


# Guia de preparação AIF-C01

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.

**5**
capítulos por domínio

**32**
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 AIF-C01: 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 65 questões e 90 minutos, no formato do exame. Referência de aprovação: 70%.

| Domínio (capítulo) | Peso | Neste guia |
|---|---|---|
| 1. Aplicações de modelos de base | 28% | 18 conceitos · 14 pegadinhas |
| 2. Diretrizes de IA responsável | 14% | 18 conceitos · 14 pegadinhas |
| 3. Fundamentos de IA e ML | 20% | 18 conceitos · 14 pegadinhas |
| 4. Fundamentos de IA generativa | 24% | 18 conceitos · 14 pegadinhas |
| 5. Segurança, conformidade e governança para soluções de IA | 14% | 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 · Aplicações de modelos de base

**Uma empresa lançou um agente de autoatendimento para remarcação de consultas. A diretoria quer medir se o agente gera valor para o negócio, e não só se o texto é bem escrito. Qual métrica é a mais alinhada a esse objetivo?**

A. Pontuação BLEU das respostas contra textos de referência
B. Taxa de conclusão de tarefas sem transferência para um atendente
C. Número médio de tokens gerados por resposta do agente
D. Perplexidade do modelo de base medida no conjunto de validação do domínio

D2 · Aplicações de modelos de base

**Uma empresa de atendimento ao cliente quer reduzir o tempo de resposta usando modelos de base na AWS. Eles desejam gerar rascunhos de respostas a partir de e-mails recebidos, mas precisam garantir que nenhum dado sensível dos clientes seja usado para treinar modelos públicos. Qual abordagem é MAIS apropriada?**

A. Armazenar os e-mails em um bucket Amazon S3 público e usar um modelo open-source executado localmente pelos atendentes em seus computadores pessoais
B. Adotar modelo do Bedrock só para inferência, com os e-mails enviados sob controles de privacidade e criptografia
C. Treinar um novo modelo de linguagem do zero em instâncias do Amazon EC2 utilizando os e-mails históricos dos clientes, sem usar serviços gerenciados
D. Exportar os e-mails para um provedor externo de IA generativa e permitir que o provedor utilize os dados

D3 · Aplicações de modelos de base

**Uma equipe quer que um assistente baseado no Amazon Bedrock responda como um atendente da empresa, em português formal e sem opinar sobre concorrentes, em todas as conversas. Onde essas instruções devem ficar para valer de forma consistente?**

A. Numa base de conhecimento com documentos sobre os concorrentes
B. No limite de tokens de saída da resposta
C. No prompt de sistema (system prompt) enviado em cada chamada
D. Na temperatura, definindo um valor intermediário

D4 · Diretrizes de IA responsável

**Uma organização está definindo políticas internas para o uso de IA generativa por seus funcionários. Entre as iniciativas abaixo, qual representa de forma mais clara a aplicação de princípios de IA responsável?**

A. Permitir o uso irrestrito de IA generativa, desde que os funcionários assinem um termo afirmando que assumem responsabilidade total pelos resultados.
B. Limitar o uso de IA generativa só à equipe de TI, sem necessidade de registrar como as ferramentas são usadas ou quais dados são inseridos.
C. Adotar diretrizes com revisão humana de saídas críticas, proteção de dados, transparência e canal de reporte
D. Proibir por completo o uso de ferramentas de IA generativa para evitar riscos legais.

D5 · Diretrizes de IA responsável

**Uma equipe de desenvolvimento está criando um chatbot com Amazon Bedrock para responder dúvidas de clientes. Para seguir diretrizes de IA responsável, eles querem reduzir o risco de o modelo gerar respostas preconceituosas ou discriminatórias. Qual ação é mais alinhada com esse objetivo?**

A. Aumentar o número de parâmetros do modelo para que ele se torne mais preciso e reduza automaticamente cada viés
B. Confiar nas configurações padrão do provedor de modelo, já que esses modelos são treinados para não gerar conteúdo enviesado
C. Remover as referências a dados sensíveis (como gênero ou etnia) dos registros de auditoria para simplificar o monitoramento
D. Adotar moderação de conteúdo e filtros de segurança, com revisão de prompts e respostas para achar vieses

D6 · Diretrizes de IA responsável

**Uma organização deseja garantir que seus sistemas de IA generativa reduzam o risco de produzir conteúdo discriminatório ou ofensivo. Qual prática está mais alinhada às diretrizes de IA responsável para esse objetivo?**

A. Confiar apenas nas políticas de uso do provedor de nuvem, sem processos internos de revisão ou monitoramento
B. Reduzir o tamanho do modelo de IA para minimizar a chance de respostas ofensivas
C. Permitir que os usuários sinalizem problemas, mas não registrar incidentes para evitar burocracia
D. Testar com dados diversos e usar guardrails para monitorar e bloquear respostas inadequadas

D7 · Fundamentos de IA e ML

**Um departamento financeiro recebe notas fiscais digitalizadas e precisa extrair automaticamente campos como CNPJ, data e valores, além das linhas da tabela de itens, para lançar no ERP. Qual serviço gerenciado da AWS é o mais indicado?**

A. Amazon Polly, com leitura em voz alta do conteúdo
B. Amazon Textract, com análise de formulários e tabelas
C. Amazon Rekognition, com detecção de rótulos nas imagens
D. Amazon Comprehend, com análise de sentimento do documento

D8 · Fundamentos de IA e ML

**Um engenheiro de ML está explicando para um time de produto a diferença entre aprendizado supervisionado, não supervisionado e por reforço. Qual das alternativas descreve CORRETAMENTE um cenário típico de aprendizado por reforço em contraste com os outros dois paradigmas?**

A. Um modelo que recebe um grande conjunto de dados rotulados de imagens de gatos e cães, aprende um limite de decisão e prevê diretamente a classe 'gato' ou 'cão' para novas imagens.
B. Um agente que interage com um ambiente, toma ações sequenciais, recebe recompensas ou penalidades e ajusta sua política para maximizar a recompensa acumulada ao longo do tempo.
C. Um algoritmo que recebe apenas dados de transações sem rótulos, agrupa clientes com comportamentos de compra semelhantes, sem saber de antemão quais grupos existem.
D. Um modelo generativo que, a partir de um conjunto de textos de exemplo, aprende a prever a próxima palavra em uma frase, produzindo textos semelhantes aos do conjunto de treinamento.

D9 · Fundamentos de IA e ML

**No contexto de aprendizado de máquina, qual opção descreve melhor o objetivo principal da etapa de 'validação' durante o desenvolvimento de um modelo de classificação?**

A. Coletar e rotular novos dados para melhorar a qualidade do conjunto de treino.
B. Treinar o modelo nos dados de entrada para minimizar a função de perda escolhida.
C. Ajustar hiperparâmetros e comparar versões com dados separados do treino
D. Medir o desempenho do modelo em dados nunca vistos para estimar seu comportamento em produção.

D10 · Fundamentos de IA generativa

**Uma equipe de desenvolvimento quer um ambiente com agentes de IA que transforme um pedido em linguagem natural em especificações (requisitos, design e lista de tarefas) antes de gerar o código. Qual ferramenta, citada no guia do AIF-C01, tem esse foco?**

A. Kiro, com desenvolvimento orientado por especificações
B. AWS Step Functions, com fluxos gerados a partir do pedido
C. Amazon Bedrock Prompt Management, com versões de prompts
D. Amazon SageMaker Canvas, com modelos de ML criados sem código

D11 · Fundamentos de IA generativa

**Em uma base de conhecimento do Bedrock, a equipe dividiu os manuais em trechos muito grandes, de várias páginas. As respostas passaram a trazer informação irrelevante e o custo por pergunta subiu. Qual ajuste é o mais indicado?**

A. Aumentar a temperatura do modelo para ele filtrar o conteúdo recebido
B. Reduzir o tamanho dos trechos (chunks), com alguma sobreposição entre eles
C. Remover a etapa de recuperação e enviar o manual completo no prompt
D. Trocar o modelo de embeddings por um modelo de geração de imagens

D12 · Fundamentos de IA generativa

**Uma empresa precisa classificar milhões de mensagens curtas de texto por dia com a menor latência e o menor custo possíveis; a tarefa é simples e não envolve imagens. Entre os modelos Amazon Nova no Bedrock, qual é o ponto de partida mais adequado?**

A. Amazon Nova Pro, modelo multimodal para tarefas complexas
B. Amazon Nova Reel, modelo de geração de vídeos
C. Amazon Nova Canvas, modelo de geração de imagens
D. Amazon Nova Micro, modelo só de texto e de menor latência

01

CAPÍTULO 1 · 28% DA PROVA

# Aplicações de modelos de base

Este é o domínio de maior peso, e eu preciso escolher a técnica certa para cada cenário: engenharia de prompt, RAG com Knowledge Bases, ajuste fino, pré-treino contínuo, RLHF, destilação ou agentes. A prova traz enunciados longos com vários requisitos e quatro arquiteturas; quase sempre há uma alternativa de treinar do zero, outra de bucket público ou de enviar tudo no prompt, que são distratores. Também cai muito parâmetro de inferência (temperatura, top-p, max tokens, stop sequence), modos de inferência (sob demanda, lote, Provisioned Throughput, streaming, cache de prompts) e métricas de avaliação (ROUGE, BLEU, avaliação humana, LLM como juiz, avaliação de RAG). Onde mais erro: achar que temperatura resolve tamanho de contexto e confundir quando usar ajuste fino ou RAG.

**Neste capítulo**
18 tópicos-chave · 14 pegadinhas · 6 macetes · 14 gatilhos · 4 questões comentadas

## Tópicos-chave

### Critérios de seleção de modelo

Custo, modalidade, latência, suporte a idiomas, tamanho e complexidade do modelo, personalização e tamanho de entrada e saída. A escolha certa é testar os candidatos com dados reais e comparar qualidade, latência e custo por chamada; o maior, o mais novo ou o de janela maior não é automaticamente o melhor.

### Parâmetros de inferência

Temperatura e top-p controlam a aleatoriedade: reduzir dá saídas mais consistentes, aumentar dá variedade e tende a aumentar invenções. Max tokens limita o tamanho da resposta (não aumenta o que cabe na entrada); stop sequence encerra a geração num marcador; top-k igual a um torna a escolha gulosa, sem definir onde parar.

### Prompt de sistema e templates

Instruções persistentes de persona, tom, idioma e limites (como não opinar sobre concorrentes) vão no prompt de sistema enviado em cada chamada. Templates padronizados numa camada de orquestração (por exemplo, AWS Lambda) mantêm o tom e permitem trocar o modelo em tempo de execução.

### Técnicas de prompt

Zero-shot só tem a instrução; one-shot (single-shot) traz um exemplo; few-shot traz vários exemplos e dá formato consistente sem treinar. Chain-of-thought pede raciocínio passo a passo e ajuda em problemas de várias etapas, como cálculo de prazos com feriados.

### Prompt negativo em imagens

No Amazon Nova Canvas, o que não se quer na imagem vai no prompt negativo (negativeText), sem 'não' ou 'sem'. Negar no prompt principal mantém as palavras e tende a gerar exatamente o que se queria evitar.

### RAG e Amazon Bedrock Knowledge Bases

Os documentos são divididos em trechos, convertidos em embeddings e indexados; a cada pergunta, os trechos relevantes vão ao modelo junto com a pergunta. É a escolha para dados internos que mudam, respostas com citação das fontes e redução de alucinação, sem retreinar o modelo.

### Armazenamentos vetoriais

Para busca por similaridade sem gerenciar servidores: coleção de busca vetorial do Amazon OpenSearch Serverless. Para quem já usa Aurora PostgreSQL: extensão pgvector. DynamoDB com GSI, SQS, EBS e S3 Glacier Deep Archive não são armazenamentos vetoriais.

### Chunking

Trechos grandes demais trazem informação irrelevante e aumentam o custo por pergunta; reduza o tamanho dos chunks, com alguma sobreposição. Quando o manual excede a janela de contexto, divida em trechos com embeddings e recupere só os relevantes.

### Abordagens de personalização

Pré-treino contínuo (adaptação de domínio) usa texto não rotulado para vocabulário do domínio; ajuste fino supervisionado usa pares rotulados para tom e formato; RLHF usa avaliadores que comparam pares de respostas para ensinar preferências de estilo; destilação treina um modelo menor (aluno) a partir de um maior (professor). Treinar do zero é o mais caro e quase nunca é a resposta.

### Qualidade dos dados de ajuste

Antes de ajustar, faça curadoria: balancear as classes ou produtos, corrigir respostas desatualizadas e remover dados pessoais. Duplicar exemplos majoritários agrava o viés e mais épocas sobre dados ruins reforçam erros e aumentam o sobreajuste.

### Agentes do Amazon Bedrock e uso de ferramentas

Agentes com grupos de ação ligados a APIs (por exemplo, via AWS Lambda) consultam reservas, verificam disponibilidade e executam a remarcação. No padrão seguro, o modelo decide qual ferramenta chamar e com quais parâmetros, e um serviço intermediário valida e executa; o modelo não executa SQL ou HTTP direto.

### Métricas automáticas de texto

ROUGE mede sobreposição de n-gramas com resumos de referência; BLEU mede sobreposição de n-gramas com traduções de referência. RMSE, acurácia, AUC-ROC e silhueta não avaliam texto livre.

### Avaliação humana, LLM como juiz e avaliação de RAG

Critérios subjetivos como tom da marca e utilidade pedem avaliação com trabalhadores humanos (pode ser a própria equipe). Para milhares de respostas em utilidade, correção e completude sem custo humano, use um modelo como juiz. Para saber se o erro está na recuperação ou na geração, use a avaliação de RAG do Bedrock sobre a base de conhecimento.

### Modos de inferência no Bedrock

Sob demanda cobra por tokens; inferência em lote processa grandes volumes de forma assíncrona com arquivos no S3, a custo menor; Provisioned Throughput reserva capacidade dedicada para volume alto e constante e pode executar modelos personalizados. Streaming exibe os tokens enquanto são gerados e melhora a experiência percebida.

### Cache de prompts

Para um prefixo longo repetido em toda chamada (o mesmo contrato com perguntas diferentes), o prefixo fica em cache, os tokens lidos do cache custam menos e a resposta sai mais rápido. Deixe o conteúdo fixo no início e o variável no fim.

### Bedrock Prompt Management

Gerencia prompts com variáveis, testes de variantes e versões, resolvendo o caso de equipes editando prompts no código sem saber o que está em produção nem como voltar a uma versão anterior.

### Amazon Q Business e SageMaker JumpStart

Q Business é um assistente corporativo que responde sobre conteúdos do SharePoint, Confluence e S3 por conectores, respeitando as permissões; Q Developer é para desenvolvimento de software. JumpStart implanta com poucos cliques, num endpoint da própria conta, um modelo de código aberto que não está no Bedrock.

### Riscos de aplicações com modelos de base

Envenenamento: dados manipulados no repositório de ajuste mudam o comportamento do modelo. Injeção de prompt: entrada que tenta sobrescrever as instruções na inferência; jailbreak: prompt para burlar as salvaguardas; exposição: o modelo revela o prompt de sistema ou dados sensíveis.

## Macetes para memorizar

### Escada de personalização

Prompt, RAG, ajuste fino, pré-treino contínuo, do zero: cada degrau custa mais; suba só se o de baixo não resolver.

### RAG x ajuste fino

RAG traz os fatos; ajuste fino ensina a forma.

### ROUGE x BLEU

ROUGE = Resumo; BLEU = traduções Bilíngues.

### Modos de inferência

Lote = lento e barato; Provisionado = pesado e constante; Cache = contexto repetido; Streaming = sensação de rapidez.

### Parâmetros

Temperatura = criatividade; max tokens = tamanho da saída; stop sequence = onde parar; janela de contexto = quanto cabe na entrada.

### Prompt negativo

O que eu não quero vai no negativo, escrito sem 'não'.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **RAG**<br>Fatos internos que mudam com frequência, necessidade de citar fontes, sem retreinar. | **Ajuste fino**<br>Ensinar tom de voz e formato padronizado com milhares de pares rotulados, quando os fatos já estão disponíveis. |
| **Pré-treino contínuo (adaptação de domínio)**<br>Milhões de documentos não rotulados para aprender o vocabulário do domínio. | **Ajuste fino supervisionado**<br>Pares de pergunta e resposta rotulados para uma tarefa e estilo. |
| **RLHF**<br>Mudar tom e estilo com avaliadores que comparam pares de respostas. | **RAG**<br>Acrescentar conhecimento recuperado ao prompt; não ensina preferências de estilo. |

| | |
|---|---|
| **Inferência em lote**<br>Grande volume periódico sem resposta imediata (2 milhões de resumos por semana), menor custo. | **Provisioned Throughput**<br>Tráfego contínuo em tempo real que exige capacidade garantida e previsível. |
| **Cache de prompts**<br>Mesmo documento ou instruções longas repetidos em chamadas interativas. | **Inferência em lote**<br>Volume grande sem pressa, processado de forma assíncrona. |
| **ROUGE**<br>Resumos comparados a resumos de referência. | **BLEU**<br>Traduções comparadas a traduções de referência. |
| **Amazon Bedrock**<br>Modelos de base gerenciados por API, sem gerenciar infraestrutura. | **Amazon SageMaker JumpStart**<br>Modelo de código aberto fora do Bedrock, implantado num endpoint da própria conta com mais controle. |

## Números que caem

| Item | Valor |
|---|---|
| Peso do domínio na prova | **28% (o maior)** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Treinar um modelo do zero dá total personalização e é a solução mais robusta.
**Como não cair:** É caro, demorado e desnecessário na imensa maioria dos cenários; prefira prompt, RAG ou ajuste fino sobre um modelo de base.

