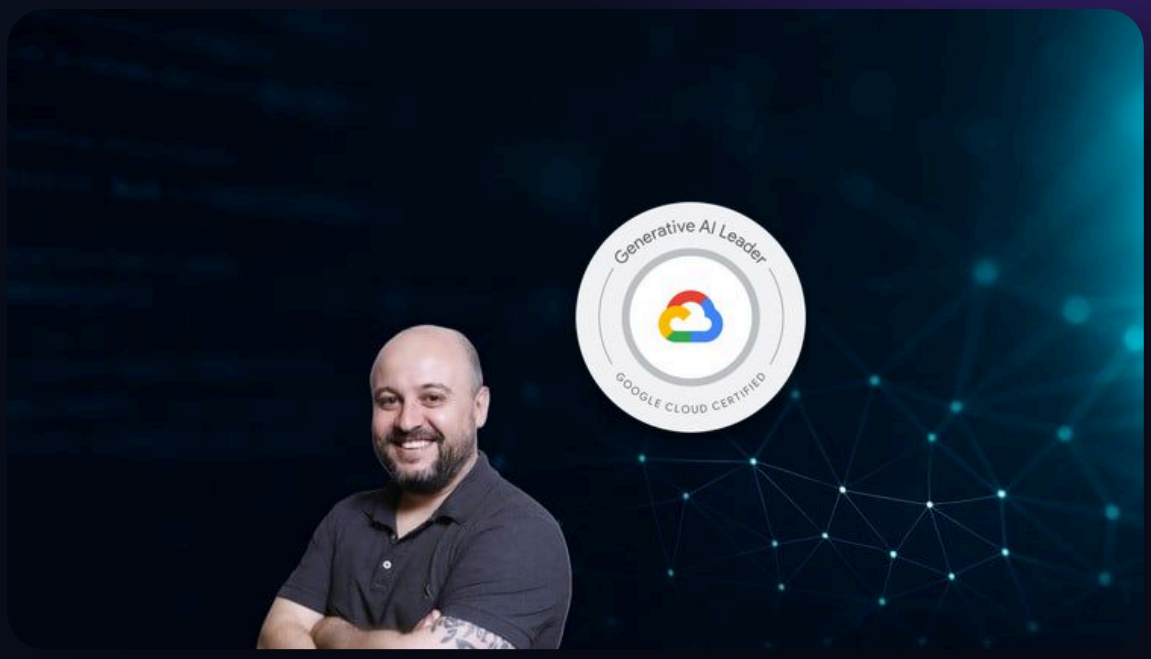
Strecht Simulados · Guia de preparação



Guia de preparação Google Generative AI Leader

Tudo o que a prova cobra, organizado por domínio: conceitos-chave, comparações que confundem, pegadinhas reais, gatilhos do enunciado, questões comentadas e um plano para chegar pronto ao exame.

4 capítulos por domínio	**28** questões comentadas	**70%** nota de corte
6 provas completas no curso		

ANTES DE COMEÇAR

Como usar este guia

Este guia foi feito para ser estudado junto com as 6 provas completas do curso Google Generative AI Leader: o guia explica o que cai e onde estão as armadilhas; as provas treinam o ritmo e mostram o que você ainda não domina.

Quando	O que fazer
Semana 1	Faça o teste diagnóstico (página seguinte) e leia os capítulos 1 e 2. No fim da semana, faça a Prova 1 do curso no tempo do exame.
Semana 2	Leia os demais capítulos, começando pelos domínios em que você errou mais no diagnóstico. Faça as provas 2 e 3.
Semana 3	Releia os alertas de pegadinha e as tabelas de comparação. Faça as provas 4 e 5 e revise cada erro pela explicação.
Semana 4	Estude a folha de cola, faça a Prova 6 e refaça as provas abaixo de 80%. Agende o exame quando estiver acima de 80% em todas (a nota de corte é 70%).

Mapa do exame

Cada prova do curso tem 50 questões e 90 minutos, no formato do exame. Referência de aprovação: 70%.

Domínio (capítulo)	Peso	Neste guia
1. Estratégias de Negócio para Soluções de IA Generativa	15%	18 conceitos · 14 pegadinhas
2. Fundamentos de IA Generativa	30%	18 conceitos · 14 pegadinhas
3. Ofertas de IA Generativa do Google Cloud	35%	18 conceitos · 14 pegadinhas
4. Técnicas para Melhorar a Saída dos Modelos	20%	18 conceitos · 14 pegadinhas

Como ler os capítulos: os **tópicos-chave** são o que mais cai; a **tabela de memorização** separa o que a prova costuma confundir; os **alertas de prova** mostram as armadilhas das alternativas erradas; os **gatilhos** ligam palavras do enunciado à resposta; e o **checklist** diz se você está pronto para o domínio.

Estude um pouco todo dia, de graça

Instagram @strecht.simulados

Um post de prova por dia: questão comentada, pegadinhas e o "não confunda" de cada certificação.

Seguir no Instagram →

YouTube Strecht Simulados

Shorts diários com pegadinhas de prova e, aos sábados, um simulado comentado.

Inscrever-se no canal →

DIAGNÓSTICO

Você já sabe isso?

Responda às 12 questões sem consultar nada. As respostas, com a explicação e o capítulo para revisar, estão no fim do guia. Comece o estudo pelos capítulos em que errou.

D1 · Estratégias de Negócio para Soluções de IA Generativa

Antes de iniciar o projeto de IA, a diretoria e as áreas combinam como o sucesso será medido. Por que isso é importante?

A. Alinha expectativas e decisões
B. Garante o sucesso do projeto
C. Elimina riscos técnicos
D. Dispensa acompanhar resultados

D2 · Estratégias de Negócio para Soluções de IA Generativa

Ao aplicar o SAIF, uma empresa usa os mesmos controles de identidade, rede e dados já aplicados aos outros sistemas também nos sistemas de IA. Qual elemento do framework é esse?

A. Abrir os modelos à internet
B. Tratar a IA fora da governança
C. Estender a segurança existente à IA
D. Eliminar os controles de identidade

D3 · Estratégias de Negócio para Soluções de IA Generativa

Um invasor insere exemplos manipulados nos dados de treino para alterar o comportamento do modelo. Qual risco é esse?

A. Envenenamento de dados
B. Negação de serviço
C. Força bruta de senhas
D. Injeção de SQL

D4 · Fundamentos de IA Generativa

Uma solicitação tem 30 mil tokens de entrada e o modelo tem janela de contexto de 32 mil. O que pode acontecer?

A. A entrada é dobrada automaticamente
B. A resposta pode ter tamanho ilimitado
C. O modelo ignora a janela nesse caso
D. Sobra pouco espaço para a resposta

D5 · Fundamentos de IA Generativa

Ao preparar documentos longos para RAG, a equipe os divide em trechos menores antes de criar os embeddings. Por que isso é feito?

A. Para aumentar o custo de armazenamento
B. Para apagar informações dos documentos
C. Para recuperar trechos relevantes
D. Para impedir o uso dos documentos

D6 · Fundamentos de IA Generativa

Uma startup quer um modelo aberto e leve do Google para rodar e ajustar na própria infraestrutura. Qual família atende?

A. Imagen
B. Veo
C. Chirp
D. Gemma

D7 · Ofertas de IA Generativa do Google Cloud

Uma empresa quer dar aos funcionários um ponto único para pesquisar informações em vários sistemas corporativos e usar agentes de IA sobre esses dados. Qual oferta do Google atende?

A. Cloud Storage Archive
B. Cloud Interconnect
C. Gemini Enterprise
D. Cloud Deploy

D8 · Ofertas de IA Generativa do Google Cloud

Uma equipe quer testar prompts, comparar respostas de modelos e ajustar parâmetros em uma interface visual antes de integrar à aplicação. O que usar?

A. Agent Studio
B. Cloud DNS
C. Cloud Billing
D. Cloud Scheduler

D9 · Ofertas de IA Generativa do Google Cloud

Um assistente precisa responder sobre notícias do dia com fontes públicas atualizadas. Qual recurso da Agent Platform ajuda?

A. Exportação para o BigQuery
B. Temperatura igual a zero
C. Aumento do número de réplicas do app
D. Grounding com a Pesquisa Google

D10 · Técnicas para Melhorar a Saída dos Modelos

Um assistente funciona bem em perguntas comuns, mas falha em pedidos raros e incomuns. Qual limitação é essa?

A. Casos de borda pouco representados
B. Latência baixa da rede corporativa
C. Janela de contexto grande demais
D. Temperatura igual a zero

D11 · Técnicas para Melhorar a Saída dos Modelos

O assistente respondeu que a versão mais recente de um software é uma que já foi substituída. Qual é a causa provável?

A. A data de corte do modelo
B. O excesso de filtros de segurança
C. O uso de delimitadores no prompt
D. A temperatura igual a zero

D12 · Técnicas para Melhorar a Saída dos Modelos

Um regulador pede que o banco explique exatamente como um LLM chegou a uma resposta. Qual limitação dificulta isso?

A. Suporte a vários idiomas
B. Latência baixa das respostas
C. Janela de contexto grande demais no prompt
D. Baixa explicabilidade interna dos LLMs

01

CAPÍTULO 1 · 15% DA PROVA

Estratégias de Negócio para Soluções de IA Generativa

Aqui eu preciso pensar como líder: escolher o primeiro caso de uso, medir retorno, organizar a adoção e, principalmente, aplicar segurança e IA responsável. O que mais se repete é o Secure AI Framework (SAIF) e os riscos específicos de IA (injeção de prompt direta e indireta, envenenamento de dados, roubo de pesos), as soluções Model Armor e Sensitive Data Protection, e os princípios de IA responsável (equidade, privacidade, segurança, transparência e responsabilização), além de explicabilidade. Também caem linha de base de KPIs, custo total, patrocínio executivo, centro de excelência e gestão da mudança. As alternativas erradas costumam ser extremas (proibir tudo, aprovar tudo automaticamente, desligar logs) ou irrelevantes (cor da interface, logotipo). Onde mais erro: separar necessidades de recursos ao escolher a solução e distinguir equidade de explicabilidade.

Neste capítulo

18 tópicos-chave · 14 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

Secure AI Framework (SAIF)

Framework conceitual do Google para proteger sistemas de IA contra riscos como injeção de prompt, envenenamento de dados e roubo ou exfiltração de modelos. Entre seus elementos: estender as fundações de segurança existentes (identidade, rede, dados) ao ecossistema de IA e automatizar defesas para acompanhar a velocidade de ameaças novas e existentes.

Injeção de prompt

Usuários inserem instruções maliciosas para fazer o assistente ignorar suas regras. Na injeção indireta, as instruções escondidas vêm de conteúdo recuperado, como um documento do RAG que manda revelar dados internos. O Model Armor inspeciona prompts e respostas contra injeção de prompt e outros riscos.

Envenenamento de dados e roubo de modelo

Envenenamento é inserir exemplos manipulados nos dados de treino para alterar o comportamento do modelo. Em um modelo ajustado com dados estratégicos, o risco que envolve o próprio modelo é o roubo dos pesos. Ambos são riscos específicos de ML tratados pelo SAIF, diferentes de ataques clássicos como injeção de SQL ou força bruta.

Sensitive Data Protection

Antigo Cloud DLP: identifica, classifica e mascara dados sensíveis, como CPFs e dados de saúde em documentos enviados a um assistente. Junto com Model Armor, IAM e VPC Service Controls, apoia a segurança de aplicações de IA e dos dados que elas usam.

Menor privilégio para agentes

Agentes devem ter só as permissões necessárias, com IAM e auditoria; uma conta de serviço de agente com papel de proprietário viola o princípio. Ações de alto impacto, como reembolsos de valor alto, pedem aprovação humana acima de um limite que o agente não pode alterar.

Princípios de IA responsável

O Google Cloud destaca equidade, privacidade, segurança, transparência e responsabilização. Entre os princípios de IA publicados pelo Google está evitar criar ou reforçar vieses injustos. Latência, disponibilidade e escalabilidade não são princípios de IA responsável.

Equidade (fairness)

Vieses nos dados podem gerar resultados injustos, como triagem de currículos que favorece um grupo por padrões históricos. Testar se o desempenho é parecido entre grupos de clientes (métricas por grupo) é a forma de detectar e tratar esses vieses.

Explicabilidade e transparência

Explicabilidade é poder entender e justificar uma decisão do modelo, como explicar ao cliente por que o crédito foi recusado. Transparência inclui deixar claro quando e como a IA é usada, por exemplo informando o uso de IA nas respostas aos clientes.

Responsabilização e auditoria

Responsabilização exige donos claros, governança e mecanismos de correção: se ninguém sabe quem responde pelos erros em produção, esse princípio está faltando. Registros de auditoria das interações (respostas dadas e ações executadas pelo agente) dão rastreabilidade e permitem investigar incidentes.

