

# Diseño e Implementación de un Agente Inteligente Local para la Gestión Proactiva de Tareas Académicas con Monitoreo de Moodle

Antonio Baruc Maldonado Alfaro, Edson David Texon Lozano y Max Roet Ramirez Hernandez  
Licenciatura en Ingeniería en Inteligencia Artificial  
Universidad de Xalapa

**Resumen**—En este trabajo se presenta el diseño e implementación de un agente inteligente local orientado a la gestión proactiva de tareas académicas. El agente permite registrar recordatorios mediante lenguaje natural, interpretar expresiones temporales absolutas y relativas, mantener persistencia mediante una base de datos local y ejecutar acciones autónomas mediante notificaciones del sistema operativo. Asimismo, se integra un módulo de monitoreo periódico de un entorno académico tipo Moodle, capaz de detectar la aparición de nuevas tareas y generar recordatorios automáticos sin intervención del usuario. El sistema fue desarrollado bajo una arquitectura modular con servicios desacoplados, priorizando un enfoque *offline-first* y la robustez ante fallos comunes. Las pruebas realizadas demuestran la autonomía del agente, su capacidad de percepción del entorno y su utilidad práctica en contextos académicos.

**Index Terms**—Agentes inteligentes, FastAPI, SQLite, APScheduler, Moodle, notificaciones, sistemas autónomos.

## I. INTRODUCCIÓN

En el ámbito académico, los estudiantes suelen enfrentarse a múltiples tareas, evaluaciones y entregas distribuidas en diferentes materias y plataformas digitales. Un problema recurrente no es únicamente recordar una tarea específica, sino recordar revisar de manera constante los entornos académicos donde dichas tareas son publicadas. Esta situación incrementa la carga cognitiva del estudiante y eleva la probabilidad de incumplimientos [2].

Las aplicaciones tradicionales de gestión de tareas suelen depender completamente de la interacción explícita del usuario, funcionando como sistemas pasivos. En contraste, un agente inteligente puede actuar de forma autónoma, monitoreando el entorno, manteniendo memoria interna y ejecutando acciones sin intervención directa. Bajo este contexto, se propone el desarrollo de un agente inteligente local que asista al estudiante de manera proactiva en la gestión de tareas académicas [1].

## II. OBJETIVO

El objetivo principal de este proyecto es diseñar e implementar un agente inteligente local capaz de:

- Interpretar instrucciones en lenguaje natural para el registro de tareas.

- Procesar expresiones temporales absolutas y relativas.
- Mantener persistencia del estado mediante una base de datos local.
- Ejecutar acciones autónomas basadas en el tiempo.
- Monitorear un entorno académico tipo Moodle para detectar nuevas tareas.
- Notificar al usuario de forma proactiva sin requerir interacción constante.

## III. CLASIFICACIÓN DEL AGENTE

El sistema desarrollado se clasifica como un **agente inteligente autónomo orientado a objetivos**, con características reactivas y deliberativas [2]:

- **Autónomo**: opera sin intervención continua del usuario.
- **Reactivo**: responde a eventos temporales y cambios en el entorno.
- **Deliberativo**: decide cuándo notificar y evita acciones redundantes.
- **Persistente**: mantiene memoria interna mediante almacenamiento local.

Estas propiedades permiten distinguir al sistema de una aplicación convencional de gestión de tareas.

## IV. ARQUITECTURA DEL SISTEMA

El sistema se implementó bajo una arquitectura modular, separando responsabilidades para facilitar mantenimiento y escalabilidad [3].

### IV-A. API

Se utiliza FastAPI como servidor local, proporcionando endpoints para la interacción con el usuario y la interfaz web.

### IV-B. Persistencia

La información de las tareas se almacena en una base de datos SQLite local, garantizando ligereza y confiabilidad [4].

### IV-C. Herramientas del Agente

Las acciones del agente se encapsulan en una capa de herramientas, permitiendo agregar, listar y completar recordatorios sin acoplar la lógica central.

#### IV-D. Scheduler

El comportamiento autónomo del agente es habilitado mediante APScheduler, el cual ejecuta de forma periódica los módulos de revisión temporal, monitoreo de Moodle, resumen proactivo y limpieza de duplicados [5].

### V. PROCESAMIENTO DE LENGUAJE NATURAL

El agente interpreta instrucciones en español mediante un enfoque ligero y local. Se detectan expresiones temporales absolutas y relativas utilizando técnicas de análisis determinista, reduciendo la dependencia de servicios externos y mejorando la confiabilidad del sistema [6].

### VI. MONITOREO DE MOODLE

Se implementó un módulo denominado *Moodle Watcher*, encargado de monitorear periódicamente un entorno académico simulado. Este módulo sigue el ciclo percepción–decisión–acción característico de los agentes inteligentes [1].

### VII. NOTIFICACIONES Y COMPORTAMIENTO AUTÓNOMO

El agente emite notificaciones locales cuando se cumple una condición temporal o se detecta un cambio relevante en el entorno académico, utilizando las capacidades del sistema operativo para alertar al usuario de forma inmediata [7].

### VIII. PRUEBAS Y VALIDACIÓN

Las pruebas realizadas confirman que el agente actúa de forma autónoma, confiable y persistente, manteniendo coherencia en su estado incluso tras reinicios del sistema.

### IX. LIMITACIONES Y TRABAJO FUTURO

La automatización completa de la entrega de archivos a Moodle no se implementó debido a restricciones de autenticación y seguridad. Como trabajo futuro se propone la integración real mediante APIs oficiales y la extensión del sistema a notificaciones móviles.

### X. REQUISITOS DE INSTALACIÓN Y EJECUCIÓN

El sistema requiere Windows 10 o superior y Python 3.9 o superior. Las dependencias principales incluyen FastAPI, Uvicorn, APScheduler, Plyer, Dateparser y Jinja2.

```
pip install fastapi uvicorn apscheduler plyer dateparser
jinja2
```

```
uvicorn main:app --host 0.0.0.0 --port 8000 --reload
```

### REFERENCIAS

- [1] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.
- [2] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley, 2009.
- [3] FastAPI Documentation, “FastAPI: Modern, Fast Web Framework for Building APIs with Python.” [Online].
- [4] SQLite Documentation, “SQLite: Small, Fast, Reliable.” [Online].
- [5] APScheduler Documentation, “Advanced Python Scheduler.” [Online].
- [6] Dateparser Documentation, “Human Readable Date Parsing.” [Online].
- [7] Plyer Documentation, “Platform-independent API for Notifications.” [Online].

### XI. CONCLUSIÓN

Se desarrolló un agente inteligente local capaz de asistir proactivamente en la gestión de tareas académicas. El sistema demuestra que un enfoque autónomo y local puede reducir la carga cognitiva del estudiante y servir como base para futuras integraciones académicas más complejas.