándonos tanto en el acceso como en el currículum.

3.1.2. Adaptaciones de acceso

Las adaptaciones de acceso permiten al alumno acceder a los diferentes componentes del currículum. No implica una adaptación del currículum sino una adaptación en los recursos materiales, espaciales o de comunicación para que los alumnos con necesidades educativas especiales puedan desarrollar el currículum. Dicha adaptación puede tomar a su vez diferentes tipos.

3.1.2.1. Acceso espacial

Hacen referencia a las adaptaciones en relación con el espacio. Destacan las siguientes:

- **Adaptación en la sonorización del aula:** Significa que las aulas deben tener un cierto nivel de volumen adecuado, sin que haya ruidos continuos, sin eco, etc. Dichas clases son especialmente adecuadas para estudiantes que tienen impedimentos auditivos o visuales o que requieren, por su propia condición especial, un entorno con pocos estímulos auditivos que les distraiga.
- **Adaptación en la iluminación del aula:** Las aulas bien iluminadas requieren que estas no tengan sombras, deben poseer ventanales que suministren luz natural o en su lugar luz artificial. Estos requisitos son necesarios para los estudiantes con discapacidades sensoriales.
- **Adaptación en el espacio físico:** Es todo aquello relacionado con la ausencia de barreras arquitectónicas: braille en las puertas, ascensores, pasamanos, remate de las escaleras rugoso, etc. En esta sección también se encuentran los aspectos relacionados con la ubicación del aula (sin escaleras, lugares poco ruidosos) y con la disposición del estudiante dentro del aula (al lado de un enchufe, del profesor, de la puerta, etc.)

3.1.2.2. Acceso material

Se adaptan materiales de uso frecuente o se introducen nuevos instrumentos que faciliten el desarrollo del aprendizaje.

- **Material adaptado:** Se refieren a materiales que se usan habitualmente, se adaptan para un uso apropiado por parte de los alumnos con necesidades especiales, ejemplo de ello es plastificar un libro o hacer dibujos con relieve.
- **Materiales específicos:** Los materiales específicos deben superar las dificultades de los niños, por ejemplo el mobiliario, las sillas y mesas deben de estar adaptadas, los comunicadores electrónicos con salidas de voz o escrita, etc.

3.1.2.3. Acceso de comunicación

Algunos alumnos son incapaces de comprender o expresarse por medio del lenguaje oral, o su nivel no es suficiente para poder comunicarse correctamente. Dichos alumnos requieren estudiar y usar códigos de comunicación suplementarios al lenguaje oral, o alguna alternativa al mismo. Aprender dichos códigos de comunicación facilitarán el acceso a los elementos curriculares ordinarios, les concederá una herramienta imprescindible tanto para el desarrollo de algunas competencias y aprendizajes de diferentes contenidos, como para relacionarse y comunicarse con el resto de personas. De esta manera, podemos destacar los siguientes sistemas que mejoran la entrada a la comunicación:

- **Sistemas alternativos a la comunicación:** En estos sistemas no se requiere del lenguaje oral para su utilización. Ejemplo de estos procedimientos son los símbolos pictográficos para la comunicación, Sistema de comunicación Bliss (sistema gráfico), la lengua de signos española, etc. los gestos, lenguaje corporal, lenguaje por señales, tableros de comunicación
- **Sistemas complementarios a la comunicación:** No sustituyen al lenguaje oral, solo lo acompaña añadiendo más información visual. Se destacan la palabra complementada (para que las personas con discapacidad auditiva puedan distinguir entre fonemas con la misma articulación), la comunicación bimodal (Es un sistema de comunicación que utiliza el habla y los signos al mismo tiempo)

3.1.3. Adaptaciones del currículo

Como se ha mencionado anteriormente, la adaptación curricular es un conjunto de refinamientos y cambios en los elementos del currículum para adaptar la respuesta educativa a las necesidades formativas de los estudiantes especiales.

Las adaptaciones curriculares se dividen en dos tipos, las adaptaciones curriculares significativas y las adaptaciones curriculares no significativas.

