



Universidad de las Américas

Nombre:

Aspiazu Jennifer

Velásquez Roberth

Carrera:

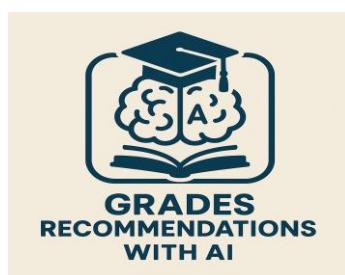
Ingeniería en Inteligencia Artificial

Materia: Programación

Facultad de Ingeniería y Ciencias Aplicadas

TEMA:

sistema SIGE-IA (Sistema Inteligente de Gestión Estudiantil)



Informe del Proyecto SIGE-IA

1. Introducción

En el contexto educativo ecuatoriano, particularmente en zonas urbanas como Quito, existen problemáticas asociadas al rendimiento estudiantil, como la desmotivación, la falta de seguimiento emocional, y la carencia de herramientas que apoyen tanto al docente como al estudiante en la mejora continua. El presente proyecto, SIGE-IA

(Sistema Inteligente de Gestión Estudiantil), propone una solución digital que integra inteligencia artificial para registrar, analizar, predecir y mejorar el desempeño de los estudiantes de forma personalizada y accesible. Este proyecto presenta el desarrollo de SIGE-IA, una aplicación de escritorio con interfaz gráfica (Tkinter) que integra inteligencia artificial simbólica y aprendizaje automático para asistir tanto a estudiantes como docentes.

2. Objetivos del trabajo

Objetivo general:

Diseñar un sistema inteligente que permita registrar, analizar, predecir y mejorar el rendimiento estudiantil mediante técnicas de IA, considerando las restricciones sociales, económicas y ambientales.

- Desarrollar una solución de software que permita gestionar, analizar y mejorar el rendimiento académico de estudiantes.
- Aplicar técnicas de IA simbólica y Machine Learning para generar recomendaciones, predicciones y análisis.
- Respetar restricciones sociales (acceso según rol), ambientales (uso digital sin impresiones innecesarias) y económicas (software libre, multiplataforma).

Objetivos específicos:

- Identificar estudiantes con riesgo de fracaso académico.
- Generar recomendaciones personalizadas de mejora.
- Incorporar una evaluación emocional del estudiante.
- Predecir el estado académico con modelos de IA.

3. Formulación del problema e identificación de variables

Campo	Descripción
Nombre del sistema	SIGE-IA (Sistema Inteligente de Gestión Estudiantil con IA)
Usuarios	Docentes y estudiantes
Objetivo del sistema	Automatizar el análisis, seguimiento y mejora del rendimiento estudiantil
Variables de entrada	Nombre, materias y notas, promedio, respuestas emocionales, imagen OCR
Variables de salida	Estado final (Aprobado/Reprobado), medalla, recomendaciones, boletín PDF

Procesos asociados	Registro, evaluación, análisis, predicción, exportación de informes
Restricciones externas	Acceso por rol, lenguaje comprensible, consideraciones de privacidad
Limitaciones internas	Uso de Python, Tkinter, JSON para almacenamiento, OCR limitado por imagen

4. Planteamiento de alternativas de solución

Alternativa 1: Uso de Python con programación modular, archivos JSON y modelo IA de clasificación simple. Interfaz gráfica con Tkinter, Aplicación de consola con IA simbólica y OCR.

- Programación estructurada/modular
- Entrenamiento de modelo IA con sklearn
- Lectura de imagen con OCR para ingreso de datos
- Evaluación emocional con radio buttons
- Uso de JSON para almacenar datos
- Inferencias por reglas tipo IF-THEN
- OCR para leer notas desde imagen
- Módulo de predicción con scikit-learn

- **Alternativa 2:** Aplicación gráfica con interfaz Tkinter + IA
- Todo lo de la alternativa 1
- GUI amigable con acceso por rol
- Simulador de mejora, evaluación emocional, chat educativo IA
- Exportación de boletines y reportes desde interfaz

5. Análisis de alternativas

Criterio	Alternativa 1 (Consola)	Alternativa 2 (Tkinter + GUI)
Facilidad de uso	Baja	Alta
Curva de aprendizaje usuario	Media	Baja
Estética e interacción	Mínima	Alta (GUI intuitiva)
Modularidad y escalabilidad	Media	Alta
Integración de IA y gráficos	Limitada	Completa

Criterio	Alternativa 1	Alternativa 2
Facilidad de implementación	Alta	Media
Requerimientos técnicos	Básicos	Avanzados
Escalabilidad	Media	Alta
Claridad del código	Alta	Media
Tiempo de desarrollo	Bajo	Alto

a. ¿Cuál sería más fácil de implementar y por qué?

a. La más fácil de implementar es la alternativa 1, pero es menos amigable para el usuario

b. ¿Qué aspectos son importantes para una solución eficiente?

