Universidad de Cádiz

Vicerrectorado de Profesorado

Unidad de Formación e Innovación Docente

Hospital Real
Plaza Falla, 8 | 11003 Cádiz
Tel. 956 015 728
http://www.uca.es
innovacion.docente@uca.es

MEMORIA FINAL INNOVA[1]
COMPROMISOS Y RESULTADOS PROYECTOS DE INNOVACIÓN Y MEJORA DOCENTE
2024/2025

Identificación del proyecto	
Código	**sol-202400284957-tra**
Título	**Integración de NLP y Tecnología Móvil en la Educación para la Autoevaluación Efectiva**
Responsable	**Sara Balderas-Díaz**

1. Describa los resultados obtenidos a la luz de los objetivos y compromisos que adquirió en la solicitud de su proyecto. Incluya tantas tablas como objetivos contempló.

Objetivo nº 1	***Desarrollar una interfaz interactiva en Telegram***
Actividades previstas en la solicitud del proyecto:	*Este objetivo comprende las siguientes tareas:* ● *Diseñar la interfaz de usuario del bot de Telegram que sea intuitiva y fácil de usar para los estudiantes.* ● *Implementar las funcionalidades básicas del bot para interactuar, como menús interactivos y comandos para acceder a diferentes secciones.*
Resultados obtenidos:	*Se implementó un bot de Telegram con una interfaz sencilla e intuitiva basada en comandos (/submit, /test, /review, /report) y teclados en línea. Cada comando activa un flujo orquestado por el Telegram Bot Service y, más allá del texto libre, la interacción se apoya en botones que generan eventos callback_query con una carga codificada (p. ej., ans:Q123:2, topic:P2P) que el router del bot interpreta para continuar el diálogo (véase Figura 1). Esta combinación de comandos mínimos y botones contextuales ofreció una UI intuitiva y consistente durante las pruebas.*

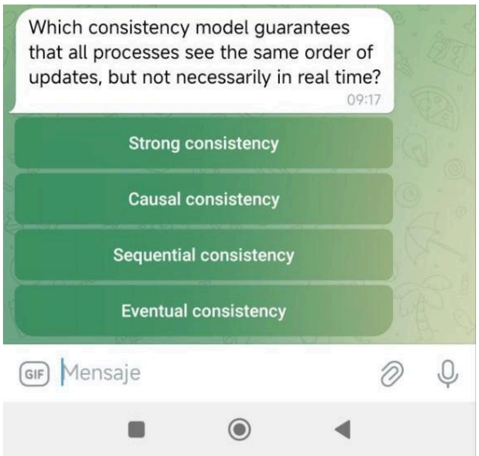


Figura 1. Ejemplo de interacción con el teclado integrado de Telegram donde el usuario responde a una pregunta de opción múltiple y cada pulsación genera un update con una carga codificada que indica la opción seleccionada.

[1] Esta memoria no debe superar las 6 páginas.

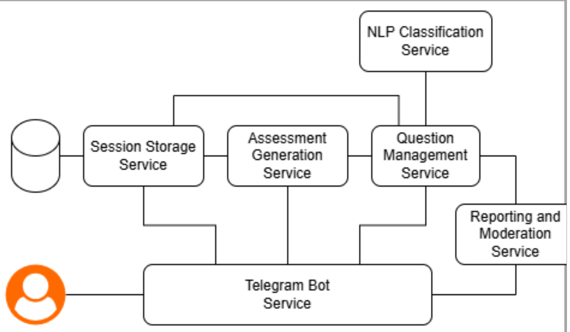
<table>
<tr><th>Objetivo nº 2</th><th>Implementar un sistema de generación de cuestionarios</th></tr>
<tr><td>Actividades previstas en la solicitud del proyecto:</td><td>Se prevé, llevar a cabo las siguientes actividades:
● Programar un sistema que permita a los estudiantes seleccionar el tema/temática y el nivel de dificultad de los cuestionarios.
● Desarrollar algoritmos que generen cuestionarios basados en los parámetros seleccionados por los usuarios.</td></tr>
<tr><td></td><td>La plataforma se organizó como microservicios que se comunican por APIs. Partiendo del punto de entrada ya descrito en el Objetivo 1, el bot encamina cada interacción al servicio adecuado (véase Figura 2). El Session Storage Service guarda el contexto por estudiante (preguntas ya hechas, preferencias, etc.)


Figura 2. Arquitectura del sistema.</td></tr>
<tr><td>Resultados obtenidos:</td><td>con una caducidad de 12 h, lo que permite retomar sesiones y evitar repeticiones. El Question Management Service valida y almacena los ítems junto con sus metadatos en una base de datos. La etiquetación por temática se hace con un módulo de Procesamiento en Lenguaje Natural (Natural Language Processing, NLP) que clasifica automáticamente las preguntas pero cuando alcanza un nivel de incertidumbre, las manda a revisión en lugar de asignarlas a ciegas.
El Assessment Generation Service construye cada test bajo demanda: toma preguntas del tema elegido, solo usa las que superan un umbral de calidad, excluye las ya vistas por el estudiante y evita duplicados para que el cuestionario sea variado; por defecto muestra 10 preguntas con opciones barajadas y entrega un resumen final de aciertos y errores. La calidad se mantiene con el servicio de reportes y moderación: el alumnado puede “marcar” una pregunta indicando si está mal, confusa, fuera de tema o inapropiada (en dos palabras: aviso rápido). El sistema ordena esos avisos por impacto y permite corregir, reclasificar o retirar ítems; esa señal ajusta el score de calidad que el generador usa para filtrar preguntas. Con este engranaje, durante el despliegue se construyeron 1.709 cuestionarios sin incidencias reseñables, mostrando robustez y escalabilidad.</td></tr>
</table>

