

Los Sistemas Tutoriales Inteligentes: Apuntes para una metodología de desarrollo

Intelligent Tutoring Systems: Notes Toward a Development Methodology

Jorge E. Gil Mateos¹

¹ Universidad de La Habana, Cuba. jorgegil@cepes.uh.cu



PARA CITAR ESTE ARTÍCULO

Gil Mateos, J. E. Los Sistemas Tutoriales Inteligentes: Apuntes para una metodología de desarrollo. *Alternativas*, 25(3). <https://doi.org/10.23878/alternativas.v25i3.457>

DOI

<https://doi.org/10.23878/alternativas.v25i3.457>

CORRESPONDENCIA

jorgegil@cepes.uh.cu



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

Av. Carlos Julio Arosemena, Km 1,5. Guayaquil, Ecuador
Teléfono: +593 4 380 4600
Correo electrónico: revista.alternativas@cu.ucsg.edu.ec
Web: www.ucsg.edu.ec



© The Autor(s), 2024

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. To view a copy of this license visit <http://creativecommons.org/licenses/by/4.0/>.

Los Sistemas Tutoriales Inteligentes: Apuntes para una metodología de desarrollo

Intelligent Tutoring Systems: Notes Toward a Development Methodology

Jorge E. Gil Mateos¹

¹ Universidad de La Habana, Cuba. jorgegil@cepes.uh.cu

RESUMEN

La puesta en funcionamiento de un Sistema Tutor Inteligente (STI) basado en chatbot para Moodle requiere un enfoque multidisciplinario que combine la pedagogía, la ingeniería del conocimiento y el desarrollo tecnológico. Este artículo presenta, en una primera parte, una metodología detallada para el desarrollo de un Sistema Tutor Inteligente integrado en Moodle, diseñado para ser adaptable a cualquier asignatura. El sistema, estructurado en los módulos clásicos de estudiante, tutor y dominio, utiliza un chatbot como interfaz principal de interacción. La metodología se divide en siete fases, desde el análisis inicial hasta la mejora continua, con especial énfasis en la construcción de una base de conocimientos semántica mediante el uso intensivo de metadatos y en la integración técnica con la plataforma Moodle. La segunda parte del trabajo: Componentes Esenciales de un Sistema Tutor Inteligente; Diseño, Metadatos y Motor Conversacional está destinada a proporcionar a desarrolladores e investigadores una hoja de ruta clara y práctica para construir tutores inteligentes que ofrezcan apoyo personalizado y escalable.

PALABRAS CLAVE

Inteligencia artificial, Tutoría inteligente, Chatbot, Metodología, Moodle

ABSTRACT

Implementing a chatbot-based Intelligent Tutoring System (ITS) for Moodle requires a multidisciplinary approach that combines pedagogy, knowledge engineering, and technological development. This article presents, in its first part, a detailed methodology for developing an Intelligent Tutoring System integrated into Moodle, designed to be adaptable to any subject. The system, structured in the classic student, tutor, and domain modules, uses a chatbot as its main interaction interface. The methodology is divided into seven phases, from initial analysis to continuous improvement, with special emphasis on building a semantic knowledge base through the intensive use of metadata and on technical integration with the Moodle platform. The second part of this work, "Essential Components of an Intelligent Tutoring System: Design, Metadata, and Conversational Engine," aims to provide developers and researchers with a clear and practical roadmap for building intelligent tutors that offer personalized and scalable support.

KEYWORDS

Artificial Intelligent, Intelligent tutorial, Chatbot, Methodology, Moodle

Introducción

El auge de la Inteligencia Artificial (IA) generativa está transformando el panorama educativo, ofreciendo oportunidades sin precedentes para la personalización del aprendizaje. Los STI, que combinan técnicas de IA con modelos pedagógicos, emergen como herramientas clave para proporcionar una tutoría personalizada, imitando la actividad de un tutor real (Woolf, 2010). Sin embargo, el desarrollo de STI adaptables a cualquier dominio del conocimiento y plenamente integrados en Sistemas de Gestión de Aprendizaje (LMS) como Moodle sigue siendo un desafío. Este artículo aborda esta brecha, proponiendo una metodología exhaustiva para construir un STI genérico cuyo núcleo es un chatbot conversacional.

