

Sistema de Micro Aprendizaje basado en Telegram: Sistema de Flashcards

Angel Colombo Vega

Resumen—El microlearning es un método de aprendizaje cada día más popular gracias a su facilidad para mantener la atención y el interés del estudiante. Además, permite aprovechar pequeños ratos de estudio de forma eficiente. Una modalidad de microlearning es el sistema de Flashcards, donde el estudiante recibe contenidos en forma de cuestionarios, los cuales cambian en función de las respuestas previas del alumno. En este TFG se ha desarrollado un software que implementa el sistema de Flashcards, haciendo uso de la extendida plataforma Telegram, con el principal objetivo de mejorar el rendimiento de estudio de los alumnos. A diferencia de otros sistemas similares, la propuesta es totalmente gratuita, de fácil acceso y posibilita que un profesor controle y revise el uso que hacen del sistema sus alumnos. En lo que respecta a la estructura, el sistema está compuesto por un bot de Telegram y un servidor api rest, éste último conectado a una base de datos. El software permite realizar preguntas tipo test siguiendo la filosofía de las Flashcards y visualizar los resultados de los estudiantes, alcanzando los objetivos planteados inicialmente.

Palabras clave—bot, cuestionarios, e-learning, flashcards, Leitner, repetición espaciada, Telegram

Abstract—Microlearning is an increasingly popular learning method thanks to its ability to maintain student attention and interest. In addition, it allows you to take advantage of small study times efficiently. One modality of microlearning is the Flashcards system, where the student receives content in the form of questionnaires, which change based on the student's previous answers. In this TFG, a software has been developed that implements the Flashcards system, making use of the extended Telegram platform, with the main objective of improving the students' study performance. Unlike other similar systems, the proposal is completely free, easily accessible and allows a teacher to control and review the use made of the system by their students. Regarding the structure, the system is made up of a Telegram bot and a rest api server, the latter connected to a database. The software allows multiple choice questions to be asked following the philosophy of Flashcards and visualize the results of the students, achieving the objectives initially set.

Index Terms—bot, quizzes, e-learning, flashcards, Leitner, spaced repetition, Telegram

1 INTRODUCCIÓN

En los últimos años la educación online ha crecido ampliamente en todos los niveles educativos, tanto en las situaciones formales como informales del aprendizaje. Debido a la evolución de las tecnologías TIC, los sistemas e-learning están cogiendo cada vez más fuerza. Son muchas las causas que están impulsando el crecimiento de este tipo de formación tales como una mayor integración de las tecnologías digitales en la vida diaria, flexibilidad de acceso, la necesidad de personalizar los contenidos y recibir feedback de manera inmediata, etc. [1].

Además, la pandemia mundial provocada por la COVID-19 ha originado la necesidad de una alternativa al método de enseñanza tradicional; un sistema que facilite el aprendizaje sin el requisito de la presencialidad. Por estos motivos, ha aumentado la necesidad de crear sistemas menos estandarizados que permitan una mayor adaptabilidad y personalización de cara a los estudiantes. Una aproximación que está en auge son los sistemas de micro

aprendizaje.

1.1 Sistema de Micro Aprendizaje

Los sistemas de micro aprendizaje buscan impartir contenidos en fragmentos pequeños, de manera frecuente y activa. De esta manera se consigue que vaya adquiriendo conocimientos poco a poco, pero de forma más profunda y duradera.

En este proyecto se propone desarrollar un sistema software que facilite el uso de esta estrategia de aprendizaje en los estudios de grado. Con esto en mente se propone construir un software basado en la extendida plataforma de mensajería llamada Telegram. Mediante la implementación de un bot se busca diseñar y crear un sistema que facilite el estudio y la adquisición de conocimientos a los alumnos. Además, se pretende que el software sirva de ayuda al profesorado permitiéndole analizar el progreso de los alumnos, de manera que sea sencillo ajustar mejor los temas a tratar en clase.

El proyecto está centrado en la parte más práctica del aprendizaje: en la resolución de problemas, cuestiones y, en definitiva, la aplicación de los conocimientos adquiridos. Para potenciar la absorción de los contenidos se busca

- E-mail de contacte: Angel.Colombo@autonoma.cat
- Menció realitzada: *Enginyeria del Software*
- Treball tutoritzat per: Daniel Ponsa Mussarra (Departament de Ciències de la Computació)
- Curs 2021/22

personalizarlos para cada alumno, de manera que practique con más frecuencia los temas en los que tiene más dificultades. También es importante que puedan realizarse los ejercicios de forma rápida, de manera que los estudiantes puedan organizar su tiempo y aprovechar las sesiones cortas de estudio.

Una técnica pedagógica que se ajusta a estos principios es el sistema de Flashcards [3]. Este propone que los conceptos se revisen de forma periódica, espaciada, en pequeñas dosis y a lo largo del tiempo. Además, permite que las preguntas sean personalizadas para cada situación.

1.2 Sistema de Flashcards

El sistema de Flashcards (o sistema de Leitner) es un método pedagógico creado por el periodista científico alemán Sebastian Leitner en la década de 1970. El sistema está basado en el método de repetición espaciada, sugerido por el psicólogo alemán Hermann Ebbinghaus en 1885. En su estudio, Ebbinghaus exploró la memoria y el aprendizaje, concluyendo que la información nueva se olvida con el tiempo, pero que este proceso no ocurre de manera lineal: olvidamos las cosas más rápido inmediatamente después de entrar en contacto con ellas que después de un tiempo. Ebbinghaus representó esta idea de manera gráfica, cómo puede verse en la Fig. 1 [2].

A grandes rasgos, el sistema de Flashcards consiste en revisar los conceptos que menos conoces y dedicar menos tiempo a lo que ya recuerdas, todo durante un periodo largo de tiempo [3]. Con este propósito, se emplean tarjetas en las que, en una de las caras se realiza una pregunta y en la otra se escribe la respuesta. En función de si la pregunta se responde correctamente o no, el sistema debe clasificar las tarjetas en distintos grupos o cajas. Las cajas están enumeradas, y la idea es que las tarjetas de las cajas más bajas se repasen más a menudo y las de las cajas más altas con menos frecuencia [6].

El mecanismo de funcionamiento habitual es el siguiente. En la primera sesión de preguntas, se colocan todas las tarjetas en la primera caja. Como puede verse en la Fig. 2, las tarjetas que se respondan correctamente se mueven a la siguiente caja y las falladas se quedan en la caja 1. Para las siguientes sesiones, las preguntas falladas se moverán a la caja 1, y las acertadas avanzan a la siguiente caja [4], [5].

El número de cajas utilizadas puede ser variable, dependiendo de la complejidad del tema a estudiar. Además, existe una variante del sistema en la que, al fallar una pregunta, la tarjeta no vuelve a la caja 1 sino a la caja anterior. En la Fig. 3 puede verse un esquema de esta variante.

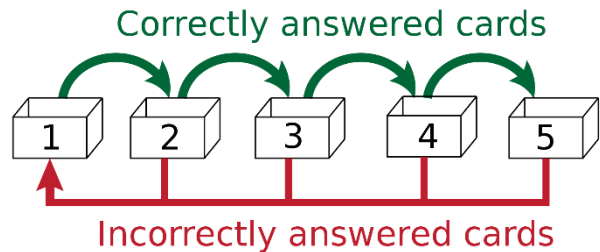


Fig. 2. Funcionamiento sistema de Flashcards [2].