3.1.3.1. Adaptaciones curriculares significativas

Según el BOE (2010), *una adaptación curricular será significativa cuando la modificación de los elementos del currículo afecten al grado de consecución de los objetivos, los contenidos y los aprendizajes imprescindibles que determinan las competencias básicas en la etapa, ciclo, grado, curso o nivel correspondiente.*

Es decir, la adaptación curricular significativa son los ajustes que se realizan en el currículum. Para su elaboración e implementación se debe seguir el criterio de menor a mayor significatividad, el enfoque sería el siguiente:

- **Inclusión.** Se introducen elementos curriculares no presentes en el currículo. Puede incorporar objetivos, contenidos, criterios de evaluación, etc conforme a las necesidades del alumno.
- **Reformulación.** Esta adaptación conlleva la amplia modificación de los elementos del currículo.
- **Temporalización fuera de ciclo.** Los alumnos con ritmo de aprendizaje lento con respecto a sus compañeros, tendrán la oportunidad de conseguir los objetivos en otro ciclo posterior, posponiendo a otras etapas algunos elementos curriculares.
- **Eliminación.** Este tipo de adaptación es la más significativa. Inicialmente se deben eliminar contenidos, a continuación, criterios de evaluación y objetivos, finalmente se propondrá quitar material.

3.1.3.2. Adaptaciones curriculares no significativas

Son adaptaciones que no modifican sustancialmente el contenido del currículo oficial, es decir, se adapta la metodología, las actividades, los tiempos, las técnicas e instrumentos de evaluación. Para su elaboración se debe seguir el criterio de menor a mayor significatividad, los aspectos serían los siguientes:

- **Cómo evaluar:** Se ajusta la manera de evaluar a las necesidades del alumno, ejemplo de ello es cuando un alumno con escayola no puede realizar un examen escrito por lo que se le adapta la forma de evaluar realizando un examen oral.
- **Metodología:** Aquí se hace mención a cómo se enseña, por ejemplo, la manera de explicar a algunos alumnos que a otros, es decir, se puede dar la situación de que un alumno requiera que le expliquen las cosas más lentamente que a otro estudiante.
- **Priorización de objetivos o contenidos:** Dentro de la planificación se podría dar más valor a unos contenidos que a otros.

- Temporalización de contenidos u objetivos: Permitir más tiempo para alcanzar algunos de los contenidos pero respetando el ciclo, ejemplo de esto es trabajar elementos de segundo en tercero sin que sea muy significativo.

3.1.4. Ejemplificación de algunas adaptaciones curriculares asociadas a diferentes necesidades

- Discapacidad motora: Es un grupo de alteraciones que se producen como consecuencia de diversas anomalías en los Sistemas que forman el movimiento. Este tipo de discapacidad requiere adaptaciones de acceso tales como rampas, pasamanos, suelos antideslizantes, etc.

En relación a las adaptaciones significativas atañen sobre todo al área de Educación Física, Música o Plástica ya que en estas modalidades se precisa del manejo de instrumentos.

Con respecto a las adaptaciones no significativas se debe adaptar la forma de evaluar ya que se debe tener en cuenta su movilidad.

- Discapacidad auditiva: Es la pérdida parcial o completa de la audición, supone la obtención del lenguaje oral por otras vías como por la visual.

Con respecto a las adaptaciones de acceso, el alumno se debe encontrar en una zona del aula en la que no haya muchas sombras ya que la adquisición de conocimientos se realiza por vía visual. Por otro lado, los alumnos con una pérdida parcial de la audición necesitan de un ambiente poco ruidoso.

En relación a las adaptaciones curriculares significativas, los profesores deberán trabajar de forma conjunta con especialistas en audición y lenguaje para que el alumno logre alcanzar los objetivos conectados con el lenguaje oral.

En cuanto a las adaptaciones curriculares no significativas, hay que tener en cuenta la manera de evaluar, además de la forma de hablar al alumno, esta debe ser de un modo claro, sin gesticular excesivamente, etc.

