



Simulação Comportamental

GOK | Inovação Digital — Pessoa Desenvolvedora Android Sênior

Como usar este material: Cada pergunta traz o que o entrevistador realmente avalia, os erros que eliminam candidatos e uma chave de resposta estruturada. Leia com atenção e pratique em voz alta — não para memorizar, mas para internalizar a estrutura.

Visão Geral das Perguntas

#	Pergunta	Nível
1	Trajetória e interesse na GOK	Aquecimento
2	Refatoração em produção sem interrupção	Intermediária
3	Trade-off entre velocidade e qualidade	Intermediária
4	Decisão arquitetural que faria diferente	Difícil
5	Plano de ação em código legado crítico	Difícil
6	Vulnerabilidade de segurança às vésperas de release	Difícil
7	Code review com desenvolvedor mais sênior	Difícil
8	Discordância técnica com decisão mantida	Difícil

Pergunta 1 — Aquecimento

"Me conta um pouco sobre sua trajetória técnica e o que te trouxe até aqui. Por que você está interessado nessa posição na GOK?"

O que o entrevistador avalia

Narrativa profissional coesa. Capacidade de conectar trajetória e vaga de forma natural — sem parecer decorado. Também serve como calibração inicial de autoconsciência e posicionamento.

Erros que eliminam candidatos

- ☐ Recitar o currículo em ordem cronológica sem filtrar o que importa para *esta* vaga
- ☐ Usar "estou procurando novos desafios" como motivação principal — é genérico e evasivo
- ☐ Não mencionar os contextos financeiros críticos que são o fio condutor da trajetória
- ☐ Falar mais de 4 minutos sem ser perguntado algo específico
- ☐ Não fazer nenhuma conexão entre experiência própria e o que a GOK faz

Sinais de uma resposta forte

- ☐ Estrutura clara: de onde veio → o que construiu → por que a GOK faz sentido agora
- ☐ Citar naturalmente os contextos críticos (Monnos, Banco Inter, Banco Master) como parte da identidade técnica
- ☐ Demonstrar que pesquisou a GOK — mencionar algo específico sobre o tipo de projeto ou contexto da empresa
- ☐ Finalizar com uma pergunta ou observação que abre diálogo

Chave de Resposta

A resposta ideal dura **entre 2 e 3 minutos** e segue uma estrutura de três atos.

Ato 1 — Âncora: onde está a identidade técnica

"Minha trajetória é quase inteiramente em Android com foco em ambientes financeiros críticos. Fui Head Android no Monnos Crypto Bank por quase cinco anos, onde conduzi as principais decisões arquiteturais do app — desde a escolha de stack até refatorações em produção sem interrupção de serviço. Depois disso, trabalhei em dois SDKs: um de pagamentos para o Banco Inter e um de segurança mobile para o Banco Master, envolvendo proteção contra engenharia reversa e criptografia de dados sensíveis."

Ato 2 — Contexto atual: por que a mudança faz sentido

"O que me motiva agora é um ambiente onde eu possa continuar operando com essa profundidade técnica, mas com mais variedade de contextos e desafios. Gosto de lugares onde qualidade de engenharia é levada a sério — não apenas entrega de features."

Ato 3 — Conexão com a GOK: por que esta empresa, não qualquer outra

"A GOK me chamou atenção por atuar em projetos de alta criticidade transacional com exigência real de performance e segurança. Esse é exatamente o ambiente em que trabalho melhor."

Pergunta 2 — Intermediária

"Me conte sobre uma situação em que você precisou conduzir uma refatoração técnica significativa em um sistema que estava em produção, sem poder parar o serviço. Como você planejou, executou e garantiu que nada quebraria?"

O que o entrevistador avalia

Maturidade técnica real em ambientes de produção. Capacidade de planejamento incremental, gestão de risco e ownership sobre decisões com consequência direta ao usuário final. A GOK opera em ambientes onde falha não é aceitável.

Erros que eliminam candidatos

- ❑ Descrever uma refatoração feita em ambiente de desenvolvimento ou sem usuários reais
- ❑ Não mencionar estratégia de rollback ou plano de contingência
- ❑ Apresentar a solução como individual, ignorando a coordenação com time e backend
- ❑ Ser vago sobre o risco técnico — “foi refatorando aos poucos” sem detalhar *como*
- ❑ Não ter resultado mensurável

Sinais de uma resposta forte

- ❑ Mencionar abordagem incremental: strangler fig pattern, feature flags, módulos isolados
- ❑ Descrever como código antigo e novo coexistiram temporariamente
- ❑ Falar sobre testes de regressão e cobertura antes de iniciar a migração
- ❑ Citar comunicação proativa com stakeholders sobre riscos e timeline
- ❑ Mencionar monitoramento pós-deploy: Crashlytics, logs, métricas de estabilidade

Chave de Resposta

Use o formato **STAR** com ênfase no *como*, não apenas no *o que*.