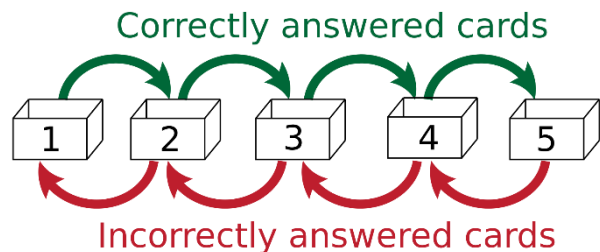


Fig. 3. Funcionamiento alternativo del sistema de Flashcards [2].

2 ESTADO DEL ARTE

Actualmente hay bastantes aplicaciones que hacen uso del sistema de Flashcards, pero muchas de ellas son demasiado simples y otras tienen demasiadas funcionalidades de pago, lo que dificulta su uso en las aulas. Entre las aplicaciones más interesantes y también bastante conocidas podemos encontrar Duolingo y Quizlet [16], [17]. Ambas tienen tanto versión web como smartphone, de forma que facilitan el acceso desde cualquier dispositivo. También tienen diferentes tipos de preguntas para que los ejercicios resulten más atractivos y entretenidos.

Duolingo está completamente enfocada al aprendizaje de idiomas, y tiene un gran abanico de cursos que los usuarios pueden realizar de manera individual. Mediante el sistema de Flashcards personaliza las preguntas de manera que puedas hacer los ejercicios que más te cuestan. La parte negativa es que un usuario no puede crear ningún tipo de curso y nadie puede ver tus progresos aparte de ti. Además, aunque es de acceso libre tiene funcionalidades de pago, número de intentos limitados por vidas, anuncios, etc.

Por otro lado, Quizlet permite a cualquier usuario crear sus propias unidades de estudio (cursos). Hace uso de tarjetas de doble cara donde se escriben la pregunta y la respuesta; en función de si aciertas o fallas te planteará unas preguntas u otras. Además, permite imágenes y diferentes tipos de pregunta (respuesta múltiple, texto, verdadero/falso, etc.). En la web también nos podemos encontrar con algunos minijuegos que hacen que responder a las preguntas sea más divertido. Tiene varias fun-



Fig. 1. La curva del olvido. Cada día consecutivo que repasamos un concepto la curva se aplanan, es decir, se nos hace más sencillo recordar [3].

ciones de pago y anuncios, pero en términos generales es muy completa. Como punto negativo, está limitado al uso individual, y el análisis de progreso que permite hacer es muy limitado.

A parte de estas dos aplicaciones, en el ámbito de la ingeniería informática encontramos el sistema UML quiz [7], una aplicación web muy útil para practicar algunos tipos de diagrama UML, pero tampoco tiene ningún tipo de registro sobre los cuestionarios realizados.

Por último, existen también algunos bots de Telegram que implementan el mecanismo de Flashcards. El más interesante es @memorizationbot [8], que permite crear fichas; en la parte frontal se escribe la pregunta y en la parte de atrás la respuesta. El bot está pensado para hacerte preguntas una vez al día. En función de tus errores te repetirá unas preguntas u otras. Aunque puede ser útil para uso propio es muy simple y limitado. No tiene ningún tipo de visualización de tu progreso ni ninguna función más allá de enseñarte las fichas que has creado.

Todas estas aplicaciones no están pensadas para analizar el progreso de un grupo de estudiantes, ni integran ningún concepto de clase o agrupación de estudiantes, por lo tanto, el análisis de resultados es muy limitado. Además, tienen muchas funcionalidades de pago que dificultan un uso generalizado de la aplicación.

En este proyecto se propone desarrollar un sistema de Flashcards que resuelva las limitaciones de los sistemas existentes. La propuesta es totalmente gratuita y de fácil acceso, puesto que está integrada en Telegram. Aunque la realización de cuestionarios se hace individualmente, la aplicación está orientada a utilizarse en un grupo de clase, de manera que el profesor pueda visualizar claramente el progreso de la clase y crear sus propios cuestionarios.

3 OBJETIVOS

Con la intención de que quedaran claras las necesidades más importantes que debía cubrir el sistema y el problema principal que debía resolver, se aplicó la metodología de análisis Logical Framework Approach [21]. En el apéndice A1 se muestran los árboles de causa-efecto y medio-fines generados. De este análisis se concluyó que los elementos más importantes para mejorar el rendimiento del estudio son:

- Dar feedback de manera constante e inmediata, tanto al alumno como al profesor.
- Realizar cuestionarios personalizados.
- Aprovechar los pequeños ratos de estudio.

Los ejercicios prácticos basados en cuestionarios ayudan a que cada alumno repase los conceptos en los que tiene más dificultades. También se busca utilizar esos momentos del día en los que el estudiante tiene poco tiempo y no lo aprovecha porque "no me da tiempo a hacer nada" o "no merece la pena". Por ejemplo, si un estudiante tiene que desplazarse a la universidad en tren, bus o similares, podría usar ese tiempo para repasar conceptos con ayuda del sistema de manera fácil, útil y práctica.

El objetivo principal del TFG es desarrollar un sistema

de Flashcards basado en Telegram para asignaturas de grado. Para alcanzarlo se deben cumplir los siguientes subobjetivos:

- Establecer un mecanismo para limitar el acceso a los contenidos a los alumnos matriculados en una asignatura.
- Establecer un mecanismo que permita tener distintos cuestionarios organizados en "temas".
- Permitir la realización de cuestionarios y problemas (siguiendo la filosofía de las Flashcards).
- Establecer un mecanismo que permita visualizar los resultados de los estudiantes.

4 METODOLOGÍA

Para la construcción correcta del software y garantizar que se alcanzan los objetivos se consideró la metodología Agile como la más adecuada. Puesto que los desarrollos software requieren de una fuerte relación con el cliente y entregas parciales del producto final para ver si se está satisfecho con la trayectoria que toma el proyecto, nos podíamos beneficiar mucho de prácticas ágiles tales como iteraciones cortas, atención a la excelencia técnica, participación del cliente y adaptabilidad a los cambios.

La metodología Agile es un término general que hace referencia a un conjunto de prácticas que basa su desarrollo en un ciclo iterativo e incremental. Estas prácticas se sustentan en el Manifiesto Agile y sus 12 principios [10].

Uno de los puntos más importantes es que todo cambio en los requisitos es bienvenido sin importar la fase del proyecto. Para conseguir adaptarse a los cambios, es necesaria la intervención del cliente en el proceso, aportando ideas y opinando sobre los resultados. Además de realizar entregas parciales del producto que mejoren las labores de seguimiento y control [11], [12].

En la Fig. 4 se muestran los pasos que se realizan en una iteración; entre los más importantes están el desarrollo de nuevas funcionalidades, integración de estas en el sistema y la entrega de resultados al cliente. En base al feedback se actúa en consecuencia y se salta a la siguiente iteración.

Puesto que el desarrollo del TFG es individual se optó por utilizar una adaptación del método Kanban [13] para

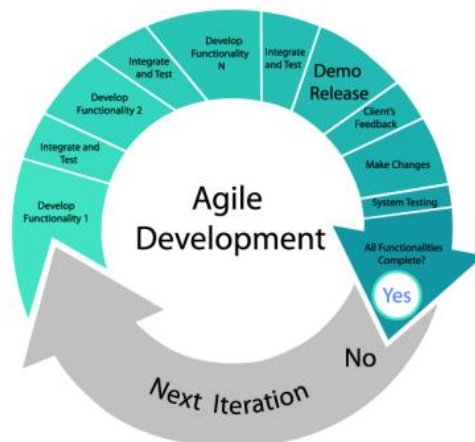


Fig. 4. Pasos que realizar en una iteración [12].

gestionar el proyecto software. Este procedimiento facilita la visualización del flujo de trabajo con ayuda de un tablero con tarjetas y columnas. También establece la duración de las tareas que se encuentran en proceso, descripciones precisas de cada una, etc. Para aplicar el método Kanban se creó un tablero en Trello, facilitando la organización de las tareas en distintas tarjetas.