Privacidade: minimização e consentimento

Minimização de dados é coletar só o necessário; coletar mais para o caso de ser útil viola o princípio. Usar gravações de clientes para treinar exige obter consentimento e informar. Residência de dados, como manter dados em regiões da UE, influencia a escolha de regiões e serviços.

Governança de dados antes de conectar

Antes de ligar repositórios a um assistente, classifique os dados e revise acessos. Soluções corporativas devem respeitar as permissões de acesso; um assistente que mostra conteúdo confidencial a quem não deveria indica que esse controle faltou.

IA de consumo fora das políticas

Funcionários que colam dados de clientes em ferramentas de IA de consumo expõem dados sem controles corporativos. Ferramentas corporativas, com contratos e controles, protegem melhor os dados da empresa.

Red teaming e monitoramento contínuo

Red teaming é provocar respostas perigosas de propósito antes do lançamento para achar falhas de segurança e de conteúdo. Depois do lançamento, a governança mantém monitoramento contínuo de desempenho, segurança e impactos, porque riscos surgem no uso real.

Necessidades x recursos

Antes de escolher a solução, o Google sugere avaliar as necessidades (escala, personalização, interação do usuário, privacidade, latência e conectividade) e os recursos (pessoas, dinheiro e tempo). Quantos usuários, quanto personalizar, quão sensíveis são os dados e qual latência é aceitável são necessidades.

Escolha do caso de uso e comprar x construir

O primeiro caso de uso deve ter valor claro, dados disponíveis e viabilidade técnica, não ser o mais complexo nem o que usa o maior modelo. Para necessidades genéricas, como assistente de escrita comum a várias áreas, comprar pronto (Gemini no Workspace) costuma ser mais rápido e barato que construir.

Medição de valor e ROI

Defina critérios de sucesso e KPIs antes do projeto para alinhar expectativas, e registre a linha de base dos indicadores antes do piloto. KPIs de negócio, como redução do tempo médio de atendimento, mostram impacto melhor que métricas técnicas (parâmetros, GPUs, tokens). Adoção e satisfação dos usuários complementam o ROI, e o custo total inclui consumo, licenças, integração, mudança de processos, capacitação e governança.

Organização da adoção

O patrocinador executivo dá direção, prioridade, orçamento e alinhamento estratégico. Combinar patrocínio da diretoria (de cima para baixo) com iniciativas das equipes (de baixo para cima) alinha estratégia e ideias práticas. Um centro de excelência coordena padrões, reuso, capacitação e governança quando várias áreas avançam sem orientação comum. Resistência por medo e falta de preparo pede gestão da mudança com comunicação e capacitação.

Escalar com segurança e custo

A sequência que reduz riscos é piloto, avaliação com métricas e expansão gradual. Se o piloto custa demais usando um modelo grande em tarefas simples, teste um modelo menor e mais barato para equilibrar custo e qualidade.

Macetes para memorizar

💡 Riscos de IA no SAIF

IER: Injeção (no prompt), Envenenamento (no treino), Roubo (dos pesos).

💡 Ferramentas de proteção

Armor protege o prompt; Sensitive Data Protection esconde o dado.

💡 Princípios de IA responsável

EPSTR: Equidade, Privacidade, Segurança, Transparência, Responsabilização.

💡 Equidade x explicabilidade

Equidade = é justo entre grupos? Explicabilidade = consigo explicar o porquê?

💡 Necessidades x recursos

Recursos são 3 Ps: Pessoas, Patrimônio (dinheiro) e Prazo (tempo); escala, personalização, interação, privacidade, latência e conectividade são necessidades.

💡 Medir valor

Linha de base antes, KPI de negócio durante, custo total na conta.

💡 Escalar

Pilota, Avalia, Expande (P-A-E).

Tabela de memorização: não confunda

Quando é...	...e quando é
Equidade Desempenho parecido entre grupos; triagem que favorece um grupo.	**Explicabilidade** Explicar por que o modelo recusou um crédito.

Model Armor Filtrar injeção de prompt e outros riscos em prompts e respostas.	**Sensitive Data Protection** Identificar e mascarar CPFs, dados de saúde e outros dados sensíveis.
Injeção de prompt direta O próprio usuário escreve instruções maliciosas no prompt.	**Injeção indireta de prompt** As instruções vêm escondidas em conteúdo recuperado, como um documento do RAG.
Necessidades do projeto Escala, personalização, interação do usuário, privacidade, latência e conectividade.	**Recursos do projeto** Pessoas, dinheiro e tempo.
Comprar solução pronta Necessidade genérica comum a várias áreas, como escrita de e-mails e documentos.	**Construir solução própria** Necessidade específica com personalização que a oferta pronta não cobre.
KPIs de negócio Mostrar impacto e ROI: tempo médio de atendimento, satisfação, adoção.	**Métricas técnicas** Parâmetros do modelo, GPUs, tokens por prompt: não medem valor para o negócio.
Envenenamento de dados Exemplos manipulados entram nos dados de treino e alteram o comportamento.	**Roubo dos pesos do modelo** Exfiltração do modelo ajustado com dados estratégicos.

Números que caem

Item	Valor
Peso do domínio na prova	**cerca de 15%**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Marcar injeção de SQL, força bruta ou negação de serviço para dados de treino manipulados.
Como não cair: Dados de treino manipulados é envenenamento de dados, risco específico de ML.

⚠ **Alerta de prova:** Tratar instruções escondidas em um documento recuperado como falha técnica comum.
Como não cair: Conteúdo recuperado que manipula o modelo é injeção indireta de prompt.

⚠ **Alerta de prova:** Escolher os recursos de pessoas e verba quando o enunciado lista usuários, personalização, sensibilidade dos dados e latência.
Como não cair: Esses itens são necessidades; recursos são pessoas, dinheiro e tempo.

⚠ **Alerta de prova:** Usar ROI já obtido como critério para escolher a solução.
Como não cair: O retorno é medido depois da implantação, contra uma linha de base registrada antes.

⚠ **Alerta de prova:** Medir o impacto com número de parâmetros, GPUs ou tokens por prompt.
Como não cair: Impacto no negócio aparece em KPIs operacionais, adoção e satisfação.

⚠ **Alerta de prova:** Contar só o preço por token no custo do projeto.
Como não cair: O custo total inclui licenças, integração, mudança de processos, capacitação e governança.

⚠ **Alerta de prova:** Acreditar que definir sucesso antes ou combinar diretoria e equipes garante o sucesso e elimina riscos.
Como não cair: Isso alinha expectativas; não há garantia, e governança e medição continuam.

⚠ **Alerta de prova:** Dar mais papéis à conta de serviço do agente para evitar erros de acesso futuros.
Como não cair: O certo é menor privilégio, com IAM e auditoria.

⚠ **Alerta de prova:** Permitir que o agente aprove qualquer valor ou mude o próprio limite.
Como não cair: Ações de alto impacto exigem aprovação humana acima de um limite fixo.

⚠ **Alerta de prova:** Responder que contar usuários ativos por mês basta após o lançamento.
Como não cair: É insuficiente; a governança mantém monitoramento contínuo de riscos.

⚠ **Alerta de prova:** Proibir a IA em todas as áreas para resolver adoção desorganizada.
Como não cair: Crie um centro de excelência com padrões, reuso, capacitação e governança.

⚠ **Alerta de prova:** Lançar para toda a empresa de uma vez ou manter piloto contínuo sem decidir.
Como não cair: Piloto, avaliação com métricas e expansão gradual.

⚠ **Alerta de prova:** Tomar latência baixa, alta disponibilidade ou escalabilidade como princípio de IA responsável.
Como não cair: Os princípios são equidade, privacidade, segurança, transparência e responsabilização.

⚠ **Alerta de prova:** Treinar um modelo próprio do zero para um assistente de escrita genérico.
Como não cair: Necessidade comum: comprar pronto, como o Gemini no Workspace.

Como cai na prova

📝 **Como cai:** Cenário de ataque ou incidente e pergunta Qual risco é esse?
Como resolver: Veja onde o ataque entra: no prompt (injeção), em documento recuperado (injeção indireta), nos dados de treino (envenenamento) ou no modelo ajustado (roubo dos pesos). Descarte ataques clássicos como SQL e força bruta.

📝 **Como cai:** Cenário ético e pergunta Qual princípio está em jogo?
Como resolver: Mapeie o verbo: explicar decisão = explicabilidade; resultado diferente por grupo = equidade; informar uso de IA = transparência; quem responde = responsabilização; coletar demais = minimização.

📝 **Como cai:** Qual a melhor forma de começar, medir ou escalar a adoção.
Como resolver: A certa é equilibrada: valor e viabilidade, linha de base, KPIs de negócio, piloto e expansão gradual. Descarte extremos (proibir, lançar tudo, esperar a tecnologia parar de mudar).

📝 **Como cai:** Controle para agente ou assistente com acesso a sistemas e documentos.
Como resolver: Pense em menor privilégio, respeito às permissões, aprovação humana em alto impacto e registros de auditoria; descarte desligar logs ou dar mais papéis.

📝 **Como cai:** Escolha DUAS soluções de segurança ou princípios.
Como resolver: Marque os produtos de segurança de IA (Model Armor, Sensitive Data Protection) ou princípios reais (privacidade, responsabilização); descarte apps de produtividade e opostos como opacidade total.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
provocar respostas perigosas de propósito antes do lançamento	**Red teaming**
instruções maliciosas para ignorar as regras	**Injeção de prompt; Model Armor**
instrução escondida em documento recuperado	**Injeção indireta de prompt**
exemplos manipulados nos dados de treino	**Envenenamento de dados**
mascarar CPFs e dados de saúde	**Sensitive Data Protection**
mesmos controles de identidade, rede e dados aplicados à IA	**SAIF: estender fundações de segurança**
explicar por que o crédito foi recusado	**Explicabilidade**
desempenho por grupo / favorece um grupo	**Equidade**

ninguém sabe quem responde pelos erros	**Responsabilização**
coleta mais dados do que precisa	**Minimização de dados**
dados só em regiões da UE	**Residência de dados**
conta de serviço do agente como proprietário	**Menor privilégio**
antes do piloto, registrar para medir retorno	**Linha de base dos indicadores**
várias áreas sem padrões comuns	**Centro de excelência em IA**

🎯 Se você lembrar de só uma coisa deste capítulo...

- SAIF trata injeção de prompt, envenenamento de dados e roubo de modelos; estende a segurança existente e automatiza defesas.
- Model Armor filtra prompts e respostas; Sensitive Data Protection mascara dados sensíveis.
- Princípios: equidade, privacidade, segurança, transparência e responsabilização.
- Linha de base antes do piloto; KPIs de negócio e custo total para o ROI.
- Agente: menor privilégio, aprovação humana acima de um limite e auditoria.

Essenciais para a prova (checklist)

☐ Sei os riscos específicos de IA e o que o SAIF recomenda.

☐ Sei diferenciar injeção de prompt direta e indireta.

☐ Sei quando usar Model Armor e Sensitive Data Protection.

☐ Sei associar cada cenário ético ao princípio de IA responsável certo.

☐ Sei separar necessidades (escala, personalização, interação, privacidade, latência, conectividade) de recursos (pessoas, dinheiro, tempo).

☐ Sei como escolher o primeiro caso de uso e quando comprar pronto.

☐ Sei medir valor com linha de base, KPIs de negócio, adoção, satisfação e custo total.

☐ Sei o papel do patrocinador executivo, do centro de excelência e da gestão da mudança.

☐ Sei os controles para agentes: menor privilégio, aprovação humana e auditoria.

Questões de revisão do capítulo

R1.1 · Estratégias de Negócio para Soluções de IA Generativa

Documentos enviados a um assistente de IA podem conter CPFs e dados de saúde. Qual serviço do Google Cloud ajuda a identificar e mascarar esses dados?

A. Cloud CDN com cache
B. Cloud Scheduler Jobs
C. Cloud Router BGP
D. Sensitive Data Protection

R1.2 · Estratégias de Negócio para Soluções de IA Generativa

Um banco precisa explicar aos clientes por que o modelo recusou um crédito. Qual princípio está em jogo?

A. Compressão
B. Explicabilidade
C. Paralelismo
D. Criatividade máxima

R1.3 · Estratégias de Negócio para Soluções de IA Generativa

Uma empresa combina patrocínio da diretoria com iniciativas que surgem das equipes para adotar IA generativa. Qual é a vantagem?

A. Elimina a necessidade de governança
B. Alinha estratégia e ideias práticas
C. Garante sucesso sem riscos
D. Dispensa medir resultados

R1.4 · Estratégias de Negócio para Soluções de IA Generativa

Uma central implantou IA para apoiar atendentes. Qual indicador mostra melhor o impacto no negócio?

