



AGENTE DE ENSEÑANZA DE ESTADISTICA BASICA

DANIELA XIMENA NIÑO BARBOSA

DIEGO ALEXANDER GIRALDO TIRADO

JOSE MANUEL JIMENEZ GOMEZ

PROYECTO FINAL

RPOCESAMIENTO DEL LENGUAJE NATURAL

UNIVERSIDAD EAFIT

ESCUELA DE CIENCIAS APLICADAS E INGENIERÍA

MAESTRÍA EN CIENCIAS DE LOS DATOS Y LA ANALÍTICA

MEDELLÍN

2024

1. Descripción del Agente

Para muchos estudiantes universitarios, la materia de Estadística Básica y Probabilidad representa un gran desafío. La enseñanza en los salones de clase no se adapta al ritmo o estilo de aprendizaje de cada persona, y buscar ayuda fuera del aula puede a veces ser complicado por tiempo o dinero. En este contexto, se hace necesario un recurso educativo accesible, flexible y adaptado a las necesidades reales de cada estudiante.

Por esta razón se diseñó y desarrolló un agente educativo para acompañar a estudiantes universitarios de primeros semestres que están cursando la materia de Estadística Básica y Probabilidad. Su función principal es ayudarlos a entender los temas fundamentales de forma clara y personalizada, adaptándose al ritmo y necesidades de cada estudiante.

El agente cuenta con dos modalidades: libre y guiada. La modalidad libre busca simplemente contestar las preguntas de los estudiantes a manera de conversación, mientras que la forma guiada busca evaluar al estudiante para identificar sus fortalezas y debilidades, de modo que genere para el estudiante un plan de enseñanza personalizado enfocado en los temas que debe reforzar.

Los estudiantes pueden interactuar con este agente mediante una aplicación, donde pueden ir cambiando entre ambos modos e interactuar con el agente en forma de chat.

2. Flujo del Agente

El flujo del agente de enseñanza en términos generales se puede ver en la *Figura 1*, en esta sección se va a explicar el diagrama y además se va a incluir explicaciones más detalladas del flujo que no se encuentra en el diagrama.

El agente comienza recibiendo la selección del modo por parte del estudiante, el cual puede seleccionar el modo libre o guiado. En caso de que el estudiante elija el modo libre, el agente y el usuario van a interactuar a modo de chat, por lo cual, el agente va a esperar a que el usuario le haga una pregunta, posteriormente el agente va a responder y volverá a esperar a que el usuario le haga otra pregunta, repitiendo el ciclo indefinidamente.

En el caso contrario, donde el estudiante elige el modo guiado, el proceso es más complicado. Primero, el agente va a generar un examen de nivel, en este caso, existen tres niveles compuestos de la siguiente manera:

- El **nivel básico** está compuesto por los temas: fundamentos de la estadística, tipos y niveles de datos, diseño de estudios y muestreo, y representación gráfica de datos.
- El **nivel intermedio** está compuesto por los temas: medidas de tendencia central, medidas de dispersión, resumen de datos con percentiles y gráficos, e identificación de valores atípicos y análisis exploratorio.
- El **nivel avanzado** está compuesto por los temas: fundamentos de probabilidad, eventos compuestos e independencia, probabilidad condicional y teorema de Bayes, y distribuciones discretas y binomial.

El agente va a realizar cuatro preguntas por cada nivel y cada pregunta la va a calificar en un rango de 0 a 5, se va a considerar superado el nivel cuando el estudiante obtenga una calificación promedio por encima de 3. Si un usuario gana las preguntas para un nivel se procede a realizar las preguntas para el siguiente nivel, si el usuario pasa todos los niveles se considera el examen como concluido. Por otra parte, en caso de que el usuario falle las preguntas de algún nivel, no se van a realizar las preguntas del siguiente nivel, sino que se va a considerar el examen como concluido.

Al haber terminado el examen, el agente va a generar un feedback de las respuestas del estudiante. Posteriormente, si el estudiante pasó todos los niveles del examen, el agente va a redirigirlo al modo libre, ya que el estudiante tiene todos los conceptos claros y no necesita una guía porque ya domina todos los temas. En el caso donde el estudiante falló algún nivel, el agente le va a dar la posibilidad de repetir el examen o de no repetirlo y obtener su plan de estudios. El plan de estudios se va a generar teniendo en cuenta el resultado del examen, por lo que le va a mostrar al estudiante sus temas fuertes y débiles, y se va a enfocar en realizar un para abordar las debilidades.