Etapa	O que cobrir
Situação	Qual era o app, qual era o problema técnico, qual era o risco de parar o serviço
Tarefa	Qual era a sua responsabilidade direta — não a do time, a sua
Ação	Como mapeou dependências, qual foi a estratégia incremental, como criou ou usou testes para garantir paridade de comportamento, como coordenou com backend, QA e produto

Etapa	O que cobrir
Resultado	O que mudou objetivamente — estabilidade, tempo de build, velocidade de entrega, redução de crashes

Dica estratégica: A Monnos é o cenário natural para essa resposta. Quase cinco anos de liderança técnica em um crypto bank inevitavelmente envolveram refatorações em produção. Parta dali — com detalhes reais e específicos.

Pergunta 3 — Intermediária

“Você já trabalhou em projetos com pressão explícita por velocidade de entrega e, ao mesmo tempo, expectativa alta de qualidade. Como você navega esse trade-off na prática — não em teoria?”

O que o entrevistador avalia

Pragmatismo com responsabilidade. A GOK atende projetos críticos em contexto consultivo — o candidato vai enfrentar pressão de cliente e pressão técnica simultaneamente. O entrevistador quer saber se você sabe tomar decisões difíceis, comunicar riscos e defender qualidade sem ser inflexível ou ingênuo.

Erros que eliminam candidatos

- ❑ Resposta purista: *“nunca aceito débito técnico”* — sinaliza inexperiência com contextos reais de produto
- ❑ Resposta capitulante: *“entregávamos o que o produto pedia”* — sinaliza ausência de ownership técnico
- ❑ Não ter exemplo concreto — responder em teoria é insuficiente para uma pergunta que começa com *“na prática”*
- ❑ Não mencionar comunicação de riscos — o entrevistador quer ver que você sabe escalar a conversa certa com a pessoa certa

Sinais de uma resposta forte

- ☐ Diferenciar débito técnico deliberado de débito accidental — saber quando aceitar e quando resistir
- ☐ Mencionar o conceito de “decisão registrada”: documentar o débito aceito com contexto e custo futuro estimado
- ☐ Descrever como priorizou o que garantia não-regressão mesmo em entregas rápidas
- ☐ Ter um exemplo de quando disse “não” ou “não assim” — e como conduziu essa conversa

Chave de Resposta

A resposta ideal tem três partes:

1. Seu framework mental

Como você distingue o que pode ser feito rápido do que não pode. Não toda velocidade é débito — mas alguns atalhos têm custo composto.

Raciocínio esperado: “Há uma diferença entre simplificar uma implementação hoje sabendo que vai refatorar depois, e pular tratamento de erro em um fluxo crítico por causa do prazo. O primeiro é pragmatismo. O segundo é risco assumido sem consciência.”

2. Um exemplo real com tensão genuína

Uma situação específica onde havia pressão real, onde você fez uma escolha difícil — e as consequências dessa escolha.

3. Como você comunica isso

Quem são seus interlocutores quando há conflito entre velocidade e qualidade — produto, tech lead, cliente. Como você apresenta o risco de forma que a decisão seja informada, não unilateral.

Pergunta 4 — Difícil

“No Monnos você ficou quase cinco anos como Head Android. Me dê um exemplo de uma decisão arquitetural que você tomou nesse período que, olhando para trás, faria diferente — e por quê.”

O que o entrevistador avalia

Autoconsciência técnica e capacidade real de aprendizado. Candidatos que não conseguem criticar suas próprias decisões passadas são um sinal de alerta — ou nunca tomaram decisões reais, ou não têm maturidade para revisitá-las.

Erros que eliminam candidatos

- ☐ Resposta falsa-modesta: *“trabalhei demais”* ou *“poderia ter documentado mais”* — não é uma decisão arquitetural
- ☐ Evasão total: *“fiz as melhores escolhas possíveis com o contexto”* — sinaliza arrogância ou medo
- ☐ Arrependimento genérico que qualquer pessoa poderia ter — sem especificidade técnica
- ☐ Não explicar por que a decisão fazia sentido na época — isso demonstra pensamento situado, não apenas hindsight

Sinais de uma resposta forte

- ☐ Identificar uma decisão real com nome, contexto e consequência
- ☐ Explicar o raciocínio que levou àquela decisão na época
- ☐ Descrever o que aprendeu e como isso mudou sua forma de decidir
- ☐ Conectar o aprendizado com algo que você aplica ativamente hoje

Chave de Resposta

Esta pergunta exige honestidade e especificidade. Não existe resposta “certa” — existe resposta verdadeira ou inventada, e entrevistadores sêniores percebem a diferença.

