

Checklist de System Design

Um roteiro para estruturar respostas em entrevistas e revisões de arquitetura. Use as caixas durante a preparação e registre hipóteses, decisões e pendências.

Problema	Data	Tempo disponível

1. Enquadrar o problema

Alinhe o que será projetado antes de escolher tecnologias.

- ☐ Resumi o objetivo do sistema em uma frase e confirmei o entendimento.
- ☐ Identifiquei os atores principais e os fluxos críticos de uso.
- ☐ Separei requisitos funcionais de requisitos não funcionais.
- ☐ Perguntei sobre escala: usuários, requisições, volume de dados e crescimento.
- ☐ Registrei restrições relevantes: latência, disponibilidade, consistência, custo e região.
- ☐ Declarei as hipóteses que precisei adotar e deixei claro o que ficou fora do escopo.

2. Estimar capacidade

Use aproximações simples para orientar decisões de arquitetura.

- ☐ Calculei a ordem de grandeza de requisições por segundo e o pico esperado.
- ☐ Estimeei leitura versus escrita e identifiquei o caminho predominante.
- ☐ Estimeei armazenamento por dia ou por ano, incluindo retenção e replicação.
- ☐ Considerei tamanho médio de payload e largura de banda quando isso muda o desenho.
- ☐ Expliquei quais estimativas influenciaram escolhas concretas.

Desenhe o caminho principal primeiro

Mostre os blocos essenciais, conecte o fluxo ponta a ponta e só então aprofunde os pontos críticos.

3. Definir contratos e dados

Modele as interfaces que sustentam os fluxos principais.

- ☐ Defini as operações ou endpoints mais importantes e seus dados de entrada e saída.
- ☐ Escolhi identificadores, chaves de idempotência e estratégia de paginação quando necessário.
- ☐ Modelei as entidades essenciais e expliquei os relacionamentos.
- ☐ Justifiquei SQL, NoSQL ou uma combinação com base nos padrões de acesso.
- ☐ Considerei índices, particionamento, replicação e retenção de dados.
- ☐ Expliquei onde a consistência forte é necessária e onde a consistência eventual é aceitável.

4. Montar a arquitetura

Conecte cliente, entrada, serviços, dados e processamento assíncrono.

- ☐ Desenhei o fluxo principal de leitura e escrita de ponta a ponta.
- ☐ Posicionei balanceamento, gateway, CDN e cache apenas onde resolvem um requisito.
- ☐ Defini responsabilidades claras para serviços e evitei divisões sem benefício.
- ☐ Usei filas ou eventos quando desacoplamento, absorção de picos ou retry são necessários.
- ☐ Considerei cache invalidation, TTL, cache stampede e fonte de verdade.
- ☐ Expliquei como o desenho se comporta no pico e qual componente limita a escala.

Fechando com falhas, trade-offs e evolução

Uma boa resposta torna explícito o que pode falhar, como recuperar e quais decisões mudariam com novos requisitos.

5. Projetar para falhas

Trate falhas como parte do fluxo normal do sistema.

- ☐ Identifiquei pontos únicos de falha e as dependências mais críticas.
- ☐ Defini timeout, retry com backoff e limites para evitar efeito cascata.
- ☐ Considerei duplicidade, ordenação, idempotência e dead-letter queue em fluxos assíncronos.
- ☐ Expliquei recuperação, replicação, failover e perda de dados aceitável.
- ☐ Incluí logs, métricas, traces e alertas ligados aos objetivos do sistema.
- ☐ Considerei autenticação, autorização, criptografia, abuso e proteção de dados.

6. Concluir a resposta

Faça uma síntese clara em vez de terminar no último detalhe técnico.

- ☐ Recapitulei o fluxo, os componentes e as principais decisões.
- ☐ Nomeei pelo menos dois trade-offs e as alternativas descartadas.
- ☐ Expliquei como a arquitetura evolui se a escala ou os requisitos mudarem.
- ☐ Validei se todos os requisitos prioritários foram atendidos.
- ☐ Reservei tempo para perguntas e aprofundei o ponto escolhido pelo entrevistador.

Pergunta de fechamento

“Qual parte desta solução você gostaria que eu aprofundasse?”

Este checklist ajuda a organizar a conversa. Adapte a profundidade ao problema e ao tempo disponível.