Para poner en funcionamiento un Sistema Tutor Inteligente (STI) basado en chatbot integrado en Moodle, que sea capaz de adaptarse a cualquier asignatura y curso, es necesario seguir una metodología estructurada que contemple el diseño, desarrollo e implementación del sistema, así como la participación de los diferentes roles de usuario. A continuación, se describen los pasos y procedimientos requeridos, con especial énfasis en la construcción de la base de conocimientos y las acciones específicas de cada rol.

La metodología general para el desarrollo del STI es un proceso que se divide en fases iterativas que abarcan desde el análisis inicial hasta la mejora continua. Cada fase involucra a roles específicos y produce entregables clave.

A continuación se describen las fases que conforman la metodología, así como los diferentes roles involucrados, las acciones por rol, los entregables, etc.

Fase 1: Análisis y definición de requerimientos

Objetivo: Comprender las necesidades de todos los actores y establecer los lineamientos funcionales y técnicos del STI.

En esta fase intervienen los roles de: profesor experto en contenidos y diseño de cursos, profesor acompañante, supervisor pedagógico, y el equipo técnico/desarrollador.

Acciones por rol:

- Profesor experto en contenidos y diseño de cursos: Define los objetivos pedagógicos del curso, la estructura general, los tipos de materiales que se utilizarán (bibliografía, videos, actividades, recur-

sos, etc.) y las posibles interacciones que los estudiantes podrían requerir con el tutor.

- Profesor acompañante: Aporta su experiencia sobre las dudas frecuentes de los estudiantes, las dificultades típicas y los momentos en que suele necesitarse apoyo adicional.
- Supervisor pedagógico: Establece los criterios de calidad pedagógica y las métricas de seguimiento que deberá cumplir el STI.
- Equipo técnico/desarrollador: Analiza la integración con Moodle, los requisitos de infraestructura y las limitaciones tecnológicas.
- En esta fase se producen diversos entregables, a saber:
 - Documento de especificación funcional del STI.
 - Definición de los casos de uso (por ejemplo: consultar dudas sobre un tema, pedir ejercicios, solicitar explicaciones alternativas, etc.).
 - Mapa de contenidos del curso y su estructura.

Fase 2: Diseño del sistema y arquitectura

Objetivo: Definir la arquitectura del STI, los módulos que lo componen y la interacción entre ellos.

En este trabajo se considerará un sistema estructurado en los tres módulos clásicos: estudiante, profesor y dominio, cuya descripción es la siguiente:

- Módulo del estudiante: Gestiona el perfil del estudiante, su progreso, preferencias y estilos de aprendizaje, y el historial de interacciones.
- Módulo del profesor: Permite a los profesores configurar el STI, cargar materiales, definir reglas de tutoría y supervisar el desempeño.
- Base de conocimientos: Almacena y organiza todo el contenido del curso, así como las relaciones conceptuales y las estrategias pedagógicas.

Acciones por rol:

- Profesor experto: Colabora en el diseño de la ontología del curso (conceptos, relaciones, jerarquías) y en la definición de las estrategias de tutoría (cuándo y cómo

intervenir). En esta fase resultan muy importantes documentos tales como planes de estudio, planes de clase, etc.

- Equipo técnico: Define la arquitectura de software (por ejemplo, uso de motores de NLP, bases de datos, APIs de Moodle) y la estructura de la base de conocimientos.

Entregables:

- Diagrama de arquitectura del STI.
- Diseño de la base de datos y de la base de conocimientos.
- Especificación de la interfaz de comunicación con Moodle.

Fase 3: Construcción de la base de conocimientos

Objetivo: Crear el repositorio central que contendrá toda la información que el STI utilizará para responder y guiar a los estudiantes.