En el proyecto que nos ocupa se planeó una reunión semanal con el tutor, que es un cliente potencial de la aplicación, que ayudó a redirigir el desarrollo en la dirección correcta en todo momento, proponiendo cambios cuando era necesario. Para realizar estas reuniones se utilizó la herramienta de Microsoft Teams.

Con el objetivo de facilitar la aplicación de la metodología se utilizó un repositorio de GitHub con el código del Bot [22]. De esta manera garantizamos que tenemos una copia del proyecto en la nube y podemos llevar un mejor control de las actualizaciones. Además, el repositorio facilita mostrar los resultados al cliente y volver a versiones anteriores en caso de necesidad.

Por último, para realizar la documentación se crearon documentos en Google Drive, garantizando así copias en la nube y comodidad para acceder desde cualquier dispositivo.

5 PLANIFICACIÓN

En este apartado se definen los pasos y actividades que se consideraron necesarios para la realización del proyecto. Esta planificación tenía el propósito de facilitar la ejecución y el control de las tareas, y alcanzar así los objetivos comentados anteriormente.

Con el objetivo de facilitar la ejecución y control de las tareas, y ajustarse a la metodología Kanban, se dividió el proyecto en varias iteraciones/sprints. Cada fase tiene una funcionalidad central y unas tareas asociadas que deben realizarse para alcanzar los objetivos. Esta organización de las tareas puede verse con detalle en la Tabla 1.

Para extraer datos que pudieran ayudar a la hora de planificar la faena se realizó un gráfico en el que se especificó el tiempo aproximado dedicado a cada grupo de tareas. Dentro de la implementación se incluyeron las actividades de diseño y testeo. En la Fig. 5 pudimos ver que el grueso de las tareas era de implementación y documentación. Por otro lado, debido a las entregas parciales de informes nos encontramos con un porcentaje importante de documentación. Finalmente, la obtención de requisitos ayudó a fijar los objetivos y prioridades del trabajo.

La planificación inicial del proyecto se muestra en el diagrama de Gantt que puede verse en la Fig.6 [14]. La primera fase de estudio preliminar no se incluyó en el diagrama ya que se orientó a documentarse y recabar información sobre todo lo relacionado con el proyecto. Por lo tanto, el desarrollo del bot empieza en la fase de realización de sesiones/cuestionarios.

Es importante remarcar que todas las tareas de implementación incluyen un procedimiento de pruebas para garantizar el buen funcionamiento del software. Asimismo, todas las tareas de documentación comprenden su

TABLA 1
FASES DEL PROYECTO, TAREAS Y TIEMPO APROXIMADO
(300 HORAS)

ESTUDIO PRELIMINAR (52 HORAS)
10h. Realizar estudio del problema
8h. Realizar estudio de viabilidad
14h. Realizar análisis de requisitos
4h. Escoger herramientas de desarrollo
16h. Redactar informe inicial
REALIZACIÓN DE SESIONES/CUESTIONARIOS (80 HORAS)
2h. Dar de alta bot
8h. Implementar funciones de prueba
16h. Realizar cuestionario (sin sistema de Flaschards)
8h. Consultar resultado cuestionario
30h. Realizar cuestionario (mediante sistema Flaschards)
16h. Redactar informe de Progreso I
GESTIÓN Y PERSONALIZACIÓN DE SESIONES (86 HORAS)
8h. Diseñar BD
14h. Implementar BD
22h. Implementar sistema de gestión de contenidos
26h. Gestionar acceso usuarios
16h. Redactar informe de Progreso II
ANÁLISIS DE RESULTADOS (82 HORAS)
28h. Consultar progreso alumnado
10h. Crear incidencia/comentario
6h. Consultar incidencia/comentario
10h. Implementar sistema de avisos
16h. Redactar informe final
8h. Preparar presentación
4h. Realizar poster

revisión y corrección.

En cuanto a los tres *sprint* que componen la planificación, el primero se centró en la funcionalidad principal del proyecto, el sistema de Flashcards. La idea es que se puedan realizar cuestionarios y que las preguntas que el software plantea sean en función de tus respuestas anteriores.

El segundo *sprint* se enfocó a la configuración de las sesiones y la integración con la base de datos. El profesor debe poder crear preguntas y cuestionarios por temas, entre otras opciones. Además, el sistema debe separar entre usuarios profesores y estudiantes.

Por último, tenemos el tercer *sprint*, que se dedicó a mejorar el sistema base ya construido, añadiendo funcionalidades extras. La idea de esta iteración era trabajar en nuevos casos de uso, en función del tiempo disponible, tales como consultar resultados de los estudiantes, enviar avisos, informar sobre incidencias, valorar la dificultad de un cuestionario, etc.

En la ejecución del trabajo planificado, se llegó a la conclusión de que se necesitaba más tiempo para realizar las tareas de realización de cuestionarios. La previsión inicial fue muy optimista y no tenía en cuenta el análisis de requisitos, el cual se ha alargado más de lo esperado. Además, la primera iteración abarcaba la implementación del núcleo del

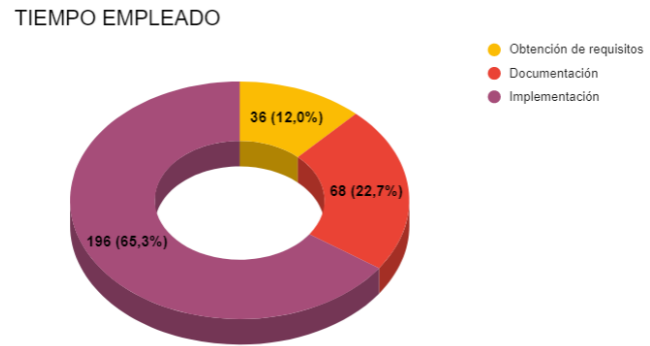


Fig. 5. Horas dedicadas a cada grupo de tareas.

proyecto, el sistema de Flashcards, por lo que era importante asegurarse de que funcionaba de la manera adecuada. También se necesitó más tiempo del previsto para redactar el informe de progreso I. Por estos motivos se decidió alargar un par de semanas la primera iteración y, por lo tanto, se redujo el alcance de la tercera iteración ya que se consideró la menos crítica para el cumplimiento de los objetivos. Concretamente, se descartaron las tareas de gestión de incidencias y el sistema de avisos del tercer sprint.

A inicios de diciembre el bot ya tenía algunas funcionalidades implementadas y testeadas, pero el sistema de Flashcards aún no estaba lo suficiente avanzado. Por esta fecha se decidió que era preferible desarrollar el sistema de Flashcards con la base de datos ya diseñada e implementada. Es por esto por lo que estas tareas se movieron a la primera iteración. Por lo tanto, se redujo el alcance de la tercera iteración, donde se planteaban funcionalidades no esenciales, para conseguir un sistema de Flashcards to-

talmente funcional y operativo.

6 RESULTADOS

Para facilitar la exposición de los resultados se ha optado por realizar la explicación por fases, igual que se ha hecho en la planificación. Es decir, primero se expondrán los resultados de la fase preliminar y después los resultados obtenidos en cada sprint.