Para finalizar, el agente va a explicarle al estudiante sus temas débiles, tema por tema. Al acabar la explicación, el estudiante repetirá el examen comenzando por el nivel donde quedó, es decir, si para la primera vez que realizó el examen superó el nivel básico y falló el intermedio, para la segunda vez que lo realice no debe volver a presentar el nivel básico, sino que comienza por el nivel intermedio.

Cabe aclarar que desde la interfaz el estudiante puede cambiar en cualquier momento entre el modo libre y guiado. Por ejemplo, si el usuario tiene alguna duda en la explicación tema por tema del modo guiado, puede pasarse al modo libre para hacer una pregunta, y luego volver nuevamente al modo guiado para continuar con la explicación del siguiente tema.

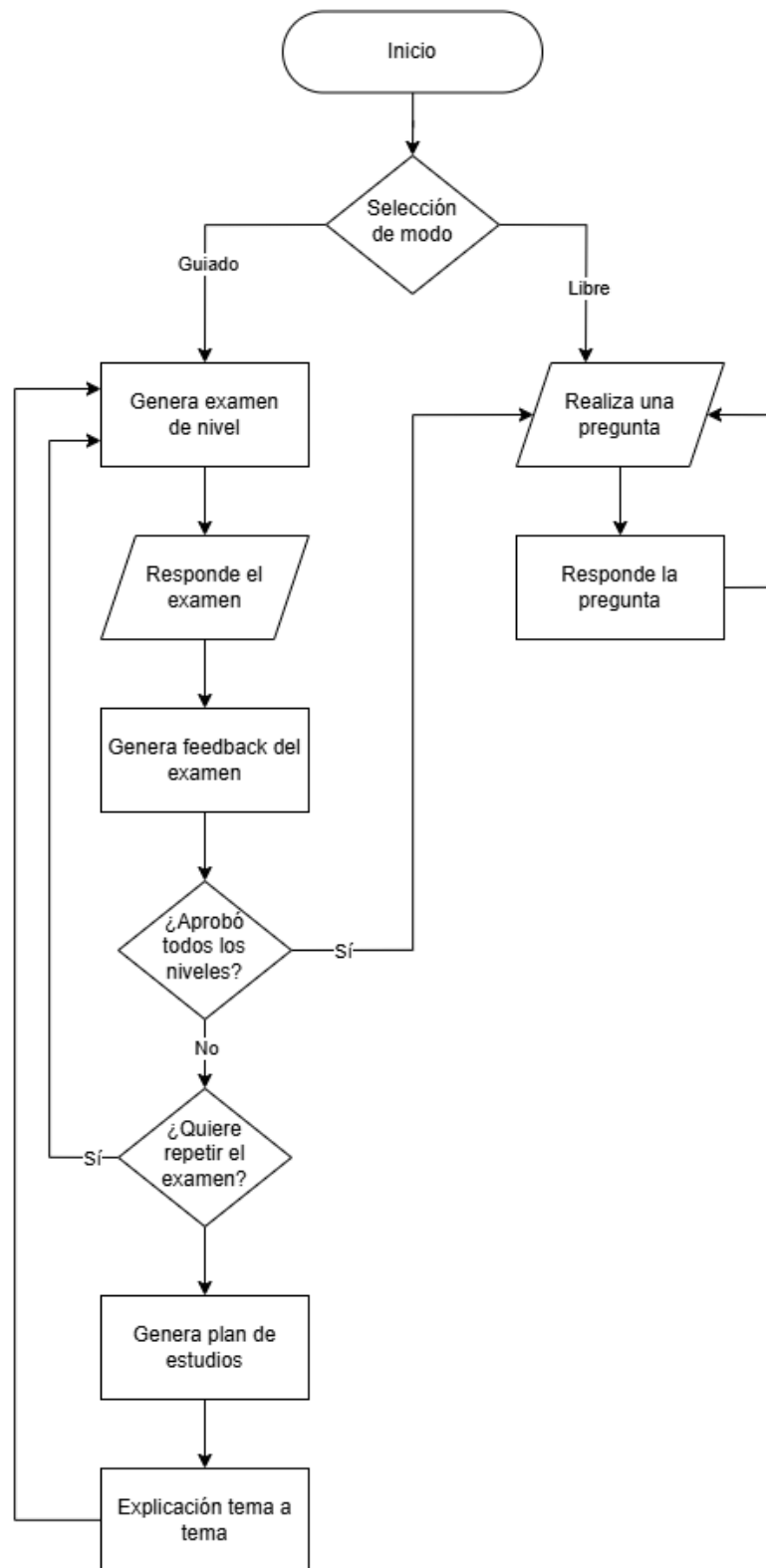


Figura 1. Diagrama de flujo del agente

3. Arquitectura del agente

En la *Figura 2* se encuentra la arquitectura del agente de enseñanza, donde se pueden ver los diferentes módulos de la arquitectura y las herramientas con las que se desarrollaron.

