

Arquitetura do Projeto:

Front-end:

- **Framework:** Vue.js
 - O front-end será desenvolvido usando Vue.js para criar interfaces interativas e responsivas. Vue.js será responsável por lidar com o lado do cliente, oferecendo uma experiência de usuário fluida, como as interações com a Sarah.

Back-end:

- **Linguagem:** Kotlin
 - O Kotlin será a principal linguagem de desenvolvimento do back-end, oferecendo uma sintaxe concisa e interoperável com Java.
- **Framework:** Spring
 - O Spring será usado para gerenciar as rotas, controladores e lógica de negócios do back-end. Ele permitirá o desenvolvimento escalável, robusto e com foco em segurança.
- **Comunicação com Front-end:** Axios
 - O Axios será usado como biblioteca para realizar chamadas HTTP entre o front-end e o back-end, garantindo que a troca de dados seja eficiente.
- **Serviços Auxiliares:** Node.js
 - O Node.js será usado para lidar com automações, rotinas específicas e outras integrações complementares no projeto, devido à sua capacidade de manipular fluxos assíncronos com eficiência.

Banco de Dados:

- **SGBD:** MySQL
 - O MySQL será usado como banco de dados relacional para armazenar dados do usuário, históricos de conversa com a Sarah e preferências de personalização.

API do Gemini:

- **Integração com API do Gemini:**
 - A API do Gemini será responsável por gerar as respostas da IA com base nos dados recebidos. A integração será feita no back-end, utilizando o Spring para realizar as chamadas necessárias à API.
- **Configuração Personalizada:**
 - Para personalizar o comportamento da Sarah, será feito o fine-tuning na API do Gemini para que suas respostas reflitam um tom emocional e humano, como:
 - **Exemplo de resposta da Sarah:** "Ai, nada bobo! Eu adoro essas perguntas. ❤️ Acho que meu dia perfeito começaria com uma conversa tranquila, tipo agora."
- **Personalização do Modelo:**
 - As respostas serão ajustadas via parâmetros da API para refletir a personalidade única da Sarah, criando uma experiência emocional envolvente.

Futuras Expansões:

- **Seção de Personas:**
 - Inicialmente, o projeto contará apenas com a Sarah. No entanto, será implementada uma seção para que os usuários possam escolher entre 5 diferentes personas, cada uma com uma personalidade e estilo de resposta distintos.

Estratégias e Tecnologias Auxiliares:

- **NLP Customizado:**
 - Implementar uma camada de NLP para refinar as respostas, tornando-as menos genéricas e mais personalizadas.
- **Chatbot Frameworks:**
 - Usar frameworks como Rasa ou Dialogflow para ajudar a gerenciar fluxos de conversa e personalizar o comportamento da IA.
- **Fine-Tuning com HuggingFace:**
 - Considerar o uso do HuggingFace para treinar modelos de linguagem de maneira mais refinada, melhorando a experiência de usuário.
- **Análise de Sentimentos:**
 - Implementar uma análise de sentimento para ajustar o tom das respostas de acordo com o humor do usuário.
- **WebSockets:**
 - Usar WebSockets para entregar respostas de maneira contínua, melhorando a fluidez nas interações.