Objetivo nº 3	***Crear un banco de preguntas colaborativo***
Actividades previstas en la solicitud del proyecto:	*Se desempeñarán las siguientes tareas:* ● *Implementar una base de datos segura y escalable para almacenar las preguntas creadas tanto por estudiantes como profesores.* ● *Desarrollar una interfaz en el bot para que los estudiantes puedan fácilmente añadir preguntas para sus cuestionarios.*
Resultados obtenidos:	*Se consolidó un banco de preguntas gestionado por el Question Management Service sobre PostgreSQL, que cubre todo el ciclo de vida del ítem: valida el envío (comprobaciones de esquema y semántica como número de opciones, rangos de índices, unicidad y longitudes), almacena de forma persistente con un identificador único y ofrece lectura/edición mediante endpoints que permiten recuperar por temática y acceder a metadatos, entre otros. Además, cada pregunta recibe etiquetas temáticas automáticas (asignadas por el clasificador) y*

	un indicador dinámico de calidad calculado con señales de uso y feedback; las preguntas con baja calidad se excluyen de futuras autoevaluaciones o se marcan para moderación. Durante las pruebas, se crearon 519 preguntas tipo MCQ, automáticamente categorizadas, con la siguiente distribución por temas: T1 - 140 preguntas, T2 - 112 preguntas, T3 - 112 preguntas, T4 - 86 preguntas y T5 - 69 preguntas.

Objetivo nº 4	***Integrar tecnología de procesamiento de lenguaje natural para la clasificación de preguntas***
Actividades previstas en la solicitud del proyecto:	*Se desempeñarán las siguientes tareas:* ● *Implementar tecnologías NLP para analizar y clasificar preguntas automáticamente según la temática.* ● *Configurar el sistema para que actualice automáticamente las categorías de preguntas basadas en el contenido.*
Resultados obtenidos:	*Para evitar el etiquetado manual, cada pregunta nueva se procesa de forma automática. Primero se limpia el texto para eliminar ruido y unificar formato. Después el enunciado se convierte en un vector numérico que captura su significado utilizando Sentence BERT en la variante all MiniLM L6 v2. Con ese vector, un modelo de regresión logística decide a qué tema pertenece entre varias opciones posibles. El modelo devuelve también una medida de confianza entre 0 y 1. Si esa confianza es menor que 0,65, el sistema no fija el tema y envía la pregunta a revisión con la etiqueta Uncertain. De este modo se automatiza aquello en lo que no existe incertidumbre y se reserva para el profesorado solo lo que genera dudas, lo que mantiene la coherencia temática y reduce la carga manual.*

Objetivo nº 5	***Mejorar la calidad del contenido educativo mediante la colaboración y el feedback***
Actividades previstas en la solicitud del proyecto:	Mejorar la calidad del contenido educativo mediante la colaboración y el *feedback:* *Se desempeñarán las siguientes tareas:* ● *Desarrollar funcionalidades que permitan a los estudiantes dar feedback sobre las preguntas de los cuestionarios.* ● *Implementar mecanismos para corregir errores y problemas de redacción en las preguntas propuestas por los estudiantes.*
Resultados obtenidos:	*Se habilitó el reporte de incidencias mediante el comando /report con una taxonomía controlada de motivos: respuesta incorrecta; redacción ambigua; fuera de tema; y contenido inapropiado. Cada pregunta mantiene una puntuación de calidad dinámica que combina frecuencia y severidad de reportes junto con el rendimiento del estudiantado. Cuando la calidad cae por debajo del umbral, la pregunta se excluye de nuevas pruebas hasta su corrección. El equipo docente dispone de una lista priorizada para editar, corregir o reclasificar con rapidez. El efecto neto fue una reducción de ruido y una mejora sostenida del banco de preguntas.*

Objetivo nº 6	***Realizar una prueba piloto de la plataforma***
Actividades previstas en la	*Se desempeñarán las siguientes tareas:* ● *Seleccionar un grupo de estudiantes de Sistemas Distribuidos para*

solicitud del proyecto:	*participar en la prueba piloto.* ● *Desarrollar un protocolo de prueba que incluya diferentes escenarios de uso de la plataforma para evaluar su rendimiento y experiencia de usuario.* ● *Implementar una prueba piloto para recoger datos reales de uso.* ● *Recopilar y analizar los feedbacks de los estudiantes y profesores participantes para identificar problemas, áreas de mejora y aspectos de interés de la plataforma.*
Resultados obtenidos:	*Para validar utilidad y escalabilidad, se llevó a cabo un piloto con carga controlada en el que se emplearon agentes automatizados que reproducían el patrón de comportamiento (elección de temática, tiempos de realización y proporción de aciertos/errores) de los estudiantes de la asignatura de Sistemas Distribuidos. Con este enfoque se observaron preferencias de temática concentradas en los bloques iniciales (T1 y T2) y, en menor medida, T3–T5, lo que ayudó a dimensionar servicios y colas internas. En total, la plataforma sostuvo 1.709 ejecuciones de cuestionarios sobre un repositorio de 519 preguntas, sin degradación apreciable del rendimiento, activando de forma coordinada los módulos de generación, gestión de preguntas, clasificación, sesiones y moderación.* *Además, se ejecutó una batería de pruebas end-to-end para asegurar el correcto funcionamiento: verificación de no repetición por historial de sesión (TTL 12 h), barajado de opciones y cómputo de resultados; aplicación del umbral de calidad (≥ 0,75) en la selección; gestión de casos con baja confianza en la clasificación (derivación a revisión); y circuito de reportes y moderación que ajusta el quality score y filtra preguntas en el generador. No se detectaron errores críticos, lo que confirma la robustez del orquestado de microservicios y la preparación del sistema para su uso con estudiantes.*

2. Realice una breve valoración sobre la influencia del proyecto ejecutado en la evolución de las asignaturas implicadas.