- Discapacidad intelectual: El funcionamiento intelectual es significativamente más bajo que la media. En este tipo de discapacidad no es muy relevante las adaptaciones de acceso, pero podemos destacar el posicionamiento del alumno en el aula, de manera que se encuentre en una zona donde no tenga muchas distracciones.

Las adaptaciones curriculares significativas se aplicarán en función de su nivel de competencia curricular.

En relación a las adaptaciones curriculares no significativas se centrarán en la metodología, como por ejemplo se incentivará la motivación y el refuerzo positivo.

- Espectro autista: Discapacidades del desarrollo en el cerebro. En relación a las adaptaciones de acceso al espacio se precisa no realizar grandes cambios en la disposición del mobiliario. También destacamos las adaptaciones de comunicación, ya que las personas autistas se caracterizan por la ausencia de comunicación, para ayudar a romper la barrera de la comunicación lo que realizan es asociar palabras con gestos e impulsar un refuerzo positivo.

En cuanto a las adaptaciones curriculares significativas, se debe introducir o priorizar el contenido en lo que respecta a la comunicación o rediseñar los objetivos o elementos que no alcancen.

Las adaptaciones curriculares no significativas se centran en la metodología. Las actividades deben de ser consistentes, con una estructura y organización claras.

- Altas capacidades intelectuales: En este caso no se requiere de adaptaciones de acceso ya que este tipo de alumnos no tienen dificultades para acceder al currículum.

Con respecto a las adaptaciones curriculares significativas lo que deberían realizar los profesores es ampliar el currículum añadiendo objetivos y contenidos.

Las adaptaciones curriculares no significativas hacen hincapié en la metodología por ejemplo, haciendo actividades de ampliación.

3.1.5. Conclusión

Tras una exhaustiva documentación acerca de las adaptaciones y aportar ejemplos de ello, determinamos que las adaptaciones han de ser individualizadas ya que cada alumno tiene sus propias dificultades. El profesorado ha de ser el encargado de la adaptación, pero su realización es algo tediosa y no disponen de herramientas que faciliten la modificación del material. Por ello, en este TFG nos centraremos exclusivamente en las adaptaciones curriculares no significativas, realizando una aplicación para que el profesorado pueda adaptar el material. Por otra parte, nuestro enfoque está en las adaptaciones mencionadas anteriormente, ya que son las únicas que podemos tratar, puesto que no tenemos la potestad de modificar y/o acceder al currículum.

3.2. Herramientas existentes para adaptaciones curriculares

Widgit Symbols¹ es un conjunto de símbolos coloridos y sencillos que cubren un amplio vocabulario de palabras y admite 17 idiomas, principalmente el inglés. Los propios creadores de Widgit también ofrecen herramientas² que permiten combinar la escritura de texto con pictogramas utilizando los Widgit Symbols. Por ejemplo, InPrint 3³ permite utilizar plantillas y editarlas para relacionar texto con pictogramas facilitando la adaptación de cualquier información, documento o recurso didáctico.

EducaPlay⁴ es una web que permite crear actividades interactivas o juegos didácticos. Entre las actividades que se pueden realizar en EducaPlay hay sopas de letras, pruebas, juegos de memoria, mapas interactivos, ejercicios de relacionar columnas, etc. El problema de esta web es que está diseñada para realizar las actividades en la web y muchas de las actividades no se pueden imprimir en papel, por lo que requieren un ordenador con acceso a internet.

3.3. AdaptaMaterialEscolar1.0

Una vez introducida la adaptación curricular, pasamos a describir AdaptaMaterialEscolar1.0 con el fin de exponer la aplicación anterior y de esta forma poder comprender la nueva versión de esta.

3.3.1. Captura de requisitos

La captura de requisitos se realizó hablando directamente con el usuario final para poder conocer sus necesidades reales. Para esto se hicieron varias reuniones con 2 profesoras del Aula TEA (Trastornos del Espectro Autista) del IES Maestro Juan de Ávila de Ciudad Real.

También, se presentó la aplicación en un workshop del grupo de investigación (Natural Interaction based on Language). Los profesores pidieron a los alumnos del TFG añadir una funcionalidad que permita introducir en el papel el espacio suficiente para aquellos los alumnos con un amplio tamaño de letra.