6.1 Fase preliminar y diseño

En la fase preliminar, con el objetivo de conocer los posibles casos de uso del sistema y determinar la prioridad en su implementación se aplicó la técnica del Story Mapping [15]. Mediante esta técnica se estableció un MVP con distintos casos de uso: dar de alta alumnado y profesor, crear y modificar preguntas, crear cuestionarios, realizar test, consultar resultados, consultar progreso alumno, etc. Además, también sirvió para definir unas mejoras a realizar en el MVP, en caso de disponer del tiempo necesario, que incluyen, entre otros, la importación de alumnos y profesores mediante formularios u hojas Excel, importación de preguntas mediante un fichero de texto, visualización del progreso de forma gráfica.

Siguiendo con el análisis de requisitos se realizó un prototipo de la aplicación con sus respectivas pantallas [16], pueden verse algunas en el apéndice A3. De esta forma pudimos analizar cómo se comunica el bot con el usuario, en función de si este es estudiante o profesor. Además, el prototipo ayudó a identificar nuevos requisitos del sistema o refinar otros. Después de realizar esta tarea, se decidió que era más adecuado implementar la creación de cuestionarios y la visualización de progresos en una aplicación externa a Telegram, con el propósito de mejorar la interacción y potenciar la comodidad de uso. Esta aplicación no entra dentro de los objetivos del TFG a desarrollar.

Para mostrar cómo están contruidos el software y la base de datos se ha realizado el diagrama de clases que se encuentra en el apéndice A2. En el diseño se deja la puerta abierta a añadir nuevos tipos de preguntas, y modificarlo en caso de ser necesario para cumplir los requisitos del sistema.

Para ejemplificar de manera clara qué módulos componen el sistema y la comunicación entre ellos se realizó la Fig. 7. Se optó por una arquitectura modular, permitiendo la reutilización de las diferentes partes del sistema y evitando que algún cambio en alguno de los módulos provoque la necesidad de rehacer todo el sistema. Por lo tanto, se separó completamente el front-end, que contiene la implementación del bot de Telegram, del back-end, que incluye la construcción del sistema de Flashcards. De esta forma, si fuera necesario cambiar la base de datos no habría necesidad de modificar el bot y viceversa. Asimismo, sería posible tener sistemas alternativos trabajando sobre la misma base de datos, como por ejemplo una aplicación de smartphone o una página web en lugar de Telegram.

Por un lado, el front-end está implementado en Python 3.9.7 y utiliza la librería python-telegram-bot [19] que

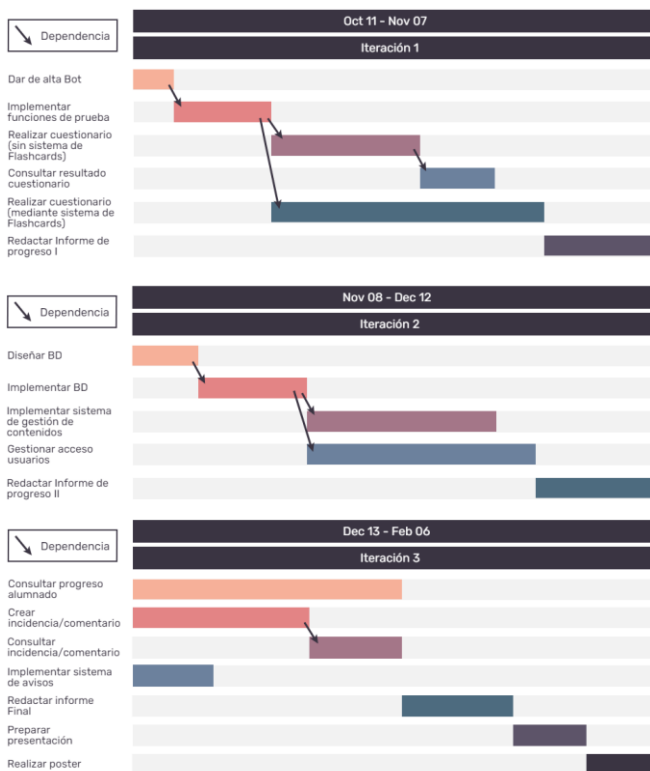


Fig. 6. Diagrama de Gantt. Planificación inicial.

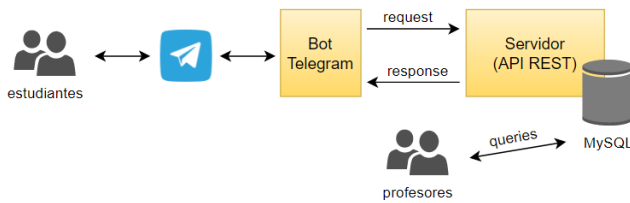


Fig. 7. Comunicación entre los módulos del sistema. Los estudiantes realizan cuestionarios comunicándose con el bot a través de Telegram y los profesores interactúan directamente con MySQL para dar de alta los contenidos.

proporciona una interfaz de Python para la API Telegram Bot API [20]. El bot interactúa con el estudiante a través de Telegram y este mismo bot hace peticiones a el servidor API REST para obtener los datos que necesita. Los datos se transportan en formato JSON, ya que es un formato eficiente y práctico.

Por otro lado, el back-end está formado por el servidor que responde a las peticiones del bot. Con este propósito se ha construido una API REST utilizando el framework Flask de Python. El servidor está conectado a una base de datos MySQL de donde extrae los cuestionarios, datos de usuario, etc. y responde a las peticiones del bot con estos datos.

Para desarrollar el sistema se decidió utilizar Python por su sencillez y popularidad, ya que esto facilita encontrar documentación detallada sobre distintas librerías y funciones. Además, es un lenguaje de código abierto y multidisciplinar, por lo que puede utilizarse para una gran variedad de aplicaciones diferentes como Big Data, IA, desarrollo web, etc. Esto permite mucho margen de maniobra en caso de querer ampliar las posibilidades del proyecto.

Para la base de datos se optó por el sistema MySQL por su facilidad de uso y su alto rendimiento. También son una ventaja su escalabilidad, su gestión de la memoria y sus distintas capas de seguridad.

6.2 Primer sprint

Antes de empezar con el desarrollo de las funcionalidades principales, se realizó un estudio de viabilidad para estudiar cómo afecta al comportamiento del bot el acceso simultáneo de varios usuarios [22]. Este factor de concurrencia es crítico para el desarrollo del TFG, puesto que el sistema está orientado a utilizarse en distintos grupos de clase y con varios estudiantes de manera simultánea. Gracias a esta prueba se pudieron identificar posibles problemas y a solucionarlos antes de que el software estuviera demasiado avanzado.

En cuanto al desarrollo, se consiguió un bot capaz de identificar al estudiante y que permitía realizar cuestionarios sobre diversos temas y asignaturas. Estas preguntas se habían dado de alta anteriormente mediante consultas SQL, de esta manera el sistema era capaz de indicar si la pregunta se había respondido correctamente o no, consiguiendo así dar feedback al estudiante. No obstante, las respuestas del alumno todavía no se registraban en la base de datos, con lo que el sistema de Leitner aún no estaba operativo. Para implementarlo, era necesario emu-

lar el mecanismo de guardar las preguntas en distintas cajas, por lo tanto, la tarea se realizó en el siguiente sprint, una vez que la base de datos ya estuviera diseñada en su totalidad.

6.3 Segundo sprint

En el segundo *sprint*, se realizó una reunión con algunos compañeros de la mención de ingeniería del Software junto con algunos profesores. Se llevó a cabo con el objetivo de debatir sobre la estructura y el diseño de la base de datos y entre todos construimos el diagrama de clases. A continuación, escribí un script para crear las tablas de la base de datos y dar de alta algunos datos para poder consultarlos desde el servidor.