⚠ **Alerta de prova:** Fine-tuning recorrente a cada documento novo mantém o assistente atualizado.
**Como não cair:** Documentos que mudam pedem RAG; ajuste fino a cada atualização é caro, lento e não cita fontes.

⚠ **Alerta de prova:** Enviar o manual inteiro e aumentar o limite de tokens de saída resolve o excesso de contexto.
**Como não cair:** O limite de saída não aumenta o que cabe na entrada; divida em trechos e recupere os relevantes.

⚠ **Alerta de prova:** Temperatura ajuda a processar textos longos, acelerar ou filtrar conteúdo.
**Como não cair:** Temperatura só controla aleatoriedade; aumentar tende a aumentar invenções.

⚠ **Alerta de prova:** Basta pedir no prompt que o modelo não invente.
**Como não cair:** Sem as fontes corretas isso ajuda pouco; use RAG com os textos oficiais e citações.

⚠ **Alerta de prova:** Repetir 'sem pessoas' no prompt da imagem ou aumentar a escala de orientação remove pessoas.
**Como não cair:** Isso reforça os termos citados; coloque 'pessoas, logotipos' no prompt negativo.

⚠ **Alerta de prova:** Provisioned Throughput elimina a cobrança por token do contrato repetido.
**Como não cair:** Ele cobra por hora capacidade dedicada e só compensa com volume alto e constante; para prefixo repetido use cache de prompts.

⚠ **Alerta de prova:** Inferência em lote melhora a espera de usuários num chat.
**Como não cair:** Lote é assíncrono e piora a conversa; use a API de streaming.

⚠ **Alerta de prova:** Temperatura zero ou top-k igual a um fazem o modelo parar no marcador.
**Como não cair:** Onde terminar se define com stop sequence.

⚠ **Alerta de prova:** Guardrail ou base de conhecimento executam a remarcação do voo.
**Como não cair:** Guardrail filtra conteúdo e base de conhecimento dá contexto; executar ações é de Agentes do Bedrock com grupos de ação.

⚠ **Alerta de prova:** Colocar documentos públicos e confidenciais no prompt e pedir ao modelo que filtre.
**Como não cair:** Controle de acesso não se delega ao modelo: metadados de acesso nos trechos e checagem de autorização antes de enviar ao modelo.

⚠ **Alerta de prova:** Duplicar exemplos do produto majoritário ou aumentar épocas compensa dados ruins.
**Como não cair:** Isso agrava desequilíbrio e sobreajuste; cure os dados primeiro.

⚠ **Alerta de prova:** Model Monitor ou métricas do CloudWatch mostram se a falha está na recuperação.
**Como não cair:** Use a avaliação de RAG do Bedrock; Model Monitor é para endpoints do SageMaker AI e latência é métrica operacional.

⚠ **Alerta de prova:** Jailbreak, injeção de prompt e envenenamento são a mesma coisa.
**Como não cair:** Envenenamento altera os dados de treino ou ajuste; injeção e jailbreak acontecem na entrada durante a inferência, sem alterar dados de ajuste.

## Como cai na prova

📝 **Como cai:** Enunciado longo com requisitos numerados (1), (2), (3) e quatro arquiteturas.
**Como resolver:** Associe cada requisito a um componente (dados internos = RAG; tom = prompt de sistema ou ajuste fino; ações = agentes; segurança = endpoint privado e IAM). Elimine as que treinam do zero, expõem dados ou ignoram um requisito; fica a que cobre todos com serviço gerenciado.

📝 **Como cai:** Qual técnica de personalização usar.
**Como resolver:** Olhe o dado disponível: texto não rotulado = pré-treino contínuo; pares rotulados = ajuste fino; comparações de preferência = RLHF; modelo grande caro = destilação; dados que mudam = RAG; sem treinar = prompt.

📝 **Como cai:** Qual parâmetro ou técnica de prompt ajustar.
**Como resolver:** Variação = temperatura/top-p; tamanho = max tokens; parar num marcador = stop sequence; raciocínio de várias etapas = chain-of-thought; formato consistente sem treino = few-shot.

📝 **Como cai:** Qual métrica ou forma de avaliação usar.
**Como resolver:** Resumo com referência = ROUGE; tradução com referência = BLEU; critérios subjetivos = avaliação humana; volume grande subjetivo = LLM como juiz; base de conhecimento errando = avaliação de RAG; valor para o negócio = taxa de conclusão sem transferência.

📝 **Como cai:** Custo e latência de uma aplicação no Bedrock.
**Como resolver:** Prefixo repetido = cache de prompts; sem pressa = lote; alto e constante = Provisioned Throughput; espera percebida = streaming; tarefa simples em volume = modelo menor.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| documentos internos atualizados, responder com citação das fontes | **RAG com Amazon Bedrock Knowledge Bases** |
| milhões de documentos não rotulados, vocabulário do domínio | **Pré-treino contínuo (adaptação de domínio)** |
| milhares de pares rotulados, tom da marca e formato padronizado | **Ajuste fino supervisionado** |
| avaliadores comparam pares de respostas e indicam a melhor | **RLHF** |
| qualidade do modelo grande com custo e latência de modelo pequeno | **Destilação de modelo** |
| o mesmo documento longo enviado em toda chamada | **Cache de prompts** |
| milhões de documentos por semana, sem resposta imediata | **Inferência em lote do Bedrock** |
| usuários esperam segundos sem ver nada | **API de streaming** |

| | |
|---|---|
| respostas variam muito entre execuções | **Reduzir temperatura e, se preciso, top-p** |
| consultar a reserva e efetuar a remarcação chamando APIs internas | **Agentes do Amazon Bedrock com grupos de ação** |
| SharePoint, Confluence e S3 respeitando permissões | **Amazon Q Business** |
| sobreposição com resumos de referência | **ROUGE (tradução de referência: BLEU)** |
| milhares de respostas em utilidade e completude, sem avaliação humana | **LLM como juiz** |
| prompts editados no código sem saber a versão em produção | **Amazon Bedrock Prompt Management** |

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

- Treinar do zero quase nunca é a resposta.
- Temperatura muda só a aleatoriedade: não muda o tamanho do contexto, a velocidade nem os tokens enviados.
- Dados que mudam = RAG; tom e formato com pares rotulados = ajuste fino.
- Ações reais em sistemas = Agentes do Bedrock com grupos de ação.
- Controle de acesso em RAG é aplicado antes de enviar trechos ao modelo, nunca pedido ao modelo.

## Essenciais para a prova (checklist)

- ☐ Sei os critérios de seleção de modelo e por que testar com dados reais.
- ☐ Sei o efeito de temperatura, top-p, top-k, max tokens e stop sequence.
- ☐ Sei diferenciar zero-shot, one-shot, few-shot, chain-of-thought e prompt negativo.
- ☐ Sei montar um RAG com Knowledge Bases, chunking e armazenamento vetorial adequado.
- ☐ Sei escolher entre pré-treino contínuo, ajuste fino, RLHF, destilação e RAG.
- ☐ Sei quando usar inferência em lote, Provisioned Throughput, streaming e cache de prompts.
- ☐ Sei escolher entre ROUGE, BLEU, avaliação humana, LLM como juiz e avaliação de RAG.
- ☐ Sei para que servem Agentes do Bedrock, Prompt Management, Q Business e JumpStart.
- ☐ Sei distinguir envenenamento, injeção de prompt, jailbreak e exposição.

## Questões de revisão do capítulo

R1.1 · Aplicações de modelos de base

**Uma equipe quer criar um sistema interno de perguntas e respostas para funcionários usando documentos corporativos (políticas, manuais e FAQs). Eles desejam aproveitar modelos de base no Amazon Bedrock para fornecer respostas atualizadas e alinhadas ao conteúdo interno, sem precisar treinar um novo modelo do zero. Qual abordagem é mais adequada para esse caso de uso?**

A. Treinar completamente um novo modelo de linguagem do zero usando só os documentos internos
B. Usar o modelo de base no Amazon Bedrock sem fornecer contexto adicional, confiando que o modelo terá todo o conhecimento necessário
C. Converter os documentos internos em prompts manuais extensos e reutilizar sempre o mesmo prompt estático nas requisições ao modelo de base
D. Adotar modelo de base no Amazon Bedrock com RAG sobre uma base de conhecimento indexada em busca vetorial

R1.2 · Aplicações de modelos de base

**Uma empresa de comércio eletrônico deseja usar modelos de base na AWS para melhorar a experiência do cliente em seu site. O objetivo é sugerir descrições de produtos mais atrativas, permitir que clientes façam perguntas em linguagem natural sobre produtos e, futuramente, integrar um chatbot de suporte. Considerando o uso do Amazon Bedrock, qual abordagem é mais adequada para começar esse projeto de forma rápida e escalável, mantendo flexibilidade para evoluir a solução?**

A. Selecionar um único modelo de base no Amazon Bedrock, integrá-lo diretamente à aplicação e ajustar hiperparâmetros à mão toda vez que surgir um novo caso de uso
B. Testar modelos no playground do Amazon Bedrock, definir os prompts de cada caso de uso e integrar o modelo escolhido à aplicação pelo SDK
C. Treinar um novo modelo de linguagem do zero em instâncias GPU do Amazon EC2 e, após o treinamento, disponibilizá-lo por meio de endpoints gerenciados pelo Amazon SageMaker
D. Armazenar as descrições e perguntas de clientes em um bucket do Amazon S3 e usar consultas SQL no Amazon Athena para gerar automaticamente respostas em linguagem natural

R1.3 · Aplicações de modelos de base

**Um e-commerce quer que a busca do site encontre produtos pelo significado da consulta, por exemplo retornando 'tênis de corrida' para 'calçado para correr', e não só por palavras idênticas. Qual abordagem com modelos de base atende a esse requisito?**

A. Gerar embeddings dos produtos e das consultas e buscar por similaridade vetorial
B. Criar um índice por palavras-chave com sinônimos cadastrados à mão para cada produto
C. Treinar um classificador de sentimento sobre as avaliações dos produtos
D. Usar um modelo de geração de imagens para criar fotos a partir das consultas

R1.4 · Aplicações de modelos de base

**Uma equipe de ML está desenvolvendo um sistema de atendimento ao cliente multilíngue usando modelos de base na AWS. O sistema deve: (1) entender perguntas em diferentes idiomas, (2) consultar dados proprietários internos (bases de conhecimento, FAQs privadas) e (3) responder em linguagem natural, mantendo rastreabilidade de qual documento interno foi usado para gerar cada resposta. A empresa quer minimizar riscos de vazamento de dados sensíveis para fora do ambiente AWS e reduzir alucinações. Qual abordagem arquitetural é MAIS adequada para esses requisitos?**

A. Utilizar diretamente um modelo de base genérico hospedado fora da AWS via API pública, enviando o texto completo dos documentos internos a cada requisição para permitir que o modelo responda com base nesses dados
B. Só ajustar (fine-tuning) um modelo de base no Amazon Bedrock com os documentos internos
C. RAG no Bedrock com embeddings em OpenSearch Serverless ou Aurora e prompts que citam as fontes
D. Treinar do zero um modelo de linguagem proprietário em instâncias Amazon EC2 com os documentos internos, expondo o modelo por uma API customizada, sem usar serviços gerenciados ou mecanismos adicionais de busca

## Respostas comentadas

### R1.1. D) Adotar modelo de base no Amazon Bedrock com RAG sobre uma base de conhecimento indexada em busca vetorial

Por quê: Esta alternativa está correta porque a abordagem de Recuperação Aumentada por Geração (RAG) permite que o modelo de base consulte uma base de conhecimento atualizada (indexada em um mecanismo de busca vetorial) e use esse conteúdo como contexto para gerar respostas. Assim, o modelo continua genérico, mas passa a responder com base nos documentos corporativos, sem exigir treinamento do zero ou fine-tuning pesado.

✗ A) Incorreta. Treinar um modelo do zero é caro, demorado e desnecessário para esse caso. Os modelos de base já fornecem capacidades linguísticas avançadas; o ideal é apenas conectá-los aos dados internos via RAG ou, se necessário, fine-tuning.
✗ B) Incorreta. Usar o modelo sem contexto adicional faz com que ele responda com base em conhecimento geral, não nas políticas e documentos específicos da empresa, o que pode gerar respostas incorretas ou desalinhadas com as regras internas.
✗ C) Incorreta. Criar um único prompt estático com todos os documentos é inviável (limite de tokens), difícil de manter e não escala. Além disso, não há mecanismo eficiente de busca ou seleção dinâmica do conteúdo relevante para cada pergunta.

> **Dica de prova:** Quando o objetivo é usar documentos próprios e manter o modelo sempre atualizado, pense em RAG: indexar dados em um repositório vetorial e enviar apenas os trechos relevantes como contexto ao modelo de base, em vez de treinar tudo de novo.

**R1.2. B) Testar modelos no playground do Amazon Bedrock, definir os prompts de cada caso de uso e integrar o modelo escolhido à aplicação pelo SDK**

Por quê: Esta alternativa é a mais adequada porque aproveita diretamente o Amazon Bedrock para uso de modelos de base pré-treinados, permitindo experimentar rapidamente vários modelos no playground, ajustar prompts para diferentes tarefas (descrições, perguntas e respostas, chatbot) e, em seguida, integrar o modelo escolhido via SDK de forma escalável. Isso reduz tempo de implementação, evita o custo de treinar modelos do zero e mantém flexibilidade para trocar ou complementar modelos no futuro.

✗ A) Incorreta. Foca em um único modelo desde o início e em ajustes manuais de hiperparâmetros, o que reduz a flexibilidade e não aproveita bem a capacidade do Amazon Bedrock de experimentar múltiplos modelos de base de forma simples.

✗ C) Incorreta. Propõe treinar um modelo do zero em EC2/SageMaker, o que é caro, complexo e desnecessário para o objetivo de começar rápido com modelos de base já disponíveis no Amazon Bedrock.

✗ D) Incorreta. Amazon S3 e Athena não geram linguagem natural; são serviços de armazenamento e consulta de dados. Eles podem apoiar a solução, mas não substituem modelos de base nem o uso do Amazon Bedrock para geração de texto e interação em...

> **Dica de prova:** Quando o objetivo é começar rápido com IA generativa na AWS, priorize serviços gerenciados com modelos pré-treinados, como Amazon Bedrock. Treinar modelos do zero geralmente só faz sentido em cenários muito específicos, com alto volume de dados proprietários e requisitos especiais.

**R1.3. A) Gerar embeddings dos produtos e das consultas e buscar por similaridade vetorial**

Por quê: Embeddings representam textos como vetores em que significados próximos ficam próximos; comparando o vetor da consulta com os dos produtos, por exemplo com Amazon Titan Embeddings e um índice vetorial no OpenSearch, a busca encontra resultados semanticamente relacionados. Sinônimos à mão não escalam, gerar imagens não ajuda a busca textual e sentimento não mede relevância.

✗ B) Incorreta. Exige manutenção manual e não cobre variações de linguagem.

✗ C) Incorreta. Sentimento não indica se o produto corresponde à consulta.

✗ D) Incorreta. Gerar imagens não melhora a busca por texto.

> **Dica de prova:** Busca por significado: embeddings e busca vetorial.

**R1.4. C) RAG no Bedrock com embeddings em OpenSearch Serverless ou Aurora e prompts que citam as fontes**

Por quê: Esta alternativa é a mais adequada porque combina Retrieval-Augmented Generation (RAG) com modelos de base no Amazon Bedrock, mantendo os dados dentro da AWS e oferecendo rastreabilidade. Ao usar uma base vetorial (por exemplo, em Amazon OpenSearch Serverless ou Aurora com suporte a vetores) para indexar os documentos internos, o sistema primeiro recupera os trechos mais relevantes e só então os injeta no prompt do modelo.

✗ A) Incorreta. Enviar documentos internos a um modelo fora da AWS por API pública aumenta o risco de vazamento de dados e reduz o controle sobre privacidade e conformidade.

✗ B) Incorreta. Apenas fine-tuning com documentos internos não garante que o modelo citará corretamente as fontes e não resolve bem o problema de rastreabilidade.

✗ D) Incorreta. Treinar um modelo do zero é caro, demorado e complexo, especialmente para suporte multilíngue. Sem um mecanismo de recuperação (RAG) e sem estrutura para citar fontes, as alucinações tendem a ser mais difíceis de controlar, e...

**Dica de prova:** Quando o requisito inclui: (a) uso de conhecimento proprietário atualizado, (b) redução de alucinações e (c) necessidade de citar ou rastrear fontes, pense em RAG com armazenamento vetorial + modelo de base gerativo.

02

CAPÍTULO 2 · 14% DA PROVA

# Diretrizes de IA responsável

Aqui eu preciso reconhecer a resposta que combina salvaguarda técnica, revisão humana, transparência e monitoramento contínuo desde antes da produção. A prova traz cenários sensíveis (saúde, crédito, triagem de currículos, setor público, menores de idade) e pergunta a MELHOR prática; as erradas são extremas (proibir tudo, desligar logs, liberar tudo) ou terceirizam a responsabilidade (termos de uso, provedor do modelo, só jurídico). As ferramentas cobradas são Amazon Bedrock Guardrails, SageMaker Clarify, Model Monitor, Model Cards, Amazon A2I e avaliação humana do Bedrock. Onde mais erro: achar que modelo maior ou remover atributos sensíveis resolve viés.