Etapa	O que cobrir
Contexto da decisão	Quando foi, qual era o estado do produto, quais eram as pressões e restrições da época
A decisão em si	O que foi escolhido — arquitetura, tecnologia ou processo
Por que fazia sentido então	Mostrar que foi uma decisão com trade-offs conscientes, não ignorância
O que ficou claro depois	Qual consequência real evidenciou o problema — escala, manutenção, estabilidade
O que faria diferente	A resposta técnica alternativa, com o <i>porquê</i> fundamentado

Atenção: Não exagere o erro para parecer humilde, nem o minimize para parecer competente. O entrevistador quer raciocínio maduro — não performance de autohumildade.

Pergunta 5 — Difícil

"Imagine que você acabou de ser alocado em um projeto na GOK. Ao entrar no repositório, você percebe arquitetura altamente acoplada, sem testes, com lógica de negócio dentro de Activities — e o app está em produção com usuários ativos. Qual é seu plano de ação nos primeiros 30 dias? Como você decide o que tocar primeiro sem quebrar o que está funcionando?"

O que o entrevistador avalia

Capacidade de diagnóstico técnico sem julgamento precipitado, planejamento incremental em ambientes com risco real e postura de ownership sem caos. Esse cenário é exatamente o que a GOK encontra em projetos de clientes — código legado crítico que não pode parar.

Erros que eliminam candidatos

- ☐ *“Proporia reescrever o app do zero”* — sinal imediato de inexperiência com produção real
- ☐ Começar a tocar no código antes de entender o domínio e os fluxos críticos
- ☐ Não mencionar comunicação com o time ou produto antes de agir
- ☐ Tratar todos os problemas como equivalentes — sem priorização baseada em risco
- ☐ Não mencionar testes como pré-requisito para qualquer mudança estrutural

Sinais de uma resposta forte

- ☐ Começar pela compreensão, não pela solução: mapear fluxos críticos, entender o negócio, conversar com quem mantém o código
- ☐ Diferenciar risco imediato (crashes, falhas de segurança) de débito técnico de longo prazo
- ☐ Mencionar a estratégia de adicionar testes como primeiro passo antes de refatorar
- ☐ Falar em mudanças incrementais e validadas — não grandes PRs que ninguém consegue revisar
- ☐ Comunicar proativamente a produto e stakeholders o que foi encontrado e qual é o plano

Chave de Resposta

Semana 1 — Diagnóstico sem tocar no código

- Rodar o app em diferentes condições: fluxo feliz, falha de rede, estados de erro
- Ler o código para entender o domínio — não para julgar, mas para aprender
- Identificar os fluxos críticos de negócio: o que não pode falhar sob nenhuma hipótese
- Conversar com quem construiu ou manteve o código — há contexto que não está no repositório
- Mapear pontos de maior instabilidade: Crashlytics, issues conhecidos, código que ninguém toca com medo

Semana 2 — Priorização e plano técnico

- Criar um mapa de risco: probabilidade de quebrar × impacto se quebrar
- Identificar onde não há testes nos fluxos críticos — esse é o maior risco para qualquer mudança
- Propor ao time um backlog técnico com justificativa de negócio para cada item — não apenas *“precisa refatorar”*

- Separar o que é bug ou risco imediato do que é melhoria arquitetural de médio prazo

Semanas 3 e 4 — Primeiras intervenções cirúrgicas

- Adicionar testes nos fluxos mais críticos *antes* de refatorar qualquer coisa
- Fazer a menor mudança possível que reduz o maior risco — sem tentar resolver tudo de uma vez
- Revisar cada mudança com o time — não agir de forma isolada em código que outros dependem
- Documentar as decisões tomadas e o estado encontrado para criar memória técnica do projeto

Dica estratégica: Use esta frase na resposta: *“Minha regra em legado é: primeiro entender, depois adicionar testes, depois refatorar. Nunca na ordem inversa.”*

Pergunta 6 — Difícil

“Você identificou um problema de segurança relevante em um app em produção — algo que expõe dados sensíveis de usuários. O time de produto está em sprint final, com release marcada para dois dias. Como você age?”