La base de conocimientos es donde se almacenarán todos los insumos necesarios para el curso en cuestión, a saber:

- Materiales del curso (textos, presentaciones, videos, audios, simulaciones, etc.).
- Metadatos asociados: tema, palabras clave, nivel de dificultad, tipo de recurso. Los metadatos tienen una relevancia notable pues son los descriptores de clasificación para búsqueda y recuperación de información en el momento que el usuario lo requiera.
- Relaciones semánticas entre conceptos (por ejemplo, “tema A es prerequisite de tema B”). Las relaciones semánticas pueden ser tantas y tan disímiles como el contenido lo requiera.
- Preguntas frecuentes y sus respuestas elaboradas por el profesor.
- Estrategias pedagógicas: rutas de aprendizaje, ejercicios recomendados según el nivel del estudiante, o según los estilos de aprendizaje.
- Reglas de inferencia para generar respuestas personalizadas.
- Proceso de construcción de la base de conocimientos:
 - Recopilación de materiales: El profesor experto sube todos los recursos del curso a la plataforma (Moodle) y los etiqueta con metadatos.

- Extracción y estructuración: Mediante herramientas semiautomáticas (procesamiento de lenguaje natural, minería de textos) se extraen conceptos, definiciones y relaciones. Se crea una ontología del dominio.
- Validación: El profesor experto revisa y ajusta la ontología y las relaciones para garantizar su precisión pedagógica.
- Incorporación de conocimiento tácito: El profesor acompañante añade explicaciones alternativas, ejemplos y respuestas a preguntas frecuentes basadas en su experiencia.
- Integración de recursos multimedia: Se indexan videos, audios y simulaciones para que el chatbot pueda recomendarlos según el contexto.

Acciones por rol:

- Profesor experto: Supervisa la calidad y completitud de la base de conocimientos.
- Profesor acompañante: Enriquece la base con interacciones reales y casos prácticos.
- Equipo técnico: Implementa los algoritmos de procesamiento y las interfaces de carga.

Entregables:

- Base de conocimientos estructurada y poblada con los contenidos del primer curso piloto.
- Documentación de la ontología y las reglas de tutoría.

Fase 4: Implementación del chatbot y su integración en Moodle

Objetivo: Desarrollar el motor del chatbot que utilizará la base de conocimientos para interactuar con los estudiantes, e integrarlo en el entorno Moodle.

Componentes del chatbot:

- Módulo de comprensión del lenguaje: Interpreta las preguntas de los estudiantes (en lenguaje natural) y las convierte en consultas a la base de conocimientos.
- Módulo de diálogo: Gestiona el flujo de la conversación, mantiene el contexto y decide las respuestas.

- Módulo de generación de respuestas: Construye respuestas en lenguaje natural, combinando fragmentos de texto, recomendaciones de recursos o acciones (Ej. abrir un foro).

Integración con Moodle:

- El chatbot se incrusta en las páginas del curso mediante un bloque o plugin.
- Accede a los datos del estudiante (progreso, calificaciones) para personalizar las interacciones.
- Puede enviar notificaciones o sugerencias basadas en el comportamiento del estudiante.

Acciones por rol:

- Equipo técnico: Desarrolla e integra el chatbot.
- Profesor acompañante: Participa en pruebas de usabilidad y ajusta el tono y estilo de las respuestas.
- Supervisor pedagógico: Verifica que el chatbot cumple con las directrices pedagógicas.

Entregables:

- Chatbot funcional integrado en Moodle.
- Manual de uso para profesores y estudiantes.

Fase 5: Pruebas y validación

Objetivo: Verificar el correcto funcionamiento del STI en un entorno controlado y realizar ajustes.

Tipos de pruebas:

- Pruebas técnicas: Rendimiento, integración con Moodle, manejo de concurrencia. Esta prueba es de tipo más técnico pues se ocupa de medir la capacidad de Chatbot de operar, a la vez, con una cantidad considerable de estudiantes.
- Pruebas pedagógicas: Eficacia de las respuestas, adecuación al nivel del estudiante, capacidad de guiar el aprendizaje.
- Pruebas de usuario: Con un grupo reducido de estudiantes y profesores.

Acciones por rol:

- Estudiantes (grupo piloto): Interactúan con el chatbot y proporcionan retroalimentación.
- Profesor acompañante: Supervisa las interacciones y evalúa la calidad de las respuestas.
- Supervisor pedagógico: Analiza los registros de interacción y verifica el cumplimiento de los objetivos.
- Equipo técnico: Corrige errores y optimiza el sistema.

Entregables:

- Informe de pruebas y validación.
- Lista de mejoras a implementar.