Análisis del impacto de la innovación en las asignaturas relacionadas con el proyecto
El proyecto pone a disposición un ecosistema complementario al aula: preguntas creadas por estudiantes, generación automática de cuestionarios bajo demanda y un canal móvil y familiar como Telegram. Este formato, ligero y ubicuo, permite practicar en cualquier momento y lugar, con retroalimentación inmediata y uso de preguntas etiquetadas por temática. En la práctica, favorece repasos breves desde el teléfono integrables en la rutina de estudio, mientras los mecanismos de calidad y moderación preservan la coherencia del repositorio. En cuanto a rendimiento, los análisis por intentos muestran tendencias ascendentes al repetir cuestionarios (véanse la Figura 3) donde las medianas, de cuestionarios generados con el tema 1 y tema 5 (seleccionados como muestra de ejemplo) suben con los reintentos y las medias por tema se sitúan en 7,70 (T1) y 7,71 (T5) respectivamente. Estos resultados muestran que aun con diferencias por complejidad del contenido y calidad de los ítems, la práctica iterativa con feedback automático tiende a mejorar el desempeño. En conjunto, la disponibilidad de un sistema móvil de autoevaluación con contenido generado por estudiantes favorece la práctica de conceptos trabajados en clase y la asimilación progresiva de los mismos, y ofrece señales útiles para orientar la docencia posterior.

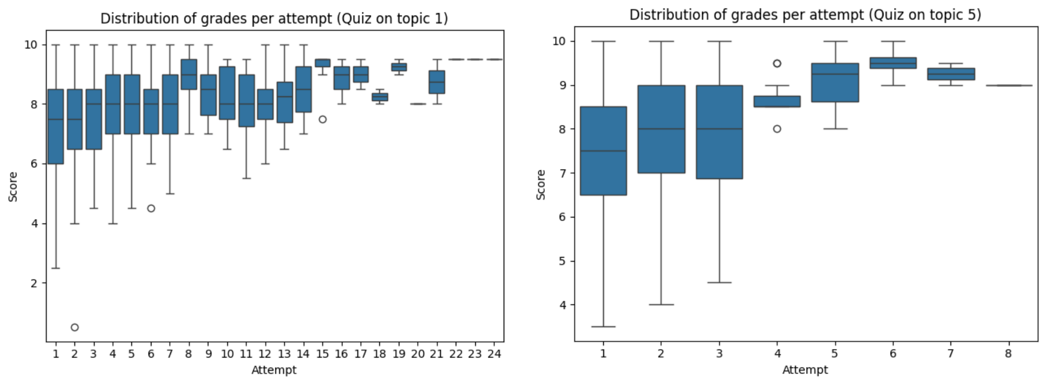


Figura 3. Calificaciones obtenidas y progresión en el aprendizaje en cuestionarios basados en temas.

3. Incluya en la siguiente tabla el número de alumnado matriculado y el de respuestas recibidas en cada opción y realice una valoración crítica sobre la influencia que el proyecto ha ejercido en la opinión del alumnado.

Opinión del alumnado al inicio del proyecto				
Número de alumnado matriculado:				
Valoración del grado de dificultad ***que cree que va a tener*** *en la comprensión de los contenidos y/o en la adquisición de competencias asociadas a la asignatura en la que se enmarca el proyecto de innovación docente*				
Ninguna dificultad	Poca dificultad	Dificultad media	Bastante dificultad	Mucha dificultad
12	18	87	14	1
Opinión del alumnado en la etapa final del proyecto				
Valoración del grado de dificultad ***que ha tenido*** *en la comprensión de los contenidos y/o en la adquisición de competencias asociadas a la asignatura en la que se enmarca el proyecto de innovación docente*				
Ninguna dificultad	Poca dificultad	Dificultad media	Bastante dificultad	Mucha dificultad
12	28	36	10	2
Los elementos de innovación y mejora docente aplicados en esta asignatura han favorecido mi comprensión de los contenidos y/o la adquisición de competencias asociadas a la asignatura				
Nada de acuerdo	Poco de acuerdo	Ni en acuerdo ni en desacuerdo	Muy de acuerdo	Completamente de acuerdo
5	15	28	28	12
En el caso de la participación de profesorado invitado				
La participación del profesorado invitado ha supuesto un gran beneficio en mi formación				
Nada de acuerdo	Poco de acuerdo	Ni en acuerdo ni en desacuerdo	Muy de acuerdo	Completamente de acuerdo
Valoración crítica sobre la influencia que ha ejercido el proyecto en la opinión del alumnado				
La percepción de dificultad mejora de forma apreciable. En la encuesta inicial con 122 respuestas, la opción dificultad media concentra el 71,3%, mientras que la baja dificultad suma el 16,4%. En la encuesta final con 88 respuestas, baja dificultad asciende al 45,5% y la dificultad media desciende al 40,9%, con la franja de alta dificultad estable en torno al 13,6%. El patrón indica desplazamiento				

desde la zona media hacia menor dificultad percibida, consistente con práctica espaciada y retroalimentación inmediata. La valoración de la innovación es moderadamente favorable y heterogénea. Un 45,5% declara estar muy o completamente de acuerdo en que la intervención ayudó, un 31,8% se mantiene neutral y un 22,7% discrepa en distinto grado. Debe considerarse el descenso de participación de 122 a 88 respuestas y el carácter autoinformado de los indicadores. Con todo, el balance es positivo en términos de experiencia percibida, aunque persiste un subgrupo con mayor dificultad.