O que o entrevistador avalia

Postura de ownership em situação de conflito real entre velocidade de negócio e responsabilidade técnica. Em ambientes financeiros críticos — contexto direto da GOK — problemas de segurança não são negociáveis. O entrevistador quer ver se você tem coragem técnica, sabe escalar corretamente e não se paralisa nem age de forma unilateral destrutiva.

Erros que eliminam candidatos

- ☐ *“Não mencionaria antes da release para não travar o time”* — capitulação com risco real ao usuário

- ☐ “Bloquearia o release imediatamente sem consultar ninguém” — ação unilateral sem contexto do risco real
- ☐ Não saber comunicar a severidade do problema de forma objetiva para não-técnicos
- ☐ Não ter opinião própria — “deixaria o time decidir” sem oferecer uma recomendação fundamentada
- ☐ Não mencionar documentação ou rastreabilidade da decisão tomada

Sinais de uma resposta forte

- ☐ Avaliar a severidade real antes de agir — nem todo problema de segurança tem o mesmo impacto imediato
- ☐ Escalar imediatamente para as pessoas certas com fatos, não com alarme
- ☐ Apresentar opções com trade-offs claros, não um ultimato
- ☐ Deixar a decisão informada nas mãos de quem tem autoridade — mas registrar a posição técnica por escrito
- ☐ Ter clareza sobre o que é negociável (prazo) e o que não é (exposição de dados de usuários)

Chave de Resposta

Passo 1 — Caracterizar o risco antes de agir

Qual é a superfície de ataque? Precisa de acesso físico ao device? É explorável remotamente? Afeta todos os usuários ou um subconjunto? Há dados já comprometidos ou é um risco futuro? Essa análise leva minutos e muda completamente o peso da decisão.

Passo 2 — Escalar com clareza

Não é o momento de resolver sozinho. É o momento de colocar as pessoas certas na mesma sala com a informação correta.

Modelo de comunicação: “Encontrei um problema de segurança com severidade X. Aqui está o que ele expõe, aqui está o que seria necessário para explorá-lo, e aqui estão as opções que vejo.”

Passo 3 — Apresentar opções, não ultimatoss

Opção	Descrição
A — Hotfix cirúrgico	Pode ser feito em X horas; release adiada em Y dias
B — Mitigação temporária	Reduz o risco enquanto a correção definitiva vai para a próxima sprint
C — Release com risco conhecido	Requer decisão explícita e registrada de quem tem autoridade

Passo 4 — Registrar a posição técnica

Independente da decisão final, a recomendação técnica deve estar documentada. Não como proteção pessoal — como governança. Em ambientes transacionais críticos, rastreabilidade de decisões é parte do processo.

Dica estratégica: Use esta frase na resposta: *“Meu trabalho não é bloquear o release — é garantir que a decisão de fazer o release seja tomada com todas as informações na mesa.”*

Pergunta 7 — Difícil

“Como você conduziria um code review em que o autor do código é um desenvolvedor mais sênior que você — ou alguém com mais tempo de casa no projeto — e você identificou um problema real na abordagem dele?”

O que o entrevistador avalia

Maturidade interpessoal em contexto técnico de alta tensão. A GOK valoriza code review como prática real — não como formalidade. O entrevistador quer saber se você mantém a qualidade técnica sem deixar que hierarquia informal ou dinâmicas de time silenciem feedbacks importantes.

Erros que eliminam candidatos

- ☐ *"Aprovaria mesmo discordando para não criar conflito"* — ausência de ownership e cumplicidade com o problema
- ☐ *"Apontaria o erro publicamente para o time ver"* — postura destrutiva que revela falta de inteligência relacional
- ☐ Não ter um exemplo real de quando fez isso — resposta puramente teórica é insuficiente
- ☐ Tratar todo feedback como uma questão de "ter razão" — não demonstra abertura para estar errado também
- ☐ Não reconhecer que pode haver contexto que você não viu

Sinais de uma resposta forte

- ☐ Separar clareza técnica de postura combativa — é possível ser direto sem ser agressivo
- ☐ Fazer perguntas antes de afirmar
- ☐ Ter exemplos reais — um onde o feedback foi bem recebido e um onde gerou tensão
- ☐ Reconhecer que review é conversa, não julgamento
- ☐ Demonstrar abertura genuína para estar errado e aprender com a resposta do autor

Chave de Resposta

Atenção: Code review não é sobre hierarquia — é sobre o código e o usuário que vai usar o sistema. Um problema real não deixa de ser um problema porque o autor é sênior.