Fase 6: Despliegue y monitoreo

Objetivo: Poner el STI a disposición de todos los estudiantes y establecer mecanismos de supervisión continua.

Acciones por rol:

- Profesor experto: Asegura que el curso está listo y que el STI está correctamente configurado.
- Profesor acompañante: Durante el desarrollo del curso, atiende las consultas que el chatbot no pueda resolver y retroalimenta al sistema.
- Supervisor pedagógico: Monitorea indicadores clave (número de interacciones, preguntas no respondidas, satisfacción del estudiante) y genera informes periódicos.
- Equipo técnico: Mantiene el sistema, aplica actualizaciones y resuelve incidencias.

Entregables:

- STI en producción.
- Panel de monitoreo para supervisores y profesores.

Fase 7: Evaluación y mejora continua

Objetivo: Evaluar el impacto del STI en el aprendizaje y realizar mejoras iterativas.

Métodos de evaluación:

- Análisis de los registros de interacción para identificar patrones de uso y dificultades.

- Encuestas de satisfacción a estudiantes y profesores.
- Comparación de resultados académicos con cursos anteriores.

Acciones por rol:

- Profesor experto: Propone ajustes en los contenidos o en la estructura de la base de conocimientos.
- Profesor acompañante: Aporta nuevas preguntas frecuentes y casos.
- Supervisor pedagógico: Coordina el proceso de mejora y asegura la calidad.
- Equipo técnico: Implementa las mejoras en el sistema.

Entregables:

- Informe de evaluación y plan de mejoras.
- Base de conocimientos actualizada.

Componentes Esenciales de un Sistema Tutor Inteligente: Diseño, Metadatos y Motor Conversacional

A continuación, y por su importancia, se desglosa la arquitectura interna de un Sistema Tutor Inteligente (STI) genérico para Moodle, analizando en detalle cada uno de sus componentes fundamentales. Se describe la función, el diseño y la interacción del Módulo del Estudiante, el Módulo del Profesor, la Base de Conocimientos y el Motor del Chatbot. Se presta especial atención al papel vertebrador de los metadatos en la organización semántica de los recursos y a la lógica del motor conversacional que personaliza la tutoría.

Se busca ofrecer una visión integral de la ingeniería detrás de un STI moderno, útil para investigadores y desarrolladores que buscan comprender la complejidad y el potencial de estos sistemas.

La arquitectura de un STI se articula tradicionalmente en torno, como se mencionó antes, a tres módulos fundamentales: el módulo del estudiante, el módulo del tutor (o pedagógico) y el módulo del dominio (o experto) (Sleman & Brown, 1982).

En un STI moderno, esta arquitectura se materializa en componentes de software con responsabilidades bien definidas que interactúan para ofrecer una experiencia de aprendizaje adaptativa.

Veamos a continuación cada uno de estos tres módulos en detalles:

El Módulo del Estudiante: Perfil, Progreso y Modelo de Conocimiento

- Funciones: Almacena información estática (perfil, preferencias) y dinámica (progreso, historial de interacciones) del estudiante.
- Modelo de Conocimiento: Implementa una representación del nivel de dominio del estudiante sobre cada concepto de la ontología. Se puede utilizar un modelo de overlay (comparación con el conocimiento experto) o uno más complejo basado en redes bayesianas (Brusilovsky & Millán, 2007). Esta información es crucial para personalizar las respuestas del chatbot.

El Módulo del Profesor: Configuración, Supervisión y Gestión de Contenido

- Interfaz de Autoría: Proporciona herramientas para que el profesor experto configure el STI sin necesidad de programación. Esto incluye:
 - Gestión de la ontología del curso (crear conceptos y relaciones).
 - Carga y etiquetado de recursos con metadatos.
 - Definición de reglas pedagógicas en un formato comprensible.
 - Edición del banco de FAQs.
- Panel de Control: Ofrece al profesor acompañante y al supervisor una visión del rendimiento del STI y de las interacciones de los estudiantes, permitiendo la detección temprana de dificultades (Baker & Inventado, 2014).