A. Número de parâmetros do modelo
B. Número de GPUs usadas no treino
C. Redução do tempo médio de atendimento
D. Quantidade média de tokens por prompt

Respostas comentadas

R1.1. D) Sensitive Data Protection

Por quê: O Sensitive Data Protection (antigo Cloud DLP) classifica e mascara dados sensíveis.

✗ A) Incorreta. Acelera conteúdo.
✗ B) Incorreta. Agenda tarefas.
✗ C) Incorreta. Troca rotas.

> **Dica de prova:** Mascarar dados pessoais: Sensitive Data Protection.

R1.2. B) Explicabilidade

Por quê: Explicabilidade ajuda a entender e justificar decisões de modelos.

✗ A) Incorreta. Não é o foco.
✗ C) Incorreta. Não é o foco.
✗ D) Incorreta. Não é o foco.

> **Dica de prova:** Explicar a decisão: explicabilidade.

R1.3. B) Alinha estratégia e ideias práticas

Por quê: Abordagens de cima para baixo e de baixo para cima se complementam na adoção.

✗ A) Incorreta. A governança continua necessária.
✗ C) Incorreta. Não há garantia.
✗ D) Incorreta. Medir continua necessário.

> **Dica de prova:** Diretoria e equipes juntas: alinhamento e engajamento.

R1.4. C) Redução do tempo médio de atendimento

Por quê: KPIs de negócio, como tempo de atendimento e satisfação, mostram o retorno da IA.

✗ A) Incorreta. Não mede o valor para o negócio.
✗ B) Incorreta. É métrica técnica.
✗ D) Incorreta. É métrica técnica.

> **Dica de prova:** Impacto no negócio: KPIs operacionais.

02

CAPÍTULO 2 · 30% DA PROVA

Fundamentos de IA Generativa

Aqui eu preciso dominar o vocabulário básico que sustenta a prova inteira: a hierarquia IA > ML > IA generativa, o que é modelo de fundação, LLM, token, janela de contexto, embeddings e multimodalidade. A prova cobra muito a distinção entre tipos de aprendizado (supervisionado, não supervisionado, reforço), tipos de dados (estruturado, semiestruturado, não estruturado) e entre ML preditivo e IA generativa. Também cai o panorama em camadas do Google (infraestrutura, modelos, plataforma, agentes, aplicações) e qual família de modelo gera o quê (Gemini, Gemma, Imagen, Veo, Chirp, Gemini Nano). Onde mais erro: alternativas que invertem conceitos (ajuste fino amplo e pré-treino específico, ML dentro da IA generativa) e cenários de previsão que parecem IA generativa mas são ML preditivo. As questões são curtas e de cenário, e quase sempre a palavra-chave do enunciado entrega a resposta.

Neste capítulo

18 tópicos-chave · 14 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

IA, ML e IA generativa

São conceitos aninhados: IA é o campo geral, machine learning é a parte que aprende padrões com dados e a IA generativa é um tipo de ML que cria conteúdo novo a partir desses padrões. Deep learning usa redes neurais com muitas camadas e é a base dos modelos generativos modernos. A prova oferece alternativas que invertem a ordem ou dizem que são sinônimos.

Modelo de fundação

Modelo pré-treinado em dados amplos que pode ser adaptado a muitas tarefas e generaliza para tarefas novas, muitas vezes em zero-shot. Um LLM é um modelo de fundação focado em linguagem, mas nem todo modelo de fundação é LLM: há modelos de fundação de imagem, áudio e outras modalidades. Um modelo específico (por exemplo, só para classificar notas fiscais) pode ser menor e mais preciso naquela tarefa, mas não é versátil.

LLM e token

De forma simplificada, o LLM gera texto prevendo o próximo token repetidas vezes, com base no contexto e no que aprendeu; não copia respostas prontas nem segue regras fixas de gramática. Token é um pedaço de texto processado; entrada e saída são medidas e cobradas em tokens. Prompts mais longos e respostas mais longas aumentam o custo.

Janela de contexto

Limita quantos tokens o modelo considera e costuma somar entrada e saída. Se a entrada ocupa quase toda a janela, sobra pouco espaço para a resposta; o limite continua valendo e a resposta não pode ter tamanho ilimitado. Para analisar de uma vez um documento enorme com anexos, a característica decisiva é o tamanho da janela de contexto.

Amostragem probabilística

Modelos generativos escolhem tokens por probabilidade; com temperatura acima de zero, o mesmo prompt pode gerar respostas um pouco diferentes. Isso não significa que o modelo foi retreinado entre as chamadas nem que o prompt foi corrompido.

Supervisionado, não supervisionado e por reforço

Supervisionado usa dados rotulados com saída conhecida (e-mails já classificados como reclamação, elogio ou dúvida). Não supervisionado não usa rótulos: agrupa clientes por semelhança (clusterização) e detecta anomalias, como transações incomuns sem exemplos de fraude. Por reforço, um agente aprende ações maximizando recompensas, como controlar a climatização economizando energia.

Rotulagem de dados

Treinar um modelo supervisionado sem rótulos esbarra no custo e no tempo da rotulagem, que é um gargalo comum. Rótulos de qualidade costumam ser criados ou validados por especialistas humanos do domínio; deixar o próprio modelo rotular sem revisão pode propagar erros.

Tipos de dados

Estruturados seguem esquema tabular rígido (tabela de vendas, cadastro em SQL, planilha de estoque). Semiestruturados, como JSON e XML, têm campos marcados mas sem esquema rígido de tabela. Não estruturados incluem PDF, texto livre, áudio, imagem e vídeo, a matéria-prima comum da IA generativa. Dado sintético é uma origem (gerado artificialmente), não um formato.

Embeddings e busca semântica

Embeddings são vetores numéricos em que conteúdos de significado parecido ficam próximos. Permitem a busca semântica, que acha o artigo Encerrar assinatura para a pergunta como cancelar meu plano, sem as mesmas palavras. Também alimentam classificação, agrupamento e recomendações. Em RAG, documentos longos são divididos em trechos (chunking) antes dos embeddings para recuperar só o trecho relevante e caber na janela.

Multimodal e nativamente multimodal

Modelo multimodal processa texto, imagem, áudio e vídeo, por exemplo entendendo a foto de um produto danificado junto com o texto da reclamação, ou descrevendo imagens para pessoas com deficiência visual. Dizer que o Gemini é nativamente multimodal significa que foi treinado com vários tipos de dados juntos; aceitar texto e usar outro modelo para imagens descreve uma solução não nativa.

Famílias de modelos do Google

Gemini é o modelo multimodal, com variantes leves (rápidas e baratas para volume) e avançadas (para raciocínio sofisticado); Gemini Nano roda no dispositivo, mesmo sem conexão. Gemma é a família de modelos abertos do Google, derivada da pesquisa do Gemini. Imagen gera imagens a partir de texto, Veo gera vídeos e Chirp trata de fala. O Google hoje recomenda os modelos de imagem do Gemini na Agent Platform e marcou os endpoints do Imagen como descontinuados, mas o guia do exame ainda cita o Imagen como o modelo de texto para imagem.

Panorama em camadas da IA generativa

Infraestrutura: hardware e software de base, como TPUs, GPUs, servidores e o AI Hypercomputer. Modelos: os algoritmos treinados, como Gemini, Gemma, Imagen e Veo. Plataforma: APIs, gestão de dados e implantação de modelos, como a Gemini Enterprise Agent Platform. Agentes: software que observa, raciocina e usa ferramentas para atingir um objetivo. Aplicações: a parte com que o usuário interage, como o app Gemini, o Google Workspace com Gemini e o Gemini Enterprise.

Agente de IA

Diferente de um chatbot que só responde perguntas de uma lista, o agente raciocina, planeja e usa ferramentas para agir em sistemas. Componentes típicos: um modelo que raciocina e decide os passos, ferramentas e uma camada de orquestração. O Google o descreve como o software que aprende a melhor forma de atingir um objetivo com base nas entradas e nas ferramentas disponíveis.

Pré-treino, ajuste fino e variações

Pré-treino usa grandes volumes de dados gerais; ajuste fino parte de um modelo pronto e usa dados específicos para especializá-lo, alterando pesos (o prompt não altera pesos). Ajuste supervisionado usa pares de entrada e saída, como pergunta e resposta. Ajuste eficiente em parâmetros (adaptadores) mexe em poucos pesos e reduz custo e tempo. RLHF usa avaliações humanas para alinhar o modelo às preferências das pessoas. Destilação treina um modelo menor para imitar um maior, com custo e latência menores.

Ciclo de vida de ML

Sequência típica: ingerir, preparar, treinar, implantar e gerenciar. Pub/Sub recebe eventos em fluxo na ingestão; Dataflow, BigQuery e Dataproc preparam dados. O Model Registry guarda versões de modelos com metadados, e o Model Monitoring detecta desvio (drift) e distorção entre os dados de treino e os de produção, quando o comportamento dos clientes muda e as previsões pioram.

Treino, inferência, on-line e lote

No treino o modelo aprende com dados; na inferência ele é usado para gerar previsões ou conteúdo. Respostas em tempo real a cada solicitação usam um endpoint de previsão on-line; grandes volumes sem exigência de resposta imediata usam previsão em lote. O conjunto de teste separado do treino mede o desempenho em dados novos (generalização).

ML preditivo x IA generativa

Prever cancelamento (churn), recomendar filmes pelo histórico e detectar fraude são aplicações clássicas de ML preditivo. IA generativa cria conteúdo novo: rascunhos de descrições, resumos de relatórios, variações de anúncios para testes A/B, dados sintéticos, casos de teste, explicação de código legado e extração de campos de texto livre. Modelos de difusão geram imagens revertendo um processo de ruído em várias etapas.

Qualidade e governança dos dados

Duplicados, campos vazios e formatos inconsistentes derrubam a qualidade das respostas: precisão, completude, consistência e relevância afetam o desempenho. Antes de treinar com registros de atendimento, dados pessoais como nomes, CPFs e telefones devem ser removidos ou mascarados. Dados de terceiros exigem verificar direitos de uso e licenças.

Macetes para memorizar

💡 Hierarquia

Bonecas russas: IA por fora, ML no meio, IA generativa dentro.

💡 Camadas do panorama

Do chip ao usuário: Infra, Modelos, Plataforma, Agentes, Aplicações (I-M-P-A-A).

💡 Tipos de aprendizado

Rótulo = Supervisionado; sem rótulo = agrupa Sozinho; prêmio = Reforço.

💡 Famílias de modelos

Imagen faz imagem, Veo faz vídeo (eu vejo), Chirp pia (fala), Gemma é a gema aberta, Nano cabe no bolso.

💡 Tipos de dados

Tabela = estruturado; etiqueta (JSON/XML) = semi; o resto (PDF, áudio, foto) = não estruturado.

💡 Ciclo de ML

I-P-T-I-G: Ingerir, Preparar, Treinar, Implantar, Gerenciar. Dado antes de treino, treino antes de implantar.

💡 Prever x gerar

Se o verbo é prever, recomendar ou detectar, é preditivo; se é criar, resumir ou redigir, é generativo.

Tabela de memorização: não confunda

Quando é...	...e quando é
Aprendizado supervisionado Há rótulos e saída conhecida: classificar e-mails já marcados, classificar fraude com exemplos rotulados.	**Aprendizado não supervisionado** Não há rótulos: agrupar clientes por comportamento ou detectar transações incomuns sem exemplos de fraude.
ML preditivo Prever churn, recomendar filmes, detectar fraude, prever vendas a partir do histórico.	**IA generativa** Criar conteúdo novo: textos, resumos, imagens, código, variações de anúncios, dados sintéticos.
Engenharia de prompts Orientar o modelo sem alterá-lo; não muda os pesos.	**Ajuste fino** Adaptar o modelo a uma tarefa ou estilo atualizando parâmetros com dados específicos.
Previsão on-line (endpoint) Aplicativo que precisa de resposta em tempo real a cada solicitação.	**Previsão em lote** Milhões de documentos uma vez por mês, sem exigência de resposta imediata.
Modelo aberto (Gemma) Quer controle e portabilidade para hospedar e ajustar na própria infraestrutura (ainda exige onde executar e tem custo de operação).	**Modelo proprietário gerenciado** Quer reduzir o esforço de operação e consumir por API.
LLM Gera texto novo; pode alucinar e nem sempre cita fontes.	**Mecanismo de busca tradicional** Localiza e ordena documentos e páginas que já existem.
RPA (regras fixas) Automatizar passos repetitivos definidos, sem aprender com dados.	**IA generativa** Interpretar linguagem e criar conteúdo; um assistente com LLM entende variações na forma de perguntar.

Números que caem

Item	Valor
Peso do domínio na prova	**cerca de 30%**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Achar que todo modelo de fundação é um LLM.
Como não cair: LLM é um modelo de fundação de linguagem; há modelos de fundação de imagem, áudio e outras modalidades.