A partir de estas reuniones se decidió que la aplicación fuera web, además de no ser necesario un sistema de iniciar sesión, y se capturó una lista de funcionalidades que se deberían implementar. Se llegó a la conclusión de que la aplicación fuera web ya que no todos los profesores tienen un ordenador

¹https://www.widgit.com/about-symbols/widgit_symbol_set.htm

²<https://www.widgit.com/products/index.htm>

³<https://www.widgit.com/products/inprint/index.htm>

⁴<https://es.educaplay.com/?lang=es>

personal en el que puedan instalar aplicaciones. Por otro lado, se decidió que no hacía falta iniciar sesión en la aplicación, para que los profesores no tengan que usar su información personal. Para poder priorizar las funcionalidades, se le envió una lista con todos los requisitos a las profesoras, para que ellas les dieran una importancia del 1 al 3, siendo el 1 la menor importancia y el 3 la mayor importancia. Luego, el equipo de desarrollo les dio una dificultad del 1 al 3, siendo el 1 la mayor dificultad y el 3 la menor dificultad. A continuación, se hizo una tabla con las importancias y dificultades medias de cada requisito. Estos 2 valores se multiplicaron para obtener un resultado, según el cual se ordenaron las funcionalidades de mayor prioridad a menor.

3.3.2. Diseño de la aplicación

Para realizar el diseño de la aplicación, el equipo empezó haciendo un boceto en papel entre todos. Luego, llevaron a cabo un diseño en Moqups, una aplicación que permite crear prototipos, diagramas, etc.

Para conseguir mejorar el diseño, decidieron que cada miembro del equipo debería realizar un prototipo por su cuenta para luego ponerlos todos en común, y poder ver las diferentes perspectivas. Se hizo así para que cada persona pudiera plasmar sus ideas sin ser influenciada por las ideas de los demás.

Finalmente, se realizó el diseño final de la aplicación después de poner en común los prototipos de todo el equipo. Una vez terminado, se le envió a las profesoras del Aula TEA para poder recibir feedback, el cual fue positivo.

3.3.3. Implementación

Las funcionalidades implementadas fueron las siguientes:

- Subir un documento fuente PDF, a partir del cual se puede realizar las adaptaciones, ejercicios, etc.
- Editor, en el que se pueden añadir y modificar las adaptaciones. También sirve como editor de texto, en el que se puede cambiar la fuente de letra, el color, posicionamiento del texto, etc.
- Buscador de pictogramas que permite añadirlos al editor.
- Ejercicio de completar huecos, en el que a partir de un texto puedes seleccionar las palabras que deben ser completadas.
- Ejercicio de definiciones.
- Ejercicio de desarrollo, en el que se puede crear un enunciado y añadir un cierto número de líneas para la respuesta.

- Sopa de letras, con ciertas palabras dadas por el usuario y un cierto tamaño.
- Ejercicio de verdadero o falso.

Para desarrollar AdaptaMaterialEscolar1.0, se utilizó React, una biblioteca de JavaScript de código abierto mantenida por Meta para crear interfaces de usuario. React nos permite crear un front-end interactivo y complejo de una forma mucho más fácil que utilizando Javascript puro. Se basa en una arquitectura de componentes reutilizables que tienen su propio estado y que juntos forman una aplicación completa.

Para simplificar la gestión del estado de la aplicación, se utilizó Redux, un patrón de arquitectura de datos que permite controlar el estado de la aplicación de manera predecible, reduciendo el número de relaciones entre componentes de la aplicación, manteniendo un flujo de información sencillo.

Para las pruebas de unidad se utilizó Jest, una librería de JavaScript para testeo mantenida por Meta.

En vez de implementar un editor de texto, se decidió utilizar una librería externa, ya que desarrollar un editor desde cero llevaría demasiado trabajo y sería ineficiente porque ya existen editores que se pueden reutilizar. En este caso se utilizó CKEditor.