**Neste capítulo**

18 tópicos-chave · 14 pegadinhas · 6 macetes · 12 gatilhos · 4 questões comentadas

## Tópicos-chave

### Segurança desde o projeto

Riscos são avaliados e mitigados ANTES da produção: testes, avaliação de impacto, filtros e escalonamento para humanos. Esperar reclamações de clientes para corrigir é reativo e viola o princípio.

### Responsabilidade compartilhada

Usar modelo de um grande provedor ou configurações padrão não dispensa avaliação no seu contexto, dados e usuários. A organização continua responsável por políticas, testes, monitoramento e mitigação.

### Amazon Bedrock Guardrails

Aplica filtros de conteúdo (ódio, violência, conteúdo sexual), tópicos negados (temas sensíveis fora do escopo), filtros de informações sensíveis (PII) e verificação de fundamentação contextual. Atua na origem (entradas e saídas do modelo), não só na interface.

### Revisão humana (human-in-the-loop) e Amazon A2I

Em decisões de alto risco, humanos revisam as saídas antes do uso; em saúde, o médico continua responsável e o sistema é apoio, não decisão clínica. O fluxo exige uso aceitável definido, revisores treinados sobre viés e segurança e decisões registradas, sem depender de um único revisor.

### SageMaker Clarify

Detecta viés nos dados e nas previsões e explica modelos por atribuição de atributos baseada em valores de Shapley (SHAP), de forma global e para previsões individuais. É a resposta quando o regulador pede quais atributos pesaram numa decisão específica de crédito.

### Model Monitor com Clarify

Acompanha deriva de dados, de qualidade e de viés em produção, com alarmes no CloudWatch. Junto com Model Registry e CloudTrail, forma o pipeline de MLOps com rastreabilidade e responsabilidade.

### SageMaker Model Cards

Documentam propósito, casos de uso pretendidos, dados de treinamento, versões, hiperparâmetros, avaliações e limitações conhecidas. É a ferramenta de transparência e documentação do modelo.

### Viés e variáveis proxy

Remover gênero, raça ou idade não elimina viés, porque proxies como CEP, universidade ou histórico profissional carregam a mesma informação. Dados históricos de contratações bem-sucedidas podem amplificar discriminação passada.

### Equidade medida por grupo

Defina métricas de equidade, teste com dados representativos e diversos (inclusive por idioma e região) e meça o desempenho por grupo de pacientes, candidatos ou clientes. Avaliação subjetiva caso a caso, sem métricas padronizadas, não serve.

### Transparência ao usuário

Avise de forma clara e visível que o conteúdo é gerado por IA, explique o propósito e como pedir atendimento humano, e oriente a revisar antes de usar. Aviso só nos termos de uso, opcional ou escondido não atende.

### Contestação e canal de reporte

Em crédito e recrutamento, informe os afetados, explique decisões automatizadas e permita contestação. Ofereça canal para reportar erros e registre incidentes para análise de causa raiz.

### Logs com privacidade

Logs são necessários para auditoria, rastreabilidade e investigação; a abordagem responsável é anonimizar ou pseudonimizar, criptografar e controlar o acesso, não desligar. Monitorar não fere privacidade quando os dados são protegidos.

### Avaliação de impacto algorítmico

Define, documenta e revisa periodicamente riscos e impactos a grupos vulneráveis, com planos de mitigação, critérios de desligamento do modelo e responsáveis por cada ação. É multidisciplinar e contínua, não só jurídica nem só financeira.

### Política de uso aceitável

Política clara do que pode ou não ser inserido (sem dados sensíveis ou sigilosos), casos de uso permitidos e proibidos, revisão humana em tarefas críticas, registro de uso e treinamento sobre alucinação, viés e privacidade.

### Governança multidisciplinar

Princípios viram requisitos mensuráveis integrados a ferramentas (Clarify, CloudTrail) e revisões, envolvendo negócio, jurídico, segurança e ML ao longo do ciclo de vida. Só um código de ética assinado, sem mudar pipelines, não é prática.

### Propriedade intelectual

Imagens geradas que reproduzem elementos protegidos de obras de terceiros são risco de violação de propriedade intelectual; mitigue com revisão humana e regras para o conteúdo gerado.

### Interpretabilidade x desempenho

Exigir que todo modelo seja totalmente interpretável é extremo e irrealista; o equilíbrio é explicabilidade proporcional ao risco. Em saúde, modelos simples são mais transparentes e facilitam explicar, validar e rastrear decisões, mas nenhum modelo é imune a viés.

### Avaliação de modelos com humanos no Bedrock

Avaliadores humanos analisam respostas quanto a viés, tom e adequação, alimentando ajustes de prompts e regras, junto com filtros de conteúdo e monitoramento.

## Macetes para memorizar

### 💡 Fórmula da resposta certa

Filtro técnico + humano no loop + transparência + monitoramento contínuo.

### 💡 Linha do tempo

Antes testa, durante filtra, depois monitora.

### 💡 Ferramentas

Clarify clareia (viés e porquê); Monitor vigia (deriva); Cards contam a história do modelo; A2I põe o humano.

### 💡 Logs

Não desligue o log: proteja o log.

**Palavras suspeitas**

Só, sempre, confiar, desligar, eliminar completamente, proibir tudo: alternativa quase sempre errada.

**Tamanho**

Tamanho não é ética: maior ou menor não resolve viés.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **SageMaker Clarify**<br>Explicar por que uma previsão individual saiu assim (SHAP) e medir viés. | **SageMaker Model Monitor**<br>Acompanhar desvios de dados, qualidade e viés ao longo do tempo em produção. |
| **SageMaker Clarify**<br>Viés e explicabilidade de um modelo treinado. | **SageMaker Ground Truth**<br>Criar conjuntos de dados rotulados por pessoas. |
| **Amazon Bedrock Guardrails**<br>Bloquear automaticamente conteúdo tóxico, temas negados e PII em tempo de inferência. | **Revisão humana (Amazon A2I)**<br>Decisões de alto risco que precisam de validação de um especialista antes do uso. |
| **SageMaker Model Cards**<br>Documentar propósito, dados, limitações e avaliações do modelo. | **AWS CloudTrail**<br>Registrar quem chamou quais APIs, para auditoria. |
| **Transparência**<br>Informar ao usuário que há IA, o propósito, os limites e como pedir humano. | **Explicabilidade**<br>Explicar quais atributos levaram a uma decisão específica. |
| **Modelo simples e interpretável**<br>Contextos regulados que exigem explicar, validar e rastrear cada decisão. | **Rede neural profunda**<br>Quando o ganho de desempenho compensa e há explicabilidade proporcional ao risco. |

## Números que caem

| Item | Valor |
|---|---|
| Peso do domínio na prova | **14%** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Aumentar o tamanho ou os parâmetros do modelo reduz viés e alucinação automaticamente.
**Como não cair:** Tamanho não é ética: modelos maiores podem amplificar vieses e gerar saídas mais convincentes, porém incorretas.

⚠ **Alerta de prova:** Reduzir o tamanho do modelo diminui conteúdo ofensivo e danos sociais.
**Como não cair:** Sem curadoria, testes e filtros, o tamanho não resolve viés nem toxicidade.

⚠ **Alerta de prova:** Desligar logs protege a privacidade e é a opção responsável.
**Como não cair:** Desligar logs elimina auditoria e investigação; proteja os logs com anonimização, criptografia e controle de acesso.

⚠ **Alerta de prova:** Remover atributos sensíveis elimina a discriminação.
**Como não cair:** Proxies carregam a mesma informação; meça equidade por grupo e mitigue.

⚠ **Alerta de prova:** O provedor do modelo de base já garante neutralidade e segurança.
**Como não cair:** Responsabilidade compartilhada: avalie no seu contexto com testes, guardrails e monitoramento.

⚠ **Alerta de prova:** Termos de uso extensos ou um aviso genérico isentam a empresa.
**Como não cair:** Não substituem controles técnicos e de processo nem reduzem o risco real.

⚠ **Alerta de prova:** Proibir totalmente a IA generativa é a política mais responsável.
**Como não cair:** É evitação; IA responsável busca uso seguro com política, controles e treinamento.

⚠ **Alerta de prova:** Delegar IA responsável só ao jurídico, revisando antes do lançamento.
**Como não cair:** Exige governança multidisciplinar e contínua ao longo do ciclo de vida.

⚠ **Alerta de prova:** Exigir que todo modelo seja totalmente interpretável, proibindo caixa-preta.
**Como não cair:** É extremo e irrealista; use explicabilidade proporcional ao risco.

⚠ **Alerta de prova:** Monitorar latência, custo, throughput e acurácia basta para IA responsável.
**Como não cair:** Faltam métricas de equidade, segurança e impacto em grupos.

⚠ **Alerta de prova:** Treinar só com dados históricos internos garante neutralidade.
**Como não cair:** Dados históricos refletem práticas passadas e podem amplificar viés.

⚠ **Alerta de prova:** Aumentar a temperatura reduz estereótipos repetidos.
**Como não cair:** Temperatura muda aleatoriedade, não é mecanismo de mitigação de viés.

⚠ **Alerta de prova:** Esconder que a resposta é de IA evita desconfiança.
**Como não cair:** Fere a transparência: o cliente precisa saber que a resposta envolve IA, mesmo quando há revisão humana.

⚠ **Alerta de prova:** Amazon Inspector detecta viés em modelos.
**Como não cair:** Inspector gerencia vulnerabilidades de instâncias e contêineres; viés e explicabilidade são do SageMaker Clarify.

## Como cai na prova

📝 **Como cai:** Cenário sensível pedindo a MELHOR ação de IA responsável.
**Como resolver:** Elimine as extremas (proibir, liberar tudo, desligar logs) e as que terceirizam (termos de uso, provedor, só jurídico). Escolha a que reúne salvaguarda técnica, revisão humana, transparência e monitoramento contínuo, antes da produção.

📝 **Como cai:** Pedido de explicar decisão, medir viés ou documentar modelo.
**Como resolver:** Explicar decisão ou medir viés = Clarify; viés ao longo do tempo = Model Monitor com Clarify; documentar = Model Cards; auditar chamadas = CloudTrail.

📝 **Como cai:** Modelo gerando estereótipos ou conteúdo nocivo nos testes.
**Como resolver:** Ajuste modelo, dados ou prompt e aplique filtros antes de ampliar o uso; documente limites e monitore em produção. Não aceite o comportamento nem mantenha em produção para não atrasar.

📝 **Como cai:** Requisito de transparência exigido pelo jurídico.
**Como resolver:** A resposta informa o cliente final na interface: conteúdo de IA, propósito e como pedir humano; log interno ou revisão sem avisar o cliente não basta.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| regulador quer saber quais atributos pesaram nesta decisão de crédito | **SageMaker Clarify com SHAP** |
| viés e deriva em produção, com alarmes | **Model Monitor com Clarify e CloudWatch** |
| documentar propósito, dados, limitações e versões do modelo | **SageMaker Model Cards** |
| decisões críticas validadas por especialista | **Revisão humana com Amazon A2I** |

| | |
|---|---|
| bloquear linguagem ofensiva e restringir temas sensíveis | **Bedrock Guardrails: filtros de conteúdo e tópicos negados** |
| triagem de currículos ou concessão de crédito | **Testes de viés por grupo, métricas de equidade, revisão humana e contestação** |
| assistente que apoia médicos | **Revisão humana obrigatória, testes por grupo de pacientes e aviso de que é apoio, não decisão** |
| imagens geradas reproduzem obras de terceiros | **Risco de propriedade intelectual: revisão humana e regras** |
| transparência para o cliente final | **Aviso visível de IA, propósito e como pedir atendimento humano** |
| critérios subjetivos como viés e tom ao comparar modelos | **Avaliação de modelos com trabalhadores humanos no Bedrock** |
| transformar princípios em prática | **Requisitos mensuráveis integrados a ferramentas e revisões** |
| avaliação de impacto algorítmico | **Riscos a grupos vulneráveis, mitigação, critérios de desligamento e responsáveis** |

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

- Responsabilidade compartilhada: o provedor do modelo não resolve o seu contexto.
- Remover atributos sensíveis não remove viés (proxies).
- Logs ficam ligados e protegidos; nunca desligados por privacidade.

## Essenciais para a prova (checklist)

☐ Sei reconhecer as quatro marcas da resposta certa: salvaguarda técnica, humano, transparência e monitoramento.

☐ Sei o que fazem filtros de conteúdo, tópicos negados, filtro de PII e fundamentação contextual do Bedrock Guardrails.

☐ Sei quando usar SageMaker Clarify, Model Monitor, Model Cards e Amazon A2I.

☐ Sei por que remover atributos sensíveis não elimina viés.

☐ Sei montar um fluxo de revisão humana bem governado.

☐ Sei o que deve conter uma avaliação de impacto algorítmico.

☐ Sei como dar transparência ao usuário final e permitir contestação.

☐ Sei identificar risco de propriedade intelectual em conteúdo gerado.

## Questões de revisão do capítulo

R2.1 · Diretrizes de IA responsável

**Um time de ciência de dados está criando um pipeline de MLOps na AWS para um modelo de recomendação. A liderança exige alinhamento com princípios de IA responsável, incluindo rastreabilidade, governança e monitoramento contínuo de risco. Qual abordagem está MAIS alinhada com boas práticas de IA responsável usando serviços AWS?**

A. Subir o modelo diretamente em instâncias Amazon EC2, expor via API sem autenticação para facilitar acesso e depender só de testes manuais eventuais para checar se o modelo continua adequado.
B. Desligar os logs de inferência no Amazon SageMaker para reduzir custos, mantendo só um dashboard estático de métricas de negócio em uma planilha compartilhada.
C. Usar o Model Registry para versões, o Model Monitor com o Clarify para deriva e viés em produção, alarmes no CloudWatch e trilha no CloudTrail
D. Registrar a métrica de acurácia do modelo no Amazon CloudWatch e, se houver queda de desempenho, treinar um novo modelo à mão sem registro de versões anteriores.

R2.2 · Diretrizes de IA responsável

**Uma empresa está desenvolvendo um assistente virtual baseado em IA generativa para atendimento ao cliente. Durante os testes, o modelo ocasionalmente inventa respostas técnicas que parecem confiáveis, mas são incorretas e podem levar o cliente a tomar decisões erradas. Segundo práticas de IA responsável na AWS, qual é a ação MAIS adequada a ser priorizada nesse cenário?**

A. Configurar o assistente para sempre responder com alto nível de confiança, evitando mensagens que indiquem incerteza ao usuário
B. Checar fatos em fontes internas, avisar sobre limitações do modelo e revisar respostas de alto risco
C. Treinar o modelo só com dados públicos da internet, assumindo que o grande volume de informações reduzirá as alucinações
D. Aumentar o tamanho do modelo e o número de parâmetros para reduzir a chance de erros

R2.3 · Diretrizes de IA responsável

**Uma empresa de recursos humanos está implementando um assistente de IA generativa para apoiar recrutadores na triagem inicial de currículos. Para alinhar o sistema às diretrizes de IA responsável da AWS, qual ação é a MAIS adequada durante a fase de projeto e desenvolvimento?**

A. Garantir só que o modelo tenha a menor latência possível, priorizando desempenho em vez de considerações éticas
B. Incluir uma avaliação de vieses e de representatividade nos dados de treinamento, com testes para diferentes grupos demográficos
C. Evitar cada forma de monitoramento do sistema após o lançamento para preservar a privacidade dos candidatos
D. Desligar registros (logs) de interação para reduzir custos de armazenamento, aceitando menor visibilidade sobre o comportamento do modelo

R2.4 · Diretrizes de IA responsável

**Uma empresa está definindo diretrizes internas para uso de modelos de IA generativa por seus colaboradores. Qual prática está mais alinhada com princípios de IA responsável ao permitir que funcionários usem essas ferramentas no dia a dia?**

A. Permitir o uso de IA só se os colaboradores aceitarem que as decisões serão tomadas automaticamente pelo modelo, reduzindo a intervenção humana
B. Restringir por completo o uso de IA generativa para evitar riscos, proibindo cada tipo de experimentação ou uso controlado na empresa
C. Definir políticas claras para não inserir dados sensíveis ou sigilosos, exigir revisão humana das saídas em tarefas críticas e registrar o uso de IA para auditoria
D. Orientar que os colaboradores podem inserir cada tipo de dado, inclusive informações sigilosas de clientes, desde que a saída seja revisada por um gerente

## Respostas comentadas

**R2.1. C) Usar o Model Registry para versões, o Model Monitor com o Clarify para deriva e viés em produção, alarmes no CloudWatch e trilha no CloudTrail**

Por quê: IA responsável em MLOps combina rastreabilidade de versões (Model Registry), monitoramento contínuo em produção (Model Monitor, com o Clarify para deriva de viés e de atribuição de atributos), alarmes automatizados (CloudWatch) e trilha de auditoria das ações (CloudTrail).

✗ A) Incorreta. Ignora segurança (API sem autenticação), não há versionamento estruturado nem monitoramento sistemático de risco. Depender só de testes manuais é incompatível com governança robusta e rastreabilidade em escala.

✗ B) Incorreta. Remover logs de inferência prejudica transparência, auditabilidade e a capacidade de investigar incidentes ou comportamentos indevidos.

✗ D) Incorreta. Foca apenas em acurácia, ignora viés, drift, governança de versões e trilhas de auditoria. IA responsável requer mais do que uma única métrica de performance técnica.