Acto seguido, implementé el sistema de la gestión de acceso a usuarios. Este sistema consiste en comprobar si el NIU introducido por el estudiante existe en la base de datos, y en caso afirmativo, se le pide un código de acceso. Si todos los datos son correctos, se guarda el identificador de chat del usuario (*chat_id*) para poder identificar siempre al estudiante de forma automática, sin necesidad de iniciar sesión. Como es lógico, el código de acceso se guarda de manera cifrada en la base de datos.

Además, se adaptó el código realizado en el sprint anterior para registrar los resultados de los cuestionarios en la base de datos. De esta forma, ya estaba todo preparado para la implementación del sistema de Flashcards. Para construir este mecanismo se decidió utilizar tres cajas porque es la forma más simple de comprobar que el sistema funciona. Por otro lado, el bot selecciona de qué caja coger la pregunta en base a los siguientes porcentajes: primera caja 65%, segunda caja 25% y tercera caja 10%. De todas formas, el código está adaptado para que sea sencillo cambiar el número de cajas y las probabilidades.

6.4 Tercer sprint

En el tercer *sprint*, se añadió una función para que los estudiantes pudieran ver los resultados de los cuestionarios que habían realizado anteriormente. Además, a la hora de realizar un cuestionario se pregunta al estudiante el número máximo de preguntas que quiere realizar, así el software se adapta en base al tiempo del que dispone el alumno.

Aunque a lo largo de todo el proyecto se han ido realizando pequeñas pruebas, en este sprint se ha ampliado en gran medida las funciones de prueba que hacen peticiones al servidor.

Además, se ha realizado exploratory testing [23] para experimentar con todo el sistema. En base a los resultados de este método se han realizado pequeños cambios para mejorar la interactividad con el bot y algunos errores en escenarios concretos.

Finalmente, se realizó una prueba con algunos usuarios finales con el objetivo de analizar si la interacción con el bot es satisfactoria. En esta prueba se pidió a los voluntarios realizar unas tareas dentro del sistema, sin darles ningún tipo de manual ni formación previa. De esta forma, con ayuda de un formulario de Google, pudimos ver si habían conseguido realizar las tareas o si por el contrario habían tenido algún problema o inconveniente. Ade-

más, en el formulario se dejaron espacios de texto para que las personas que estaban probando el bot pudieran sugerir ideas para mejorar el sistema.

En lo que respecta a los resultados, todos los voluntarios consiguieron llevar a cabo lo que se les pidió sin problemas. Algunos de los comentarios más recurrentes están relacionados con el uso de los comandos, cómo por ejemplo que los voluntarios no recordaban los comandos utilizables o no los escribieron bien y el bot no mostró ningún mensaje de ayuda. También ha habido confusión con el valor máximo que pide el programa en el momento de plantear cuestionarios, puesto que si le pides un número mayor de preguntas de las que hay en la base de datos no te informa de ello y simplemente te plantea todas las que hay guardadas. En menor medida hemos recogido feedback sobre la visualización de los resultados de los cuestionarios.

6.5 Sistema final

En el sistema final, el estudiante y el bot interactúan mediante comandos, un mecanismo habitual en Telegram, los cuales llaman a funciones internas del programa.

El primer paso para empezar a utilizar el sistema de Flashcards consiste en buscar al bot en la aplicación, por el nombre de usuario “@FlashcardsSystemBot”. Seguidamente, el estudiante utiliza el comando “/start” para registrarse en el sistema. En la Fig. 8 puede verse cómo es esta comunicación.

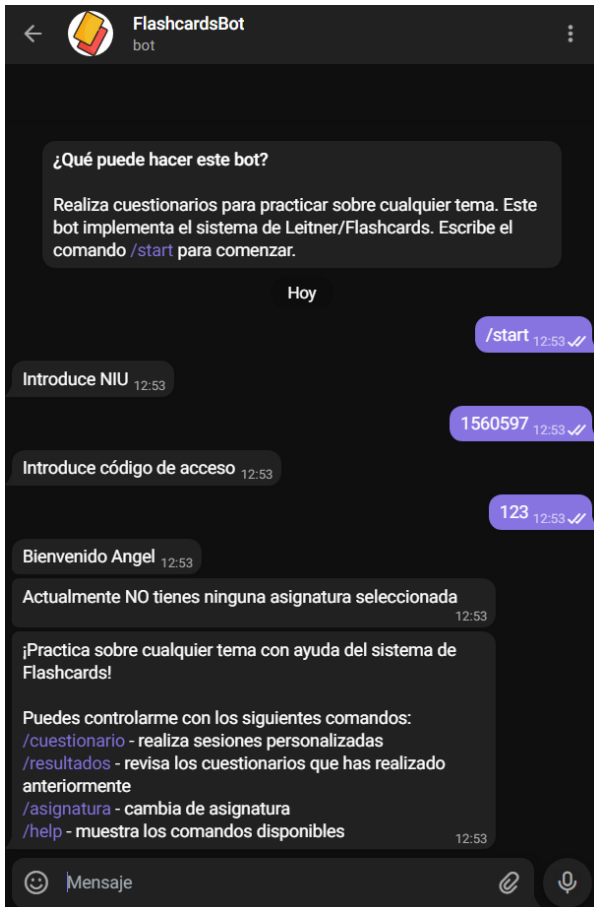


Fig. 8. Pantalla inicial + Registro estudiante

Una vez registrado en el sistema, el estudiante puede cambiar de asignatura y realizar cuestionarios para interiorizar conceptos. En la Fig. 9 se selecciona la asignatura “Inglés Profesional I” y se realiza un cuestionario sobre el tema “Irregular Verbs”. El bot plantea las preguntas al alumno en función de sus respuestas anteriores, siguiendo la filosofía de las Flashcards.

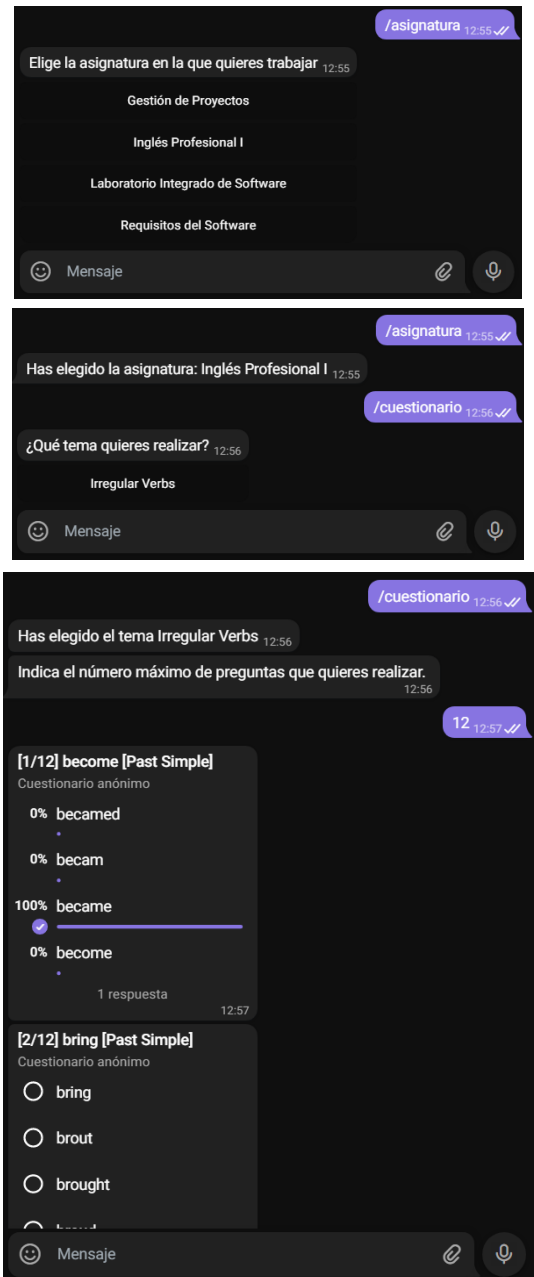


Fig. 9. Cambiar asignatura + Realizar cuestionario

Siguiendo con los distintos casos de uso del sistema, el estudiante puede consultar los resultados y respuestas de sus cuestionarios anteriores. En la Fig. 10 se muestra un ejemplo mediante el comando “/resultados”.