Son importantes: claridad del código, tiempo de ejecución, mantenibilidad y experiencia de usuario.

6. Selección de la mejor alternativa

Se elige **Alternativa 2** por su mayor alcance, facilidad de uso y mejor presentación visual. El uso de Tkinter permite separar secciones por rol, visualizar resultados, ofrecer interacción emocional y educativo (chat), y exportar PDF/TXT. Esto la hace más eficiente en términos de usabilidad y organización del código.

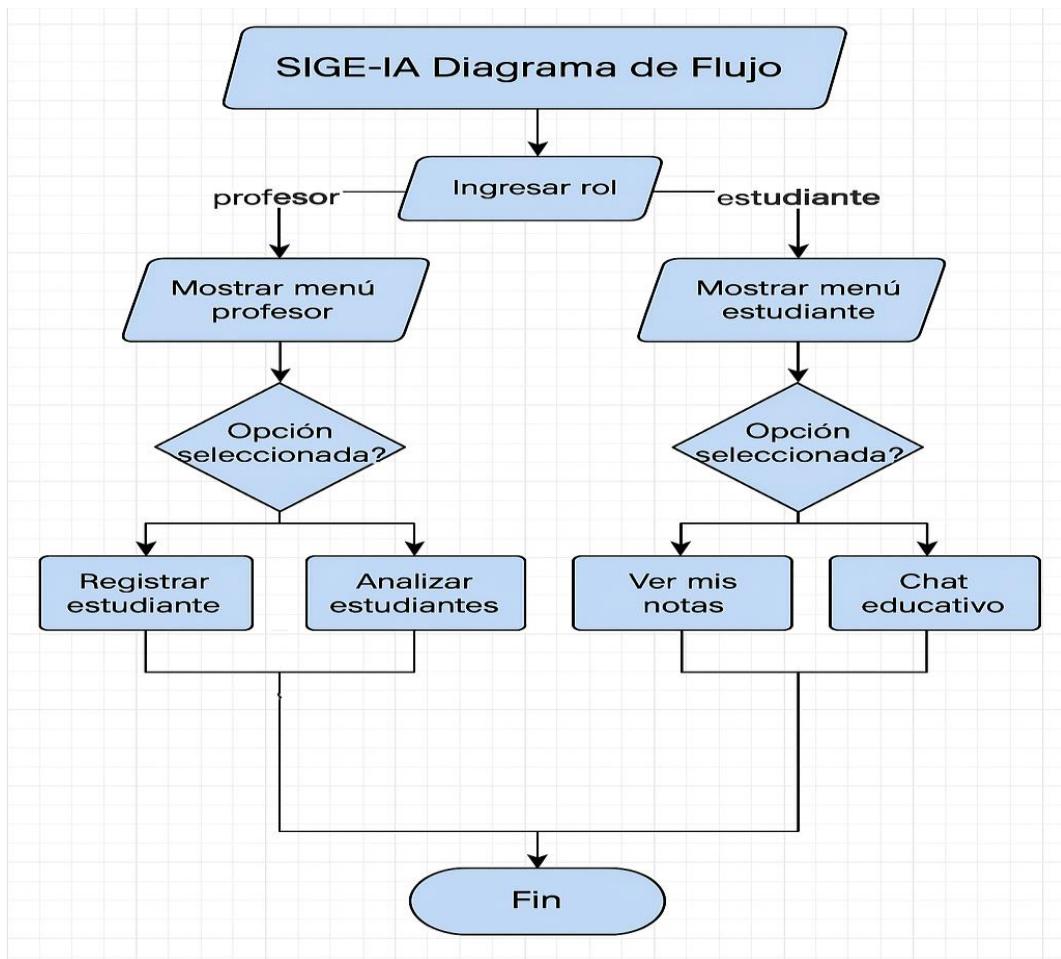
Pregunta: La eficiencia se garantiza mediante: uso de estructuras limpias, modularidad (archivos separados por lógica), uso de librerías óptimas como scikit-learn, fpdf, pytesseract, y validaciones de entrada.