Por último, para el buscador de pictogramas, se empleó la API de ARA-SAAC, la cual fue creada por el Centro Aragonés para la comunicación Aumentativa y Alternativa. Esta API nos permite hacer una petición con un término de búsqueda y nos devuelve una serie de pictogramas relacionados.

3.3.4. Evaluación

El objetivo de la evaluación fue descubrir si la aplicación es realmente útil para los profesores y si les ayuda a resolver sus problemas. Para esto se creó un examen de Ciencias Naturales adaptado usando AdaptaMaterialEscolar1.0. Luego, se replicó este examen con los profesores para mostrarles cómo se usaría la herramienta en situaciones reales.

Después de esta demostración, se les hizo una encuesta a los profesores para que pudieran dar su opinión y feedback. Este cuestionario tenía preguntas sobre usabilidad, diseño, funcionalidades y utilidad real de la aplicación. En el cuestionario se usó una serie de preguntas llamadas Escala de Usabilidad de un Sistema o SUS (System Usability Scale), que sirven para medir que tan buena es la usabilidad de un sistema.

Esta evaluación de la aplicación se realizó con varios profesores en diferentes días. Primero se hizo con las 2 profesoras del aula TEA, la orientadora y otra profesora del IES Maestro Juan de Ávila, de Ciudad Real. Luego 3 docentes de Biología, Historia y Geografía se pusieron en contacto con el equipo ya que también querían probar la aplicación. Por último, la jefa de

estudios y profesora del IES Pedro Álvarez de Sotomayor, de Manzanares, también se vio interesada ya que en su instituto había un alumno con sordera total.

Como resultado de la evaluación se llegó a la conclusión de que la aplicación sí resuelve problemas reales que tienen los profesores y que se debería seguir desarrollando.

En la encuesta se obtuvo un 99 sobre 100 en la escala SUS. También se obtuvo un 4,6 sobre 5 en estética y se recomendó que cada pestaña de las funcionalidades fuese de un color diferente para diferenciarlas mejor. También se observó que los pictogramas a veces eran muy pequeños y debían poder aumentarse de tamaño.

Se obtuvo una nueva lista de requisitos a implementar para mejorar la aplicación:

- Traducir pictogramas a lenguaje natural y viceversa.
- Cuadrícula para ejercicios de matemática.
- Poder añadir doble pauta, en vez de renglones de una línea para determinar el tamaño de letra del alumno.
- Poder recortar imágenes.
- Añadir encabezado con el nombre del centro educativo, asignatura y nombre del alumno.
- Añadir espacio para dibujar.
- Fórmulas con huecos que puedan ser rellenados por el alumno.
- Enumerar ejercicios automáticamente.
- Fuente de letra escolar.

Capítulo 4

Chicha

4.1. Metodología de desarrollo

La metodología es un marco de trabajo que permite estructurar, organizar y controlar el proceso del desarrollo software.

Dentro de la metodología encontramos dos tipos, las metodologías tradicionales y las metodologías ágiles, unas de las diferencias más notorias son la flexibilidad y la adaptabilidad.

Teniendo en cuenta lo anterior, hemos decidido aplicar una metodología ágil en nuestro proyecto. En concreto hemos implementado la metodología Kanban ya que es más flexible que el resto de metodologías ágiles existentes. Además, al ser un grupo de cuatro y tener todos el mismo nivel de conocimientos no podríamos definir roles, lo que sustenta nuestra elección.

4.1.1. Políticas explícitas

Para saber cómo está organizado el trabajo describimos las siguientes políticas explícitas:

- Límites del trabajo en curso (WIP): 2 tareas en curso por persona en todo el tablero sin tener en cuenta la columna validate. que debemos cumplir.
- Definición de *done*:
 - Tareas de memoria: Cuando hayan sido dadas por validadas por las tutoras.
 - Tareas de implementación: Cuando hayan pasado todo el plan de pruebas.
- Cuando un integrante del grupo haya terminado su tarea él será el encargado de moverla a la columna correspondiente.