> **Dica de prova:** Em questões de IA responsável na AWS, procure combinações de: (1) governança e rastreabilidade (Model Registry, CloudTrail), (2) monitoramento de viés e drift (SageMaker Clarify, métricas em CloudWatch), (3) segurança e controles de acesso (IAM, logs), e (4) processos de aprovação/observabilidade contínua, não apenas...

**R2.2. B) Checar fatos em fontes internas, avisar sobre limitações do modelo e revisar respostas de alto risco**

Por quê: Esta alternativa é a mais alinhada com diretrizes de IA responsável porque reconhece que modelos generativos podem alucinar e propõe controles práticos: checagem de fatos com bases internas confiáveis, comunicação transparente sobre as limitações do sistema e revisão humana para casos sensíveis. Isso reduz riscos de dano ao usuário, aumenta a confiabilidade e está em linha com princípios de segurança, transparência e responsabilidade compartilhada.

✗ A) Incorreta. Forçar respostas com alta confiança, mesmo quando o modelo está incerto, vai contra a transparência e aumenta o risco de danos.

✗ C) Incorreta. Usar apenas dados públicos da internet não resolve o problema de precisão; pode até introduzir mais desinformação. Falta foco em validação, curadoria de fontes e mecanismos de checagem em tempo de execução.

✗ D) Incorreta. Aumentar o tamanho do modelo não garante redução de alucinações e ainda pode ampliar riscos e custos. Não aborda diretamente governança, mitigação de risco ou mecanismos de controle de qualidade das respostas.

> **Dica de prova:** Quando a questão envolver riscos de desinformação ou alucinação em IA generativa, procure a alternativa que enfatiza: transparência sobre limitações, validação de respostas (por regras, fontes confiáveis ou humanos) e controles de risco, e não apenas maior capacidade de modelo ou mais dados.

**R2.3. B) Incluir uma avaliação de vieses e de representatividade nos dados de treinamento, com testes para diferentes grupos demográficos**

Por quê: Esta alternativa está correta porque uma prática central de IA responsável é identificar e mitigar vieses nos dados e no comportamento do modelo. Em um contexto de recrutamento, isso é crítico para evitar discriminação contra grupos específicos. Avaliar representatividade dos dados, realizar testes comparando resultados entre grupos demográficos e ajustar o sistema com base nesses achados está alinhado às diretrizes da AWS de equidade, transparência e responsabilidade no uso de IA generativa.

✗ A) Incorreta. Focar somente em latência e desempenho ignora aspectos fundamentais de IA responsável, como justiça, explicabilidade, segurança e conformidade regulatória.

✗ C) Incorreta. É importante proteger a privacidade, mas isso não significa evitar qualquer monitoramento. Observabilidade e monitoramento contínuo são necessários para detectar vieses, alucinações e comportamentos indevidos, desde que feitos...

✗ D) Incorreta. Desativar logs reduz a capacidade de auditoria, explicação de decisões e investigação de problemas. As diretrizes de IA responsável recomendam registrar interações de forma adequada e segura, permitindo rastreabilidade, análise...

> **Dica de prova:** Em questões sobre IA responsável, desconfie de alternativas que priorizam apenas desempenho (latência, throughput, custo) e ignoram avaliação de vieses, transparência, monitoramento, governança e privacidade.

**R2.4. C) Definir políticas claras para não inserir dados sensíveis ou sigilosos, exigir revisão humana das saídas em tarefas críticas e registrar o uso de IA para auditoria**

Por quê: Esta alternativa está alinhada com boas práticas de IA responsável, pois combina proteção de dados (não inserir informações sensíveis ou sigilosas), supervisão humana (revisão das saídas em tarefas críticas) e mecanismos de rastreabilidade (registro de uso para auditoria). Esses elementos ajudam a reduzir riscos de privacidade, segurança e vieses, mantendo o ser humano no controle das decisões importantes.

✗ A) Incorreta. Remover a intervenção humana em decisões importantes contraria o princípio de supervisão humana significativa, além de aumentar o risco de decisões injustas ou incorretas sem possibilidade de contestação.

✗ B) Incorreta. Proibir totalmente o uso de IA elimina oportunidades de inovação e produtividade; uma abordagem responsável busca mitigar riscos com políticas, controles e monitoramento, e não bloquear a tecnologia por completo.

✗ D) Incorreta. Permitir inserir qualquer tipo de dado, incluindo sigiloso, viola princípios básicos de privacidade e proteção de dados (como LGPD) e aumenta o risco de vazamento de informações sensíveis.

**Dica de prova:** Lembre-se da tríade para IA responsável em ambientes corporativos: (1) proteção de dados (não expor informações sensíveis), (2) supervisão humana (especialmente em decisões de alto impacto) e (3) rastreabilidade/auditoria (registrar o uso e as decisões assistidas por IA).

03

CAPÍTULO 3 · 20% DA PROVA

# Fundamentos de IA e ML

Aqui eu preciso separar com segurança os tipos de aprendizado (supervisionado, não supervisionado, por reforço e autossupervisionado), as etapas de treino, validação e teste, os sinais de sobreajuste e subajuste e as métricas certas para cada problema. A prova traz cenários de negócio curtos e pergunta qual tipo de problema é, qual métrica usar ou qual forma de inferência do SageMaker AI atende. Também cobra quando ML NÃO é a resposta: fórmula definida em lei pede regra determinística. Meus erros mais comuns: confiar em acurácia com classe rara, trocar precisão por recall e confundir validação com teste. Os serviços de IA prontos (Textract, Comprehend, Transcribe, Polly, Lex, Rekognition, Personalize) caem pelo verbo do enunciado.

**Neste capítulo**

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

## Tópicos-chave

### IA, ML, deep learning, IA generativa e IA agêntica

ML é um subconjunto da IA em que o sistema aprende padrões com dados, sem regras programadas uma a uma; deep learning é um subconjunto de ML com redes neurais profundas. A IA generativa usa deep learning para criar conteúdo novo, e a IA agêntica usa esses modelos para planejar e agir com autonomia. Nem toda IA usa redes neurais: existem sistemas baseados em regras.

### Aprendizado supervisionado

Aprende com exemplos rotulados (entrada e saída esperada). Classificação prevê categorias (spam ou não spam, gato ou cão) e regressão prevê valores contínuos (preço de imóvel, minutos de entrega, demanda futura).

### Aprendizado não supervisionado

Busca padrões em dados sem rótulos: agrupamento (segmentar clientes), redução de dimensionalidade (visualizar em 2D) e detecção de anomalias. É a escolha quando o enunciado diz que não há rótulos de fraude ou que há poucos rótulos de falha: modela-se o comportamento normal e sinalizam-se os desvios.

### Aprendizado por reforço

Um agente interage com um ambiente, toma ações sequenciais, recebe recompensas ou penalidades e ajusta sua política para maximizar a recompensa acumulada. Exemplo cobrado: alocar recursos ganhando por economia e sendo punido por violar SLA, ou um agente testando ações num jogo simulado.

### Aprendizado autossupervisionado

É o pré-treinamento de modelos de linguagem: o próprio texto fornece o alvo, porque o modelo aprende a prever o próximo token, sem rótulos humanos. Não é reforço (não há recompensa de ambiente) nem agrupamento (há um alvo explícito).

### Rótulo, feature, parâmetro e hiperparâmetro

Rótulo é a saída correta associada a cada exemplo de treino. Parâmetros (pesos) são ajustados automaticamente no treino para minimizar o erro; hiperparâmetros, como taxa de aprendizado e número de épocas, são definidos antes e controlam o treino. Valores normalizados são pré-processamento das features, não rótulos.

### Treinamento x inferência

Treinar é ajustar os parâmetros com dados para minimizar o erro entre previsão e valor real; inferência é usar o modelo já treinado em dados novos. Treino não exige tempo real e inferência pode ser em tempo real ou em lote.

### Treino, validação e teste

Treino ajusta os pesos; validação ajusta hiperparâmetros, compara versões e acompanha sobreajuste durante o desenvolvimento; teste mede o desempenho final em dados nunca vistos, só depois de treino e validação. A divisão deve manter distribuição de classes semelhante, e usar o teste para escolher features ou normalizar contamina a avaliação. Só com validação cruzada (k-fold), sem teste à parte, não há medida imparcial do pipeline final.

### Sobreajuste (overfitting)

O modelo se ajusta demais ao treino (decora) e perde capacidade de generalizar: desempenho excelente no treino e queda forte na validação. Aparece em modelos de alta capacidade com pouca regularização, como uma rede profunda contra uma regressão linear simples.

### Subajuste (underfitting)

O modelo é simples demais e não aprende nem os padrões principais: erro alto e parecido no treino e na validação. A ação é usar um modelo mais expressivo ou incluir atributos mais informativos; regularização excessiva leva a subajuste.

### Métricas de classificação

Precisão = VP / (VP + FP) mede quantos alertas estão certos; recall = VP / (VP + FN) mede quantos positivos reais foram encontrados; F1 combina os dois. Com classe rara, a acurácia engana (prever sempre 'não' dá acurácia alta) e a AUC-ROC tende a ser otimista: prefira F1 da classe minoritária e curva precisão-recall, ajustando limiares.

### Métricas de regressão

MSE, RMSE e MAE medem a distância entre previsões numéricas e valores reais; são de regressão e não servem para classificar. Prever minutos de entrega a partir de distância, horário e clima é regressão avaliada por RMSE.

### Tipos de dados

Leituras de sensores a cada minuto são série temporal; fotos e texto livre são não estruturados; tabela de cadastro é tabular (estruturado). O local de armazenamento (por exemplo, o Amazon S3) não define a estrutura do dado.

### Formas de inferência no SageMaker AI

Tempo real: endpoint persistente de baixa latência para tráfego interativo. Sem servidor: tráfego intermitente que tolera partida a frio, sem pagar instância parada. Assíncrona: fila, payloads grandes e processamento longo, com resultado gravado no S3. Transformação em lote: processa um conjunto de dados inteiro sem endpoint persistente.

### MLOps no SageMaker AI

Pipelines automatizados e versionados com Model Registry (versões e aprovação) dão reprodutibilidade. Model Monitor acompanha dados e qualidade em produção e detecta deriva para disparar retreino; Feature Store reaproveita os mesmos atributos no treino e na inferência; Ground Truth organiza rotulagem humana; Data Wrangler apoia a preparação dos dados e a análise exploratória (EDA).

### Quando ML não é a resposta

Se o resultado é definido por fórmula ou tabela fixa e precisa ser exato e auditável (cálculo de IPTU, conta pela tabela de preços), use regra determinística. ML agrega valor quando há padrão a aprender, apoiando decisões humanas, escalando e automatizando, como priorizar exames urgentes para o radiologista.

### ML tradicional x modelo de base

Para decisão tabular regulada em grande volume (crédito com dezenas de atributos e exigência de explicar cada decisão), um modelo tradicional, como árvores com boosting e explicação por atributo, é mais adequado. Um LLM é não determinístico, mais caro por decisão e difícil de explicar ao regulador.

### Métricas de negócio

Ganho em acurácia offline não diz se a mudança compensa: o diretor quer retorno sobre o investimento, custo por usuário e feedback dos clientes. Tamanho do modelo, curva de perda e épocas descrevem o treino, não o impacto.

## Macetes para memorizar

### Precisão x recall

Precisão olha para o que o modelo marcou; recall olha para o que existia de fato.

### Hierarquia

Boneca russa: IA contém ML, que contém deep learning, onde vive a IA generativa; a agêntica usa esses modelos para agir.

### Tipos de aprendizado

Tem gabarito = supervisionado; não tem gabarito = não supervisionado; tem recompensa = reforço; o texto é o próprio gabarito = autossupervisionado.

### Over x under

Over decorou a prova (vai bem no treino, mal fora); under nem estudou (vai mal em tudo).

### Serviços de IA prontos

Transcribe transcreve, Polly fala, Textract extrai de documento, Comprehend compreende texto, Rekognition reconhece imagem, Lex conversa, Translate traduz, Personalize recomenda.

### Inferência no SageMaker AI

Tempo real = conversa; sem servidor = esporádico; assíncrona = fila de arquivos pesados; lote = a base inteira de uma vez.

### Validação x teste

Validação é o treino tático durante a semana; teste é o jogo final, jogado uma vez só.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **Precisão**<br>Quando o falso positivo é o erro grave: e-mail legítimo de cliente indo para a quarentena do antispam. | **Recall (revocação)**<br>Quando o falso negativo é o erro grave: fraude ou churn que passa despercebido numa classe minoritária. |
| **Sobreajuste**<br>Erro baixo no treino e alto na validação; reduzir complexidade ou regularizar mais. | **Subajuste**<br>Erro alto e parecido no treino e na validação; modelo mais expressivo ou atributos melhores. |
| **Conjunto de validação**<br>Durante o desenvolvimento, para ajustar hiperparâmetros, comparar versões e vigiar sobreajuste. | **Conjunto de teste**<br>Uma vez, no fim, para medir o desempenho final em dados nunca vistos. |
| **Aprendizado supervisionado**<br>Há rótulos ou valores conhecidos a prever: spam, preço de imóvel, clique em anúncio, demanda de vendas. | **Aprendizado não supervisionado**<br>Não há rótulos: agrupar clientes por comportamento, reduzir dimensionalidade, achar anomalias sem rótulos de fraude. |

| | |
|---|---|
| **Inferência assíncrona**<br>Requisições grandes e demoradas que chegam ao longo do dia (laudos de centenas de MB, 20 minutos cada), com resultado no S3. | **Transformação em lote**<br>Pontuar uma base inteira de uma vez, como os 8 milhões de clientes toda madrugada, sem endpoint ligado. |
| **Inferência sem servidor**<br>Poucas requisições pequenas em horários imprevisíveis, tolerando partida a frio. | **Endpoint em tempo real**<br>Tráfego interativo e constante que exige baixa latência por requisição. |
| **Amazon Textract**<br>Extrair campos, formulários e tabelas de documentos digitalizados (CNPJ, data, itens de nota fiscal). | **Amazon Comprehend**<br>Analisar texto já extraído: sentimento, entidades, temas. |

## Números que caem

| Item | Valor |
|---|---|
| Peso do domínio na prova | **20%** |
| Precisão | **VP / (VP + FP)** |
| Recall | **VP / (VP + FN)** |
| Inferência assíncrona do SageMaker AI | **payloads de até 1 GB e processamento de até uma hora** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Acurácia de 99% com classe rara parece modelo excelente.
**Como não cair:** Um modelo que sempre prevê a classe majoritária já tem acurácia alta; olhe recall, F1 da classe minoritária e curva precisão-recall.

⚠ **Alerta de prova:** AUC-ROC é invariável ao desbalanceamento, então basta.
**Como não cair:** Com forte desbalanceamento a curva ROC tende a ser otimista; a curva precisão-recall e o F1 da classe minoritária mostram o problema.

⚠ **Alerta de prova:** Recall alto é sempre o objetivo em classificação.
**Como não cair:** Leia qual erro é grave: se marcar um legítimo como positivo é pior (antispam), priorize precisão; recall alto tende a aumentar falsos positivos.

⚠ **Alerta de prova:** Validação mede o desempenho final em produção.
**Como não cair:** Isso é papel do teste; a validação serve para ajustar hiperparâmetros e comparar versões durante o desenvolvimento.

⚠ **Alerta de prova:** Usar o conjunto de teste para selecionar features ou normalizar dados é só preparação.
**Como não cair:** Qualquer uso do teste antes da avaliação final contamina a medida, pois o modelo indiretamente vê o teste.

⚠ **Alerta de prova:** Erro alto no treino e na validação pede regularização mais forte.
**Como não cair:** Isso é subajuste; regularizar mais pioraria. Vazamento de dados produz validação boa demais, e deriva só aparece em produção.

⚠ **Alerta de prova:** Qualidade caiu porque os dados mudaram: troque por rede neural profunda ou treine mais épocas.
**Como não cair:** É deriva de dados: monitore dados e qualidade com o Model Monitor e retreine com dados que reflitam a nova distribuição.

⚠ **Alerta de prova:** Pré-treino de LLM prevendo o próximo token é aprendizado por reforço ou supervisionado com rótulos manuais.
**Como não cair:** É autossupervisionado: o alvo vem do próprio texto, sem rótulos humanos nem recompensas.

⚠ **Alerta de prova:** Detectar transações suspeitas sem rótulos com semi-supervisionado ou classificação binária.
**Como não cair:** Se o cenário diz que não há rótulos, a resposta é não supervisionado (anomalias e agrupamento); semi-supervisionado exige alguns rótulos.

⚠ **Alerta de prova:** Classificação binária ou clustering para prever um valor numérico.
**Como não cair:** Valor contínuo é regressão; clustering agrupa e classificação prevê categorias.

⚠ **Alerta de prova:** Amazon Polly transcreve gravações do SAC.
**Como não cair:** Polly faz texto para fala; para transformar áudio em texto use Amazon Transcribe e, depois, Amazon Comprehend para sentimento e entidades.

⚠ **Alerta de prova:** Amazon Rekognition ou Comprehend extrai os campos de uma nota fiscal digitalizada.
**Como não cair:** Rekognition identifica objetos, cenas, rostos e conteúdo impróprio; Comprehend analisa texto já extraído. Formulários e tabelas de documentos são do Amazon Textract.

⚠ **Alerta de prova:** Um modelo de base ou uma regressão resolve um cálculo definido em lei.
**Como não cair:** Modelos aproximam e podem errar contas; fórmula legal exata e auditável é regra determinística.