En lo que respecta al profesor, interactúa directamente con la base de datos para dar de alta los contenidos, puesto que el núcleo del TFG es Telegram. Una manera fácil de dar de alta elementos en la base de datos es mediante

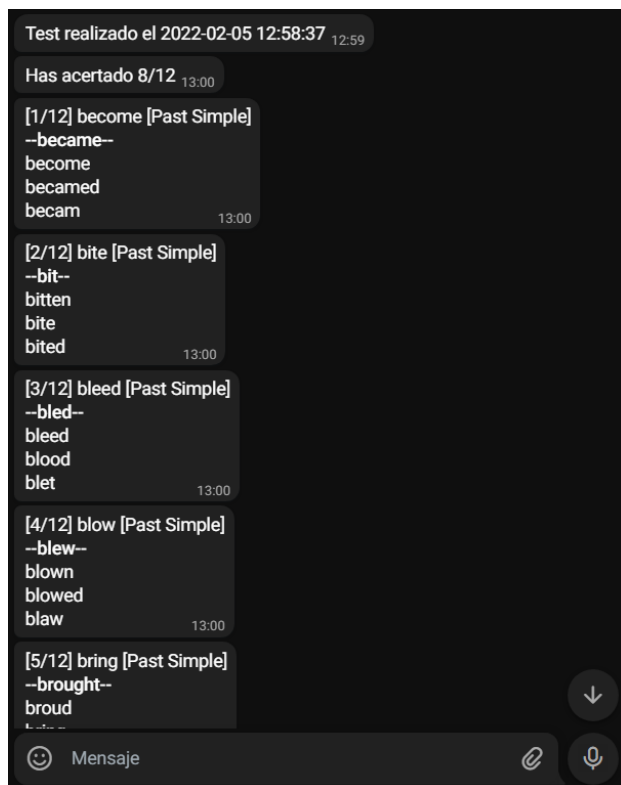
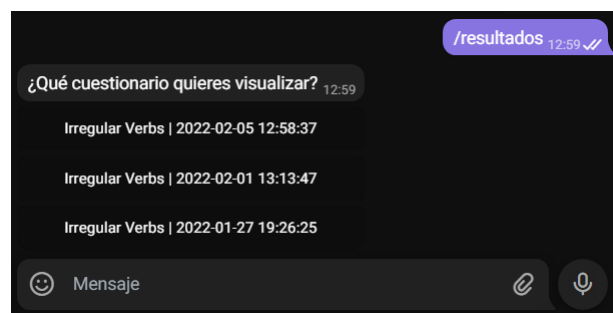


Fig. 10. Visualizar resultados de la asignatura "Inglés Profesional I"

scripts SQL. En mi caso, he utilizado la extensión MYSQL de la herramienta de desarrollo Visual Studio Code que proporciona una interfaz sencilla para modificar las tablas, vistas y otros muchos elementos.

7 CONCLUSIÓN

En este TFG se ha desarrollado un sistema que implementa el mecanismo de Flashcards, basado en la aplicación de Telegram, como soporte a la docencia en asignaturas de grado.

El sistema generado satisface todos los objetivos perseguidos: el software incluye un mecanismo que habilita el acceso únicamente a los estudiantes matriculados, tiene la capacidad para mantener el temario organizado, es capaz de plantear sesiones de preguntas test siguiendo la filosofía de las Flashcards y también permite visualizar los resultados de los estudiantes de forma simple.

Un punto que destacar del sistema desarrollado es que solo requiere que el usuario instale Telegram. Esto hace que el sistema sea accesible desde un smartphone o cual-

quier ordenador. El sistema de conversación textual es intuitivo, y los usuarios interaccionan correctamente con el sistema sin la necesidad de formación o manuales. Por otro lado, el hecho de que los mensajes intercambiados con el bot se vayan acumulando en la pantalla puede llegar a ser contraproducente si el estudiante lo utiliza para comprobar cuáles son las respuestas del cuestionario que está haciendo en ese momento.

Finalmente, como posibles extensiones del trabajo sería interesante ampliar el tipo de preguntas que puede plantear el bot, y no limitarse solamente a cuestionarios de respuesta simple. Por ejemplo, se podrían añadir preguntas que esperen una respuesta escrita o que los enunciados puedan contener imágenes, etc. También es buena idea implementar un sistema para que los estudiantes puedan comunicar incidencias, si una pregunta no se entiende o si hay algún error, de manera que los profesores puedan recibir feedback sobre las sesiones. Además, se podría hacer un sistema que facilitara introducir los contenidos en la base de datos. Por último, se podrían desarrollar representaciones gráficas del progreso de los alumnos, así como aplicar técnicas avanzadas para analizar los datos recogidos.

AGRADECIMIENTOS

En primer lugar, quiero agradecer a mi tutor Daniel Ponsa, quien con sus conocimientos y apoyo me ha guiado a través de las distintas etapas del proyecto. También agradecerle por brindarme los recursos y herramientas para facilitarme el proceso de investigación y desarrollo.

En segundo lugar, quiero agradecer a mis compañeros de carrera Adri y José Miguel por su ayuda en el proyecto, ya que siempre han estado ahí con algunas ideas y dispuestos a compartir útiles recursos para sacar el proyecto adelante.

Por último, quiero agradecer a todos los profesores que han participado en las reuniones aportando todo tipo de ideas sobre el diseño y el desarrollo del proyecto.