- Cualquier integrante del grupo puede poner una tarea en el tablero tras consultarlo con el resto.
- El 16 de mayo entrega del borrador.
- El 23 de mayo es la entrega final.
- Las reuniones se llevarán a cabo todos los domingos alrededor de las 12:00, para poner en común el trabajo realizado.
- Si se produce un error en una función, se añadirá una tarea para resolverlo.
- Los elementos del trabajo continuarán a través del flujo cuando este se ajuste a la definición de la columna. La explicación de cada columna sería la siguiente: To do: Listado de todas las tareas sin empezar.

Dev:

Doing: Tareas que se encuentran en desarrollo, ya sea la implementación de código o redactar la documentación.

Done: Tarea que se encontraba en desarrollo ha finalizado.

Test: Una vez desarrollada la tarea, se probará que cumpla con los requisitos. Validate: La tarea de memoria será comprobada por las tutoras.

Done: Las tutoras han dado el visto bueno a la tarea de memoria o cuando un integrante del grupo haya finalizado su tarea de código.

4.1.2. Clases de servicios

Las clases de servicio que empleamos serán las siguientes:

- Expedite: Tareas que necesitan ser gestionadas de manera acelerada o urgente.
- Fixed Delivery Date: Tareas con fecha fija que debemos cumplir.
- Standard: Tareas que ya ha hecho antes el equipo y que no tienen una fecha fija.
- Intangible: Tareas que son nuevas, se desconoce el tiempo que se le va dedicar y el riesgo que suponen.

4.2. Plan de Pruebas

4.2.1. Tipos de pruebas

Previamente hemos distinguido dos tipos de tareas que habrá en el proyecto, las tareas relacionadas con la memoria y las de implementación. Cada tipo de tarea se probará o revisará de manera diferente.

4.2.2. Pruebas relacionadas con la memoria

La revisión de la memoria consistirá en la búsqueda de faltas ortográficas y errores gramaticales, y comprobación de la claridad del texto. También, se confirmará que no falta información. La revisión de la memoria la realizarán todos integrantes del equipo siguiendo los siguientes pasos:

1. Cuando haya una tarea de memoria en la columna de *DEV:DONE*, el miembro del equipo que vaya a revisarla se la asignará y la llevara a la columna de *TESTING*.
2. Tras acabar de revisar la tarea de memoria, la moverá de vuelta a la columna de *DEV:DONE*.
3. Cuando todos los miembros del equipo hayan revisado la tarea, el último revisor se encargará de mover la tarea a la columna de *VALIDATE*.

4.2.3. Pruebas de implementación

Las pruebas de implementación consistirán en dos tipos de pruebas: las pruebas unitarias y las pruebas de integración. Las pruebas de una tarea de implementación concreta las realizará algún miembro del equipo que no haya participado en el desarrollo de esta y las hará cuando la tarea se encuentre en la columna de *TESTING*. La ventaja de que las pruebas las realice un miembro que no se haya visto involucrado en el desarrollo de la tarea es que puede sacar más casos de prueba que aquellos miembros que han implementado la tarea y conocen el código.

Para ambos tipos de pruebas, unitarias y de integración, se utilizará la herramienta Jest ([ver sección de herramientas](#)).

4.2.3.1. Pruebas Unitarias

Las pruebas unitarias son pruebas, mayoritariamente automatizadas, que verifican la funcionalidad de una unidad software (componente, clase o método), de forma aislada. Esta verificación no se debe ver afectada por otros casos de prueba que se hayan ejecutado anteriormente o por dependencias con otros módulos de la aplicación. Para evitar que las dependencias de un

módulo, por ejemplo el acceso a una base de datos o una petición a una API externa, afecten al resultado de la prueba se pueden usar *stubs* para simular esas dependencias.

La razón por la que vamos a realizar pruebas unitarias es porque permiten detectar errores en fases tempranas y evitar que los errores se propaguen a fases posteriores, aumentando la calidad del software. Además, facilitan los cambios, ya que se puede comprobar rápidamente que estos no han afectado al funcionamiento esperado de la aplicación.