⚠ **Alerta de prova:** Escolher ML é um tipo de IA generativa, que é parte da IA.
Como não cair: A ordem correta é IA contém ML, que contém a IA generativa; não são sinônimos.

⚠ **Alerta de prova:** Marcar ajuste fino é amplo e pré-treino especializa, ou o prompt muda os pesos.
Como não cair: São inversões: pré-treino é amplo, ajuste fino especializa; ajuste fino altera pesos e prompt não.

⚠ **Alerta de prova:** Classificar JSON e XML como dados estruturados ou não estruturados.
Como não cair: Têm campos marcados sem esquema rígido de tabela: são semiestruturados.

⚠ **Alerta de prova:** Marcar dados sintéticos para descrever PDFs, áudios e fotos reais.
Como não cair: Sintético é a origem (gerado artificialmente). PDF, áudio e imagem reais são dados não estruturados.

⚠ **Alerta de prova:** Explicar respostas diferentes para o mesmo prompt com o modelo foi retreinado entre as chamadas.
Como não cair: O modelo não muda a cada chamada; a variação vem da amostragem probabilística com temperatura acima de zero.

⚠ **Alerta de prova:** Achar que, com janela de 32 mil tokens e entrada de 30 mil, a resposta pode ter tamanho ilimitado ou que o modelo ignora a janela.
Como não cair: A janela costuma somar entrada e saída; sobra pouco espaço para a resposta.

⚠ **Alerta de prova:** Definir nativamente multimodal como aceitar texto e usar outro modelo para imagens.
Como não cair: Nativamente multimodal é ter sido treinado com várias modalidades juntas.

⚠ **Alerta de prova:** Trocar as famílias: Veo para imagem estática, Imagen para vídeo, Gemma para fala, Chirp para imagem.
Como não cair: Imagen: imagem. Veo: vídeo. Chirp: fala. Gemma: modelo aberto de linguagem.

⚠ **Alerta de prova:** Responder IA generativa de imagens ou geração de vídeo para prever cancelamentos ou recomendar filmes.
Como não cair: Prever e recomendar a partir do histórico é ML preditivo (classificação ou recomendação).

⚠ **Alerta de prova:** Pôr Gemini, Gemma, Imagen e Veo na camada de agentes ou o app Gemini na camada de modelos.
Como não cair: Modelos são os algoritmos treinados; agentes orquestram modelos com ferramentas; aplicações são o que o usuário final usa.

⚠ **Alerta de prova:** Escolher o maior modelo do catálogo para um chatbot de alto volume com perguntas simples.
Como não cair: Equilibre custo, latência e qualidade: variante leve para volume, avançada só para raciocínio sofisticado.

⚠ **Alerta de prova:** Pensar que o LLM sempre cita a fonte e a busca é que inventa respostas.
Como não cair: Quem pode alucinar é o LLM, e ele nem sempre cita fontes; a busca localiza documentos existentes.

⚠ **Alerta de prova:** Deixar o próprio modelo rotular os dados sem revisão para economizar.
Como não cair: Rótulos de qualidade vêm de especialistas humanos do domínio; o modelo sem revisão pode propagar erros.

Como cai na prova

📝 **Como cai:** Cenário de negócio que pede o tipo de IA ou de aprendizado (churn, agrupar clientes, climatização com recompensas).
Como resolver: Procure a palavra-chave: rótulos, sem rótulos, recompensas, prever ou criar. Elimine as alternativas de geração de conteúdo quando o verbo é prever.

📝 **Como cai:** Pergunta de camada do panorama com uma lista de produtos ou componentes.
Como resolver: Pergunte quem usa: usuário final = aplicações; equipe técnica implantando = plataforma; chip e servidor = infraestrutura; Gemini/Gemma/Imagen/Veo = modelos; algo que raciocina e age com ferramentas = agentes.

📝 **Como cai:** Qual família de modelo do Google atende (imagem, vídeo, fala, aberto, celular).
Como resolver: Associe a saída pedida ao modelo e descarte os que geram outra modalidade.

📝 **Como cai:** Definição conceitual com alternativas invertidas (pré-treino x ajuste fino, treino x inferência, prompt x pesos).
Como resolver: Leia cada alternativa procurando a inversão; a certa mantém a ordem lógica (geral antes do específico, treino antes da inferência).

📝 **Como cai:** Escolha DUAS sobre casos de uso, dados não estruturados ou fatores de custo.
Como resolver: As erradas costumam ser absurdas ou de infraestrutura (cor do console, backup, cabo de rede); marque as duas que tratam de conteúdo, modalidade, custo ou tokens.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
dados já classificados / rotulados	**Aprendizado supervisionado**
agrupar sem categorias definidas / transações incomuns sem rótulos	**Aprendizado não supervisionado (clusterização, detecção de anomalias)**
testando ações e recebendo recompensas	**Aprendizado por reforço**
mesmo sentido com palavras diferentes / significado em números	**Embeddings e busca semântica**
TPUs, GPUs, servidores	**Camada de infraestrutura**
APIs, gestão de dados e implantação de modelos sem administrar servidores	**Camada de plataforma**
observa o ambiente, raciocina e usa ferramentas	**Camada de agentes / agente de IA**
funcionários usam sem saber nada de modelos	**Camada de aplicações**
IA no celular mesmo sem conexão	**Gemini Nano**
modelo aberto do Google para hospedar e ajustar com controle e portabilidade	**Gemma**
previsões pioram meses depois porque o comportamento mudou	**Desvio (drift); Model Monitoring**
modelo menor imitando um maior	**Destilação**
avaliadores humanos indicam as melhores respostas	**RLHF (reforço com feedback humano)**
adaptar sem retreinar todos os pesos	**Ajuste eficiente em parâmetros**

🎯 Se você lembrar de só uma coisa deste capítulo...

- IA contém ML, que contém a IA generativa; LLM é um tipo de modelo de fundação, não o contrário.
- Janela de contexto costuma somar entrada e saída; custo por token cresce com prompt e com resposta.
- Churn, recomendação e fraude são ML preditivo, não IA generativa.
- Camadas: infraestrutura, modelos, plataforma, agentes, aplicações.
- Imagen = imagem, Veo = vídeo, Chirp = fala, Gemma = aberto, Gemini Nano = no dispositivo.

Essenciais para a prova (checklist)

- ☐ Sei ordenar IA, ML, deep learning e IA generativa.
- ☐ Sei diferenciar supervisionado, não supervisionado e reforço por um cenário.
- ☐ Sei classificar dados como estruturados, semiestruturados ou não estruturados.
- ☐ Sei explicar token, janela de contexto e por que o mesmo prompt varia.
- ☐ Sei para que servem embeddings e o chunking em RAG.
- ☐ Sei encaixar qualquer produto em uma das cinco camadas do panorama.
- ☐ Sei qual família de modelo do Google gera imagem, vídeo, fala, roda no celular ou é aberta.
- ☐ Sei distinguir pré-treino, ajuste fino, ajuste eficiente em parâmetros, RLHF e destilação.
- ☐ Sei a ordem do ciclo de ML e o que é drift.
- ☐ Sei quando usar previsão on-line ou em lote.

Questões de revisão do capítulo

R2.1 · Fundamentos de IA Generativa

Meses após a implantação, as previsões de um modelo pioram porque o comportamento dos clientes mudou. O que está acontecendo?

A. Aumento da janela de contexto
B. Troca da interface do console
C. Desvio dos dados em produção
D. Redução do número de tokens

R2.2 · Fundamentos de IA Generativa

Uma empresa quer prever quais clientes vão cancelar o contrato no próximo mês. Esse problema é típico de qual tipo de IA?

A. IA generativa de imagens
B. ML preditivo de classificação
C. Síntese de voz a partir de texto
D. Modelo de geração de vídeo

R2.3 · Fundamentos de IA Generativa

O que caracteriza o deep learning dentro do machine learning?

A. Regras escritas por especialistas
B. Redes neurais com muitas camadas
C. Planilhas com fórmulas prontas
D. Consultas SQL em tabelas fixas

R2.4 · Fundamentos de IA Generativa

Um usuário envia o mesmo prompt duas vezes e recebe respostas um pouco diferentes. Por que isso acontece?

A. O modelo foi retreinado entre as chamadas
B. A internet alterou o texto da resposta
C. O prompt foi corrompido no envio
D. A geração usa amostragem probabilística

Respostas comentadas

R2.1. C) Desvio dos dados em produção

Por quê: O desvio (drift) exige monitorar e retreinar ou ajustar o modelo.

✗ A) Incorreta. Não explica a piora.
✗ B) Incorreta. Não explica a piora.
✗ D) Incorreta. Não explica a piora.

> **Dica de prova:** Modelo piora com o tempo: desvio.

R2.2. B) ML preditivo de classificação

Por quê: Prever cancelamento é classificação preditiva; IA generativa cria conteúdo novo.

✗ A) Incorreta. Cria imagens, não prevê cancelamentos.
✗ C) Incorreta. Gera áudio.
✗ D) Incorreta. Cria vídeos.

> **Dica de prova:** Prever churn: ML preditivo.

R2.3. B) Redes neurais com muitas camadas

Por quê: Deep learning usa redes neurais profundas e é a base dos modelos generativos modernos.

✗ A) Incorreta. Não é aprendizado de máquina.
✗ C) Incorreta. Não é aprendizado de máquina.
✗ D) Incorreta. Não é aprendizado de máquina.

> **Dica de prova:** Deep learning: redes neurais profundas.

R2.4. D) A geração usa amostragem probabilística

Por quê: Modelos generativos escolhem tokens por probabilidade; com temperatura acima de zero, há variação.

✗ A) Incorreta. O modelo não muda a cada chamada.
✗ B) Incorreta. Não é a causa.
✗ C) Incorreta. Não é a causa.

> **Dica de prova:** Respostas diferentes: amostragem probabilística.

03

CAPÍTULO 3 · 35% DA PROVA

Ofertas de IA Generativa do Google Cloud

É o domínio de maior peso e o que mais exige memória de produtos: preciso ligar cada necessidade ao produto certo do Google e conhecer os nomes novos. A prova usa os nomes vigentes: Gemini Enterprise Agent Platform (antes Vertex AI), Agent Studio, Agent Search, Agent Runtime, Gemini Notebook, Gemini Enterprise for Customer Experience e Customer Experience Agent Studio. Quase todo enunciado é um cenário (quem é o usuário, o que precisa, quanto quer desenvolver) com uma resposta certa e três distratores de outros serviços do Google Cloud. Onde mais erro: pares parecidos como Google AI Studio x Agent Studio, Agent Assist x CX Agent Studio x CX Insights, AutoML x treinamento personalizado e funções x extensões. Se eu identificar o perfil (usuário final sem código, analista de negócio, desenvolvedor, cientista de dados) metade da questão está resolvida.

Neste capítulo

18 tópicos-chave · 14 pegadinhas · 8 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

Gemini Enterprise Agent Platform

Nome atual da Vertex AI: plataforma com modelos, ferramentas e controles de segurança e governança para levar IA generativa à produção, com IAM, controles de rede, residência de dados, endpoints regionais, auditoria e SLAs corporativos. É por isso que empresas a escolhem em vez de ferramentas de consumo. Não é a única forma de usar o Gemini: também há o Google AI Studio e o app Gemini.

Model Garden

Catálogo de modelos da Agent Platform, com mais de 200 modelos do Google, de parceiros (como o Claude) e abertos, refletindo a abordagem aberta do Google Cloud. Simplifica ajustar e implantar modelos abertos como o Gemma em um endpoint gerenciado com poucos cliques. Oferece modelos prontos para usar e ajustar, não um treino totalmente personalizado.

Agent Studio

Antes Vertex AI Studio: ambiente visual da Agent Platform para prototipar prompts, comparar respostas de modelos e ajustar parâmetros. Aceita prompts multimodais (texto, imagem, vídeo, PDF), salva e compartilha prompts, gera o código do prompt testado e faz ajuste fino gerenciado de modelos Gemini com exemplos. Também cria agentes em tela visual de baixo código, com exportação da lógica para o Agent Development Kit.

Google AI Studio

Ferramenta gratuita para prototipar rapidamente com a API do Gemini usando uma chave de API, sem projeto corporativo completo. Para produção em escala com governança e controles do Google Cloud, o caminho é o Agent Studio na Agent Platform.

Agent Search, RAG Engine e Vector Search

Agent Search (antes Vertex AI Search) oferece busca com IA e RAG pronto sobre os dados da empresa, com pouco desenvolvimento; indexa sites, documentos e dados estruturados (catálogo tabular) e gera respostas com citações. Repositórios de dados (data stores) fundamentam agentes em conteúdo próprio, como manuais internos. O RAG Engine orquestra ingestão, indexação e recuperação; o Vector Search acha vetores semelhantes entre milhões com baixa latência; o BigQuery tem busca vetorial sem mover os dados; a API de embeddings da Agent Platform gera embeddings de texto e multimodais.

