**Requerimientos Funcionales**

1. **Registro de Respuestas:**
   * El sistema debe permitir al usuario ingresar respuestas a preguntas de entrevista.
2. **Evaluación de Respuestas con IA:**
   * El sistema debe enviar las respuestas del usuario al backend.
   * El backend debe construir un prompt y enviarlo a la API de OpenAI.
   * La API debe retornar una evaluación en lenguaje natural.
3. **Visualización del Resultado:**
   * El sistema debe mostrar al usuario una retroalimentación corta con una recomendación final (ej: "Listo para el puesto" o "Debe mejorar...").
4. **Gestión del Flujo de Preguntas:**
   * El sistema debe presentar al menos 3 preguntas estándar de entrevista al usuario.
5. **Interfaz de Usuario Amigable:**
   * El frontend debe mostrar campos claros para escribir las respuestas y un botón para enviar.
6. **Manejo de Estados:**
   * El sistema debe mostrar mensajes como "Analizando..." mientras se procesa la solicitud.
7. **Consumo de API con Seguridad:**
   * El backend debe incluir una clave de API segura para comunicarse con OpenAI.

**Requerimientos No Funcionales**

1. **Rendimiento:**
   * El sistema debe procesar la evaluación en menos de 5 segundos en condiciones normales.
2. **Escalabilidad:**
   * El sistema debe ser modular y permitir agregar más preguntas o ajustar el modelo de IA fácilmente.
3. **Portabilidad:**
   * El sistema debe poder ejecutarse tanto en entornos locales como en servidores cloud.
4. **Seguridad:**
   * La clave de la API de OpenAI no debe estar expuesta en el frontend.
   * El backend debe validar el contenido de las respuestas antes de enviarlas a la IA.
5. **Mantenibilidad:**
   * El código debe estar dividido en capas (controller, service, model) y seguir buenas prácticas de arquitectura.
6. **Compatibilidad:**
   * El frontend debe ser accesible desde los navegadores modernos (Chrome, Firefox, Edge).
7. **Interfaz Intuitiva:**
   * El usuario no debe necesitar conocimientos técnicos para interactuar con el bot.

Principio del formulario

Final del formulario