4. Describa las medidas de difusión a las que se comprometió en la solicitud y las que ha llevado a cabo[2].

Descripción de las medidas comprometidas en la solicitud
Para garantizar que los profesores de nuestra comunidad estén bien informados sobre las conclusiones y ventajas de nuestro proyecto de innovación docente, nos comprometemos a organizar una charla o taller en la Escuela Superior de Ingeniería. En este evento, abordaremos las metodologías utilizadas, los resultados alcanzados y las enseñanzas extraídas durante el desarrollo del proyecto. Anunciaremos la fecha y hora del taller con suficiente antelación para maximizar la asistencia y participación de todos los interesados. Durante la sesión, proporcionaremos una exposición detallada de todos los elementos cruciales del proyecto para asegurar una comprensión integral. Este enfoque tiene como objetivo no solo difundir conocimientos y promover la incorporación de estas innovaciones en diferentes ámbitos académicos, sino también fomentar un diálogo constructivo que pueda ser el catalizador de futuras iniciativas educativas dentro de nuestra institución.
Descripción de las medidas que se han llevado a cabo
El jueves, 11 de septiembre, a las 18:30 h, se celebró en el seminario FS06 de la ESI la charla abierta al profesorado anunciada en la solicitud. En ella se presentaron los objetivos, arquitectura de la plataforma, pruebas realizadas, resultados, aprendizajes y materiales del proyecto. Como resultado también del mismo, se ha elaborado y enviado un artículo al Simposio Internacional de Informática Educativa (SIIE 2025), que ha sido aceptada con el título: “A Telegram-Based Platform for Self-Assessment, NLP-Driven Classification and Collaborative Question Generation” y se adjunta como anexo. Dicha contribución, será presentada del 20 al 22 de noviembre de 2025, en Viseu, Portugal. Este resultado cumple la medida prevista de difusión académica en el foro de referencia mencionado en la solicitud y garantiza su proyección internacional conforme a la política de publicación del congreso. Se adjunta artículo aceptado.

[2] Si en la solicitud no indicó compromiso de difusión de resultados, este criterio no se tendrá en cuenta en la evaluación.

A Telegram-Based Platform for Self-Assessment: NLP-Driven Classification and Collaborative Question Generation

Gabriel Guerrero-Contreras*
Dept. of Computer Science and Eng.
University of Cadiz
Av. Universidad de Cádiz, 10
Puerto Real, 11519, Cádiz, Spain
gabriel.guerrero@uca.es

Sara Balderas-Díaz
Dept. of Computer Science and Eng.
University of Cadiz
Av. Universidad de Cádiz, 10
Puerto Real, 11519, Cádiz, Spain
sara.balderas@uca.es

***Abstract*—The increasing adoption of conversational interfaces and semantic technologies in education is enabling new forms of scalable, personalised, and reflective assessment. In this context, a self-assessment platform has been designed and deployed within Telegram to promote formative learning through collaborative content creation and iterative practice. The system allows students to generate multiple-choice questions (MCQs), which are automatically classified into instructional topics using Natural Language Processing (NLP) techniques. These questions are reused to construct personalized quizzes, offering immediate feedback and supporting iterative improvement. During a semester-long deployment involving over 149 participants, students authored 519 MCQs and completed 1,709 quizzes. Results indicate high engagement, consistent score improvements across attempts, and effective reuse of learner-generated content, demonstrating the potential of NLP-assisted, mobile-friendly assessment to support self-regulated learning and performance growth.**

***Index Terms*—Artificial Intelligence (AI), Mobile Learning, Natural Language Processing (NLP), Self-Assessment, Telegram Bots**

I. Introduction

The integration of Artificial Intelligence (AI) into educational systems is significantly transforming how learning processes are conceptualized, implemented, and evaluated [1]. Within this landscape, Natural Language Processing (NLP) has emerged as a key technological enabler, allowing machines to process and interpret human language in ways that support automated analysis of educational content and learner input [2], [3].

One of the persistent challenges in education is the development of scalable and responsive assessment strategies that can adapt to diverse learner needs [4]. Traditional formative assessment methods, although pedagogically sound, often require considerable human effort and lack the immediacy needed to provide timely, actionable feedback in dynamic learning environments [2]. To address this gap, assessment systems must evolve beyond static testing models. They should incorporate mechanisms that engage learners continuously, deliver instant feedback, and adapt to their individual learning profiles [4]. Achieving this level of adaptability requires more than technical infrastructure. It also demands alignment with how learners process and reflect on knowledge [3]. This requires support for internal cognitive processes that help students monitor and regulate their own understanding. In this regard, research in educational psychology emphasizes the importance of metacognitive development, whereby students become aware of their own learning strategies, recognize areas of weakness, and adjust their approaches accordingly [5].

In addition, assessment tools that facilitate this kind of reflective engagement not only measure performance but also contribute directly to the learning process. When combined with personalization features such as adapting content difficulty or focusing on specific topics based on prior outcomes, these systems can create a more effective and equitable learning experience for diverse student populations. This aligns with inclusive access initiatives, such as adaptive digital heritage platforms and multimodal learning systems that ensure accessibility for all learners [6], [7].

In parallel, recent advances in AI have led to the emergence of tools that facilitate adaptive assessment, collaborative content creation, and learning analytics. In this context, NLP, in particular, allows for semantic classification, question validation, and content organization, offering educators and learners efficient ways to assess and improve understanding [8]. These capabilities can be applied to construct intelligent educational platforms that reduce manual overhead while enhancing the quality and relevance of feedback [3].

This paper presents a self-assessment platform that combines a conversational interface, implemented through a Telegram bot, with an NLP-based topic classification system and a learning analytics module. The platform enables students to create multiple-choice questions, which are automatically categorized by topic using semantic similarity techniques. Interaction data are used to generate detailed reports that inform both learners and instructors, contributing to a more reflective and data-informed educational process.

The proposed system contributes to ongoing research in

AI-driven education by integrating automated language understanding, mobile accessibility, and performance visualization. It also aligns with international objectives such as the Sustainable Development Goals (SDGs), particularly Goal 4 on Quality Education and Goal 9 on Innovation and Infrastructure. These connections highlight the broader relevance of the platform as a scalable and equitable solution for enhancing teaching and learning through AI [9].

The remainder of this paper is organized as follows. Section II reviews related work on the application of NLP and conversational agents in educational settings. Section III describes the architecture and functionalities of the self-assessment platform, including the Telegram-based interface and NLP-driven question classification. Section IV presents the deployment results, analyzing student participation, quiz performance, and learning progression. Finally, Section V summarizes key findings and outlines directions for future work.