Agent Runtime, Sessions e Memory Bank

Agent Runtime (antes Agent Engine) é o ambiente gerenciado da Agent Platform para implantar, operar e escalar agentes sem administrar servidores. Aceita agentes feitos com frameworks abertos como LangChain, LangGraph, AG2 e LlamaIndex, além do Agent Development Kit. Sessions guardam o contexto de cada conversa e Memory Bank dá memória persistente entre conversas.

ADK, Agent Garden e A2A

Agent Development Kit (ADK) é o framework aberto do Google, em Python, para construir agentes e sistemas multiagente; exige programação. Agent Garden oferece exemplos prontos de agentes e ferramentas para acelerar projetos. O protocolo aberto Agent2Agent (A2A) padroniza a comunicação e a colaboração entre agentes de fornecedores diferentes.

Ferramentas de agentes

Na classificação do Google: extensões conectam o agente a APIs externas e são executadas pelo próprio agente; funções descrevem chamadas que a aplicação cliente executa; repositórios de dados dão acesso a informações; plugins adicionam habilidades. O produto Vertex AI Extensions foi descontinuado em 2026, mas o guia mantém extensões como categoria. Na chamada de função, o modelo gera uma solicitação estruturada que a aplicação executa; a execução de código permite cálculos exatos; o grounding com a Pesquisa Google traz fatos públicos recentes.

Tipos de agente conversacional

Determinístico segue caminhos e ações predefinidos; generativo é definido por objetivos e instruções em linguagem natural com LLMs; híbrido combina os dois, útil quando parte do processo exige fluxo fixo e auditável (autenticação, pagamentos) e parte pede conversa livre.

Gemini Enterprise

Ponto único para funcionários pesquisarem em vários sistemas e usarem agentes sobre os dados da empresa: busca multimodal unificada, conectores nativos para Drive, CRM e chamados que respeitam as permissões de origem. O Workflow Builder (antes Agent Designer) permite que usuários de negócio criem agentes e fluxos sem código. A Gemini Notebook API gerencia por código os notebooks do Gemini Notebook Enterprise.

Gemini Notebook

Antes NotebookLM: assistente de pesquisa que responde com base só nas fontes carregadas pelo usuário, com citações, e gera resumos em áudio no estilo podcast. A versão Gemini Notebook Enterprise roda com os controles de dados, segurança e identidade do Google Cloud. Junto com o Google Workspace com Gemini, é oferta pronta, sem desenvolvimento.

Google Workspace com Gemini

No guia do exame, Gemini for Google Workspace: IA dentro do Gmail (redigir e resumir conversas), Documentos (rascunhos), Planilhas (organizar dados, fórmulas, tabelas), Apresentações (slides e imagens), Meet (notas, resumos, legendas traduzidas e tradução de fala), Chat (resumir conversas), Drive (resumir arquivos de uma pasta) e Google Vids (vídeos com IA). O console de administração libera o Gemini por unidade organizacional para pilotos. O conteúdo da organização não treina modelos fora do domínio sem permissão e o Gemini respeita as permissões existentes.

App Gemini, Gems e Deep Research

O app Gemini é o assistente de conversa de uso geral, disponível também para contas corporativas. Gems são versões personalizadas do Gemini com instruções fixas, mas não implantam aplicações. Deep Research pesquisa dezenas de fontes na web e monta um relatório com referências.

Gemini Enterprise for Customer Experience

Antes Customer Engagement Suite: atendimento omnicanal com IA. Customer Experience Agent Studio (evolução do Dialogflow CX; no guia, Conversational Agents) cria agentes virtuais por voz, chat e telefonia, combinando IA generativa e fluxos determinísticos e transferindo para humano quando preciso. Agent Assist apoia o atendente em tempo real com sugestões de resposta, artigos da base e resumo pós-chamada. Customer Experience Insights (antes Conversational Insights) analisa conversas já gravadas, com sentimento e motivos de contato. Pode rodar sobre o Contact Center as a Service (CCaaS). O AI Commerce Search (antes Vertex AI Search for commerce), hoje parte da mesma oferta, traz busca, navegação e recomendações para varejo.

APIs pré-treinadas

Resolvem tarefas comuns sem treinar modelos e servem como ferramentas para agentes; são descobertas e ativadas na API Library do console. Vision analisa imagens estáticas e lê texto (OCR); Video Intelligence detecta objetos, cenas e textos em vídeos; Natural Language extrai sentimento, categorias e entidades; Speech-to-Text transcreve áudio com modelos como o Chirp; Text-to-Speech gera vozes naturais; Cloud Translation traduz textos e documentos inteiros mantendo a formatação; Document AI classifica e extrai campos, com extratores personalizados para layouts próprios.

AutoML, treinamento personalizado e MLOps

AutoML treina modelos (como classificação com dados tabulares) com pouco conhecimento técnico e sem código de treino; treinamento personalizado dá controle total com o framework preferido (TensorFlow, PyTorch). Agent Platform Pipelines automatiza e repete preparação, treino, avaliação e implantação. Feature Store centraliza e serve features compartilhadas com baixa latência. Colab Enterprise oferece notebooks gerenciados, com Gemini para gerar e explicar código. O serviço de avaliação de IA generativa compara modelos com métricas como coerência e fidelidade às fontes.

Infraestrutura e implantação

Cloud TPU são aceleradores do Google para treinar e servir modelos grandes; o AI Hypercomputer integra TPUs, GPUs, armazenamento, rede e software para IA. A vantagem de ter toda a pilha é a integração e otimização de ponta a ponta. Cloud Run hospeda apps em contêiner com escala automática; inferência em lote atende grandes volumes assíncronos com custo menor; Google Distributed Cloud executa IA em ambiente isolado por soberania; Veo e Lyria (música) estão na Agent Platform.

Compromissos e segurança do Google Cloud

O Google não usa os dados do cliente para treinar ou ajustar modelos sem permissão ou instrução prévia, e oferece indenização de propriedade intelectual para dados de treino e resultados gerados, sob condições. SynthID marca de forma imperceptível conteúdo gerado e permite detectá-lo. Security Command Center dá visibilidade e proteção para modelos, agentes e dados de IA; VPC Service Controls cria perímetro contra exfiltração. Gemini Cloud Assist ajuda a operar a nuvem, Gemini Code Assist é focado em código na IDE, Gemini em Security Operations investiga ameaças e Gemini no BigQuery gera SQL por linguagem natural.

Macetes para memorizar

Nomes novos

Onde era Vertex AI, agora é Agent: Agent Platform, Agent Studio, Agent Search, Agent Runtime (antes Agent Engine). NotebookLM virou Gemini Notebook.

Estúdios

AI Studio = Amostra grátis (protótipo); Agent Studio = Ambiente de produção.

Trio do atendimento

Studio Cria o agente, Assist Ajuda o atendente, Insights Investiga depois.

Ferramentas de agente

Extensão: o agente Executa. Função: o cliente Faz a chamada.

Tipos de agente

Determinístico = trilho; generativo = instrução em linguagem natural; híbrido = trilho onde precisa, conversa onde pode.

APIs pré-treinadas

Vision vê foto, Video Intelligence vê filme, Natural Language lê sentimento, Speech-to-Text ouve, Text-to-Speech fala, Document AI extrai campos.

Treino de modelos

AutoML = automático e pouco código; personalizado = controle total; Pipelines = esteira; Feature Store = despensa de features; Model Registry = arquivo de versões.

Gardens

Model Garden tem modelos; Agent Garden tem agentes de exemplo.

Tabela de memorização: não confunda

Quando é...	...e quando é
Google AI Studio Protótipo rápido e gratuito com a API do Gemini e chave de API.	**Agent Studio (Agent Platform)** Aplicação de produção, em escala, com governança e controles do Google Cloud.
AutoML Pouco conhecimento técnico, sem escrever código de treino (por exemplo, classificador com dados tabulares).	**Treinamento personalizado** Cientistas de dados que querem controle total com o framework de ML preferido.
Customer Experience Agent Studio Agente virtual que atende o cliente por chat, voz ou telefone e transfere para humano.	**Agent Assist** Apoio ao atendente humano em tempo real: sugestões, artigos da base e resumo da conversa.
Agent Assist Durante a conversa, ajudando o atendente.	**Customer Experience Insights** Depois: analisar milhares de chamadas gravadas para motivos de contato e sentimento.

Extensões do agente A chamada à API externa é executada pelo próprio agente.	**Funções (chamada de função)** O modelo gera a solicitação estruturada e quem executa é a aplicação cliente.
Agent Search Busca e RAG sobre os dados da empresa para clientes ou aplicações, com pouco desenvolvimento.	**Gemini Notebook** Pesquisador carrega os próprios documentos e pergunta com respostas baseadas só nessas fontes; resumo em áudio.
Gemini Cloud Assist Entender custos, diagnosticar problemas e gerar configurações no console do Google Cloud.	**Gemini Code Assist** Ajuda com código dentro da IDE.

Números que caem

Item	Valor
Modelos no Model Garden	**mais de 200 modelos do Google, de parceiros e abertos**
Peso do domínio na prova	**cerca de 35% (o maior)**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Indicar Google AI Studio com chave pessoal para empresa regulada levar a aplicação à produção.
Como não cair: AI Studio é gratuito e para protótipo; produção com controles é Agent Studio na Agent Platform.

⚠ **Alerta de prova:** Escolher Gems personalizados ou Gemini Notebook para implantar uma aplicação.
Como não cair: Gems personalizam o assistente do usuário e o Notebook é assistente de pesquisa; nenhum implanta aplicações.

⚠ **Alerta de prova:** Marcar Customer Experience Insights para criar o chatbot do site.
Como não cair: Insights analisa conversas já ocorridas; quem cria o agente virtual é o CX Agent Studio.

⚠ **Alerta de prova:** Usar Cloud Vision API para indexar acervo de vídeos.
Como não cair: Vision analisa imagens estáticas; conteúdo de vídeo ao longo do tempo é Video Intelligence API.

⚠ **Alerta de prova:** Escolher Gemini Code Assist para operar custos e configurações da nuvem.
Como não cair: Code Assist é código na IDE; operar e otimizar recursos no console é Gemini Cloud Assist.

⚠ **Alerta de prova:** Chamar de funções do cliente a API externa que o próprio agente executa.
Como não cair: Se quem executa é o agente, é extensão; se é a aplicação cliente, é função.

⚠ **Alerta de prova:** Marcar AutoML sem código quando o cenário pede controle total do treino, ou Model Garden com um clique.
Como não cair: AutoML reduz o controle; Model Garden entrega modelos prontos. Controle total é treinamento personalizado.

⚠ **Alerta de prova:** Achar que Agent Platform Pipelines treina o modelo sozinho.
Como não cair: Pipelines orquestra etapas de ML; quem treina é AutoML ou treinamento personalizado.

⚠ **Alerta de prova:** Responder que na Agent Platform só há modelos do Google ou que é preciso usar só ferramentas do Google.
Como não cair: Model Garden tem modelos de parceiros e abertos, e o Agent Runtime aceita LangChain, LangGraph, AG2 e LlamaIndex.

⚠ **Alerta de prova:** Indicar Agent Development Kit ou Pipelines com SDK para uma analista que não sabe programar.
Como não cair: Sem código no Gemini Enterprise é o Workflow Builder; ADK é framework para desenvolvedores.

⚠ **Alerta de prova:** Escolher Agent Search ou Workflow Builder para gerenciar notebooks por código.
Como não cair: Notebooks criados, compartilhados e gerenciados por programação: Gemini Notebook API.

⚠ **Alerta de prova:** Marcar agente generativo quando o banco quer fluxo fixo na autenticação e conversa livre nas dúvidas.
Como não cair: Fluxo fixo + conversa livre = agente híbrido; só instruções em linguagem natural = generativo; só caminhos predefinidos = determinístico.

⚠ **Alerta de prova:** Responder que o Agent Studio aceita só texto ou que exige terminal.
Como não cair: O Agent Studio aceita prompts multimodais com PDF, imagem e vídeo na interface visual.

⚠ **Alerta de prova:** Tentar trazer notícias do dia com temperatura zero, mais réplicas ou exportação para o BigQuery.
Como não cair: Fatos públicos atuais vêm de grounding com a Pesquisa Google.

Como cai na prova

📝 **Como cai:** Cenário com uma necessidade de negócio e quatro produtos, três deles serviços de infraestrutura sem relação (Cloud DNS, Cloud NAT, Cloud Armor, Cloud Billing).
Como resolver: Descarte os serviços de rede, custo e armazenamento e escolha o único produto de IA que faz exatamente o verbo pedido (resumir, buscar, transcrever, gerar).

