**Requerimientos funcionales del proyecto de tutor inteligente:**

1. Registro e inicio de sesión:

* El sistema debe permitir el inicio de sesión y la gestión de contraseñas.
* El sistema debe ofrecer perfiles diferenciados para estudiantes y docentes.

| Nombre o identificador | 1 - Inicio de sesión | | |
| --- | --- | --- | --- |
| Resumen | El usuario debe poder iniciar sesión con su usuario y contraseña | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| user | String | El usuario debe estar registrado en el entorno de la universidad Icesi. |
| password | String | El usuario debe estar registrado en el entorno de la universidad Icesi. |
| Actividades generales necesarias para obtener los resultados | El usuario debe ingresar los datos solicitados | | |
| Resultado o postcondición | Se inicia sesión en el perfil del usuario y se accede al chatbot con el perfil selecionado. | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
|  |  |  |

1. Gestión de perfiles:

* El sistema debe ofrecer perfiles diferenciados para estudiantes y docentes.
* Los perfiles de estudiantes deben registrar el progreso, los ejercicios resueltos y las áreas de dificultad.
* Los perfiles de docentes deben permitir la visualización del avance de los estudiantes y la generación de reportes.

| Nombre o identificador | 2 - Gestión de conversaciones individuales | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe poder permitir a los estudiantes de la universidad el iniciar sesión para ver el chatbot y poder acceder a sus conversaciones pasadas. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Consultar la base de datos y traer las conversaciones más recientes | | |
| Resultado o postcondición | Resumen de las conversaciones pasados. | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| Resumen |  |  |

| Nombre o identificador | 2.1 - Gestión de conversaciones de docente | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe permitir a los docentes de la universidad iniciar sesión y ver los perfiles de sus estudiantes con las principales debilidades, preguntas y métricas indicadas por el chatbot. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | Consultar la base de datos y traer información relevante al usuario, por ejemplo los cursos del docente. | | |
| Resultado o postcondición | Despliega menu con lista de conversaciones | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| Lista de estudiantes | Array de Objetos | [{ id: int, nombre: string, perfil: objeto }] |
| Metricas de Estudiante | Objeto JSON | { debilidades: array, preguntas: array, métricas: objeto } |
|  |  |  |

1. Asistencia personalizada:

El sistema debe desplegar un chatbot basado en una IA externa para asistir tanto a docentes como a estudiantes en diferentes áreas:

* **Docentes:** Podrán iniciar sesión y retroalimentar el chatbot para mejora
* **Estudiantes:** Podrán interactuar con el chatbot para recibir asistencia en:
  + **Errores de código:** Identificación de errores de sintaxis y lógica en programas desarrollados.
  + **Evaluación de código:** Compilación y ejecución de fragmentos de código en Java, proporcionando retroalimentación inmediata sobre la corrección y oportunidades de optimización.
  + **Asistencia con conceptos clave:** Explicaciones sobre programación orientada a objetos, ciclos, funciones y polimorfismo, con ejemplos prácticos en Java.

El chatbot no proporcionará respuestas completas, sino que ofrecerá orientación y explicaciones para que los estudiantes lleguen a la solución por sí mismos.

| Nombre o identificador | Chatbot de asistencia personalizada. | | |
| --- | --- | --- | --- |
| Resumen | El sistema debe desplegar un chatbot proveído por una IA para resolver dudas del estudiante respecto a código, conceptos o lógica. Este chatbot no debe presentar la respuesta completa sino una retroalimentación con explicación para que el estudiante llegue a la solución. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
| user\_prompt | String | El usuario debe estar registrado en el entorno de la universidad Icesi. |
| codigo\_fuente | String | |  |  | | --- | --- | |  | En caso de evaluación de código, el usuario debe proporcionar un fragmento de código en Java. | |
| Actividades generales necesarias para obtener los resultados | 1. El usuario (docente o estudiante) debe iniciar sesión en el sistema. 2. El sistema recibe la solicitud del usuario (pregunta sobre código, evaluación de código o concepto clave) y la envía al chatbot. 3. El chatbot procesa la solicitud y devuelve una respuesta con explicaciones, ejemplos y recomendaciones. 4. El sistema presenta la respuesta al usuario en la interfaz correspondiente. | | |
| Resultado o postcondición | El chatbot responde con una explicación sobre cómo llegar a la solución de su problema de código o concepto teórico. | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| chatbot\_response | String | Mensaje de respuesta de chatbot |

1. Seguimiento del progreso:

* El sistema debe registrar el progreso individual, identificando temas dominados y áreas de mejora.
* El sistema debe permitir la generación de reportes automáticos sobre el rendimiento del estudiante, accesibles para docentes.
* El sistema debe recolectar métricas de desempeño antes y después del uso del tutor para evaluar su impacto en el aprendizaje.

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre o identificador | Seguimiento personalizado del progreso | | |
| Resumen | El sistema debe aprender de las conversaciones tratadas con los estudiantes para perfilarlos de acuerdo a sus debilidades y así poder identificar sus puntos débiles para notificar al profesor a cargo y así poder reforzar al estudiante donde lo necesita. | | |
| Entradas | Nombre entrada | Tipo de dato | Condición de selección o repetición |
|  |  |  |
| Actividades generales necesarias para obtener los resultados | El curso debe tener alumnos que hayan interactuado con el chatbot  (añadir actividades necesarias) | | |
| Resultado o postcondición | Actualizar debilidades grupales e individuales | | |
| Salidas | Nombre salida | Tipo de dato | Formato |
| Metricas(editar) |  |  |

**Requerimientos no funcionales.**

**RNF 1: Rendimiento**

* El Sistema debe responder a las consultas en un tiempo inferior a los 10 segundos en condiciones normales de uso]
* Debe soportar al menos 75 usuarios concurrentes sin verse considerablemente afectado.

**RNF 2: Seguridad**

* El acceso a los datos debe requerir autenticación respectiva al rol asignado.
* El sistema debe implementar un filtrado de inputs para prevenir ataques contra la base de datos
* El sistema cumplirá con normas de privacidad y protección de datos vigentes (Ley 1581 de 2012 y Decreto 1377 de 2013). Política de tratamiento de datos de ICESI

**RNF 3: Disponibilidad**

* El sistema tendrá una disponibilidad mínima del 99% con monitoreo constante.
* En caso de falla, el sistema deberá recuperarse en un plazo de un día hábil o menos.

**RNF 4: Mantenibilidad**

* El código seguirá estándares SOLID y Clean Code para facilitar su mantenimiento.
* El sistema usara logs centralizados para su monitoreo.

**RNF 5: Restricciones Técnicas**

* El backend se debe desarrollar en Python usando Django Framework
* La base de datos se hará en Postgres SQL
* El despliegue se realizara en Render

RNF 6: IA

* Uso de Chatgpt API para la ia.
* El chatbot debe tener un tiempo de respuesta adecuado
* La IA debe manejar una tasa de error mínima una vez cumpla cierto números de mensajes para el buen funcionamiento del chat
* La precisión debe ser máxima debido al uso exclusivo de material de la universidad