II. Related Work

AI has increasingly been integrated into educational systems to automate assessment, support personalized learning, and enhance instructional feedback mechanisms [1]. Among AI subfields, NLP plays a central role in interpreting learner-generated content, enabling systems to assess conceptual understanding, generate feedback, and classify responses based on semantic similarity [2], [4].

Recent studies demonstrate the capability of NLP-based models such as RoBERTa, ELECTRA, and ALBERT to accurately evaluate short-answer responses by assessing the semantic validity and confidence of student justifications [8]. These models have shown accuracies above 90%, even when applied to small datasets, making them suitable for scalable formative assessment [8]. Unlike traditional grading systems that prioritize correctness, NLP systems facilitate deeper insights into student thinking, thus enabling reflective learning processes [3].

Another line of work focuses on the quality and evaluation of student-generated content. Large Language Models (LLMs) have been used to assess the clarity, accuracy, and pedagogical value of student-created questions and explanations [10], [11]. Learner-sourcing strategies that integrate automated evaluation support collaborative content creation while preserving instructional integrity at scale [10]. Comparative analyses with peer-review models reveal that NLP can augment or partially automate quality assurance mechanisms in student-led content generation [11].

Conversational agents (CAs) and educational chatbots have become popular channels for delivering such intelligent support. These systems use NLP to interpret user input and provide real-time, context-aware responses [12]. Studies in K–12 and higher education have shown that CAs can improve learning outcomes, increase engagement, and reduce learner anxiety [3]. Rule-based systems have been increasingly replaced by adaptive models capable of semantic parsing and dialog management, allowing for personalized interactions that mirror human tutoring [12].

AI-driven feedback systems have also been deployed in simulation-based learning environments. In teacher education, adaptive systems have been used to guide users through structured decision-making processes using metacognitive cues [3]. These platforms not only scaffold reflection but also contribute to the development of higher-order thinking skills such as critical reasoning and strategic planning.

Despite these advancements, few systems provide a lightweight and mobile-friendly solution for student-driven content creation. This work addresses that gap by introducing a Telegram-based chatbot that allows students to create their own questions, which are then automatically classified by topic using NLP. The platform delivers immediate feedback and tracks performance, supporting ongoing self-assessment in an accessible and scalable format.

III. System Overview

The platform is organized as a set of containerized microservices coordinated via RESTful APIs, following a stateless event-driven architecture. User interaction occurs exclusively through the Telegram Bot interface, which exposes a minimal set of command-based workflows. Students interact with the system by issuing structured commands: `/submit` initiates the creation of a new self-assessment question; `/test` launches a topic-specific evaluation session; `/review` retrieves previously failed questions; and `/report` enables users to flag problematic questions. Each command triggers a dedicated control flow orchestrated by the Telegram Bot Service and routed to backend services according to its semantics. Beyond textual commands, the interface supports rich interaction via Telegram inline keyboards (Figure 1). These are used during test execution, topic selection, and reporting flows. When a user clicks a button, Telegram issues a `callback_query` update containing a compact payload in the field. This payload encodes the intended action and is parsed by the bot's routing logic. For instance, `ans:Q123:2` denotes the selection of option 2 for question Q123 or `topic:P2P` indicates the chosen subject for test generation.

Figure 2 illustrates the high-level architecture and inter-service communication.

A. Telegram Bot Service

It is implemented as a stateless microservice that exposes a single HTTPS endpoint (`/webhook`) registered as a webhook with the Telegram Bot API[1]. This component serves as the main entry point of the system and routes user interactions to the appropriate backend services.

The service processes two types of updates as defined by the Telegram Bot API: (i) text commands, represented by `message.text`, and (ii) callback interactions, represented by `callback_query`.

Text commands correspond to plain-text messages issued by the user, prefixed with a slash (e.g., `/submit`, `/test`).

[1] https://core.telegram.org/bots/api

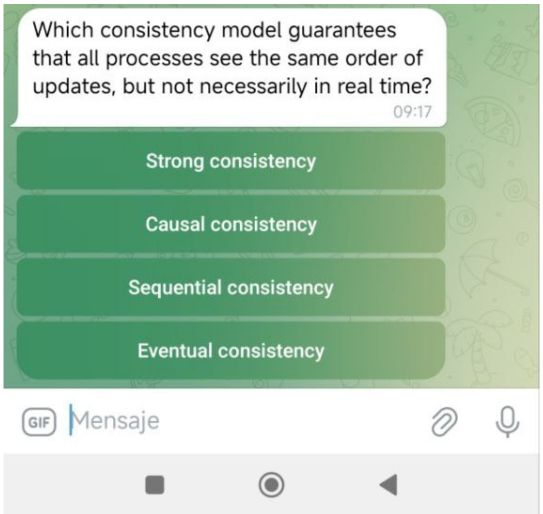


Fig. 1: Example of a Telegram inline keyboard interaction. The user is prompted with a multiple-choice question, and each response triggers a `callback_query` update containing an encoded payload indicating the selected option.

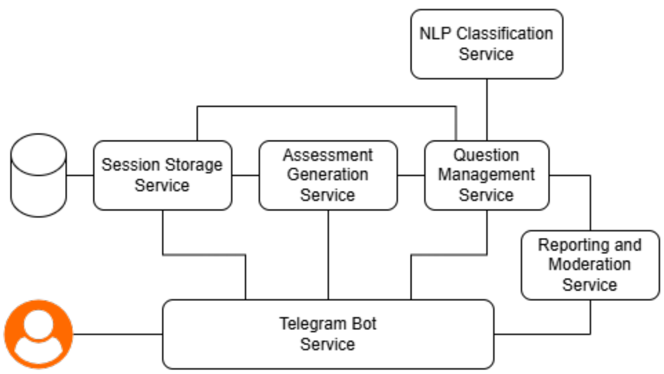


Fig. 2: Architecture of the self-assessment platform based on Telegram.