📝 **Como cai:** Escolha entre produtos de IA parecidos (AI Studio x Agent Studio, AutoML x personalizado, Agent Assist x CX Insights).
Como resolver: Identifique o perfil (usuário final, analista sem código, desenvolvedor, cientista de dados) e o momento (protótipo ou produção; durante ou depois da conversa).

📝 **Como cai:** Pergunta Isso é possível? com alternativas Não, só..., Só com hardware próprio, Só fora do Google Cloud.
Como resolver: Em geral a certa é Sim, com o recurso nomeado (frameworks abertos, modelos de parceiros, canal de voz, dados estruturados, controles do console de administração).

📝 **Como cai:** Escolha DUAS ofertas sem desenvolvimento ou produtos de uma suíte.
Como resolver: Sem desenvolvimento são ofertas prontas (Google Workspace com Gemini, Gemini Notebook); descarte ADK, Pipelines e Vector Search, que exigem desenvolvedores.

📝 **Como cai:** Recurso do Workspace para uma tarefa do dia a dia (e-mail, reunião, planilha, slides, vídeo).
Como resolver: Associe o aplicativo à tarefa: Gmail redige e resume e-mails, Meet faz notas e traduz fala, Planilhas organiza dados, Apresentações cria imagens nos slides, Vids cria vídeos, Drive resume pastas.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
produção com IAM, rede, residência de dados e SLA	**Gemini Enterprise Agent Platform**
catálogo com modelos do Google, de parceiros e abertos / implantar Gemma em endpoint	**Model Garden**
testar prompts, comparar modelos, ajustar parâmetros, gerar código do prompt	**Agent Studio**
protótipo rápido com chave de API, sem projeto corporativo	**Google AI Studio**
clientes pesquisam catálogo e políticas em linguagem natural com pouco desenvolvimento	**Agent Search**

funcionários buscam em vários sistemas respeitando permissões	**Gemini Enterprise com conectores**
agente lembra preferências entre conversas	**Memory Bank (Agent Runtime)**
agentes de fornecedores diferentes colaborando	**Protocolo Agent2Agent (A2A)**
sugestões ao atendente em tempo real / resumo pós-chamada	**Agent Assist**
analisar chamadas já gravadas	**Customer Experience Insights**
perguntas sobre os próprios documentos com citações / podcast das apostilas	**Gemini Notebook**
identificar se imagem foi gerada por IA do Google	**SynthID**
medo de processo por direitos autorais	**Indenização de propriedade intelectual do Google Cloud**
ambiente isolado da internet por soberania	**Google Distributed Cloud**

🎯 **Se você lembrar de só uma coisa deste capítulo...**

- Produção com controles corporativos = Agent Platform; protótipo grátis = Google AI Studio.
- Dados do cliente não treinam modelos do Google sem permissão; Workspace com Gemini respeita permissões.
- CX Agent Studio atende o cliente, Agent Assist ajuda o atendente, CX Insights analisa o que já aconteceu.
- Sem código no Gemini Enterprise = Workflow Builder; com código = ADK.
- Extensão é executada pelo agente; função é executada pela aplicação cliente.

Essenciais para a prova (checklist)

☐ Sei os nomes atuais e antigos: Agent Platform (Vertex AI), Agent Studio, Agent Search, Agent Runtime (Agent Engine), Gemini Notebook (NotebookLM), Workflow Builder (Agent Designer).

☐ Sei quando indicar Google AI Studio e quando indicar Agent Studio.

☐ Sei o que fazem Model Garden, Agent Garden, ADK e o protocolo A2A.

☐ Sei separar Agent Search, RAG Engine, Vector Search e busca vetorial no BigQuery.

☐ Sei os três produtos do Gemini Enterprise for Customer Experience e o papel do CCaaS.

☐ Sei associar cada API pré-treinada (Vision, Video Intelligence, Natural Language, Speech, Translation, Document AI) à tarefa.

☐ Sei diferenciar extensões, funções, repositórios de dados e plugins.

☐ Sei diferenciar agente determinístico, generativo e híbrido.

☐ Sei citar os compromissos do Google Cloud: dados sem treino sem permissão, indenização de IP, SynthID.

☐ Sei diferenciar AutoML, treinamento personalizado, Pipelines, Feature Store e Model Registry.

Questões de revisão do capítulo

R3.1 · Ofertas de IA Generativa do Google Cloud

Uma URA precisa falar com os clientes usando vozes naturais geradas a partir de texto. Qual serviço usar?

A. Speech-to-Text
B. Cloud Vision
C. Text-to-Speech
D. Document AI

R3.2 · Ofertas de IA Generativa do Google Cloud

Uma empresa quer que o agente virtual atenda também clientes por telefone, com fala natural. Isso é possível com o Customer Experience Agent Studio?

A. Só após treinar um modelo do zero
B. Não, o produto atende só texto
C. Sim, com canal de voz e telefonia
D. Só em aplicativos móveis offline

R3.3 · Ofertas de IA Generativa do Google Cloud

Uma central de atendimento quer sugerir respostas em tempo real aos atendentes humanos durante as conversas. Qual solução do Google Cloud atende?

A. Gemma
B. Cloud Armor
C. Imagen
D. Agent Assist

R3.4 · Ofertas de IA Generativa do Google Cloud

Uma empresa pergunta se os dados que envia aos modelos da Agent Platform serão usados pelo Google para treinar modelos. Qual é o compromisso do Google Cloud sobre esses dados?

A. Publica-os após trinta dias de uso
B. Não os usa para treinar sem permissão
C. Só os protege em contas gratuitas
D. Usa-os para treinar o modelo público

Respostas comentadas

R3.1. C) Text-to-Speech

Por quê: O Text-to-Speech gera áudio com vozes naturais a partir de texto.

✗ A) Incorreta. Converte fala em texto.

✗ B) Incorreta. Analisa imagens.

✗ D) Incorreta. Extrai dados de documentos.

> **Dica de prova:** Texto em voz: Text-to-Speech.

R3.2. C) Sim, com canal de voz e telefonia

Por quê: O Customer Experience Agent Studio (antes Conversational Agents) atende por voz e texto, com integração a telefonia.

✗ A) Incorreta. Não é necessário.

✗ B) Incorreta. Também atende voz.

✗ D) Incorreta. Não é o caso.

> **Dica de prova:** Agente por telefone: canal de voz.

R3.3. D) Agent Assist

Por quê: O Agent Assist faz parte do Gemini Enterprise for Customer Experience (antes Customer Engagement Suite) e apoia atendentes em tempo real.

✗ A) Incorreta. É uma família de modelos abertos.

✗ B) Incorreta. Protege aplicações web.

✗ C) Incorreta. Gera imagens.

> **Dica de prova:** Apoio ao atendente humano: Agent Assist.

R3.4. B) Não os usa para treinar sem permissão

Por quê: Pela restrição de treinamento prevista nos termos do Google Cloud, o Google não usa os dados do cliente para treinar ou ajustar modelos sem permissão ou instrução prévia.

✗ A) Incorreta. Não é o compromisso.

✗ C) Incorreta. Não é o compromisso.

✗ D) Incorreta. Contraria o compromisso do Google Cloud.

> **Dica de prova:** Dados do cliente: não treinam modelos sem permissão.

04

CAPÍTULO 4 · 20% DA PROVA

Técnicas para Melhorar a Saída dos Modelos

Aqui eu preciso saber escolher a técnica certa para melhorar a resposta: técnicas de prompt (zero, one e few-shot, papel, instruções do sistema, delimitadores, cadeia de pensamento, encadeamento, ReAct, metaprompting), parâmetros de amostragem (temperatura, top-k, top-p, máximo de tokens, configurações de segurança), grounding e RAG, e quando partir para o ajuste fino. A prova também cobra as limitações dos modelos (alucinação, data de corte, viés, casos de borda, baixa explicabilidade, contas erradas) e como avaliar e manter a qualidade em produção. O enunciado descreve um sintoma e pede a correção; quase sempre aparece aumentar a temperatura como distrator. Onde mais erro: confundir few-shot com one-shot, cadeia de pensamento com encadeamento de prompts e os tipos de grounding (dados próprios, de terceiros e da web).

Neste capítulo

18 tópicos-chave · 14 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

Zero-shot, one-shot e few-shot

Zero-shot pede a tarefa sem exemplos; one-shot inclui um único exemplo de entrada e saída; few-shot inclui alguns exemplos para mostrar o padrão esperado. Exemplos devem ser claros, corretos, variados e representativos; repetir o mesmo exemplo ou usar respostas erradas ensina o padrão errado. Exemplos demais (como 50) consomem tokens e contexto e deixam as respostas lentas e caras.

Elementos de um bom prompt

Prompts eficazes especificam tarefa, contexto, formato, público, tom e limites; um pedido vago como faça um relatório gera saída genérica. Podem dizer explicitamente o que evitar (por exemplo, não incluir preços). Enviar só o contexto relevante melhora a qualidade e reduz custos; páginas irrelevantes viram ruído.

Definição de papel e instruções do sistema

Começar com Você é um consultor tributário experiente define um papel ou persona que orienta tom, estilo e nível de detalhe. Instruções do sistema definem comportamento, tom e limites persistentes em todas as conversas, sem que o usuário precise repeti-los. Combinadas com few-shot, dão regras fixas e exemplos de formato.

Delimitadores e modelos de prompt

Delimitadores, como aspas triplas ou marcações, separam instruções do conteúdo a ser processado (por exemplo, o e-mail a resumir). Para centenas de prompts parecidos, um modelo de prompt com variáveis garante consistência em escala.

Cadeia de pensamento (chain-of-thought)

Pedir que o modelo raciocine passo a passo antes da resposta final ajuda em problemas de lógica com várias etapas. O custo e a latência sobem porque o modelo gera mais tokens de saída.

Encadeamento de prompts (prompt chaining)

Divide uma tarefa complexa em etapas mais simples e controláveis: um prompt extrai dados, o próximo analisa, o seguinte redige. Resumir um relatório grande demais por partes e depois consolidar é uma forma de encadeamento.

ReAct e metaprompting

ReAct (raciocinar e agir) alterna pensar sobre o problema, chamar uma ferramenta e observar o resultado até chegar à resposta; é útil para agentes. Metaprompting é usar prompts para o modelo gerar, criticar ou melhorar outros prompts; o Agent Studio oferece ferramentas de IA para escrever e refinar prompts.

Temperatura

Temperatura baixa favorece os tokens mais prováveis e gera respostas consistentes e previsíveis, ideal para classificação e chatbot jurídico. Temperatura alta favorece variedade e criatividade, como ideias de slogans. Temperatura zero dá consistência, mas não elimina alucinações nem garante fatos corretos.

Top-k, top-p e máximo de tokens

Top-k limita a escolha aos k tokens mais prováveis; top-k igual a 1 escolhe sempre o mais provável (escolha gulosa). Top-p limita a amostragem aos tokens cuja probabilidade acumulada atinge p; top-p baixo reduz a diversidade e não define o tamanho da resposta. Respostas cortadas no meio indicam máximo de tokens de saída baixo demais. Ao testar, altere um parâmetro por vez e registre os valores.

Saída estruturada

Saída estruturada com esquema JSON faz o modelo responder em campos fixos, facilitando integrar a sistemas. Mesmo assim, uma saída que aciona sistemas (como criar pedidos) deve ter formato e valores validados antes do uso.

Configurações de segurança

Ajustam limites de bloqueio por categoria de dano: um app infantil pede filtros mais rigorosos. Devem ser calibradas ao caso de uso: se um assistente médico bloqueia perguntas legítimas sobre doenças, calibre, não desligue os controles.

Grounding e seus tipos

Grounding conecta as respostas a fontes, reduzindo alucinações. O guia diferencia grounding com dados próprios da empresa (políticas internas, via Agent Search ou RAG Engine), com dados de terceiros (dados licenciados de um provedor externo) e com dados do mundo, como os resultados da Pesquisa Google. Citações permitem checar a origem e ajudam a identificar alucinações.

RAG

Recupera trechos relevantes de fontes atualizadas e os entrega ao modelo; é a técnica para dados que mudam muito, como políticas de RH, em vez de retreinar a cada mudança. Em bases grandes e dinâmicas, recupera só os trechos úteis, enquanto a janela de contexto longa serve para poucos documentos. Instruir a responder só com o contexto recuperado e a dizer que não sabe quando faltar informação reduz alucinações. RAG só é tão bom quanto as fontes: manuais desatualizados geram respostas erradas.

Ajuste fino supervisionado

Usa pares de exemplo para adaptar o modelo a tarefas, estilo e estrutura fixos quando prompts com exemplos não bastaram. Depende de dados de qualidade e em quantidade suficiente: poucos exemplos com erros fazem o modelo herdar os problemas. Para conhecimento atual e estilo da marca ao mesmo tempo, combine RAG para as normas e ajuste fino para o estilo.

