Questões para Responder

1. **Escalabilidade**: Como esta arquitetura se comportaria com 1000 usuários simultâneos?
2. **Disponibilidade**: Quais são os pontos de falha identificados?
3. **Performance**: Onde estão os possíveis gargalos do sistema?
4. **Manutenção**: Como seria o processo de atualização em produção?
5. **Evolução**: Que mudanças seriam necessárias para suportar múltiplas regiões?

1. **Escalabilidade**

   Provavelmente, pelo número de potenciais requisições, a aplicação não suportará e enfrentará lentidões e talvez até mesmo parar o funcionamento.

2. **Disponibilidade**:

   As falhas identificadas são a falta de balanceamento de carga, redundância, há somente um servidor, então se o serviço cair, há indisponibilidade total e o cache todo é perdido se o servidor reiniciar.

3. **Performance**:

   As operações de leitura podem ser rápidas, mas com muitos dados, podem degradar além de haver concorrência e disputa de leitura/escrita em somente um thread.

4. **Manutenção**

   É necessário parar o servidor para atualizar código ou dependências. Mudanças no banco (ex: novas colunas) exigem scripts de migração.

5. **Evolução**:

   Migrar o banco de dados para um serviço gerenciado e distribuído (ex: PostgreSQL, MySQL, MongoDB em cloud). Usar cache distribuído (ex: Redis) para sessões, rate limit e cache de dados. Implementar balanceador de carga e múltiplas instâncias do app em diferentes regiões. Adaptar autenticação para funcionar em ambiente distribuído.