Para aprovechar al máximo las ventajas que ofrecen las pruebas unitarias hay que asegurarse de que cumplen las siguientes características:

- Deben ser automáticas y repetibles, es decir, se deben poder ejecutar tantas veces como uno quiera sin necesidad de intervención manual durante las pruebas.
- Deben ser rápidas, ya que si el ejecutar las pruebas unitarias es un proceso lento, no se va a llevar a cabo tantas veces como sea necesario.
- Cada caso de prueba debe estar aislado completamente del resto de casos o pruebas de otros módulos. Si no se aísla correctamente, se pueden producir resultados no consistentes, complicando la detección de errores.
- Las pruebas deben ser relevantes para el futuro y deben tratar de cubrir la totalidad del código.

4.2.3.2. Pruebas de integración

Las pruebas de integración se utilizan para comprobar que las conexiones o interfaces entre los distintos módulos, ya probados individualmente (pruebas unitarias), funcionan correctamente.

En este proyecto las pruebas de integración se realizarán de forma incremental, es decir, los módulos se integran uno a uno, a medida que estos se van desarrollando y probando. En esta estrategia, la integración se lleva a cabo de arriba abajo, siguiendo el flujo natural de la aplicación.

Se seguirá una estrategia *Top-Down* para realizar las pruebas. Utilizaremos esta estrategia, en vez de *Bottom-Up*, porque nos queremos centrar en errores de diseño y tener una aplicación funcional probada lo antes posible. Puede darse el caso de que una prueba de integración no se pueda realizar debido a que alguno de los módulos a integrar está en desarrollo o todavía se está probando individualmente. En este caso, se utilizarán *stubs* para simular la dependencia entre módulos.

4.2.4. Integración continua

Para complementar las pruebas unitarias y de integración, en este proyecto se implementará integración continua. La integración continua es la práctica de desarrollo software mediante la cual los miembros del equipo combinan su trabajo frecuentemente en un repositorio compartido. Cada integración se verifica mediante una serie de fases por las que va pasando el software y que se automatizan, esta serie de fases se conoce como *pipeline*.

El *pipeline* que se va a implementar en este proyecto constará de 3 fases principales: *Build*, *Test* y *Release*. Cuando un miembro integre una nueva versión del proyecto en el repositorio compartido, pasará primero por la fase *Build*, durante la cual se instalarán las dependencias necesarias en una máquina virtual y se preparará todo lo necesario para pasar a la fase *Test*. En la fase *Test*, se ejecutarán todas las pruebas especificadas en el plan de pruebas a la versión que se está intentando subir al repositorio. En caso de que el resultado de las pruebas sea correcto, se pasará a la fase *Release*, en caso de que se produzca un error en alguna de las pruebas se abortará la ejecución de cualquier prueba restante, no se pasará a la fase *Release*, por lo tanto no se subirán los cambios realizados en la versión que se estaba compartiendo. En la fase *Release*, se subirán los cambios automáticamente al repositorio.

Las razones por las que creemos necesaria la integración continua en este proyecto son las siguientes:

- La integración continua mejora la productividad de desarrollo, ya que libera a los desarrolladores de tener que ejecutar las pruebas manualmente y tener que esperar a que se arreglen errores en el repositorio. Si al integrar el trabajo se produce un error en la verificación se notificará a los desarrolladores y el cambio que ha producido ese error no se subirá al repositorio compartido.
- Mejora la detección de errores debido a la ejecución de pruebas de forma automática y frecuente, lo que permite a los desarrolladores descubrir los errores y arreglarlos antes de que se conviertan en problemas graves.

En este proyecto se utilizará la herramienta de GitHub Actions (ver sección de herramientas) para implementar la integración continua.