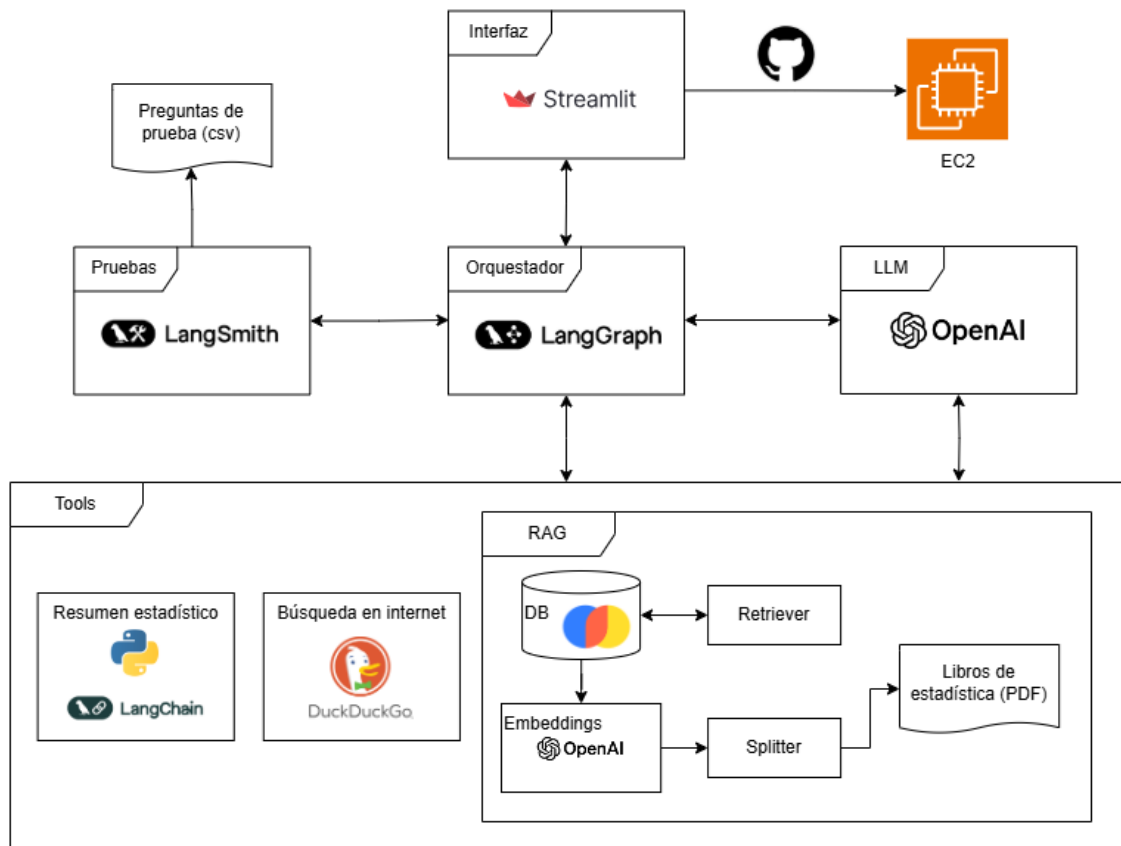


Figura 2. Arquitectura del agente

Estas son las partes de la arquitectura:

- El **orquestador** del agente se implementó con **LangGraph**, este sirve para indicarle al agente cómo actuar según la tarea que vaya a realizar o según las instrucciones que le dé el usuario.
- El **LLM** usado para generación de texto fue el modelo **GPT 4.1 de OpenAI**.
- Para las **tools** se implementaron tres:
 - Se implementó una **tool personalizada** mediante una función en **Python** y se convirtió en una tool usando **LangChain**. Esta tool permite al estudiante ingresar una lista de números y genera como output un

resumen estadístico de ese conjunto de datos. Por ejemplo, si el usuario ingresa los valores 1, 2, 3, 4, el agente le va a devolver: count = 4, mean=5, stdev=1.118, min=1, max=4, Q1=1.5, Q2=2.5, Q3=3.5.