¿Cómo garantizar que esta solución es la más eficiente?

- División del sistema en módulos reutilizables
- Pruebas de rendimiento constantes
- Documentación del código
- Control de errores e interfaces intuitivas

7. Desarrollo de la solución

1. Código en GitHub: [Enlace al repositorio cuando se suba]
https://github.com/jenn2005pardo/SIGE_IA



2. Capturas del código: se muestran y explican secciones clave como el sistema de predicción, recomendación y chat IA.
3. **Función registrar_estudiante_manual():**
 - a. Permite ingresar el nombre, promedio y las materias del estudiante manualmente desde la interfaz gráfica.
 - b. Utiliza cajas de entrada (Entry) y un Text multilinea para capturar datos.
 - c. Evalúa si el estudiante aprueba, y le asigna una medalla según rendimiento.
4. **Función ver_estudiantes():**
 - a. Muestra una lista ordenada alfabéticamente de los estudiantes.
 - b. Al hacer clic en un estudiante, se abre una ventana con su información, botones para evaluación emocional, generación de PDF y edición de notas.
5. **Función mostrar_emocional_personalizado():**
 - a. Abre un formulario con preguntas sobre motivación y bienestar del estudiante.
 - b. Captura respuestas mediante botones de selección (Radiobutton) y muestra un resumen personalizado.
6. **Función abrir_chat():**
 - a. Implementa un chat educativo con IA simbólica.

- b. Detecta palabras clave en los mensajes del usuario y responde con enlaces útiles, consejos o apoyo motivacional.

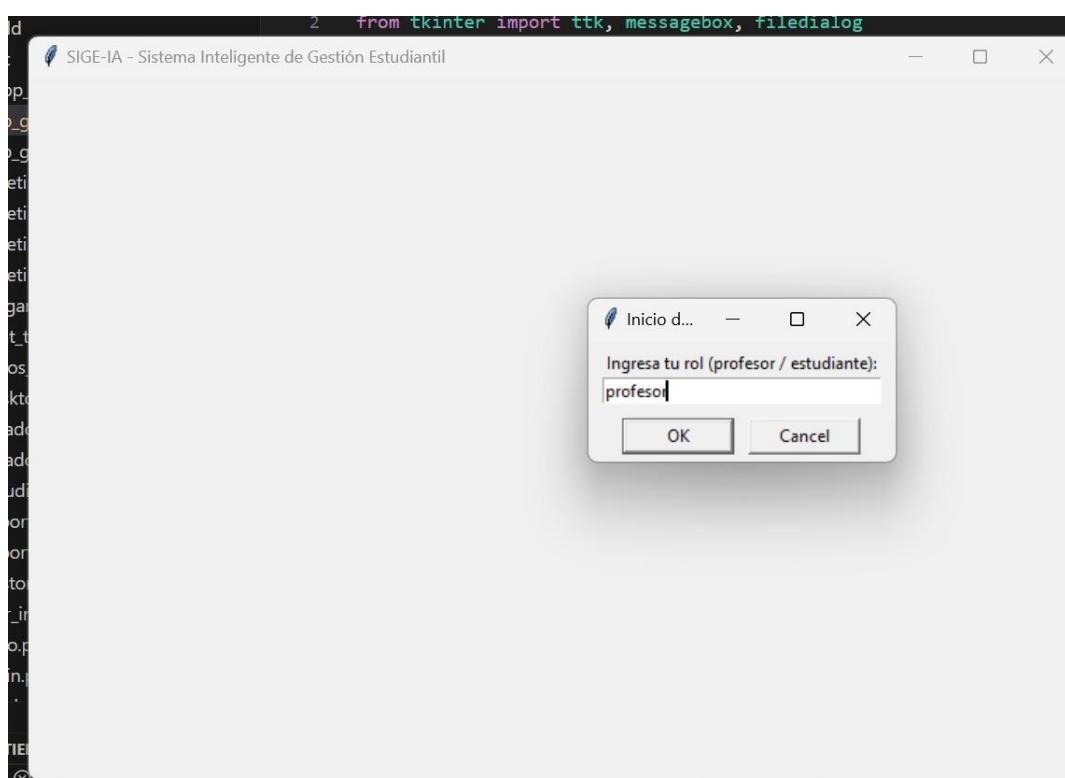
7. Control de roles (login_usuario()):

- a. Determina si el usuario es profesor o estudiante.
- b. En función de eso, bloquea o muestra funcionalidades específicas (ej. solo los profesores pueden exportar datos o reentrenar modelos).