4.3. AdaptaMaterialEscolar2.0

4.3.1. Requisitos

Adaptaciones de temario		
Funciones	Coste	Importancia
Generar un resumen a partir de un texto.	2	3
Crear esquemas que faciliten la visualización	1	3
Crear tablas que organicen el temario y/o las actividades, seleccionando contenido	1	1,5

Adaptaciones de actividades y/o exámenes		
Funciones	Coste	Importancia
Ejercicios de relacionar contenido mediante flechas.	2,5	2
Ejercicios de completar los espacios en blanco en tablas y esquemas.	1	1,2

Adaptaciones de formato		
Funciones	Coste	Importancia
Sustituir una palabra por una imagen.	2	1
Añadir una leyenda de colores con la categoría de cada tipo.	1,5	1
Añadir leyenda de colores para el tema de cada asignatura.	2	1
Estandarizar formato para títulos e índices del temario.	1	1
Añadir imágenes buscando una palabra en bases de datos de imágenes libres.	1,8	2
Enumerar ejercicios de forma automática.	1	2,5
Añadir un tipo de fuente escolar.	1	1
Crear una herramienta de recorte de imágenes para el texto original.	1	1
Añadir encabezado al texto.	3	1,5
Añadir ejercicios con espacio para dibujar.	2,5	1
Añadir ejercicios para ejercicios de cálculo con fórmulas con huecos a rellenar por el alumno.	2,5	2
Enumerar ejercicios de forma automática.	1	2,5

Propuestas de mejora		
Funciones	Coste	Importancia
Añadir un pictotraductor como funcionalidad.	1	1
Añadir la alternativa de añadir doble pauta, en vez de renglones de una única línea, para determinar el tamaño de la letra del alumno.	2	1
Exportar el documento a formato Word para hacer modificaciones.	2	3
Añadir ejercicios de matemáticas con cuadrícula para escribir los números.	2	1

Priorización	
Funciones	Prioridad
Exportar el documento a formato Word para hacer modificaciones.	6
Generar un resumen a partir de un texto.	6
Ejercicios de relacionar contenido mediante flechas.	5
Añadir ejercicios para ejercicios de cálculo con fórmulas con huecos a rellenar por el alumno.	5
Añadir encabezado al texto.	4,5
Añadir imágenes buscando una palabra en bases de datos de imágenes libres.	3,6
Crear esquemas que faciliten la visualización	3
Añadir ejercicios con espacio para dibujar.	2,5
Enumerar ejercicios de forma automática.	2,5
Sustituir una palabra por una imagen.	2
Añadir leyenda de colores para el tema de cada asignatura.	2
Añadir ejercicios de matemáticas con cuadrícula para escribir los números.	2
Añadir la alternativa de añadir doble pauta, en vez de renglones de una única línea, para determinar el tamaño de la letra del alumno.	2
Crear tablas que organicen el temario y/o las actividades, seleccionando contenido.	1,5
Añadir una leyenda de colores con la categoría de cada tipo.	1,5
Ejercicios de completar los espacios en blanco en tablas y esquemas.	1,2
Añadir un tipo de fuente escolar.	1
Estandarizar formato para títulos e índices del temario.	1
Crear una herramienta de recorte de imágenes para el texto original.	1
Añadir un pictotraductor como funcionalidad.	1

Capítulo 5

Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Chapter 5

Conclusions and Future Work

Conclusions and future lines of work.

Bibliografía

AMAZON. <https://aws.amazon.com/es/devops/continuous-integration/>. n.d.

BOE. <https://boe.es/buscar/pdf/2010/boe-a-2010-5493-consolidado.pdf>. 2010.

DIEGO J. LUQUE PARRA, J. F. R. P. *Trastornos del desarrollo y la adaptación curricular..* 2002.

EDU. <http://www.ub.edu/casosenxarxa/glossary/adaptacion-curricular-ac/>. n.d.

EDUCA2. <https://www.educa2.madrid.org/web/eoep.discapacidadmotora.madrid>. n.d.

HELP, S. T. <https://www.softwaretestinghelp.com/incremental-testing/>. 2022.

REDHAT. <https://www.redhat.com/en/topics/devops/what-cicd-pipeline>. 2022.

SAMUEL GENTO PALACIOS, S. M. L., MERCEDES SÁNCHEZ SÁINZ. *Diseño y ejecución de planes, proyectos y adaptaciones curriculares para el tratamiento educativo de la diversidad..* Universidad Nacional de Educación a Distancia, 2010.