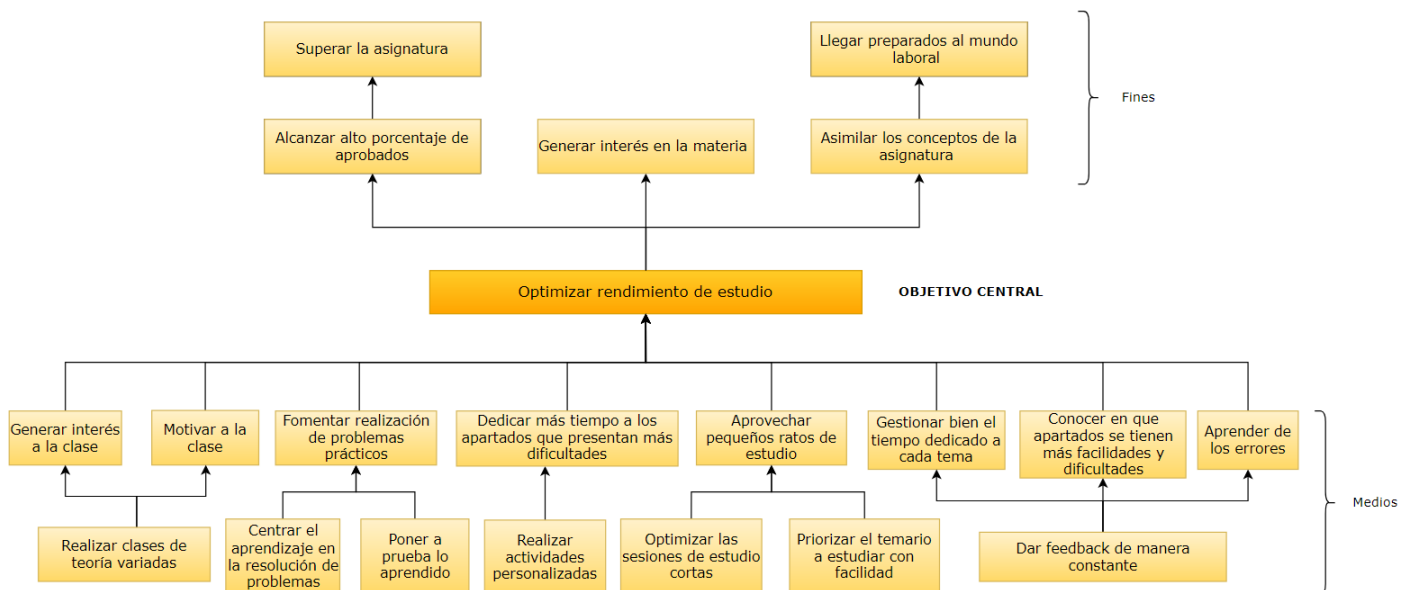
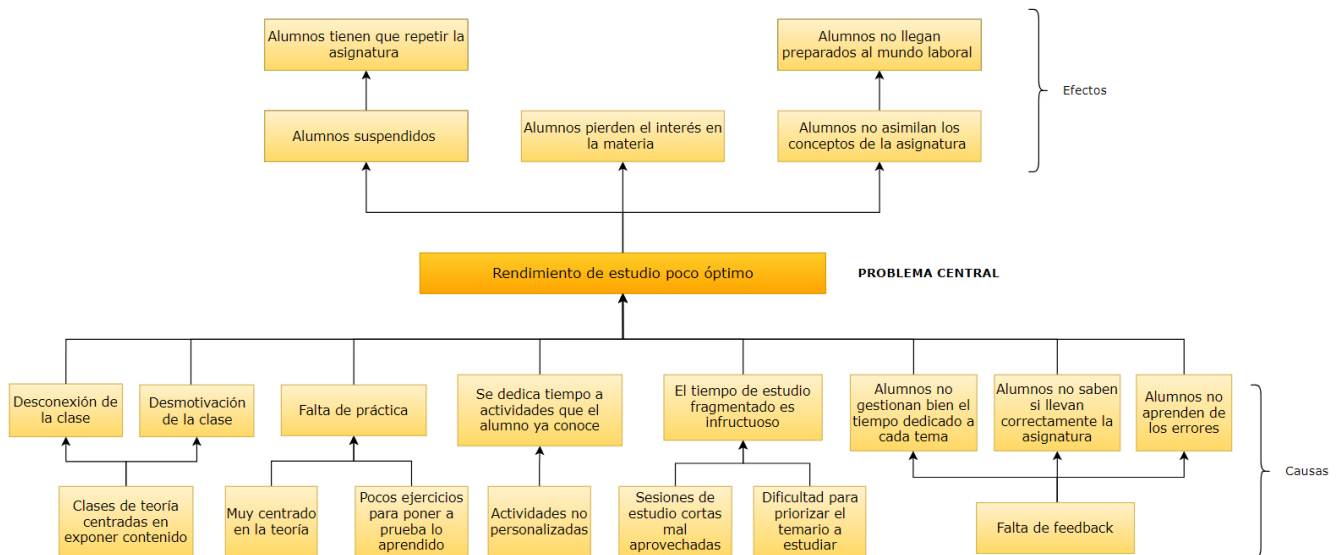
BIBLIOGRAFÍA

- [1] B. G. Salvat, "La evolución del e-learning: del aula virtual a la red," RIED. Revista Iberoamericana de Educación a Distancia, 21(2), pp. 69-82, disponible en <https://www.redalyc.org/jatsRepo/3314/331455826005/331455826005.pdf>, 2018. Fecha último acceso 01/02/2022.
- [2] A. Morris, "El sistema de repetición espaciada en el aprendizaje de idiomas," 2019. Fecha último acceso 01/02/2022. <https://blog.chatterbug.com/es/el-sistema-de-repeticion-espaciada-en-el-aprendizaje-de-idomas/>
- [3] "Leitner system," 2021. Fecha último acceso 01/02/2022. https://en.wikipedia.org/wiki/Leitner_system
- [4] "Sistema Leitner," 2016. Fecha último acceso 01/02/2022. <https://www.mnemotecnia.es/la-pastilla-verde-sistema-leitner>
- [5] S. Leitner, Así se aprende. Barcelona: Harder, 1973
- [6] A. De Luca, "Sistema Leitner para estudiar y memorizar tarjetas," 2019. Fecha último acceso 01/02/2022. <https://www.mentesliberadas.com/2019/08/14/sistema-leitner-estudiar-tarjetas/>
- [7] Business Informatics Group, Vienna University of Technology.

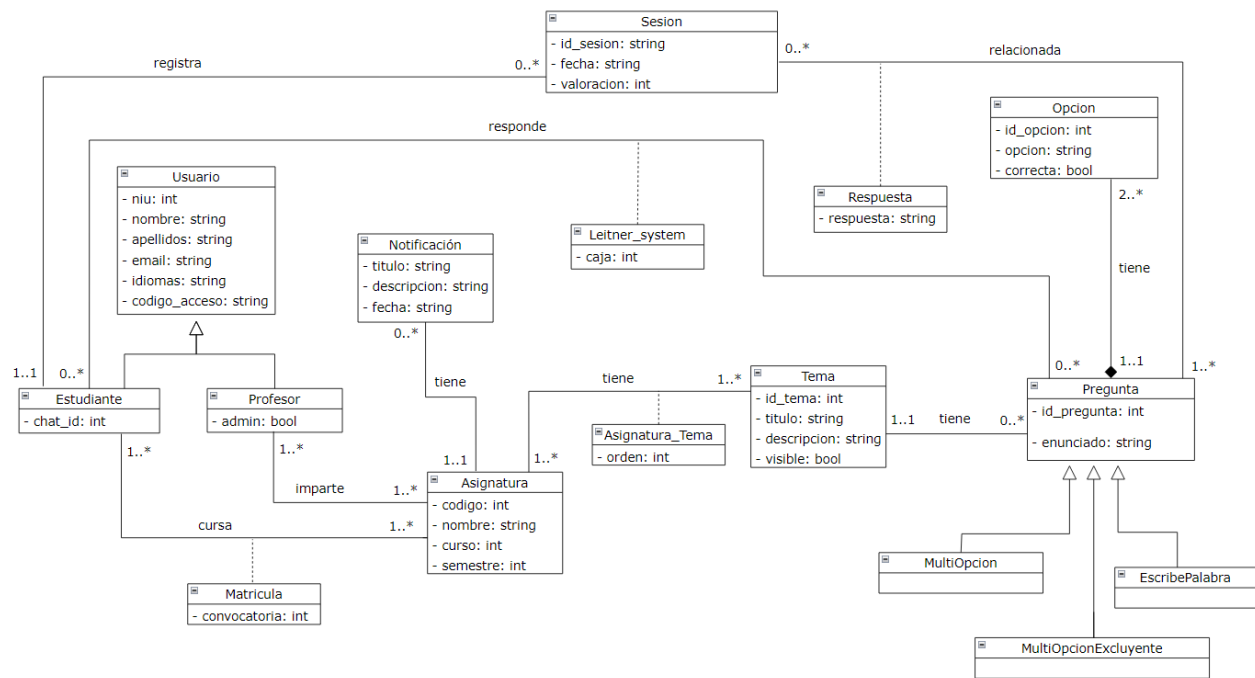
- "UML Quiz," 2021. Fecha último acceso 01/02/2022. <http://elearning.uml.ac.at>
- [8] B. van der Bijl, "Memorization Bot," Fecha último acceso 01/02/2022. <https://memorizationbot.com>
- [9] D.F. Betancourt, "Cómo hacer un árbol de objetivos: Ejemplo práctico," 2016. Fecha último acceso 01/02/2022. <https://www.ingenioempresa.com/arbol-de-objetivos/>
- [10] A. Perez, "Metodología agile: ¿Cuáles son los 12 principios de su modelo?," 2016. Fecha último acceso 01/02/2022. <https://obsbusiness.school/es/blog-project-management/metodologias-agiles/metodologia-agile-cuales-son-los-12-principios-de-su-modelo>
- [11] "Agile 101," 2021. Fecha último acceso 01/02/2022. <https://www.agilealliance.org/agile101/>
- [12] D.A.G. Hinojosa, G.D.T. Taboada. "Agile Software Development," 2016. Fecha último acceso 01/02/2022. <https://kencourses.com/tc1019fall2016/syndicated/agile-software-development-2/>
- [13] M. Burrows, Kanban from the Inside. Blue Hole Press, 2014
- [14] "¿Qué es y para qué sirve un diagrama de Gantt?," 2021. Fecha último acceso 01/02/2022. <https://www.teamleader.es/blog/diagrama-de-gantt>
- [15] "Story Mapping Flashcards," 2021. Fecha último acceso 01/02/2022. <https://microcontenttelegram.invisionapp.com/freehand/StoryMap-FlashCards-i3zqxemav>
- [16] "Prototipo Flashcards," 2021. Fecha último acceso 01/02/2022. <https://microcontenttelegram.invisionapp.com/freehand/Prototipo-Flashcards-Pe7EwLWIG>
- [17] "Duolingo," 2021. Fecha último acceso 01/02/2022. <https://www.duolingo.com>
- [18] "Quizlet," 2021. Fecha último acceso 01/02/2022. <https://quizlet.com/es>
- [19] "Welcome to Python Telegram Bot's documentation!," 2021. Fecha último acceso 01/02/2022. <https://python-telegram-bot.readthedocs.io/en/stable/index.html>
- [20] "Telegram Bot API," 2021. Fecha último acceso 01/02/2022. <https://core.telegram.org/bots/api>
- [21] European Integration Office. Guide to The Logical Framework Approach. Rep. of Serbia Gov. 2011
- [22] "GitHub, BotTelegram-Flashcards", 2022. Fecha último acceso 01/02/2022. <https://github.com/microcontents/BotTelegram-Flashcards>
- [23] D. Parmar, "Exploratory testing", 2022. Fecha último acceso 01/02/2022. <https://www.atlassian.com/continuous-delivery/software-testing/exploratory-testing>