8. Exportación de resultados:

- a. Funciones como exportar_estado() y generar_boletin_gui() permiten generar archivos .txt o .pdf.
- b. Incluye librerías como fpdf para crear documentos y filedialog para elegir rutas de guardado.

9. Capturas de ejecución: interfaz principal, ventana de registro, evaluación emocional, predicción, etc.



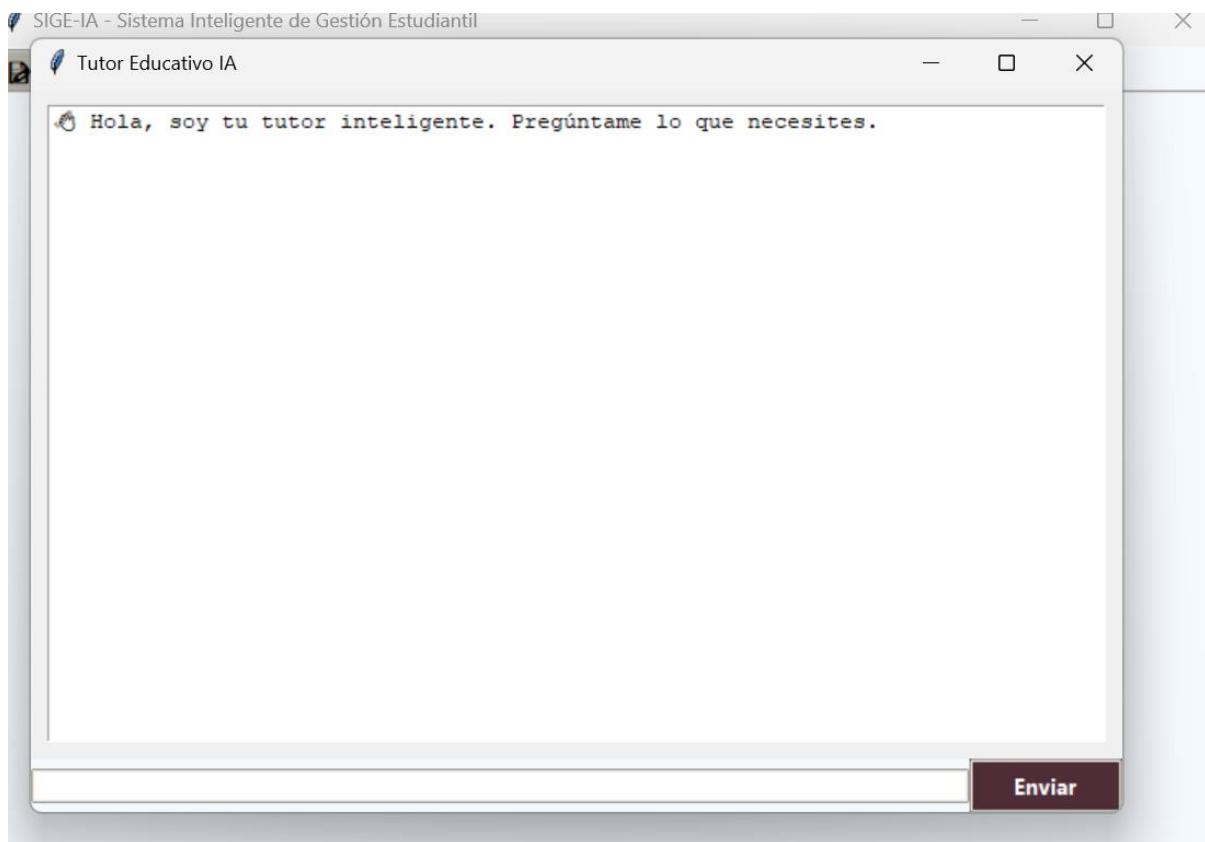
muestra un cuadro para ingresar el rol (profesor/estudiante). El comportamiento posterior depende de esa entrada.