Callback interactions are triggered when users interact with inline keyboards attached to messages (e.g., during a self-assessment test or topic selection).

B. Session Storage Service

It is responsible for maintaining temporary user-specific state across multi-step interactions with the Telegram bot. As the Telegram Bot API is inherently stateless and does not provide any native mechanism for dialogue persistence, this service provides the necessary infrastructure to support context-aware conversational flows, assessment history tracking, and personalised user experiences.

All state is maintained using a namespaced key schema of the form `session:$telegram_user_id`, with values encoded as JSON documents. These documents store both high-level session metadata and command-specific payloads. Session entries have a configurable expiration policy (12 hours of inactivity) enforced via Redis TTLs.

In addition to managing temporary input sequences, the service is also used to record persistent user-specific data relevant to the assessment flow, including:

- List of questions previously answered (with correctness outcome)
- Identifiers of reported questions to avoid redundant submissions
- User preferences

This information is accessed by the Assessment Generation Service to prevent repetition, by the Telegram Bot Service to resume interrupted dialogues, and by the Reporting and Moderation Service to determine the eligible question list for user report.

C. Question Management Service

It is responsible for managing the complete lifecycle of user-submitted questions. The service is backed by a PostgreSQL database. Upon completion of the submission process through the bot interface, the Telegram Bot Service issues a `POST` request to the service, containing a structured JSON payload (see Listing 1).

```
{
  "question_text": "Which consistency model ↩
      ↪ guarantees that all processes see the ↩
      ↪ same order of updates, but not ↩
      ↪ necessarily in real time?",
  "options": [
    "Strong consistency",
    "Causal consistency",
    "Sequential consistency",
    "Eventual consistency"
  ],
  "correct_index": 3,
  "user_id": "123456789",
  "timestamp": "2025-07-21T12:34:56Z"
}
```

Listing 1: Example of a question submission payload for the Question Management Service.

The service performs schema and semantic validation, including checks for the number of options, index range, uniqueness, and maximum character length. If the input is valid, the question is stored persistently and assigned a unique identifier.

The service also provides read access through query endpoints consumed by other components of the system. These endpoints allow retrieval of questions tagged with a specific topic, optionally filtered by report count or usage frequency. They also support fetching the full contents of a question, including its metadata and classification, and enable authorized moderators to update question content or override classification labels following manual validation.

Each question is associated with a dynamic quality score computed from the number of user reports, the average correctness rate across self-assessment sessions, and peer validation metrics. Questions with low quality scores can be automatically excluded from future assessments or flagged for manual moderation.

In addition to this, each question is also assigned one or more semantic topic labels, along with associated confidence scores, by the NLP Classification Service. This classification is computed asynchronously upon question submission and is later used by the Assessment Generation Service to ensure topical coherence during test construction.

D. NLP Classification Service

It operates as a stateless component that receives a question and returns a set of predicted labels drawn from a predefined taxonomy. The classification pipeline consists of three stages:

1) The input text is tokenized, lowercased, and normalized (i.e., removal of punctuation, Unicode standardization).
2) The processed input is transformed into a dense representation using a sentence embedding model. The implementation relies on the `all-MiniLM-L6-v2` model from the `Sentence-BERT` family.
3) The embedding is passed to a multi-class logistic regression model. The model outputs a ranked list of candidate labels with associated confidence scores. If the top score is below a defined threshold (e.g., < 0.65), the question is flagged for manual review and temporarily assigned to a fallback class (*Uncertain*).

E. Assessment Generation Service

It is responsible for dynamically constructing and delivering self-assessment tests to students upon request. It receives requests via the Telegram Bot Service with the desired topic and user identifier as parameters. The service then selects a subset of questions related to the requested topic and returns them in a structured format suitable for interactive presentation via inline keyboards. Upon reception, the service executes the following pipeline:

1) Retrieves all questions associated with the requested topic from the Question Management Service. Only questions with valid classification labels and a quality score above a threshold (i.e., 0.75) are considered.
2) The user's previous session history, stored in the Session Storage Service, is used to exclude already-answered questions. Additionally, the service ensures topic coverage diversity by penalizing repetitive phrasings or near-duplicate questions (based on cosine similarity of embeddings).
3) A fixed number of questions (default: 10) is sampled uniformly at random.
4) Each question is returned with its ID, text, and shuffled answer options.

When the last question has been answered, the service returns a summary of the results, including accuracy and a list of incorrect items.

F. Reporting and Moderation Service

It enables end-users to flag problematic content, aggregates community feedback, and allows authorized moderators to inspect and act upon reports. The service contributes to maintaining the pedagogical reliability and semantic correctness of the assessment dataset.

A user may invoke the reporting mechanism at any time using the `/report` command. The Telegram Bot Service then presents the user with a list of recently answered questions stored in the session history. Reports include a predefined reason code from a closed taxonomy (i.e., `incorrect_answer`, `ambiguous_wording`, `off_topic`, `inappropriate_content`). The system computes a dynamic *quality score* per question as a weighted function of the total number of reports, the severity of reported issues, and the overall success rate in user assessments. This score is used by the Assessment Generation Service to filter out low-quality items.

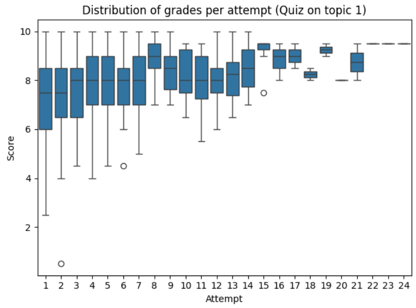
For moderation, the service exposes a restricted interface that returns a ranked list of reported questions, sorted by report frequency and impact. Authorized moderators can access the detailed report history for each question and perform corrective actions such as editing the question text or correcting the answer, reclassifying the topic labels, or disabling the question so that it is excluded from future assessments.