- También se incluyó la tool **DuckDuckGo**, la cual le permite al agente obtener información de internet, con el fin de ampliar las posibilidades de respuesta del agente. Por ejemplo, si el estudiante pregunta por las estadísticas de un partido de futbol reciente, el agente puede acceder a internet para encontrar la información del partido.
- Se implementó **RAG** para ampliar el conocimiento del agente a partir de libros universitarios de Estadística y Probabilidad en formato PDF. Se separó los libros en chunks, se usaron los **embeddings de OpenAI**, se almacenó la información con **ChromaDB** y se generó un retriever, el cual es el que devuelve información relevante de los libros al LLM. Los libros usados fueron:
 - Estadística para administración y economía, escrito por David R. Anderson, Dennis J. Sweeney, y Thomas A. Williams. Se incluyó del capítulo 1 al 7.
 - Estadística, escrito por Mario F. Triola. Se incluyó del capítulo 1 al 5.
- La **evaluación** del agente se realizó utilizando **LangSmith**, específicamente, su API para Python langsmith.evaluation y la librería QAEvalChain. Para garantizar la calidad del agente educativo desarrollado, se implementó un proceso de evaluación estructurado sobre dos funcionalidades críticas del sistema:
 - Nodo de preguntas libres: encargado de responder preguntas entregadas por el usuario.
 - Nodo de feedback: diseñado para evaluar el examen de desempeño del estudiante y entregar retroalimentación a partir de los resultados.

Para evaluar el agente se elaboraron dos datasets:

- Un dataset con preguntas y respuestas esperadas para el nodo de preguntas libres.
- Otro dataset con respuestas de estudiantes y retroalimentación esperada para el nodo de feedback.

Ambos nodos fueron evaluados automáticamente utilizando la función run_on_dataset() de LangSmith, que ejecuta las cadenas y compara los resultados generados con las respuestas esperadas.

- La **interfaz gráfica** para permitir la fácil interacción entre el agente y el estudiante se realizó usando **Streamlit**.
- Finalmente se hizo un **despliegue** de la aplicación usando el servicio **EC2** de **Amazon Web Services** y usando un pipeline de continuous delivery (CD) con **GitHub Actions**, para que cualquier usuario pueda usar la aplicación. Hay que tener en cuenta que en el momento se usó AWS Academy, por lo que la máquina se apaga cada cuatro horas.

4. Resultados y oportunidades de mejora

El agente realizado es un MVP de lo que sería un agente abierto a todo público en la web. Se logró desarrollar un agente funcional, con habilidad para expandir su conocimiento de diferentes fuentes, con la capacidad de personalizar la experiencia de cada usuario, y con una interfaz de fácil y cómoda interacción.

El proceso de evaluación evidenció un desempeño sólido por parte del agente en ambos nodos evaluados:

- **Nodo de preguntas libres:** Este componente mostró un rendimiento satisfactorio tanto en preguntas abiertas como cerradas. La mayoría de las respuestas fueron correctas, destacando la capacidad del agente para manejar conceptos fundamentales de estadística. Sin embargo, las respuestas incorrectas se concentraron principalmente en preguntas que exigían una interpretación más profunda o contextual, lo cual sugiere que el modelo puede encontrar posibilidades de mejora en la comprensión de respuestas no tan completas.
- **Nodo de feedback:** Este nodo demostró ser especialmente valioso para evaluar la comprensión del estudiante, generando en su mayoría comentarios útiles y pertinentes. Sin embargo, una parte de las retroalimentaciones fue considerada inadecuada, debido a su carácter excesivamente general o por no abordar con precisión el error cometido en la respuesta. Esto evidencia la necesidad de mejorar la especificidad, claridad y utilidad de la retroalimentación generada.

En [este](#) link se puede encontrar un demo de la interacción entre un usuario y el agente, y en [este](#) link se puede acceder al agente.

Por último, estos son algunos de los cambios o extensiones que se le podrían hacer al agente para mejorarlo en una futura implementación:

- Permitir al estudiante aceptar o rechazar el plan académico de estudio propuesto. En caso de que sea rechazado, el estudiante debería poder darle al agente una indicación de cómo mejorar el plan.
- Integrar ejercicios prácticos para complementar la explicación de los temas, de modo que la explicación tema a tema se vuelva más dinámica para el estudiante.
- Permitir al usuario ingresar documentos (PDF, imágenes, Excel, etc.) para que el agente los interprete y pueda contestar preguntas a partir de ellos.
- Mejorar la interfaz gráfica para que se vea más profesional y hacer más evaluaciones de desempeño del agente.
- Integrar más fuentes de conocimiento al RAG y agregarle más tools al agente.
- Darle acceso al agente a APIs de datasets para que este pueda generar mejores ejercicios para el estudiante y darle mejores recursos de estudio.