contiene botones para registrar estudiantes manualmente u OCR. Al hacer clic, abre formularios detallados para ingresar información.



permite ver el estado final, generar boletines PDF y predecir aprobación usando IA. Cada botón abre una ventana interactiva.



muestra una ventana de chat que responde dinámicamente a preguntas frecuentes del estudiante.



lanza un simulador que sugiere mejoras académicas a.



acceso restringido donde se pueden exportar datos y reentrenar modelos IA.



se presenta en lista ordenada alfabéticamente. Al seleccionar un estudiante, se despliega su boletín, evaluación emocional y opciones de edición.

A screenshot of a terminal window with a dark background. At the top, there are tabs for PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL (which is currently selected), and PUERTOS. There are also icons for Code, +, and other terminal functions. The main area displays a numbered list of 18 options related to student management. The user has selected option 18, "Evaluar estado emocional". A prompt asks "¿Cómo te sientes hoy? (bien, regular, mal):" followed by a cursor. At the bottom of the terminal, there is some footer text: "jenn2005pardo (Hace 1 hora) Lín. 115, col. 1 Espacios: 4 UTF-8 CRLF () Python 3.13.5 64-bit".

muestra un formulario con preguntas, y al finalizar, despliega el resumen emocional.

8. Validación de la solución

Casos de prueba:

Caso	Entrada	Acción Realizada	Resultado Esperado	Resultado Obtenido	Estado
1	Estudiante con promedio 9.5 y sin materias reprobadas	Registro manual	Aprobado, medalla de Excelencia	Aprobado, medalla de Excelencia	<input checked="" type="checkbox"/>
2	Imagen con notas legibles (OCR)	Registro por imagen	Datos reconocidos y estudiante registrado	Datos correctos en sistema	<input checked="" type="checkbox"/>
3	Estudiante con promedio 6.8 y 2 materias < 7	Registro manual	Reprobado, sin medalla	Reprobado, sin medalla	<input checked="" type="checkbox"/>
4	Promedio 7.5 en IA predictiva	Predicción	Probabilidad de aprobar alta	Mensaje positivo	<input checked="" type="checkbox"/>
5	Estudiante desmotivado (evaluación emocional)	Formulario emocional	Resumen motivacional correcto	Resultado claro en pantalla	<input checked="" type="checkbox"/>
6	Exportar boletín de PDF	PDF generado	Archivo PDF con datos completos	PDF generado y visible	<input checked="" type="checkbox"/>

- Estudiante con promedio 10 y sin materias reprobadas → "Aprobado"
- Estudiante con promedio 6.5 y dos materias con nota <7 → "Reprobado"
- Imagen escaneada cargada por OCR con 5 materias y promedio 8 → registro exitoso
- Evaluación emocional con respuestas negativas → alerta emocional mostrada

9. Conclusiones y recomendaciones

- **Conclusiones:**

- El sistema SIGE-IA cumple satisfactoriamente con los objetivos académicos propuestos.
- La integración de IA simbólica y funcionalidad gráfica mejora la experiencia de usuario.
- El control de roles garantiza la seguridad de la información.
- Las funciones de análisis emocional y chat educativo aportan valor agregado en el aprendizaje.
- **Recomendaciones:**
- Implementar autenticación con contraseña en login futuro.
- Explorar nuevas funciones como tutor virtual más avanzado usando modelos NLP reales.
- Mejorar la exportación del PDF con firmas o gráficos visuales.
- Subir la aplicación a una plataforma como Streamlit o ejecutarla en la nube para acceso remoto.

Referencias (APA7):

- Scikit-learn. (s.f.). scikit-learn: Machine Learning in Python. <https://scikit-learn.org>
- Tkinter. (s.f.). Tkinter documentation. <https://docs.python.org/3/library/tkinter.html>
- Pytesseract. (s.f.). OCR in Python with Tesseract. <https://pypi.org/project/pytesseract/>
- Psicología y Mente. (s.f.). Consejos para superar la desmotivación. <https://www.psicologiaymente.com>
- W3Schools. (s.f.). Learn Python. <https://www.w3schools.com/python/>