IV. Proof of Concept and Results

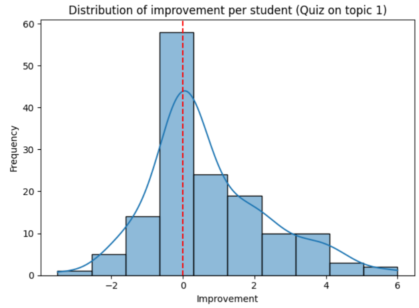
To assess the pedagogical value and usability of the proposed system, a semester-long deployment was carried out in a university-level course involving 149 students. The platform was designed to shift the traditional role of students from passive recipients of instructor-designed assessments to active contributors through a learner-sourcing model. Using the Telegram-based chatbot interface, students were able to create multiple-choice questions (MCQs), each consisting of a correct answer and three distractors, as part of their formative learning process.

A total of 519 MCQs were collaboratively generated by students over the semester. Each submission was automatically analyzed and assigned to a relevant instructional category using the platform's semantic classification engine. The resulting distribution of authored content was: `Topic 1 (Introduction)` – 140 questions, `Topic 2 (Communication Models)` – 112 questions, `Topic 3 (Synchronization)` – 112 questions, `Topic 4 (Peer-to-Peer Systems)` – 86 questions, and `Topic 5 (Distributed Resources)` – 69 questions.

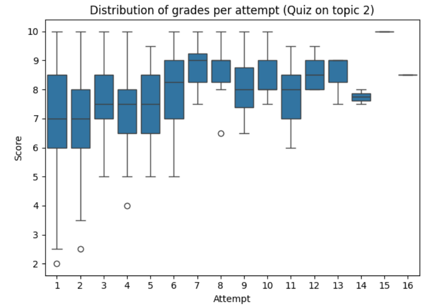
These categorized questions were subsequently reused by the system to create personalized self-assessment quizzes, that learners could access on demand. The system recorded a total of 1,709 quiz completions across all topics, indicating strong student engagement. Completion counts per topic were as follows: `Topic 1: 643`, `Topic 2: 485`, `Topic 3: 256`, `Topic 4: 174`, and `Topic 5: 151`. This usage data highlights not only the scalability and efficiency of the chatbot interface but also the educational potential of integrating student-generated content into the assessment workflow. Moreover, higher completion rates for earlier topics suggest



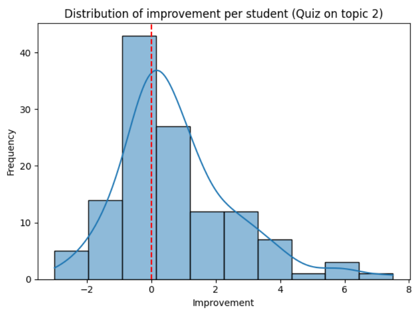
(a) Grades by attempt (T1)



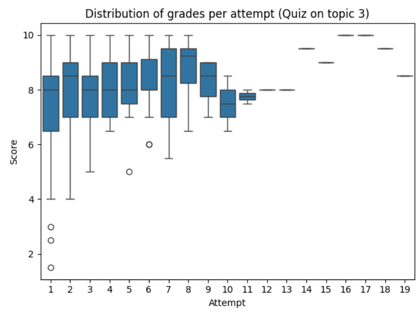
(b) Improvement distribution (T1)



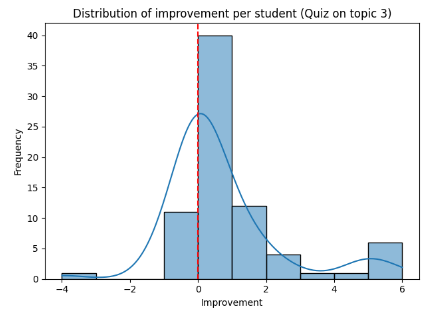
(c) Grades by attempt (T2)



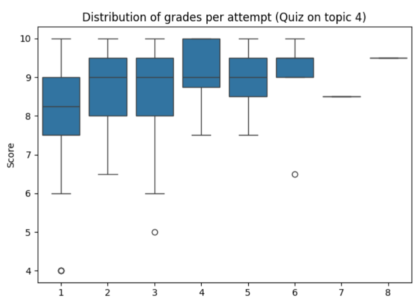
(d) Improvement distribution (T2)



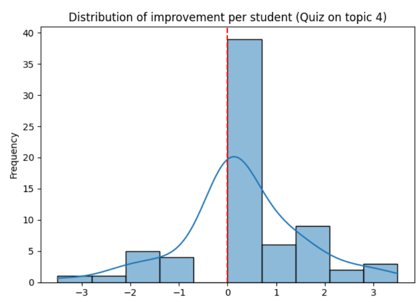
(e) Grades by attempt (T3)



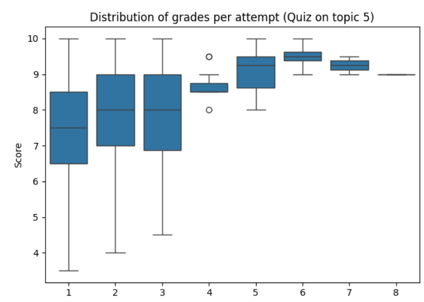
(f) Improvement distribution (T3)



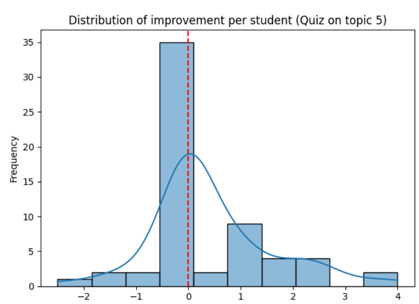
(g) Grades by attempt (T4)



(h) Improvement distribution (T4)



(i) Grades by attempt (T5)



(j) Improvement distribution (T5)

Fig. 3: Score trends and learning gains across topic-based quizzes. Each pair shows score distribution by attempt (left) and student improvement (right), highlighting iterative learning effects.

that students frequently revisited foundational material, aligning with spaced repetition strategies.