⚠ **Alerta de prova:** Inferência sem servidor serve para arquivos de centenas de MB que levam 20 minutos.
**Como não cair:** Payload grande e processamento longo pedem inferência assíncrona; sem servidor é para tráfego intermitente com requisições pequenas.

## Como cai na prova

📝 **Como cai:** Cenário de negócio perguntando qual tipo de aprendizado se aplica.
**Como resolver:** Procure a palavra-chave: há rótulos ou valores históricos conhecidos? supervisionado. O texto diz que não há rótulos? não supervisionado. Há agente, ambiente e recompensa? reforço. Depois veja se o alvo é categoria (classificação) ou número (regressão).

📝 **Como cai:** Números de uma matriz de confusão e pedido de precisão e recall.
**Como resolver:** Monte VP, FP e FN: precisão = VP / marcados como positivos; recall = VP / positivos reais. Exemplo: marcou 100, acertou 80, havia 200 fraudes: precisão 80% e recall 40%. Desconfie da alternativa que traz a acurácia disfarçada.

📝 **Como cai:** Modelo com desempenho estranho entre treino, validação e produção.
**Como resolver:** Compare os erros: treino bom e validação ruim = sobreajuste; ambos ruins e parecidos = subajuste; validação boa demais = vazamento; piora ao longo do tempo em produção = deriva.

📝 **Como cai:** Escolha da forma de inferência no SageMaker AI.
**Como resolver:** Leia tamanho do payload, duração, padrão de tráfego e se alguém espera a resposta: interativo e constante = tempo real; esporádico = sem servidor; pesado e demorado ao longo do dia = assíncrona; base inteira periódica = lote.

📝 **Como cai:** Qual serviço de IA da AWS resolve a tarefa, sem treinar modelos.
**Como resolver:** Ache o verbo: extrair de documento, transcrever áudio, analisar sentimento, traduzir, conversar com intenção, moderar imagens, recomendar. Elimine o serviço que faz o caminho inverso (Polly x Transcribe).

📝 **Como cai:** Pergunta se ML, modelo de base ou regra é a melhor abordagem.
**Como resolver:** Resultado exato definido por regra: determinístico. Decisão tabular regulada em alto volume: ML tradicional explicável. Gerar conteúdo novo: IA generativa.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|

| | |
|---|---|
| sem rótulos, agrupar clientes com comportamento semelhante | **Aprendizado não supervisionado (clustering)** |
| agente, recompensa, penalidade, tentativa e erro | **Aprendizado por reforço** |
| prever preço, minutos ou demanda a partir de histórico com valores reais | **Regressão supervisionada, avaliada por RMSE** |
| classe minoritária, fraudes ou cancelamentos que passam despercebidos | **Recall (e F1 da classe minoritária)** |
| muito grave marcar um legítimo como positivo | **Precisão** |
| excelente no treino, queda forte na validação | **Sobreajuste** |
| toda madrugada, base inteira num arquivo no S3, ninguém consulta de dia | **Transformação em lote (batch transform)** |
| arquivos grandes, inferência de vários minutos, resultado depois | **Inferência assíncrona do SageMaker AI** |
| dezenas de requisições por dia em horários imprevisíveis | **Inferência sem servidor** |
| ninguém sabe que dados e código geraram o modelo em produção | **Pipelines versionados com SageMaker Model Registry** |
| distribuição dos novos clientes mudou e a qualidade caiu sem alarme | **Deriva de dados: Model Monitor e retreinamento** |
| mesmos atributos calculados no treino e na inferência em tempo real | **SageMaker Feature Store** |
| milhões de imagens sem rótulo e rotulagem feita por pessoas | **SageMaker Ground Truth** |
| fórmula definida em lei, resultado exato e auditável | **Regra determinística, não ML** |

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

- Com classe rara, acurácia não prova nada: recall, F1 e curva precisão-recall.
- Teste só no fim; tudo que usa o teste antes contamina a avaliação.
- Erro alto em treino e validação é subajuste, não sobreajuste.
- Fórmula exata e auditável não é caso de ML.

## Essenciais para a prova (checklist)

- ☐ Sei classificar um cenário como supervisionado, não supervisionado, por reforço ou autossupervisionado.
- ☐ Sei calcular precisão e recall a partir de VP, FP e FN.
- ☐ Sei qual métrica priorizar conforme o erro mais grave (falso positivo ou falso negativo).

- ☐ Sei o papel de treino, validação e teste e por que não usar o teste antes do fim.
- ☐ Sei diagnosticar sobreajuste, subajuste, vazamento e deriva pelos erros.
- ☐ Sei escolher entre inferência em tempo real, sem servidor, assíncrona e transformação em lote.
- ☐ Sei para que servem Model Registry, Model Monitor, Feature Store, Ground Truth e Data Wrangler.
- ☐ Sei qual serviço pronto usar: Textract, Comprehend, Transcribe, Polly, Translate, Lex, Rekognition e Personalize.
- ☐ Sei identificar quando a regra determinística é melhor que ML.

## Questões de revisão do capítulo

R3.1 · Fundamentos de IA e ML

**Em relação à diferença entre inteligência artificial (IA), aprendizado de máquina (ML) e deep learning (DL), qual afirmação descreve CORRETAMENTE a relação entre esses conceitos?**

A. IA é sempre baseada em redes neurais profundas, portanto toda IA é obrigatoriamente deep learning
B. ML é um subconjunto de IA, e DL é um subconjunto de ML, formando uma relação hierárquica entre os três
C. DL é um conceito mais amplo que IA, pois engloba as técnicas de tomada de decisão automatizada
D. IA, ML e DL são sinônimos e podem ser usados de forma intercambiável em todo contexto técnico

R3.2 · Fundamentos de IA e ML

**Em um experimento controlado, um cientista de dados treina três modelos diferentes para prever churn de clientes: uma regressão logística, uma árvore de decisão e um modelo de rede neural profunda. O conjunto de dados é altamente desbalanceado (apenas 5% dos clientes cancelam) e o objetivo de negócio é identificar o máximo possível de clientes que irão cancelar, mesmo aceitando mais falsos positivos. Qual MÉTRICA e QUAL ESTRATÉGIA de avaliação combinadas são MAIS adequadas para comparar esses modelos nesse contexto?**

A. Usar erro quadrático médio (MSE) e RMSE, que penalizam fortemente previsões distantes do valor real em problemas de classificação binária
B. Usar AUC-ROC, pois é invariável ao desbalanceamento
C. Usar F1 na classe minoritária e curva precisão-recall, ajustando limiares para comparar os modelos
D. Usar acurácia no conjunto de teste, pois reflete diretamente a porcentagem total de previsões corretas

R3.3 · Fundamentos de IA e ML

**Uma equipe de ciência de dados está avaliando diferentes tipos de aprendizado de máquina para um novo projeto. O objetivo é detectar padrões desconhecidos em grandes volumes de dados de telemetria de dispositivos IoT, sem rótulos prévios, para posteriormente agrupar comportamentos semelhantes. Qual abordagem de aprendizado de máquina é a mais apropriada para essa fase inicial do projeto?**

A. Aprendizado supervisionado com modelo de classificação binária
B. Aprendizado por reforço com um agente interagindo com o ambiente
C. Aprendizado semi-supervisionado usando poucos rótulos e muitos dados não rotulados
D. Aprendizado não supervisionado com algoritmos de clustering

R3.4 · Fundamentos de IA e ML

**No contexto de aprendizado de máquina supervisionado, qual é o propósito principal do conjunto de dados de validação durante o treinamento de um modelo?**

A. Ajustar hiperparâmetros e acompanhar o desempenho do modelo para evitar overfitting durante o treinamento
B. Gerar novos exemplos sintéticos para aumentar o tamanho do conjunto de treinamento
C. Fornecer exemplos rotulados ao algoritmo para que ele aprenda os padrões básicos dos dados
D. Avaliar o desempenho final do modelo em dados nunca vistos após o término de todo o processo de treinamento

## Respostas comentadas

### R3.1. B) ML é um subconjunto de IA, e DL é um subconjunto de ML, formando uma relação hierárquica entre os três

Por quê: Esta alternativa está correta porque reflete a relação hierárquica mais aceita na área: Inteligência Artificial (IA) é o campo mais amplo, que inclui qualquer técnica computacional para simular comportamentos inteligentes. Dentro de IA, o Aprendizado de Máquina (ML) é um subconjunto focado em algoritmos que aprendem padrões a partir de dados. Já o Deep Learning (DL) é um subconjunto de ML que utiliza redes neurais profundas com múltiplas camadas para aprender representações mais complexas dos dados.

✗ A) Incorreta. Porque nem toda IA usa redes neurais profundas. Há sistemas de IA baseados em regras, lógica, árvores de decisão e outros métodos que não envolvem deep learning.

✗ C) Incorreta. Porque DL não é mais amplo que IA. DL é apenas uma técnica específica (embora poderosa) dentro do conjunto de técnicas de ML, que por sua vez está dentro de IA.

✗ D) Incorreta. Porque IA, ML e DL não são sinônimos; tratam de níveis diferentes de abstração. Usá-los como equivalentes gera confusão técnica e de requisitos de projeto.

> **Dica de prova:** Lembre a hierarquia como "bonecas russas": IA (maior) > ML (dentro de IA) > DL (dentro de ML). Se a técnica envolve redes neurais profundas, ela é DL; se envolve algoritmos que aprendem com dados (árvores, regressões, SVM, redes neurais), é ML; se simula inteligência de forma geral (com ou sem aprendizado), é IA.

### R3.2. C) Usar F1 na classe minoritária e curva precisão-recall, ajustando limiares para comparar os modelos

Por quê: Esta alternativa é a mais adequada porque, em cenários de classes altamente desbalanceadas, métricas globais como acurácia tendem a mascarar o desempenho real na classe de interesse (churn). O F1-score para a classe positiva combina precisão e recall em uma única métrica harmônica, o que ajuda a avaliar se o modelo está conseguindo encontrar a maioria dos clientes que vão cancelar sem degradar demais a precisão.

✗ A) Incorreta. MSE e RMSE são métricas típicas de problemas de regressão. Embora possam ser computadas em classificadores probabilísticos, não são as mais interpretáveis nem otimizadas para o objetivo de maximizar a detecção da classe positiva...

✗ B) Incorreta. AUC-ROC pode ser útil, mas em situações de forte desbalanceamento a curva ROC tende a ser otimista, pois a taxa de falsos positivos é calculada sobre a classe majoritária.

✗ D) Incorreta. Acurácia é enganosa em dados desbalanceados. Um modelo que sempre prevê 'não churn' teria acurácia de 95%, mas seria inútil para o objetivo de identificar quem vai cancelar.

> **Dica de prova:** Em problemas de classificação com classes desbalanceadas, desconfie da acurácia alta: verifique sempre métricas focadas na classe de interesse (precision, recall, F1) e prefira curvas Precision-Recall à ROC quando a classe positiva é rara.

### R3.3. D) Aprendizado não supervisionado com algoritmos de clustering

Por quê: O problema descreve explicitamente a ausência de rótulos ('sem rótulos prévios') e a necessidade de 'agrupar comportamentos semelhantes'. Isso caracteriza um cenário clássico de aprendizado não supervisionado, em que algoritmos de clustering (como k-means, DBSCAN ou clustering hierárquico) são usados para encontrar estruturas e padrões ocultos nos dados sem depender de classes pré-definidas.

✗ A) Incorreta. Aprendizado supervisionado com classificação binária exige dados rotulados (por exemplo, 'normal' vs. 'anômalo').

✗ B) Incorreta. Aprendizado por reforço envolve um agente que toma ações em um ambiente e recebe recompensas ou punições. O cenário fala em analisar telemetria histórica de IoT, não em controlar ativamente o ambiente nem em receber sinais de...

✗ C) Incorreta. Aprendizado semi-supervisionado combina poucos dados rotulados com muitos não rotulados. O caso descreve explicitamente uma fase inicial sem rótulos;

> **Dica de prova:** Quando o enunciado mencionar 'sem rótulos', 'descobrir padrões', 'agrupar dados semelhantes' ou 'estrutura oculta nos dados', pense em aprendizado não supervisionado, especialmente clustering e redução de dimensionalidade.

**R3.4. A) Ajustar hiperparâmetros e acompanhar o desempenho do modelo para evitar overfitting durante o treinamento**

Por quê: Esta alternativa está correta porque o conjunto de validação é usado principalmente para monitorar o desempenho do modelo durante o treinamento e ajustar hiperparâmetros (como taxa de aprendizado, número de árvores, número de épocas etc.). Ele ajuda a identificar quando o modelo começa a decorar o conjunto de treinamento (overfitting), permitindo técnicas como early stopping e seleção do melhor conjunto de hiperparâmetros.

✗ B) Incorreta. Diz respeito a técnicas de aumento de dados (data augmentation) ou geração sintética, que não são a função principal de um conjunto de validação.

✗ C) Incorreta. Refere-se ao conjunto de treinamento, que é de fato usado para o modelo aprender padrões a partir de exemplos rotulados.

✗ D) Incorreta. Descreve o papel do conjunto de teste, não o de validação. O teste é usado apenas ao final, para medir o desempenho em dados totalmente novos.

**Dica de prova:** Lembre-se da sigla intuitiva: Treino-Validação-Teste = Aprender-Ajustar-Avaliar. O modelo aprende no treino, é ajustado (hiperparâmetros) na validação e é avaliado de forma final no teste.

04

CAPÍTULO 4 · 24% DA PROVA

# Fundamentos de IA generativa

Neste domínio eu preciso dominar o que é IA generativa, modelo de base, LLM, token, embedding, janela de contexto e amostragem, além das limitações (alucinação, não determinismo, baixa interpretabilidade). A versão 1.1 do guia acrescentou IA agêntica: uso de ferramentas, MCP, multiagente, memória e engenharia de contexto, e serviços como Amazon Bedrock AgentCore, Strands Agents, Kiro, Amazon Quick e AWS Transform. Muitas perguntas são conceituais ('qual descreve melhor') e o erro vem de alternativas absolutas ('elimina completamente', 'sempre', '100%'). Também cai custo e capacidade: cobrança por token, perfis de inferência entre regiões e Provisioned Throughput.

**Neste capítulo**
18 tópicos-chave · 14 pegadinhas · 6 macetes · 14 gatilhos · 4 questões comentadas

## Tópicos-chave

### IA generativa

Aprende a distribuição dos dados e amostra conteúdo novo e semelhante aos exemplos (texto, imagem, áudio, código). Modelos generativos modelam p(x) ou p(x,y); discriminativos aprendem p(y|x) para prever ou classificar. IA generativa não dispensa dados de treino e não substitui todos os modelos tradicionais.

### Modelo de base (foundation model)

Modelo grande e genérico, pré-treinado em grandes volumes de dados diversos, em geral sem rótulos, e adaptável a muitas tarefas por prompts, RAG ou ajuste fino. LLMs, modelos de difusão de imagem e modelos generativos de áudio são modelos de base; um modelo de recomendação clássico não é.

### LLM

Modelo generativo de sequência treinado em muito texto para prever o próximo token e, a partir de instruções, produzir texto coerente. Aprende padrões e representações, não regras fixas, e não é um banco de respostas; pode repetir trechos, mas o comportamento principal é gerar combinações novas.

### Transformador e atenção

Os LLMs se tornaram dominantes porque o mecanismo de atenção captura dependências de longo alcance em paralelo, escalando em grandes volumes de texto. Transformadores precisam de muito mais dados, não menos, e também alucinam.

### Token e janela de contexto

Token é a unidade básica de entrada e saída (pedaços de palavras, palavras ou símbolos) e a unidade de cobrança. A janela de contexto define quanto texto o modelo considera; janela maior mantém coerência em documentos longos, mas aumenta o custo e não garante respostas corretas.

### Embeddings

Vetores numéricos que representam o significado do conteúdo; documentos e perguntas viram vetores e a busca usa os mais próximos. Numa base de conhecimento, um modelo de embeddings indexa e um modelo de texto redige a resposta.

### Amostragem e não determinismo

O modelo escolhe o próximo token a partir de uma distribuição de probabilidades, então o mesmo prompt pode gerar respostas diferentes com a temperatura padrão. Escolher sempre o token mais provável (greedy) é determinístico.

### Alucinação

Conteúdo plausível, dito com confiança, mas parcial ou totalmente incorreto. Não é sobreajuste, ocorre em modelos proprietários e abertos, e não some com mais dados ou modelo menor; reduz-se com RAG em fontes confiáveis, citação das fontes e verificação de fundamentação contextual do Bedrock Guardrails.

### Baixa interpretabilidade

Com bilhões de parâmetros em redes profundas, é difícil explicar como um LLM chegou a uma resposta, ao contrário de uma regressão logística. Não confunda com deriva, custo ou latência.

### Ciclo de vida de um modelo de base

Seleção de dados e modelo, pré-treino, ajuste fino, avaliação, implantação e feedback. O ajuste fino parte de um modelo já pré-treinado, e a avaliação depende de um modelo treinado.

### Vantagem de usar modelos prontos via API

Permite experimentar rapidamente modelos e casos de uso com menor custo e complexidade, sem grandes volumes de dados e infraestrutura. Não garante qualidade sem ajustes, não elimina segurança e governança de dados e é cobrado por uso (tokens de entrada e saída no sob demanda).

### IA agêntica

Planeja etapas, usa ferramentas e age com autonomia para cumprir um objetivo, usando modelos de base como motor de raciocínio. Para processos com etapas fixas, um fluxo orquestrado é mais previsível e auditável que um agente autônomo.