APÉNDICE

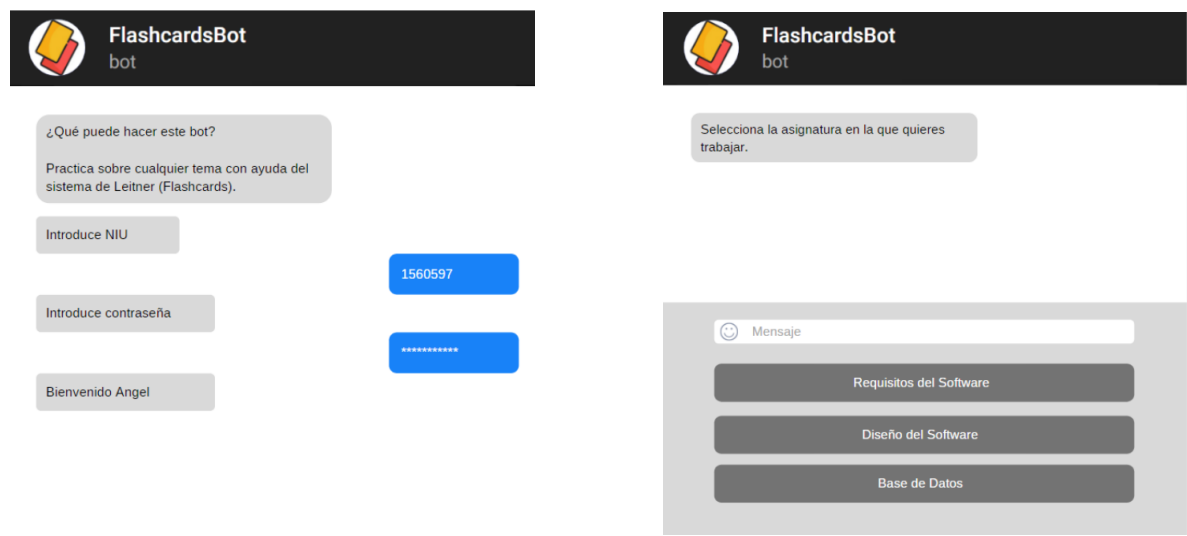
A1. ÁRBOL DE EFECTOS Y CAUSAS / ÁRBOL DE MEDIOS Y FINES



A2. DIAGRAMA DE CLASES

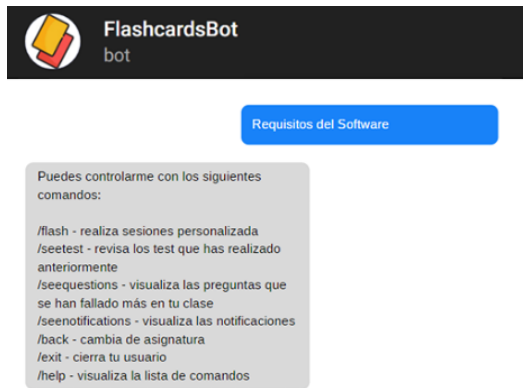


A3. PROTOTIPO

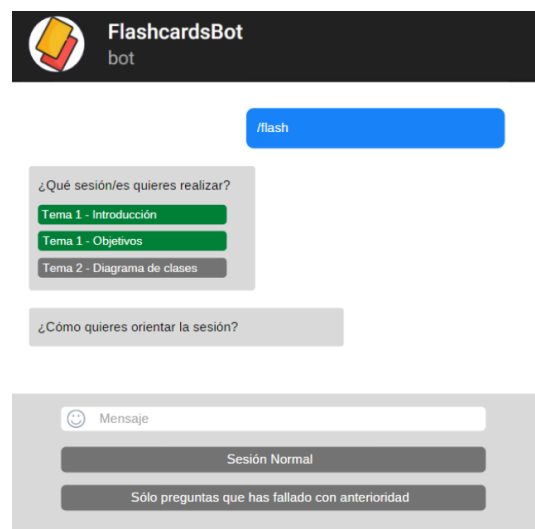


1. Registrar estudiante

2. Seleccionar asignatura



3. Mensaje de ayuda



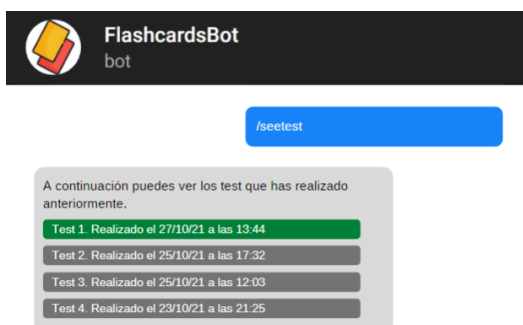
4. Realizar cuestionario (Parte 1)



5. Realizar cuestionario (Parte 2)



6. Realizar cuestionario (Parte 3)



7. Ver resultados (Parte 1)



8. Ver resultados (Parte 2)