1. Começar com uma pergunta, não com uma afirmação

"Você considerou o comportamento desse código quando X acontece?" ou "Estou vendo que não há tratamento para Y — tem algum contexto que eu possa ter perdido?"

Isso abre espaço para o autor explicar algo que o revisor não viu e coloca o foco no problema técnico, não no julgamento da pessoa.

2. Se a pergunta confirmar o problema, ser direto

"Entendo o raciocínio, mas acredito que isso vai causar Z em produção quando X acontecer. Aqui está o que eu proporia como alternativa — o que você acha?"

Direto, técnico, com solução na mesa. Sem passivo-agressividade, sem suavização que obscurece o ponto.

3. Se houver resistência, escalar o argumento — não o conflito

Se não há consenso, a resposta não é ceder nem insistir em loop. É trazer um terceiro com mais contexto ou propor que o time discuta — tornando a decisão explícita e documentada.

4. Aceitar estar errado com a mesma abertura

Se a resposta do autor revelar contexto que o revisor não tinha, reconheça com clareza. Quem faz boas perguntas e aceita boas respostas é mais respeitado do que quem tem sempre razão.

Dica estratégica: Use esta frase na resposta: *"Hierarquia não é argumento técnico. Se o código tem um problema, meu trabalho é levantar — com respeito, com contexto e com uma proposta alternativa."*

Pergunta 8 — Difícil

"Conte sobre uma situação em que você discordou de uma decisão técnica tomada pelo time ou por uma liderança — e a decisão foi mantida mesmo assim. Como você agiu depois?"

O que o entrevistador avalia

Resiliência profissional e capacidade de operar com ownership mesmo quando não concordou com a direção. Em contexto consultivo — como na GOK — você frequentemente vai trabalhar em decisões técnicas que já foram tomadas antes de você chegar. O entrevistador quer saber se você sabota, se abdica ou se age de forma madura.

Erros que eliminam candidatos

- ☐ *“Nunca discordei de nenhuma decisão técnica do time”* — não é crível e sinaliza que nunca teve voz real
- ☐ *“Disse o que pensava e fiz do jeito deles sabendo que estava errado”* — passivo, sem aprendizado
- ☐ Demonstrar ressentimento ou julgamento sobre quem tomou a decisão
- ☐ Não ter um exemplo real — resposta genérica em primeira pessoa do plural
- ☐ Não deixar claro o que você fez de forma diferente após a decisão ser mantida

Sinais de uma resposta forte

- ☐ Ter articulado a discordância de forma clara e técnica na época — não ficou em silêncio
- ☐ Registrar a posição de alguma forma (comentário no PR, documento, conversa registrada) sem drama
- ☐ Após a decisão mantida, comprometer-se genuinamente com a execução — sem sabotar passivamente
- ☐ Usar a experiência para aprender algo — sobre o contexto que não tinha ou sobre como influenciar melhor
- ☐ Distinguir decisões que mudaria retrospectivamente das que ainda discorda, mas respeita o processo

Chave de Resposta

Use o formato **STAR com foco no “depois”** — essa é a parte que mais interessa ao entrevistador.

Situação Qual era o contexto técnico — qual era a decisão, quem estava envolvido, qual era o impacto esperado.

Tarefa Qual era o seu papel — era uma decisão que impactava diretamente o seu trabalho? Você tinha responsabilidade técnica sobre aquela parte?

Ação — em dois momentos

Antes da decisão ser mantida: Como você articulou a discordância. Qual foi o argumento técnico, com quem conversou, que alternativas apresentou. A qualidade do argumento revela que você age com método, não com emoção.

Depois da decisão ser mantida:

- Se comprometeu com a execução com a mesma qualidade que teria na sua própria decisão?
- Monitorou os resultados e revisitou com dados quando a hipótese se confirmou?
- Aprendeu que havia contexto que não tinha e revisou sua análise?

Resultado O que aconteceu com a decisão ao longo do tempo — confirmou sua preocupação? Não confirmou? O que você aprendeu sobre o processo de decisão técnica em times?

Dica estratégica: Use esta frase na resposta: *“Discordar e depois executar com excelência não são contraditórios. Meu compromisso é com o produto e com o time — não com estar certo.”*

*Career Agent Pro — Material personalizado para Pessoa Desenvolvedora Android Sênior
/ GOK / Inovação Digital*

Career Agent PRO

A preparação perfeita para cada vaga de emprego. Inteligência aplicada ao seu contexto e à empresa que você quer entrar.

 www.career-agent-pro.com

 support@main.career-agent-pro.com