### Uso de ferramentas e MCP

No uso de ferramentas (tool use ou function calling), o modelo recebe a descrição da ferramenta, decide chamá-la e usa o resultado, acessando dados que mudam a cada momento. O Model Context Protocol (MCP) padroniza a conexão de agentes a sistemas como CRM e chamados, em vez de uma integração por combinação de agente e ferramenta.

### Colaboração multiagente

Um agente supervisor entende o pedido e distribui subtarefas para agentes especializados, como contestar cobrança e ajustar limite na mesma conversa. Só vale quando o problema exige especialização e coordenação; vários agentes aumentam a complexidade.

### Engenharia de contexto

Selecionar e resumir o que entra na janela a cada turno (prompt de sistema, histórico, documentos, saídas de ferramentas). Excesso de conteúdo irrelevante piora as respostas e o custo; janela maior ou mexer na temperatura não resolve.

### Amazon Bedrock AgentCore

Coloca agentes em produção sem gerenciar servidores, com isolamento de sessão, identidade, memória e observabilidade, inclusive para agentes feitos com LangGraph e Strands Agents. O AgentCore Memory guarda o contexto da conversa e preferências entre sessões.

### Strands Agents, Kiro, Amazon Quick e AWS Transform

Strands Agents é um SDK de agentes de código aberto em Python: você define modelo, prompt e ferramentas e o modelo planeja. Kiro é um ambiente com agentes que transforma o pedido em especificações (requisitos, design e tarefas) antes do código. Amazon Quick atende a área de negócio com perguntas em linguagem natural, painéis e agentes; AWS Transform usa agentes para modernizar .NET Framework e mainframe.

### Custo e capacidade no Bedrock

No sob demanda o custo depende dos tokens de entrada e saída; prompt de sistema longo e histórico inteiro a cada mensagem inflam a fatura. Perfis de inferência entre regiões ampliam a capacidade sob demanda sem compromisso (geográfico para manter residência, global pode sair da geografia); Provisioned Throughput reserva capacidade dedicada cobrada por hora.

## Macetes para memorizar

### 💡 Discriminativo x generativo

Discriminativo decide sobre o que existe; generativo cria o que não existia.

### 💡 Fine-tuning x prompt

Ajuste fino mexe nos pesos; prompt mexe nas palavras.

**Agente**

Agente = cérebro (modelo de base) + mãos (ferramentas) + plano (autonomia).

**Custo sob demanda**

Paga o que entra e o que sai: tokens de entrada + tokens de saída.

**Agente ou fluxo**

Caminho fixo, fluxo; caminho incerto, agente.

**Família Amazon Nova**

Micro = só texto, rápido e barato; Pro = multimodal para tarefas complexas; Canvas = imagem; Reel = vídeo.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **Modelo discriminativo**<br>Prever ou classificar dados existentes: spam, fraude, demanda. | **Modelo generativo**<br>Criar conteúdo novo a partir de poucas palavras-chave ou instruções, como descrições de produtos. |
| **Ajuste fino (fine-tuning)**<br>Especializar o modelo alterando os pesos com dados adicionais. | **Engenharia de prompt**<br>Mudar instruções e exemplos no prompt, sem mexer nos pesos; o modelo fica congelado. |
| **LLM puro**<br>Responder com o conhecimento guardado nos pesos. | **RAG**<br>Somar ao modelo a recuperação de fontes externas atualizadas na hora da pergunta, reduzindo alucinações. |
| **Agente autônomo**<br>O caminho não é conhecido antes: o modelo planeja, escolhe ferramentas e decide. | **Fluxo orquestrado com etapas definidas**<br>As etapas são sempre as mesmas (validar nota, checar política, encaminhar): mais previsível e auditável. |
| **LLM**<br>Entender e gerar texto. | **Modelo de difusão**<br>Gerar e modificar imagens e outros dados contínuos. |
| **Perfil de inferência geográfico**<br>Picos de limitação com exigência de manter o processamento numa geografia, como os EUA. | **Perfil de inferência global**<br>Máxima capacidade sem restrição de onde o processamento ocorre. |
| **Modelo personalizado sob demanda**<br>Uso baixo e irregular de um modelo ajustado: paga pelo uso. | **Provisioned Throughput**<br>Volume alto e constante que exige capacidade garantida e previsível. |

## Números que caem

| Item | Valor |
|---|---|
| Peso do domínio na prova | **24%** |

| | |
|---|---|
| Ordem do ciclo de vida de um modelo de base | **seleção de dados, seleção de modelo, pré-treinamento, ajuste fino, avaliação, implantação e feedback** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Aumentar a janela de contexto reduz custo ou garante respostas corretas.
**Como não cair:** Janela maior permite mais tokens, o que tende a aumentar custo e latência, e não elimina alucinações.

⚠ **Alerta de prova:** Modelos menores alucinam menos.
**Como não cair:** Reduzir o tamanho não elimina alucinações; o que ajuda é fundamentar em fontes confiáveis (RAG com citação) e verificar a fundamentação.

⚠ **Alerta de prova:** Alucinação é sinônimo de sobreajuste.
**Como não cair:** Sobreajuste é decorar o treino; alucinação é inventar fatos plausíveis na geração.

⚠ **Alerta de prova:** Transformadores precisam de menos dados que RNNs e não alucinam.
**Como não cair:** Precisam de muito mais dados e também alucinam; a vantagem é a atenção, que captura dependências longas em paralelo.

⚠ **Alerta de prova:** Dar exemplos no prompt (few-shot) é re-treinar o modelo.
**Como não cair:** É aprendizado em contexto: o modelo permanece congelado; só o ajuste fino muda os pesos.

⚠ **Alerta de prova:** Modelos gerenciados são sempre mais baratos, dispensam ajuste fino ou impedem controles de acesso.
**Como não cair:** Custo depende do cenário; ainda pode ser preciso ajustar prompts ou fazer ajuste fino; serviços gerenciados facilitam monitoramento e controle de acesso. O ganho real é transferir infraestrutura, atualização, segurança e escala ao provedor.

⚠ **Alerta de prova:** Perfil global de inferência resolve picos mantendo o processamento nos EUA.
**Como não cair:** O perfil global pode rotear para fora dos EUA; com restrição de residência use o perfil geográfico. Provisioned Throughput não é capacidade extra sem compromisso: reserva capacidade dedicada cobrada por hora.

⚠ **Alerta de prova:** AWS Step Functions é o SDK para um agente em que o modelo decide as ferramentas.
**Como não cair:** Step Functions orquestra fluxos definidos previamente; o agente em poucas linhas de Python é o Strands Agents.

⚠ **Alerta de prova:** Processo com três etapas fixas pede sistema multiagente ou agente autônomo.
**Como não cair:** Etapas fixas pedem fluxo orquestrado; delegar a ordem a um LLM acrescenta não determinismo, custo e risco.

⚠ **Alerta de prova:** Prompt Management, Guardrails ou Feature Store guardam preferências do cliente entre sessões.
**Como não cair:** Memória de curto e longo prazo de agentes é o Amazon Bedrock AgentCore Memory.

⚠ **Alerta de prova:** RAG, LoRA ou RLHF padronizam a conexão de agentes a ferramentas.
**Como não cair:** Quem padroniza a chamada de ferramentas e sistemas é o MCP; LoRA e RLHF são técnicas de ajuste, RAG traz documentos ao prompt.

⚠ **Alerta de prova:** Contexto inchado se resolve movendo o prompt de sistema, aumentando a temperatura ou trocando por janela maior.
**Como não cair:** Selecione e resuma o que entra a cada turno (engenharia de contexto).

⚠ **Alerta de prova:** Destilação, ajuste fino ou cache de prompts dão acesso à situação atual de um pedido.
**Como não cair:** Dado que muda a cada momento e existe só numa API vem por uso de ferramentas (function calling).

⚠ **Alerta de prova:** Trocar de região ou aumentar a temperatura reduz a fatura de um chatbot.
**Como não cair:** Custo vem dos tokens processados: encurte o prompt de sistema e resuma o histórico antigo.

## Como cai na prova

📝 **Como cai:** Pergunta conceitual: qual afirmação descreve melhor IA generativa, LLM ou modelo de base.
**Como resolver:** Procure as ideias-chave: aprende padrões estatísticos com muitos dados, prevê o próximo token, gera conteúdo novo, adaptável a várias tarefas. Elimine alternativas com regras fixas, banco de respostas, 'só texto', 'só dados rotulados' ou garantias absolutas.

📝 **Como cai:** Cenário agêntico pedindo o serviço da AWS.
**Como resolver:** Mapeie a necessidade: memória = AgentCore Memory; produção com identidade e observabilidade = AgentCore; conectar ferramentas = MCP; SDK Python = Strands Agents; especificação antes do código = Kiro; negócio sem técnico = Amazon Quick; modernização = AWS Transform.

📝 **Como cai:** Limitação de IA generativa num caso concreto.
**Como resolver:** Respostas diferentes para o mesmo prompt = não determinismo; fatos inventados = alucinação; não dá para explicar = interpretabilidade; resultado reproduzível e auditável em etapas fixas = prefira fluxo determinístico.

📝 **Como cai:** Custo ou capacidade do Bedrock.
**Como resolver:** Fatura crescendo = tokens (encurte e resuma); picos sem compromisso = entre regiões (geográfico se houver residência); uso baixo de modelo ajustado = sob demanda; volume alto e constante = Provisioned Throughput.

📝 **Como cai:** Ordenar o ciclo de vida de um modelo de base.
**Como resolver:** Seleção de dados e modelo, pré-treino, ajuste fino, avaliação, implantação e feedback; nada de ajustar antes de pré-treinar ou implantar antes de avaliar.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| prever a próxima palavra ou token em grandes volumes de texto | **LLM, pré-treino autossupervisionado** |
| vetor numérico e documentos mais próximos da pergunta | **Embeddings e busca semântica** |
| mesmo prompt, respostas diferentes | **Não determinismo da amostragem (temperatura)** |
| resposta plausível, confiante e inventada | **Alucinação: RAG com citações e fundamentação contextual** |
| difícil explicar como o LLM chegou à resposta | **Baixa interpretabilidade** |
| planeja etapas, usa ferramentas, age com autonomia | **IA agêntica** |
| padrão para conectar agentes ao CRM e ao sistema de chamados | **Model Context Protocol (MCP)** |
| lembrar o que o cliente pediu e preferências em sessões futuras | **Amazon Bedrock AgentCore Memory** |
| produção para agentes LangGraph ou Strands com identidade, isolamento e observabilidade | **Amazon Bedrock AgentCore** |
| agente em Python com poucas linhas, o modelo planeja as ferramentas | **Strands Agents** |
| pedido em linguagem natural vira requisitos, design e lista de tarefas | **Kiro** |
| área de negócio sem conhecimento técnico, perguntas sobre vendas, painéis e agentes | **Amazon Quick** |

| | |
|---|---|
| modernizar .NET Framework e mainframe com agentes | **AWS Transform** |
| erros de limite em picos, sem compromisso de prazo | **Perfil de inferência entre regiões** |

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

- Alucinação se combate com fontes: RAG com citação e verificação de fundamentação.
- Few-shot não mexe nos pesos; ajuste fino mexe.
- Agente é para caminho incerto; etapas fixas pedem fluxo orquestrado.
- Custo do Bedrock sob demanda = tokens de entrada + saída.
- Serviços novos da v1.1: AgentCore (Memory, Identity, Policy), Strands Agents, Kiro, Amazon Quick, AWS Transform.

## Essenciais para a prova (checklist)

☐ Sei diferenciar modelo generativo de discriminativo.

☐ Sei definir modelo de base, LLM, token, embedding e janela de contexto.

☐ Sei explicar alucinação, não determinismo e baixa interpretabilidade e como mitigar cada um.

☐ Sei por que a atenção dos transformadores tornou os LLMs dominantes.

☐ Sei quando usar agente autônomo, fluxo orquestrado ou multiagente.

☐ Sei o que fazem MCP, uso de ferramentas e engenharia de contexto.

☐ Sei para que servem AgentCore, AgentCore Memory, Strands Agents, Kiro, Amazon Quick e AWS Transform.

☐ Sei o que determina o custo no Bedrock sob demanda e como reduzi-lo.

☐ Sei escolher entre perfil geográfico, perfil global e Provisioned Throughput.

## Questões de revisão do capítulo

R4.1 · Fundamentos de IA generativa

**Um engenheiro de machine learning está avaliando diferentes abordagens para gerar texto com alta coerência semântica usando modelos de linguagem. Ele deseja explicar a um colega por que modelos de base (foundation models) de IA generativa conseguem produzir saídas mais ricas e contextuais em comparação com modelos clássicos de NLP baseados em técnicas como bag-of-words ou n-gramas. Qual opção descreve de forma mais precisa o fundamento técnico que diferencia os modelos de base usados em IA generativa dessas abordagens clássicas?**

A. Modelos de base aprendem embeddings e relações de contexto em muitos dados; modelos clássicos usam contagens locais sem contexto longo
B. Modelos de base utilizam regras simbólicas e gramáticas formais definidas à mão para capturar a estrutura da linguagem.
C. Modelos de base são treinados só em textos curtos e anotados à mão, o que reduz o ruído nos dados e melhora a coerência dos textos gerados em relação a modelos clássicos, que usam grandes corpora não anotados.
D. Modelos de base utilizam só algoritmos de busca e recuperação de documentos, sem realizar geração real de texto, enquanto modelos clássicos de NLP geram texto por meio de cadeias de Markov e modelos probabilísticos simples.

R4.2 · Fundamentos de IA generativa

**Um assistente precisa responder 'qual é a situação atual do meu pedido de reembolso?', informação que só existe em uma API interna. O modelo recebe a descrição da ferramenta, decide chamá-la com os parâmetros certos, a aplicação executa a chamada e devolve o resultado ao modelo. Como se chama esse mecanismo?**

A. Uso de ferramentas (tool use ou function calling)
B. Destilação do modelo a partir das respostas da API
C. Ajuste fino supervisionado com dados da API
D. Cache de prompts com o resultado da última consulta

R4.3 · Fundamentos de IA generativa

**Em um fluxo moderno de desenvolvimento com modelos de linguagem generativa (LLMs), qual opção descreve de forma mais precisa a principal diferença conceitual entre 'fine-tuning' e 'prompt engineering' em relação ao controle do comportamento do modelo?**

A. O fine-tuning é usado só para corrigir alucinações, enquanto o prompt engineering é voltado só para tarefas de classificação de texto.
B. Fine-tuning altera os pesos para especializar o modelo; prompt engineering muda as instruções sem mexer nos pesos
C. O fine-tuning substitui completamente o conjunto de dados de pré-treinamento original.
D. O fine-tuning serve só para reduzir custos de inferência, enquanto o prompt engineering é a única técnica que realmente melhora a qualidade das respostas do modelo.

R4.4 · Fundamentos de IA generativa

**Uma equipe técnica está avaliando diferentes tipos de modelos usados em IA generativa. Eles querem entender o que diferencia especificamente um modelo generativo de outros modelos de machine learning. Qual característica descreve de forma mais precisa um modelo generativo?**

A. Um modelo que aprende a distribuição dos dados e cria amostras novas semelhantes às de treino
B. Um modelo que é treinado só para reduzir o erro de previsão em problemas de séries temporais financeiras
C. Um modelo que só classifica entradas em categorias pré-definidas
D. Um modelo que aprende a mapear entradas para saídas fixas usando regras explicitamente programadas por especialistas

## Respostas comentadas

### R4.1. A) Modelos de base aprendem embeddings e relações de contexto em muitos dados; modelos clássicos usam contagens locais sem contexto longo

Por quê: Esta alternativa está correta porque descreve o aspecto central dos modelos de base em IA generativa: eles são redes neurais de grande escala (como transformers) treinadas em enormes corpora, aprendendo embeddings de alta dimensão e capturando dependências de longo alcance entre tokens. Isso permite compreensão de contexto amplo, coerência semântica e capacidade de generalização, superando abordagens tradicionais que dependem de estatísticas locais e representações esparsas.

✗ B) Incorreta. Porque modelos de base não se baseiam em regras simbólicas ou gramáticas definidas manualmente; eles aprendem padrões diretamente dos dados de forma distribuída.

✗ C) Incorreta. Porque, na prática, modelos de base são treinados em grandes volumes de dados geralmente não anotados (auto-supervisão), e não apenas em textos curtos e anotados manualmente.

✗ D) Incorreta. Porque confunde recuperação com geração. Foundation models de IA generativa realmente geram texto token a token com base em distribuições condicionais aprendidas.

> **Dica de prova:** Quando a questão falar em 'modelos de base' ou 'foundation models' em IA generativa, pense em três pilares: (1) grande escala de parâmetros e dados; (2) aprendizado de representações distribuídas (embeddings) com contexto amplo;

### R4.2. A) Uso de ferramentas (tool use ou function calling)

Por quê: Uso de ferramentas é um conceito central de IA agêntica no guia v1.1, ao lado de memória, orquestração de fluxos e padrões multiagente. No Amazon Bedrock, o modelo não chama a ferramenta diretamente: ele indica a ferramenta e os argumentos, e a aplicação executa a chamada e devolve o resultado.

✗ B) Incorreta. A destilação transfere comportamento de um modelo maior para um menor; não consulta sistemas em tempo real.

✗ C) Incorreta. O ajuste fino muda os pesos com exemplos fixos; não dá acesso a dados que mudam a cada momento, como a situação de um pedido.