To analyze the effectiveness of repeated testing, we examined the distribution of student scores and learning improvements across five distinct topic-based quizzes. Each quiz allowed multiple attempts, and students' performance progression was recorded through the platform's interaction logs. Figure 3 presents the distribution of grades (boxplots) and improvement per student (histograms) for each topic. The boxplots reveal a clear upward trend in scores with repeated attempts, indicating that students benefited from repeated attempts and immediate feedback. This pattern is particularly notable in Topic 1 (Figure 3a), Topic 3 (Figure 3e), and Topic 5 (Figure 3i), where the median score consistently increases across attempts, reinforcing the positive impact of iterative practice on learning outcomes.

Across topics, the average scores also reflect these trends, with higher means in quizzes on topics 1 (7.70), 3 (7.87), and 4 (8.53), and slightly lower averages in topics 2 (7.44) and 5 (7.71). These results suggest not only that students improve over time but also that topic complexity and question quality may influence performance.

The histograms of score improvement provide additional insight into students' learning trajectories by visualizing the distribution of the difference between final and first quiz attempts. Most distributions are right-skewed, indicating overall learning gains. Quizzes on topic 4 (Figure 3h) show a cluster of students improving by +1 to +3 points, along with a central peak at zero that suggests plateauing. Quizzes on topic 3 (Figure 3f) display a broader distribution, indicating a subset of students whose scores declined (from –1 to –4). Topic 5 (Figure 3j) also exhibits a high concentration of scores around zero, reflecting performance stagnation for many students despite a modest positive trend. These patterns may reflect differences in content difficulty, feedback effectiveness, or student engagement. The red vertical line at zero in each histogram marks the boundary between performance improvement and decline. Overall, the visualizations illustrate heterogeneous learning trajectories, with repeated practice generally fostering improvement.

Taken together, the data suggest that the combination of learner-sourced content, iterative quiz engagement, and immediate automated feedback effectively promotes both sustained participation and academic improvement. The platform's integration of question authoring, semantic categorization, and formative assessment encourages reflective learning and progressively deeper understanding.

V. Conclusions and Future Work

This work introduced a mobile-friendly self-assessment platform that leverages a Telegram chatbot, NLP-based topic classification, and learner-sourced content to support formative learning. The semester-long deployment demonstrated that students actively engaged with the system, authoring hundreds of questions and completing over 1,700 quizzes.

Analysis of quiz data revealed measurable learning gains, particularly through repeated attempts and automated feedback. These results highlight the effectiveness of combining collaborative content creation with lightweight, NLP-enhanced assessment tools.

Future work will explore adaptive difficulty tuning, multilingual support, and integration with broader learning analytics platforms to enhance personalization and scalability across educational contexts.

Acknowledgements

This work was supported by the University of Cádiz through the Grants for the Communication of Teaching Innovation Outcomes programme.

References

[1] L. Chen, P. Chen, and Z. Lin, "Artificial intelligence in education: A review," *IEEE access*, vol. 8, pp. 75264–75278, 2020.

[2] M. Liang, "Leveraging natural language processing for automated assessment and feedback production in virtual education settings," *Journal of Computational Methods in Sciences and Engineering*, vol. 25, no. 3, pp. 2502–2515, 2025.

[3] E. Bauer, M. Sailer, F. Niklas, S. Greiff, S. Sarbu-Rothsching, J. M. Zottmann, J. Kiesewetter, M. Stadler, M. R. Fischer, T. Seidel, *et al.*, "Ai-based adaptive feedback in simulations for teacher education: an experimental replication in the field," *Journal of Computer Assisted Learning*, vol. 41, no. 1, p. e13123, 2025.

[4] R. Somers, S. Cunningham-Nelson, and W. Boles, "Applying natural language processing to automatically assess student conceptual understanding from textual responses," *Australasian Journal of Educational Technology*, vol. 37, no. 5, pp. 98–115, 2021.

[5] S. F. Rivas, C. Saiz, and C. Ossa, "Metacognitive strategies and development of critical thinking in higher education," *Frontiers in psychology*, vol. 13, p. 913219, 2022.

[6] S. Balderas-Díaz, G. Guerrero-Contreras, M. Ramírez-Vela, S. Toribio-Camuñas, N. C. Gay, and A. M. Reguera, "Inclusive education and cultural heritage access: The lectpat platform multimodal online dictionary," in *2024 International Symposium on Computers in Education (SIIE)*, pp. 1–6, IEEE, 2024.

[7] G. Guerrero-Contreras, S. Balderas-Díaz, S. Toribio-Camuñas, M. L. Rodríguez-Almendros, and M. V. Hurtado-Torres, "Al-musactra: A unified digital platform for inclusive access to the cultural heritage of andalusia," in *2024 International Symposium on Computers in Education (SIIE)*, pp. 1–6, IEEE, 2024.

[8] K. Li, C. Qian, and X. Yang, "Evaluating the quality of student-generated content in learnersourcing: A large language model based approach," *Education and Information Technologies*, vol. 30, no. 2, pp. 2331–2360, 2025.

[9] World Health Organization, "Disability," 2023.

[10] A. Darvishi, H. Khosravi, A. Rahimi, S. Sadiq, and D. Gašević, "Assessing the quality of student-generated content at scale: A comparative analysis of peer-review models," *IEEE Transactions on Learning Technologies*, vol. 16, no. 1, pp. 106–120, 2022.

[11] C. Rodrigues, A. Reis, R. Pereira, P. Martins, J. Sousa, and T. Pinto, "A review of conversational agents in education," in *International Conference on Technology and Innovation in Learning, Teaching and Education*, pp. 461–467, Springer, 2022.

[12] T. Ait Baha, M. El Hajji, Y. Es-Saady, and H. Fadili, "The impact of educational chatbot on student learning experience," *Education and Information Technologies*, vol. 29, no. 8, pp. 10153–10176, 2024.