La Base de Conocimientos: El Corazón Semántico del Sistema

- Estructura Multimodal: La Base de conocimientos no es un simple repositorio de archivos, sino una estructura de datos compleja que integra:
- La Ontología: Un grafo de conceptos en Neo4j que representa el "esqueleto" del conocimiento (Robinson et al., 2015).
- El Repositorio de Recursos: Los archivos en bruto (PDF, MP4, etc.) almacenados en un sistema de objetos (como MinIO).
- El Índice Semántico: Los recursos y sus metadatos se indexan en Elasticsearch, permitiendo búsquedas por relevancia,

similitud semántica y filtrado por múltiples criterios (tipo, dificultad, concepto asociado) (Gormley & Tong, 2015).

- Los Metadatos como Vertebradores: Cada recurso está envuelto en una capa de metadatos que lo describe y lo conecta con la ontología. Esta capa de metadatos es lo que permite al chatbot "comprender" qué es un recurso, para qué sirve y a quién recomendarlo (Duval & Hodgins, 2003).

El Motor del Chatbot: Comprensión, Diálogo y Respuesta

- Módulo de Comprensión del Lenguaje Natural (NLU): Su función es traducir la pregunta del estudiante (en lenguaje natural) a una representación estructurada que el sistema pueda procesar. Para ello, utiliza modelos entrenados para reconocer la intención (ej. preguntar_definicion) y las entidades (ej. el concepto derivada) (Jurafsky & Martin, 2023).
- Gestor de Diálogo: Es el "cerebro" de la conversación. Mantiene el contexto, consulta el modelo del estudiante y la base de conocimientos, y decide la siguiente acción. Esta decisión puede basarse en reglas predefinidas o en políticas de aprendizaje automático (ej. aprendizaje por refuerzo) (Bocklisch et al., 2017). Por ejemplo, si la intención es preguntar_definicion y el nivel del estudiante en ese concepto es bajo, el gestor puede decidir no solo dar la definición, sino también recomendar un recurso de nivel básico.
- Módulo de Generación de Respuestas: Construye la respuesta final para el estudiante. Utiliza plantillas de lenguaje que se rellenan con información dinámica obtenida de la base de conocimientos (definiciones, enlaces a recursos) para ofrecer una respuesta natural y contextualizada.

Conclusiones

La metodología presentada ofrece un camino estructurado y tecnológicamente fundamentado para construir un STI genérico en Moodle. La

combinación de una arquitectura de microservicios, una base de conocimientos enriquecida con metadatos semánticos y un motor de diálogo basado en NLU permite crear un asistente virtual capaz de ofrecer una experiencia de aprendizaje personalizada y escalable a cualquier área de conocimiento.

La eficacia de un STI reside en la correcta orquestación de sus componentes internos. Un módulo de estudiante que modele con precisión el conocimiento, una base de conocimientos rica en semántica y metadatos, y un motor de diálogo que integre toda esta información para tomar decisiones pedagógicas son las claves para construir un sistema de tutoría inteligente, personalizado y escalable.

Referencias bibliográficas

- Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. En J. A. Larusson & B. White (Eds.), *Learning analytics: From research to practice* (pp. 61-75). Springer. https://doi.org/10.1007/978-1-4614-3305-7_4
- Bocklisch, T., Faulkner, J., Pawlowski, N., & Nichol, A. (2017). Rasa: Open source language understanding and dialogue management. arXiv. <https://arxiv.org/abs/1712.05181>
- Brusilovsky, P., & Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. En P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: Methods and strategies of web personalization* (pp. 3-53). Springer.
- Duval, E., & Hodgins, W. (2003). A LOM research agenda. *Proceedings of the 12th International Conference on World Wide Web*, 1-9. <https://doi.org/10.1145/775152.775158>
- Gormley, C., & Tong, Z. (2015). *Elasticsearch: The definitive guide*. O'Reilly Media.
- Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Pearson. <https://web.stanford.edu/~jurafsky/slp3/>
- Robinson, I., Webber, J., & Eifrem, E. (2015). *Graph databases: New opportunities for connected data* (2nd ed.). O'Reilly Media.
- Sleeman, D., & Brown, J. S. (Eds.). (1982). *Intelligent tutoring systems*. Academic Press.
- Woolf, B. P. (2010). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann.