✗ D) Incorreta. O cache reaproveita prefixos de prompt; devolver um resultado antigo não atende a uma consulta atualizada.

> **Dica de prova:** Modelo decide chamar uma API e usa o resultado = tool use (function calling).

**R4.3. B) Fine-tuning altera os pesos para especializar o modelo; prompt engineering muda as instruções sem mexer nos pesos**

Por quê: Esta alternativa é correta porque o fine-tuning implica treinar novamente o modelo (total ou parcialmente) em um novo conjunto de dados, atualizando pesos internos para especializá-lo em domínios, estilos ou tarefas específicas. Já o prompt engineering atua na camada de instruções e contexto fornecidos na entrada, guiando o comportamento do modelo sem qualquer alteração dos parâmetros do modelo base.

✗ A) Incorreta. É equivocada porque o fine-tuning não é usado exclusivamente para corrigir alucinações, e prompt engineering não é restrito a tarefas de classificação.

✗ C) Incorreta. O fine-tuning não substitui o conjunto de dados de pré-treinamento, mas o complementa com dados adicionais. Também não há relação direta entre prompt engineering e substituição de tokenizador;

✗ D) Incorreta. O fine-tuning não tem como objetivo principal reduzir custos de inferência; ele foca em adaptação e especialização.

> **Dica de prova:** Pense em fine-tuning como 'mudar o modelo' (ajuste de pesos) e em prompt engineering como 'mudar a conversa' (como você pergunta). Se a técnica altera parâmetros do modelo, é fine-tuning; se apenas muda texto de entrada, é prompt engineering.

**R4.4. A) Um modelo que aprende a distribuição dos dados e cria amostras novas semelhantes às de treino**

Por quê: Esta alternativa é a correta porque modelos generativos aprendem a modelar a distribuição de probabilidade dos dados (por exemplo, texto, imagens, áudio) e, a partir disso, conseguem gerar novas amostras que são coerentes e semelhantes às do conjunto de treinamento. Essa é justamente a base de modelos de difusão, VAEs, GANs e de muitos modelos de linguagem de grande porte (LLMs).

✗ B) Incorreta. Limita o objetivo ao contexto de previsão de séries temporais financeiras, que é um caso específico de modelagem preditiva, não a definição de modelo generativo.

✗ C) Incorreta. Descreve modelos discriminativos de classificação, que focam em atribuir rótulos às entradas, não em gerar novos dados.

✗ D) Incorreta. Fala de sistemas baseados em regras, que não aprendem a partir de dados estatisticamente, mas seguem lógica manualmente programada, o que não caracteriza modelos generativos modernos.

> **Dica de prova:** Quando pensar em modelos generativos, lembre-se da ideia de "aprender o padrão dos dados" para poder criar algo novo e plausível (texto, imagem, código), enquanto modelos discriminativos focam em decidir entre classes ou prever valores para entradas já dadas.

05

CAPÍTULO 5 · 14% DA PROVA

# Segurança, conformidade e governança para soluções de IA

Neste domínio eu preciso montar a combinação certa de controles: IAM de menor privilégio, CloudTrail, logs de invocação do Bedrock, KMS com chaves gerenciadas pelo cliente, PrivateLink, Guardrails para PII, SCPs do AWS Organizations e residência de dados por região. A prova traz cenários regulados (financeiro, saúde, setor público, UE) com vários requisitos, e as alternativas erradas trazem um controle que não resolve o problema pedido (WAF para PII, criptografia para bloquear conteúdo, CloudTrail para impedir ações). A v1.1 acrescentou AgentCore Identity e Policy in AgentCore. Também cai a Generative AI Security Scoping Matrix e o fato de o Bedrock não usar prompts e respostas para treinar modelos.

**Neste capítulo**
18 tópicos-chave · 14 pegadinhas · 6 macetes · 14 gatilhos · 4 questões comentadas

## Tópicos-chave

### Proteção de dados no Amazon Bedrock

O Bedrock não usa prompts e respostas para treinar modelos nem os compartilha com os provedores de modelo. Isso não dispensa controles do cliente: IAM, criptografia, logs e minimização.

### IAM com menor privilégio

Políticas baseadas em identidade liberam só as ações e os modelos aprovados, associadas a roles por aplicação; condições por tag (ABAC) refinam o acesso. Usuário raiz, chaves estáticas compartilhadas, credenciais no código ou no prompt e roles administrativas amplas são sempre erradas.

### AWS Organizations e SCPs

SCPs impõem centralmente, em várias contas, quais serviços e modelos podem ser usados. Contas separadas por time ou ambiente (produção e teste) dão isolamento e segregação melhores que VPCs na mesma conta.

### AWS CloudTrail

Registra chamadas de API: quem chamou qual modelo e quando. Com eventos de dados, bucket com Object Lock (logs imutáveis), CloudTrail Lake e Athena atende auditoria completa; mas não bloqueia nada, registra depois do fato.

### Logs de invocação do Bedrock

Guardam o conteúdo de prompts e respostas no CloudWatch Logs ou no S3, complementando o CloudTrail. Devem ser criptografados com KMS, ter acesso restrito e retenção limitada; guardar indefinidamente aumenta o risco.

### AWS KMS e chaves gerenciadas pelo cliente

Quando o requisito exige chaves gerenciadas pelo cliente, chaves gerenciadas pela AWS ou SSE-S3 não atendem; use CMKs do KMS em S3, EBS, logs e artefatos. Criptografia protege confidencialidade, mas não impede PII no prompt nem conteúdo nocivo.

### AWS PrivateLink e endpoints VPC

Acesso privado ao Bedrock e ao SageMaker sem passar pela internet; políticas de endpoint podem bloquear acesso público e saída para a internet, prevenindo exfiltração. Rede não decide quais modelos podem ser usados: isso é IAM.

### Guardrails para dados sensíveis

O filtro de informações sensíveis do Bedrock Guardrails detecta e mascara PII (e PHI) em entradas e saídas, e a API ApplyGuardrail aplica a política fora da chamada ao modelo. A verificação de fundamentação contextual bloqueia respostas não apoiadas nos trechos recuperados.

### Anonimização, pseudonimização e minimização

Mascarar ou remover PII antes de enviar o prompt (por exemplo, com uma ferramenta de DLP) e antes de gravar logs da aplicação. HTTPS protege o transporte, mas não limita quem vê nem o que o modelo recebe.

### Residência de dados

Dados de clientes europeus processados e armazenados em regiões da UE, com políticas de dados e CMKs por região, para GDPR; o mesmo raciocínio vale para LGPD por jurisdição. Uma única região global ou us-east-1 não atende residência.

### Generative AI Security Scoping Matrix

Cinco escopos, do menor ao maior controle: 1 aplicativo de consumo, 2 aplicativo corporativo de terceiros, 3 modelos pré-treinados em aplicação própria, 4 modelos ajustados, 5 modelos treinados do zero. Quanto maior o escopo, mais responsabilidades de segurança, governança e conformidade ficam com a organização.

### Amazon Macie

Descobre e classifica dados sensíveis no Amazon S3, apoiando a minimização do que um chatbot pode acessar, junto com CloudTrail para auditoria. Não controla acesso nem concede acesso a agentes.

### AWS Lake Formation

Permissões baseadas em tags do Glue Data Catalog liberam só conjuntos aprovados e aplicam controle por coluna e linha no data lake. Junto com IAM e CloudTrail, governa o que o modelo e os cientistas podem ler.

### SageMaker em ambiente protegido

Treino em VPC privada com endpoints VPC para o S3, criptografia em repouso e em trânsito com KMS e IAM de menor privilégio. ML Lineage Tracking registra a linhagem de dados, jobs e artefatos de cada modelo para comprovar em auditoria quais dados geraram o modelo.

### Amazon Bedrock AgentCore Identity

Permite que um agente acesse serviços como agenda e CRM de terceiros em nome de cada usuário, com consentimento, sem guardar senhas no código e com trilha de auditoria.

### Policy in AgentCore

Regras determinísticas avaliadas no AgentCore Gateway sobre as chamadas de ferramentas, como impedir estorno acima de um limite sem aprovação, independentemente do que o modelo decidir. Instruções no prompt e guardrails são probabilísticos ou focados em conteúdo.

### Conformidade e configuração

O escopo de certificações como ISO e SOC e o acesso privado via PrivateLink ajudam a demonstrar conformidade do serviço; AWS Artifact fornece relatórios de conformidade da AWS e AWS Config monitora a conformidade e a configuração dos recursos.

### Serviços que não resolvem o problema de IA

WAF protege aplicações web contra ataques como SQL injection; Shield protege contra DDoS; GuardDuty detecta ameaças; Inspector gerencia vulnerabilidades; Secrets Manager guarda segredos. Nenhum deles filtra PII em prompts, define modelos aprovados ou detecta viés.

## Macetes para memorizar

**Trio básico**

IAM controla quem, CloudTrail registra quem fez, KMS tranca os dados.

**Registro**

CloudTrail = quem chamou; log de invocação = o que foi dito.

### Scoping Matrix

Consumo, corporativo de terceiros, pré-treinado no meu app, ajustado, do zero: de 1 a 5, cada degrau traz mais responsabilidade.

### Controle certo para o risco

PII no conteúdo = Guardrails; rede = PrivateLink; identidade = IAM; regra de agente = Policy; dados no S3 = Macie.

### Registra x bloqueia

CloudTrail registra, não bloqueia; Policy bloqueia, não registra conteúdo.

### Distratores de segurança

WAF, Shield, GuardDuty e Inspector são segurança de infraestrutura, não governança de IA.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **AWS CloudTrail**<br>Saber quem chamou qual API e modelo, quando. | **Logs de invocação do Bedrock**<br>Guardar o conteúdo dos prompts e das respostas para auditoria. |
| **Filtro de informações sensíveis do Guardrails**<br>Detectar e mascarar CPF, cartão e outros PII no conteúdo de entradas e saídas. | **AWS KMS**<br>Criptografar dados em repouso e gerenciar chaves; não analisa o conteúdo. |
| **Chave gerenciada pelo cliente (CMK)**<br>O requisito exige controle das chaves pelo cliente (rotação, políticas, revogação). | **Chave gerenciada pela AWS**<br>Basta criptografia padrão, sem exigência de controle das chaves. |
| **Policy in AgentCore**<br>Regra de negócio que o agente não pode violar, aplicada de forma determinística. | **Instrução no prompt de sistema**<br>Orientar estilo e comportamento, sem garantia de cumprimento. |
| **Amazon Macie**<br>Descobrir e classificar dados sensíveis no S3. | **Amazon GuardDuty**<br>Detectar ameaças e atividades maliciosas na conta. |
| **IAM**<br>Controlar quem pode invocar quais modelos. | **PrivateLink e grupos de segurança**<br>Isolar o tráfego de rede e evitar a internet. |
| **Escopo 3 da Scoping Matrix**<br>Modelo pré-treinado, sem ajuste, numa aplicação própria. | **Escopo 4**<br>Modelo ajustado com os dados da própria empresa. |

## Números que caem

| Item | Valor |
|---|---|
| Peso do domínio na prova | **14%** |

| | |
|---|---|
| Escopos da Generative AI Security Scoping Matrix | **5 (1 consumo, 2 corporativo de terceiros, 3 pré-treinado em app próprio, 4 ajustado, 5 do zero)** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Desligar logs ou o CloudTrail reduz o risco de exposição.
**Como não cair:** Elimina a trilha de auditoria; mantenha os logs criptografados, com acesso restrito e retenção limitada.

⚠ **Alerta de prova:** HTTPS/TLS basta para proteger dados sensíveis enviados ao modelo.
**Como não cair:** Protege só o transporte; é preciso mascarar PII, controlar acesso e garantir que os dados não treinem modelos.

⚠ **Alerta de prova:** Criptografia em repouso impede PII nos prompts e nas respostas.
**Como não cair:** Criptografia não analisa conteúdo; use filtro de PII do Guardrails ou mascaramento antes do envio.

⚠ **Alerta de prova:** AWS WAF na frente da API bloqueia dados sensíveis.
**Como não cair:** WAF protege contra ataques web e não faz controle semântico de PII.

⚠ **Alerta de prova:** CloudTrail impede o agente de executar uma ação proibida.
**Como não cair:** CloudTrail registra depois do fato; bloqueio em tempo de chamada é Policy in AgentCore.

⚠ **Alerta de prova:** Grupos de segurança ou PrivateLink limitam quais modelos podem ser usados.
**Como não cair:** Rede isola tráfego; modelos aprovados se restringem com políticas IAM e SCPs.

⚠ **Alerta de prova:** Rodar modelos em EC2 ou usar modelos de código aberto garante que os dados não treinem modelos.
**Como não cair:** Não garante por si só e aumenta a carga operacional; o Bedrock já não usa os dados para treino.

⚠ **Alerta de prova:** Guardar logs de prompts indefinidamente é melhor para auditoria.
**Como não cair:** Aumenta a superfície de vazamento; logs seguem retenção limitada e criptografia.

⚠ **Alerta de prova:** Red teaming automático bloqueia prompts com PII.
**Como não cair:** Red teaming testa riscos e comportamento; mascarar PII é do filtro de informações sensíveis.

⚠ **Alerta de prova:** Instrução no prompt garante que o agente respeite o limite de estorno.
**Como não cair:** O modelo pode errar ou ser manipulado; use política determinística no AgentCore Gateway.

⚠ **Alerta de prova:** Uma única região global com criptografia atende GDPR e LGPD.
**Como não cair:** Residência exige processar e armazenar em regiões da jurisdição, com políticas e chaves por região.

⚠ **Alerta de prova:** Macie controla acesso, CloudTrail criptografa e IAM audita.
**Como não cair:** Papéis trocados: IAM controla acesso, CloudTrail audita e KMS criptografa; Macie descobre dados sensíveis.

⚠ **Alerta de prova:** Chaves gerenciadas pela AWS atendem exigência de chaves do cliente.
**Como não cair:** Só CMKs do KMS cumprem esse requisito.

⚠ **Alerta de prova:** Compartilhar credenciais root ou guardar credenciais no prompt simplifica com segurança.
**Como não cair:** Sempre errado: use roles IAM de menor privilégio e AgentCore Identity para acesso delegado.

## Como cai na prova

📝 **Como cai:** Cenário regulado com três ou quatro requisitos (acesso, auditoria, criptografia, residência).
**Como resolver:** Associe cada requisito a um controle: acesso = IAM ou SCP; auditoria = CloudTrail mais logs de invocação; chaves do cliente = CMK do KMS; rede privada = PrivateLink; residência = região da jurisdição. A certa cobre todos; elimine a que desliga logs, usa root, bucket público ou região única.

📝 **Como cai:** Garantia de que dados não treinam modelos.
**Como resolver:** Escolha o Bedrock (não usa prompts e respostas para treino nem compartilha com provedores) somado a controles do cliente. EC2, modelo aberto ou criptografia sozinhos não garantem.

📝 **Como cai:** Vazamento de PII em prompts ou respostas.
**Como resolver:** Mascare antes do envio (DLP ou filtro de informações sensíveis do Guardrails) e antes de gravar logs; HTTPS, KMS e WAF não analisam o conteúdo.

📝 **Como cai:** Agente com ações de risco.
**Como resolver:** Acesso em nome do usuário = AgentCore Identity; regra que não pode ser violada = Policy in AgentCore; auditoria = CloudTrail. Prompt e CloudTrail sozinhos não impedem a ação.

📝 **Como cai:** Classificar uma carga na Scoping Matrix.
**Como resolver:** App de consumo = 1; app corporativo de terceiros = 2; modelo pronto no seu app = 3; ajustado = 4; treinado do zero = 5.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| apenas modelos aprovados pelo comitê | **Políticas IAM liberando só esses modelos, mais SCPs do AWS Organizations** |
| quem chamou qual modelo e quando | **AWS CloudTrail** |
| guardar prompts e respostas para auditoria | **Logs de invocação do Bedrock, criptografados com KMS** |
| sem passar pela internet | **AWS PrivateLink e endpoints VPC** |
| mascarar CPF e cartão nas entradas e saídas | **Filtro de informações sensíveis do Bedrock Guardrails** |
| descobrir dados sensíveis em buckets | **Amazon Macie** |
| chaves gerenciadas pelo cliente | **CMKs do AWS KMS** |
| clientes europeus com dados na UE | **Regiões da UE com políticas e CMKs por região** |
| logs imutáveis e consultáveis | **CloudTrail com Object Lock, CloudTrail Lake e Athena** |
| só conjuntos de dados aprovados, controle por coluna e linha | **AWS Lake Formation com tags do Glue Data Catalog** |
| agente acessa agenda e CRM em nome do usuário, sem senhas no código | **Amazon Bedrock AgentCore Identity** |
| o agente nunca pode executar a ação acima de um limite | **Policy in AgentCore** |
| modelo pré-treinado usado num aplicativo próprio | **Escopo 3 da Generative AI Security Scoping Matrix** |
| linhagem de dados e modelos | **SageMaker ML Lineage Tracking** |

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

- O Bedrock não usa prompts e respostas para treinar modelos nem os repassa aos provedores.
- PII em prompts se resolve com mascaramento, não com criptografia ou WAF.
- Modelos aprovados = IAM e SCPs; rede não decide modelo.
- Regra que o agente não pode violar = Policy in AgentCore.