Ordem de esforço

Engenharia de prompts é o primeiro passo e é iterativa: testar, avaliar e refinar. Grounding e ajuste fino vêm quando necessários; treinar do zero é o caminho mais caro.

Limitações dos modelos

Alucinação é resposta incorreta apresentada com confiança como fato. Data de corte: sem grounding em fontes atuais, o modelo não conhece eventos posteriores ao treino. Viés reflete os dados (imagens que mostram sempre o mesmo perfil em cargos de liderança) e exige avaliar e ajustar. Casos de borda pouco representados geram falhas em pedidos raros. LLMs têm baixa explicabilidade interna, e erram contas: use ferramenta de cálculo.

Avaliação e supervisão humana

Para saber se uma mudança melhorou, use um conjunto de testes fixo com métricas; a métrica de fundamentação (groundedness) verifica se a resposta se apoia nos documentos recuperados. Feedback de útil ou não útil dos usuários orienta melhorias. Em decisões de alto impacto, como diagnósticos, e em relatórios com números críticos, exija revisão humana e confira nas fontes.

Gestão de modelos em produção

O Google recomenda acompanhar KPIs, versionar, rastrear o desempenho, detectar desvios e planejar atualizações de modelo e de segurança. O Model Registry guarda as versões e permite voltar à anterior se a nova piorar. Uma versão nova do modelo deve ser avaliada com os KPIs atuais antes de ir para produção, sem pausar o monitoramento.

Macetes para memorizar

💡 Shots

Conte os exemplos: 0 = zero-shot, 1 = one-shot, alguns = few-shot.

💡 Cadeia x encadeamento

Cadeia de pensamento = um prompt que pensa em voz alta; encadeamento = vários prompts em fila.

💡 Temperatura

Fria é firme (consistente); quente é criativa. Zero não é infalível.

💡 RAG x ajuste fino

RAG traz o que o modelo deve SABER; ajuste fino muda COMO ele escreve.

💡 Tipos de grounding

Meu (dados próprios), comprado (dados de terceiros), do mundo (Pesquisa Google).

💡 Ordem de esforço

Prompt primeiro, grounding depois, ajuste fino se precisar, treino do zero quase nunca.

💡 Top-k e top-p

Top-k conta tokens (k=1 pega o primeiro); top-p soma probabilidade. Nenhum dos dois mede tamanho.

Tabela de memorização: não confunda

Quando é...	...e quando é
One-shot O prompt traz um único exemplo de entrada e saída antes do pedido.	**Few-shot** O prompt traz alguns exemplos (por exemplo, três) do formato desejado.
Cadeia de pensamento Um único pedido para o modelo raciocinar passo a passo antes de responder.	**Encadeamento de prompts** Vários prompts em sequência, cada um com uma etapa (extrair, analisar, redigir).

RAG Conhecimento atualizado que muda com frequência (normas, políticas, manuais).	**Ajuste fino** Estilo, formato e estrutura fixos que prompts não resolveram.
Temperatura baixa Classificação, respostas previsíveis e consistentes (jurídico).	**Temperatura alta** Brainstorm e variedade criativa (slogans).
Grounding com dados próprios Dados gerados pela própria empresa, como políticas internas.	**Grounding com dados de terceiros** Dados licenciados de um provedor externo especializado, como dados de mercado.
RAG Bases muito grandes e dinâmicas: recupera só os trechos úteis.	**Janela de contexto longa** Poucos documentos que cabem inteiros na solicitação.
ReAct Agente alterna raciocínio, chamada de ferramenta e observação em ciclos.	**Cadeia de pensamento** Só raciocínio intermediário em etapas, sem usar ferramentas.

Números que caem

Item	Valor
Top-k igual a 1	**escolhe sempre o token mais provável (escolha gulosa)**
Top-p baixo (por exemplo, 0,3 em vez de 0,95)	**escolhas entre menos tokens prováveis, menos diversidade**
Peso do domínio na prova	**cerca de 20%**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Achar que temperatura zero garante fatos corretos ou que nunca alucina.
Como não cair: Temperatura zero dá consistência, não precisão; grounding e avaliação continuam necessários.

⚠ **Alerta de prova:** Usar aumentar a temperatura como solução para qualidade, contexto irrelevante, fatos atuais ou bloqueios.
Como não cair: Temperatura só controla variação; ela não traz conhecimento, não separa instruções e não trata filtros.

⚠ **Alerta de prova:** Culpar temperatura, top-k ou nível de segurança por respostas cortadas no meio.
Como não cair: Resposta cortada é limite de máximo de tokens de saída baixo demais.

⚠ **Alerta de prova:** Achar que top-p baixo gera mais tokens raros ou respostas mais longas.
Como não cair: Top-p baixo restringe a menos tokens prováveis (menos diversidade) e não define tamanho.

⚠ **Alerta de prova:** Usar ajuste fino para normas que mudam ou retreinar o modelo a cada mudança.
Como não cair: Conhecimento que muda é RAG; ajuste fino adapta estilo e exigiria retreino a cada mudança.

⚠ **Alerta de prova:** Classificar dados licenciados de um fornecedor como grounding com dados próprios ou com dados públicos da web.
Como não cair: Próprios são gerados pela empresa; web é conhecimento do mundo (Pesquisa Google); licenciados de fornecedor são dados de terceiros.

⚠ **Alerta de prova:** Usar grounding na Pesquisa Google para responder sobre políticas internas.
Como não cair: Pesquisa Google traz informação pública; políticas internas pedem grounding nos dados da empresa.

⚠ **Alerta de prova:** Desligar os controles de segurança porque o assistente bloqueou perguntas legítimas.
Como não cair: Calibre as configurações ao caso de uso; desligar expõe a riscos.

⚠ **Alerta de prova:** Marcar few-shot quando há um único exemplo, ou cadeia de pensamento quando há exemplos.
Como não cair: Conte os exemplos: zero, um (one-shot) ou alguns (few-shot). Cadeia de pensamento pede raciocínio, não exemplos.

⚠ **Alerta de prova:** Acreditar que o grounding elimina a revisão ou que o modelo corrige manuais desatualizados.
Como não cair: O grounding depende da qualidade das fontes; a revisão continua necessária.

⚠ **Alerta de prova:** Mudar vários parâmetros ou fatores de uma vez para acelerar os testes.
Como não cair: Mude um por vez e compare com um conjunto de testes fixo, senão não dá para saber o que funcionou.

⚠ **Alerta de prova:** Adotar a versão nova do modelo sem testar, ou congelar a antiga para sempre.
Como não cair: Avalie a versão nova com os KPIs atuais; versões antigas são descontinuadas e deixam de receber correções.

⚠ **Alerta de prova:** Supor que o ajuste fino corrige sozinho dados poucos e com erros.
Como não cair: O modelo herda os problemas dos dados de ajuste.

⚠ **Alerta de prova:** Fazer retreino completo do zero para desfazer uma versão ruim em produção.
Como não cair: Com versionamento no Model Registry, basta voltar à versão anterior registrada.

Como cai na prova

📝 **Como cai:** Sintoma de saída ruim (inconsistente, cortada, genérica, longa e técnica, inventada) e quatro ajustes possíveis.
Como resolver: Associe o sintoma ao controle exato: inconsistente = temperatura baixa; cortada = máximo de tokens; genérica = contexto e objetivo; longa e técnica = tamanho, público e tom; inventada = grounding e admitir que não sabe. Descarte temperatura alta.

📝 **Como cai:** Descrição de um prompt e pergunta Que técnica é essa?
Como resolver: Procure a marca: número de exemplos, Você é..., passo a passo, etapas em vários prompts, ciclo com ferramenta, prompt que gera prompt.

📝 **Como cai:** Escolha entre RAG, ajuste fino, contexto longo e retreino.
Como resolver: Pergunte se o problema é conhecimento (atual ou interno = RAG/grounding) ou comportamento (estilo e formato fixos = ajuste fino). Se o volume é pequeno, contexto longo serve.

📝 **Como cai:** Qual tipo de grounding (dados próprios, terceiros, Pesquisa Google).
Como resolver: Veja quem gera o dado: a empresa, um fornecedor licenciado ou a web pública.

📝 **Como cai:** Prática recomendada de avaliação ou de produção (versão nova, queda de qualidade, comparação de prompts).
Como resolver: A certa sempre mede de forma controlada (conjunto fixo, KPIs, um parâmetro por vez) e mantém monitoramento e versionamento; descarte confiar na primeira impressão ou testar com um exemplo.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
sem nenhum exemplo	**Zero-shot**
um único exemplo antes do pedido	**One-shot**
Você é um especialista...	**Definição de papel**

regras de tom e segurança em todas as conversas sem repetir	**Instruções do sistema**
separar instrução do texto a processar	**Delimitadores**
raciocine passo a passo	**Cadeia de pensamento (mais tokens, mais custo)**
primeiro prompt extrai, segundo analisa, terceiro redige	**Encadeamento de prompts**
pensar, chamar ferramenta, observar	**ReAct**
modelo escreve e melhora o prompt	**Metaprompting**
mesma entrada recebe categorias diferentes	**Reduzir a temperatura**
resposta cortada no meio	**Máximo de tokens de saída**
JSON com campos fixos	**Saída estruturada com esquema**
versão de software que já foi substituída / fatos recentes	**Data de corte; grounding em fontes atuais**
afirmou com confiança algo que não existe	**Alucinação**

🎯 **Se você lembrar de só uma coisa deste capítulo...**

- Aumentar a temperatura quase nunca é a resposta certa, exceto para criatividade.
- Temperatura zero não elimina alucinação; grounding e avaliação continuam.
- Dados que mudam = RAG; estilo fixo = ajuste fino; os dois juntos se precisar dos dois.
- Resposta cortada = máximo de tokens de saída.
- Em alto impacto, humano no ciclo e conferência nas fontes.

Essenciais para a prova (checklist)

- ☐ Sei identificar zero-shot, one-shot e few-shot e as boas práticas de exemplos.
- ☐ Sei quando usar definição de papel, instruções do sistema e delimitadores.
- ☐ Sei diferenciar cadeia de pensamento, encadeamento de prompts, ReAct e metaprompting.
- ☐ Sei o efeito de temperatura, top-k, top-p e máximo de tokens.
- ☐ Sei os três tipos de grounding citados no guia.
- ☐ Sei escolher entre RAG, ajuste fino e janela de contexto longa.
- ☐ Sei nomear alucinação, data de corte, viés, casos de borda e baixa explicabilidade.
- ☐ Sei como avaliar mudanças: conjunto fixo, métrica de fundamentação, um parâmetro por vez.
- ☐ Sei por que versionar no Model Registry e avaliar versões novas com KPIs.

Questões de revisão do capítulo

R4.1 · Técnicas para Melhorar a Saída dos Modelos

Um modelo classifica chamados em categorias fixas, mas a mesma entrada recebe categorias diferentes. Qual ajuste ajuda?

A. Retirar as categorias do prompt
B. Usar uma temperatura mais baixa
C. Usar uma temperatura mais alta
D. Pedir respostas mais detalhadas

R4.2 · Técnicas para Melhorar a Saída dos Modelos

Uma equipe deve decidir entre enviar documentos inteiros em uma janela de contexto longa ou usar RAG. Qual é uma vantagem do RAG em bases muito grandes e dinâmicas?

A. Dispensa uma base de documentos
B. Recupera só os trechos úteis
C. Dispensa o uso de um modelo
D. Garante respostas livres de erro

R4.3 · Técnicas para Melhorar a Saída dos Modelos

Um sistema usa o JSON gerado pelo modelo para criar pedidos. Qual cuidado é necessário antes de usar a saída?

A. Confiar na saída sem verificação
B. Aumentar a temperatura
C. Validar formato e valores
D. Desligar os logs do sistema

R4.4 · Técnicas para Melhorar a Saída dos Modelos

Um técnico envia a foto de um painel com erro junto com a pergunta "o que significa este código?". Que tipo de prompt é esse?

A. Ajuste fino supervisionado
B. Prompt multimodal
C. Prompt zero-shot só de texto
D. Sequência de parada

Respostas comentadas

R4.1. B) Usar uma temperatura mais baixa

Por quê: Tarefas determinísticas, como classificação, se beneficiam de temperatura baixa.

✗ A) Incorreta. Piora a classificação.
✗ C) Incorreta. Aumenta a variação.
✗ D) Incorreta. Não resolve a inconsistência.

> **Dica de prova:** Classificação inconsistente: temperatura baixa.

R4.2. B) Recupera só os trechos úteis

Por quê: O RAG escala para bases grandes e dinâmicas; contexto longo é útil para poucos documentos.

✗ A) Incorreta. O RAG depende de uma base de documentos.
✗ C) Incorreta. O modelo continua necessário.
✗ D) Incorreta. Não há garantia absoluta.

> **Dica de prova:** Base grande e dinâmica: RAG.

R4.3. C) Validar formato e valores

Por quê: Saídas de modelos devem ser validadas antes de acionar sistemas.

✗ A) Incorreta. O modelo pode errar.
✗ B) Incorreta. Aumenta a variação.
✗ D) Incorreta. Reduz a rastreabilidade.

> **Dica de prova:** Saída que aciona sistemas: validar.

R4.4. B) Prompt multimodal

Por quê: Prompts multimodais combinam texto com imagens, áudio ou vídeo.

✗ A) Incorreta. Não é prompt.
✗ C) Incorreta. Inclui imagem.
✗ D) Incorreta. É parâmetro.

> **Dica de prova:** Foto e pergunta juntas: prompt multimodal.

DIA DA PROVA

Estratégia de prova

A prova é para líderes, não para engenheiros: quase toda questão é um cenário de negócio curto que pede o produto, a técnica ou a prática mais adequada. Ela cobra os nomes vigentes dos produtos (Gemini Enterprise Agent Platform no lugar de Vertex AI, Agent Studio, Agent Search, Agent Runtime, Gemini Notebook, Gemini Enterprise for Customer Experience). Os distratores costumam ser serviços do Google Cloud sem relação com IA ou produtos de IA vizinhos com função parecida. Nos cenários de técnica, aumentar a temperatura é o distrator clássico; nas de estratégia, as erradas são extremas (proibir, aprovar tudo, desligar logs). A resposta certa é a que atende exatamente ao perfil do usuário e à necessidade descrita, com o menor esforço e com governança.

Como responder

✓ Gerencie o tempo: são 50 a 60 questões de múltipla escolha em 90 minutos, menos de dois minutos por questão; marque as dúvidas e volte no fim.

✓ Leia o enunciado procurando o perfil (usuário final, analista sem código, desenvolvedor, cientista de dados) e o momento (protótipo ou produção, durante ou depois do atendimento) antes de olhar as alternativas.

✓ Elimine primeiro os serviços sem relação com IA (Cloud DNS, Cloud NAT, Cloud Armor, Cloud Billing, Cloud Scheduler); normalmente sobram uma ou duas opções.

✓ Traduza nomes antigos para os atuais: se pensar em Vertex AI, procure Agent Platform; NotebookLM é Gemini Notebook; Agent Engine é Agent Runtime.

✓ Em questões de técnica, desconfie de aumentar a temperatura e de soluções de força bruta (retreinar do zero, trocar de modelo a cada erro).

✓ Em Escolha DUAS, confirme que as duas marcadas pertencem ao mesmo critério pedido e descarte as alternativas absurdas.

✓ Distribua o estudo pelos pesos: Ofertas do Google Cloud (cerca de 35%) e Fundamentos (cerca de 30%) concentram a maior parte da prova.

✓ Entre duas alternativas razoáveis, prefira a que mantém governança, supervisão humana e segurança.

Erros que derrubam candidatos

✗ Confundir Google AI Studio (protótipo grátis) com Agent Studio (produção na Agent Platform).

✗ Trocar Agent Assist, Customer Experience Agent Studio e Customer Experience Insights.

✗ Responder com nomes antigos ou achar que a Agent Platform só tem modelos do Google.

✗ Acreditar que temperatura zero elimina alucinação ou usar ajuste fino para dados que mudam.

✗ Classificar churn, recomendação e fraude como IA generativa.

✗ Confundir necessidades com recursos e equidade com explicabilidade.

✗ Ignorar as palavras sem código ou com controle total, que decidem entre Workflow Builder, ADK, AutoML e treinamento personalizado.

Na semana da prova

- ☐ Refaça as provas práticas e anote cada erro na forma de par confundido (por exemplo, Agent Assist x CX Insights).
- ☐ Revise a tabela de nomes antigos e atuais dos produtos até falar sem pensar.
- ☐ Revise os gatilhos e macetes de cada domínio, com foco em Ofertas e Fundamentos.
- ☐ Repasse SAIF, princípios de IA responsável e os parâmetros temperatura, top-k, top-p e máximo de tokens.
- ☐ Na véspera, só revisão leve dos itens de lembrar; no dia, controle o tempo para fechar as 50 a 60 questões nos 90 minutos.

REVISÃO FINAL

Folha de cola

Leia na véspera e na manhã da prova: os gatilhos e as pegadinhas mais importantes de cada domínio.

1. Estratégias de Negócio para Soluções de IA Generativa

‣ provocar respostas perigosas de propósito antes do lançamento → **Red teaming**

‣ instruções maliciosas para ignorar as regras → **Injeção de prompt; Model Armor**

‣ instrução escondida em documento recuperado → **Injeção indireta de prompt**

‣ exemplos manipulados nos dados de treino → **Envenenamento de dados**

‣ mascarar CPFs e dados de saúde → **Sensitive Data Protection**

⚠ Dados de treino manipulados é envenenamento de dados, risco específico de ML.

⚠ Conteúdo recuperado que manipula o modelo é injeção indireta de prompt.

⚠ Esses itens são necessidades; recursos são pessoas, dinheiro e tempo.

2. Fundamentos de IA Generativa

‣ dados já classificados / rotulados → **Aprendizado supervisionado**

‣ agrupar sem categorias definidas / transações incomuns sem rótulos → **Aprendizado não supervisionado (clusterização, detecção de anomalias)**

‣ testando ações e recebendo recompensas → **Aprendizado por reforço**

‣ mesmo sentido com palavras diferentes / significado em números → **Embeddings e busca semântica**

‣ TPUs, GPUs, servidores → **Camada de infraestrutura**

⚠ LLM é um modelo de fundação de linguagem; há modelos de fundação de imagem, áudio e outras modalidades.

⚠ A ordem correta é IA contém ML, que contém a IA generativa; não são sinônimos.

⚠ São inversões: pré-treino é amplo, ajuste fino especializa; ajuste fino altera pesos e prompt não.

3. Ofertas de IA Generativa do Google Cloud

‣ produção com IAM, rede, residência de dados e SLA → **Gemini Enterprise Agent Platform**

‣ catálogo com modelos do Google, de parceiros e abertos / implantar Gemma em endpoint → **Model Garden**

‣ testar prompts, comparar modelos, ajustar parâmetros, gerar código do prompt → **Agent Studio**

‣ protótipo rápido com chave de API, sem projeto corporativo → **Google AI Studio**

‣ clientes pesquisam catálogo e políticas em linguagem natural com pouco desenvolvimento → **Agent Search**

⚠ AI Studio é gratuito e para protótipo; produção com controles é Agent Studio na Agent Platform.

⚠ Gems personalizam o assistente do usuário e o Notebook é assistente de pesquisa; nenhum implanta aplicações.

⚠ Insights analisa conversas já ocorridas; quem cria o agente virtual é o CX Agent Studio.

4. Técnicas para Melhorar a Saída dos Modelos

‣ sem nenhum exemplo → **Zero-shot**

‣ um único exemplo antes do pedido → **One-shot**

‣ Você é um especialista... → **Definição de papel**

‣ regras de tom e segurança em todas as conversas sem repetir → **Instruções do sistema**

‣ separar instrução do texto a processar → **Delimitadores**

⚠ Temperatura zero dá consistência, não precisão; grounding e avaliação continuam necessários.

⚠ Temperatura só controla variação; ela não traz conhecimento, não separa instruções e não trata filtros.

⚠ Resposta cortada é limite de máximo de tokens de saída baixo demais.

GABARITO

Respostas do diagnóstico

D1. A) Alinha expectativas e decisões → revise o capítulo 1

Por quê: Critérios de sucesso e KPIs definidos cedo alinham partes interessadas.

✗ B) Incorreta. Não há garantia.
✗ C) Incorreta. Os riscos continuam.
✗ D) Incorreta. O acompanhamento continua.

> **Dica de prova:** Definir sucesso antes: alinhamento.

D2. C) Estender a segurança existente à IA → revise o capítulo 1

Por quê: O SAIF recomenda estender fundações de segurança fortes ao ecossistema de IA.

✗ A) Incorreta. Aumenta riscos.
✗ B) Incorreta. Aumenta riscos.
✗ D) Incorreta. Enfraquece a segurança.

> **Dica de prova:** Controles atuais aplicados à IA: fundações de segurança.

D3. A) Envenenamento de dados → revise o capítulo 1

Por quê: O envenenamento de dados é um risco específico de sistemas de ML tratado pelo SAIF.

✗ B) Incorreta. Busca indisponibilidade.
✗ C) Incorreta. Ataca credenciais.
✗ D) Incorreta. Ataca bancos de dados.

> **Dica de prova:** Dados de treino manipulados: envenenamento.

D4. D) Sobra pouco espaço para a resposta → revise o capítulo 2

Por quê: A janela de contexto costuma limitar tokens de entrada e de saída somados.

✗ A) Incorreta. Não é o comportamento.
✗ B) Incorreta. A janela limita o total.
✗ C) Incorreta. O limite continua valendo.

> **Dica de prova:** Entrada grande: pouco espaço para a saída.

D5. C) Para recuperar trechos relevantes → revise o capítulo 2

Por quê: A divisão em trechos (chunking) melhora a recuperação e cabe na janela de contexto.

✗ A) Incorreta. Não é o objetivo.
✗ B) Incorreta. Não é o objetivo.
✗ D) Incorreta. Não é o objetivo.

> **Dica de prova:** Dividir documentos: recuperar trechos relevantes.

D6. D) Gemma → revise o capítulo 2

Por quê: Gemma é a família de modelos abertos do Google, derivada da pesquisa do Gemini.

✗ A) Incorreta. Gera imagens.
✗ B) Incorreta. Gera vídeos.
✗ C) Incorreta. Trata de fala.

> **Dica de prova:** Modelo aberto do Google: Gemma.

D7. C) Gemini Enterprise → revise o capítulo 3

Por quê: O Gemini Enterprise reúne busca corporativa, assistente e agentes conectados aos sistemas da empresa.

✗ A) Incorreta. Guarda dados, sem busca inteligente.
✗ B) Incorreta. É um recurso de rede.
✗ D) Incorreta. Automatiza entregas de software.

> **Dica de prova:** Busca e agentes para funcionários: Gemini Enterprise.

D8. A) Agent Studio → revise o capítulo 3

Por quê: O Agent Studio (antes Vertex AI Studio) é o ambiente da Agent Platform para prototipar prompts, comparar modelos e ajustar parâmetros.

✗ B) Incorreta. Resolve nomes de domínio.
✗ C) Incorreta. Gerencia custos.
✗ D) Incorreta. Agenda tarefas.

> **Dica de prova:** Prototipar prompts: Agent Studio.

D9. D) Grounding com a Pesquisa Google → revise o capítulo 3

Por quê: O grounding com a Pesquisa Google conecta as respostas a informações públicas recentes.

✗ A) Incorreta. Não traz dados atuais.
✗ B) Incorreta. Não traz dados atuais.
✗ C) Incorreta. Não traz dados atuais.

> **Dica de prova:** Fatos atuais da web: grounding com a Pesquisa Google.

D10. A) Casos de borda pouco representados → revise o capítulo 4

Por quê: Modelos podem falhar em casos de borda; testes e supervisão ajudam a identificá-los.

✗ B) Incorreta. Não é a causa.
✗ C) Incorreta. Não é a causa.
✗ D) Incorreta. Não é a causa.

> **Dica de prova:** Falha em pedidos raros: casos de borda.

D11. A) A data de corte do modelo → revise o capítulo 4

Por quê: Sem grounding em fontes atuais, o modelo reflete o conhecimento da data de corte.

✗ B) Incorreta. Não é a causa.
✗ C) Incorreta. Não é a causa.
✗ D) Incorreta. Não é a causa.

> **Dica de prova:** Versão desatualizada: data de corte.

D12. D) Baixa explicabilidade interna dos LLMs → revise o capítulo 4

Por quê: LLMs são pouco explicáveis; citações, registros e supervisão ajudam na transparência.

✗ A) Incorreta. Não é a limitação.
✗ B) Incorreta. Não é a limitação.
✗ C) Incorreta. Não é a limitação.

> **Dica de prova:** Explicar o raciocínio do LLM: baixa explicabilidade.

Agora, treine no formato real

Faça as 6 provas completas do curso Google Generative AI Leader: 300 questões inéditas, tempo do exame, explicação de cada alternativa e link da documentação para revisar.

Ver as 6 provas completas e o cupom do mês →

Até o dia da prova, uma revisão por dia no seu feed

Instagram @strecht.simulados

Um post de prova por dia: questão comentada, pegadinhas e o "não confunda" de cada certificação.

Seguir no Instagram →

YouTube Strecht Simulados

Shorts diários com pegadinhas de prova e, aos sábados, um simulado comentado.

Inscrever-se no canal →

Material independente de preparação, sem afiliação ou endosso de Google Cloud. As questões são inéditas e não reproduzem a prova real.