## Essenciais para a prova (checklist)

- ☐ Sei restringir o uso a modelos aprovados com IAM e SCPs.
- ☐ Sei a diferença entre CloudTrail e logs de invocação do Bedrock.
- ☐ Sei quando exigir CMKs do KMS em vez de chaves gerenciadas pela AWS.
- ☐ Sei usar PrivateLink e endpoints VPC para acesso privado.
- ☐ Sei mascarar PII com o filtro de informações sensíveis do Guardrails e com pré-processamento.
- ☐ Sei atender residência de dados por região.
- ☐ Sei classificar cargas nos cinco escopos da Generative AI Security Scoping Matrix.
- ☐ Sei para que servem Macie, Lake Formation, ML Lineage Tracking, AWS Config e AWS Artifact.
- ☐ Sei quando usar AgentCore Identity e Policy in AgentCore.
- ☐ Sei descartar WAF, Shield, GuardDuty e Inspector como solução de governança de IA.

## Questões de revisão do capítulo

R5.1 · Segurança, conformidade e governança para soluções de IA

**Uma equipe de ciência de dados está treinando modelos de machine learning usando dados de clientes armazenados na nuvem. Para reduzir o risco de exposição de dados sensíveis durante o treinamento e o acesso aos artefatos do modelo, qual prática é MAIS adequada do ponto de vista de segurança e governança na AWS?**

A. Desligar logs de auditoria para reduzir custos operacionais e simplificar o ambiente de desenvolvimento
B. Armazenar credenciais de banco de dados em variáveis de ambiente no código-fonte do notebook de treinamento para facilitar o acesso
C. Utilizar credenciais de usuário raiz da conta AWS para garantir maior controle direto sobre os recursos de treinamento
D. Habilitar criptografia em repouso e em trânsito para dados e artefatos de modelos, além de usar funções IAM com permissões mínimas necessárias

R5.2 · Segurança, conformidade e governança para soluções de IA

**Uma empresa financeira quer permitir que analistas internos façam perguntas em linguagem natural sobre dados sensíveis de clientes usando um modelo de linguagem hospedado na AWS. A empresa precisa garantir que nenhum dado pessoal identificável (PII) seja enviado ao provedor do modelo fora da sua conta AWS, mantendo o controle sobre logs e auditoria. Qual abordagem é a MAIS adequada para atender a esses requisitos de segurança e governança?**

A. Exportar periodicamente os dados sensíveis para arquivos criptografados e carregá-los à mão em uma ferramenta local de IA sem integração com a AWS
B. Usar um modelo fundacional inteiramente gerenciado por terceiros via API pública na internet, configurando VPN para acesso seguro
C. Habilitar acesso direto ao banco de dados de produção a partir do modelo de linguagem, confiando só na criptografia em repouso do Amazon RDS
D. Utilizar o Amazon Bedrock, que não usa prompts e respostas para treinar modelos nem os repassa aos provedores, e auditar as chamadas com o AWS CloudTrail

R5.3 · Segurança, conformidade e governança para soluções de IA

**Uma empresa está desenvolvendo um assistente virtual usando Amazon Bedrock e deseja reduzir o risco de o modelo gerar conteúdo que viole políticas internas (por exemplo, discurso de ódio, conteúdo sexual explícito ou incentivo à violência). Qual é a abordagem MAIS apropriada para atender a esse requisito de segurança e conformidade?**

A. Configurar classificadores de segurança e filtros de conteúdo usando Guardrails for Amazon Bedrock, definindo categorias de bloqueio e ações apropriadas.
B. Desligar completamente o log de prompts e respostas no Amazon Bedrock para evitar a exposição de conteúdo sensível.
C. Utilizar prompts muito genéricos e curtos, minimizando o risco de o usuário solicitar algo que viole as políticas.
D. Aumentar o tamanho do contexto (max tokens) do modelo para que ele tenha mais informações históricas e, assim, evite respostas inadequadas.

R5.4 · Segurança, conformidade e governança para soluções de IA

**Uma empresa está desenvolvendo uma solução de IA generativa com modelos hospedados no Amazon Bedrock para clientes de setores regulados (como saúde e financeiro). Para reduzir riscos de conformidade relacionados a vazamento de dados sensíveis dos usuários nas saídas do modelo, qual abordagem é a MAIS adequada?**

A. Confiar nos termos de uso do provedor de modelo, sem configuração adicional, pois os modelos já são treinados para evitar vazamento de dados
B. Desligar todo o monitoramento das respostas do modelo para garantir total privacidade dos usuários
C. Ativar logs detalhados das requisições e respostas do modelo e armazená-los indefinidamente para auditoria futura
D. Usar filtros de conteúdo e de PII nas saídas, políticas de uso de dados e revisão humana quando necessário

## Respostas comentadas

**R5.1. D) Habilitar criptografia em repouso e em trânsito para dados e artefatos de modelos, além de usar funções IAM com permissões mínimas necessárias**

Por quê: Esta alternativa está correta porque combina dois princípios fundamentais de segurança e governança na AWS: criptografia de dados (em repouso e em trânsito) e uso de funções IAM com privilégios mínimos. Isso reduz o risco de vazamento de dados sensíveis de clientes e garante que apenas serviços e usuários autorizados possam acessar dados e artefatos de modelos, alinhando-se às boas práticas de segurança e conformidade.

✗ A) Incorreta. Desativar logs de auditoria (como AWS CloudTrail) prejudica a rastreabilidade e a governança. Para soluções de IA com dados de clientes, a auditoria é essencial para conformidade e investigação de incidentes.

✗ B) Incorreta. Armazenar credenciais em código-fonte ou em variáveis de ambiente de forma não gerenciada aumenta significativamente o risco de vazamento.

✗ C) Incorreta. É uma prática insegura usar o usuário raiz para atividades do dia a dia. O usuário raiz deve ser reservado apenas para tarefas extremamente específicas de administração da conta, não para treinamento de modelos.

> **Dica de prova:** Lembre-se do trio de boas práticas em segurança para IA na AWS: (1) criptografia de dados em repouso e em trânsito, (2) princípio do privilégio mínimo com IAM, e (3) monitoramento e auditoria contínuos (CloudTrail, CloudWatch, etc.). Se a alternativa mencionar essas práticas em conjunto, é forte candidata a correta.

**R5.2. D) Utilizar o Amazon Bedrock, que não usa prompts e respostas para treinar modelos nem os repassa aos provedores, e auditar as chamadas com o AWS CloudTrail**

Por quê: No Amazon Bedrock, prompts e respostas não são usados para treinar os modelos de base e não são compartilhados com os provedores de modelo: os modelos rodam em contas de implantação operadas pela AWS, sem acesso dos provedores. Com o CloudTrail, as chamadas de API ficam registradas para auditoria; se for preciso guardar o conteúdo dos prompts, ative o registro de invocação com controle de acesso e criptografia.

✗ A) Incorreta. Esse processo manual não oferece boa governança, rastreabilidade nem integração com mecanismos nativos de auditoria da AWS.

✗ B) Incorreta. Mesmo com VPN, usar uma API pública de terceiros fora da AWS pode violar requisitos de governança e privacidade, já que os dados sensíveis podem ser processados e armazenados fora do controle da empresa.

✗ C) Incorreta. Expor o banco de dados de produção diretamente ao modelo, mesmo com criptografia em repouso, amplia a superfície de ataque e não implementa controles explícitos de minimização, mascaramento ou segregação de dados sensíveis,...

> **Dica de prova:** Dados sensíveis + modelo gerenciado na AWS: o Bedrock não treina modelos com seus prompts nem os envia ao provedor. Auditoria de chamadas = CloudTrail.

**R5.3. A) Configurar classificadores de segurança e filtros de conteúdo usando Guardrails for Amazon Bedrock, definindo categorias de bloqueio e ações apropriadas.**

Por quê: Esta alternativa é a mais adequada porque o Guardrails for Amazon Bedrock foi projetado especificamente para aplicar políticas de segurança e conformidade em aplicações de IA generativa. Com ele, é possível configurar filtros para categorias como discurso de ódio, violência, conteúdo sexual, além de definir ações de bloqueio, substituição ou moderação adicional.

✗ B) Incorreta. Desabilitar logs pode reduzir o risco de exposição de dados, mas não reduz o risco de o modelo gerar conteúdo que viole políticas.

✗ C) Incorreta. Prompts curtos ou genéricos não garantem que usuários finais não tentem induzir o modelo a gerar conteúdo proibido.

✗ D) Incorreta. Aumentar o tamanho do contexto não tem relação direta com a prevenção de conteúdo tóxico ou inadequado. Isso apenas permite que o modelo considere mais texto, o que pode até ampliar a superfície de risco se não houver controles...

> **Dica de prova:** Para questões de segurança em IA generativa na AWS, procure termos como Guardrails for Amazon Bedrock, filtros de segurança, moderação de conteúdo e políticas de uso aceitável.

**R5.4. D) Usar filtros de conteúdo e de PII nas saídas, políticas de uso de dados e revisão humana quando necessário**

Por quê: Esta alternativa é a mais adequada porque combina mecanismos técnicos (filtros de conteúdo e PII) com controles de processo (políticas de uso de dados e revisão humana). No Amazon Bedrock, é possível aplicar camadas de segurança e governança, como filtros para dados sensíveis, evitando que o modelo exponha PII ou informações reguladas, o que ajuda na conformidade com normas como LGPD, HIPAA ou normas financeiras.

✗ A) Incorreta. Confiar apenas nos termos de uso do provedor é insuficiente. Reguladores exigem controles técnicos e organizacionais específicos; a responsabilidade pela conformidade é do cliente que opera a solução de IA.

✗ B) Incorreta. Desativar monitoramento elimina a capacidade de detectar vazamentos, abusos e violações de política. Boas práticas de governança exigem observabilidade e auditoria, com proteção adequada aos dados monitorados.

✗ C) Incorreta. Armazenar logs indefinidamente aumenta a superfície de risco de vazamento de dados sensíveis. Logs devem seguir políticas de retenção mínima necessária, criptografia e controle de acesso, não retenção ilimitada.

**Dica de prova:** Soluções de IA em ambientes regulados devem sempre combinar: (1) controles técnicos (filtros de PII, criptografia, controle de acesso), (2) políticas claras de uso de dados, e (3) monitoramento e auditoria com retenção adequada. Pense em "camadas de proteção" em vez de uma única medida.

DIA DA PROVA

# Estratégia de prova

A AIF-C01 é uma prova de cenários de negócio: descreve uma empresa, um objetivo e restrições, e pede a abordagem, o serviço ou a prática MAIS adequada. Várias alternativas são tecnicamente possíveis, mas só uma atende todos os requisitos com serviço gerenciado e menor esforço. Os distratores se repetem: treinar um modelo do zero, desligar logs, bucket público, confiar no provedor, aumentar o modelo ou mexer na temperatura para resolver o problema errado. A versão 1.1 do guia trouxe IA agêntica, engenharia de contexto, custo por token, métricas de negócio e serviços novos como AgentCore, Strands Agents, Kiro e Amazon Quick.

## Como responder

✓ Leia primeiro a última frase do enunciado para saber o que é pedido (serviço, técnica, métrica ou prática) e depois sublinhe as restrições: sem treinar, sem gerenciar infraestrutura, menor custo, dados não podem sair, resposta em tempo real.

✓ São 65 questões em 90 minutos, das quais 50 pontuadas e 15 não pontuadas: trate todas com o mesmo cuidado e controle o tempo para chegar ao fim da prova.

✓ A nota vai de 100 a 1000 e o mínimo é 700: não trave numa questão difícil; elimine as alternativas erradas, fique com a melhor restante e siga.

✓ Elimine primeiro os distratores fixos: treinar do zero, desligar logs, bucket público, credencial root, confiar só no provedor ou só nos termos de uso.

✓ Desconfie de absolutos como 'sempre', 'só', '100%', 'elimina completamente' e 'garante'; em IA quase nada é garantido.

✓ Em múltipla resposta, leia quantas alternativas são pedidas e avalie cada uma de forma independente; em ordenação, lembre do ciclo de vida (dados e modelo, pré-treino, ajuste fino, avaliação, implantação, feedback).

✓ Ache o verbo do requisito e mapeie para o serviço: extrair documento, transcrever, mascarar PII, executar ação, lembrar preferências, documentar modelo, explicar decisão.

✓ Dê atenção especial a Aplicações de modelos de base (28%) e Fundamentos de IA generativa (24%), que somam mais da metade da prova.

## Erros que derrubam candidatos

✗ Escolher ajuste fino quando os dados mudam com frequência e o certo é RAG.

✗ Achar que temperatura, janela maior ou modelo maior resolvem alucinação, custo ou viés.

✗ Confiar na acurácia com classe rara e confundir precisão com recall.

✗ Escolher criptografia, HTTPS ou WAF para resolver PII dentro do prompt.

✗ Confundir CloudTrail (quem chamou) com logs de invocação (o que foi dito) e achar que CloudTrail bloqueia ações.

✗ Escolher agente autônomo ou multiagente para processo de etapas fixas.

✗ Trocar serviços de caminho inverso, como Polly e Transcribe, ou Q Business e Q Developer.

## Na semana da prova

- ☐ Faça simulados completos cronometrados, 65 questões em 90 minutos, e revise cada erro pelo motivo da alternativa errada.
- ☐ Revise as tabelas que mais derrubam: métricas (precisão, recall, ROUGE, BLEU), formas de inferência (SageMaker AI e Bedrock) e técnicas de personalização.
- ☐ Revise os serviços novos da v1.1: AgentCore (Memory, Identity, Policy), Strands Agents, Kiro, Amazon Quick, AWS Transform, SageMaker JumpStart e MCP.
- ☐ Releia os gatilhos e pegadinhas de Segurança e de IA responsável na véspera: são domínios menores, mas de pontos fáceis.
- ☐ Na véspera, nada de conteúdo novo: revise os itens de 'lembrar' de cada domínio e durma bem.

REVISÃO FINAL

# Folha de cola

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

## 1. Aplicações de modelos de base

‣ documentos internos atualizados, responder com citação das fontes → **RAG com Amazon Bedrock Knowledge Bases**

‣ milhões de documentos não rotulados, vocabulário do domínio → **Pré-treino contínuo (adaptação de domínio)**

‣ milhares de pares rotulados, tom da marca e formato padronizado → **Ajuste fino supervisionado**

‣ avaliadores comparam pares de respostas e indicam a melhor → **RLHF**

‣ qualidade do modelo grande com custo e latência de modelo pequeno → **Destilação de modelo**

⚠ É caro, demorado e desnecessário na imensa maioria dos cenários; prefira prompt, RAG ou ajuste fino sobre um modelo de base.

⚠ Documentos que mudam pedem RAG; ajuste fino a cada atualização é caro, lento e não cita fontes.

⚠ O limite de saída não aumenta o que cabe na entrada; divida em trechos e recupere os relevantes.

## 2. Diretrizes de IA responsável

‣ regulador quer saber quais atributos pesaram nesta decisão de crédito → **SageMaker Clarify com SHAP**

‣ viés e deriva em produção, com alarmes → **Model Monitor com Clarify e CloudWatch**

‣ documentar propósito, dados, limitações e versões do modelo → **SageMaker Model Cards**

‣ decisões críticas validadas por especialista → **Revisão humana com Amazon A2I**

‣ bloquear linguagem ofensiva e restringir temas sensíveis → **Bedrock Guardrails: filtros de conteúdo e tópicos negados**

⚠ Tamanho não é ética: modelos maiores podem amplificar vieses e gerar saídas mais convincentes, porém incorretas.

⚠ Sem curadoria, testes e filtros, o tamanho não resolve viés nem toxicidade.

⚠ Desligar logs elimina auditoria e investigação; proteja os logs com anonimização, criptografia e controle de acesso.

## 3. Fundamentos de IA e ML

‣ sem rótulos, agrupar clientes com comportamento semelhante → **Aprendizado não supervisionado (clustering)**

‣ agente, recompensa, penalidade, tentativa e erro → **Aprendizado por reforço**

‣ prever preço, minutos ou demanda a partir de histórico com valores reais → **Regressão supervisionada, avaliada por RMSE**

‣ classe minoritária, fraudes ou cancelamentos que passam despercebidos → **Recall (e F1 da classe minoritária)**

‣ muito grave marcar um legítimo como positivo → **Precisão**

⚠ Um modelo que sempre prevê a classe majoritária já tem acurácia alta; olhe recall, F1 da classe minoritária e curva precisão-recall.

⚠ Com forte desbalanceamento a curva ROC tende a ser otimista; a curva precisão-recall e o F1 da classe minoritária mostram o problema.

⚠ Leia qual erro é grave: se marcar um legítimo como positivo é pior (antispam), priorize precisão; recall alto tende a aumentar falsos positivos.

## 4. Fundamentos de IA generativa

‣ prever a próxima palavra ou token em grandes volumes de texto → **LLM, pré-treino autossupervisionado**

‣ vetor numérico e documentos mais próximos da pergunta → **Embeddings e busca semântica**

‣ mesmo prompt, respostas diferentes → **Não determinismo da amostragem (temperatura)**

‣ resposta plausível, confiante e inventada → **Alucinação: RAG com citações e fundamentação contextual**

‣ difícil explicar como o LLM chegou à resposta → **Baixa interpretabilidade**

⚠ Janela maior permite mais tokens, o que tende a aumentar custo e latência, e não elimina alucinações.

⚠ Reduzir o tamanho não elimina alucinações; o que ajuda é fundamentar em fontes confiáveis (RAG com citação) e verificar a fundamentação.

⚠ Sobreajuste é decorar o treino; alucinação é inventar fatos plausíveis na geração.

## 5. Segurança, conformidade e governança para soluções de IA

‣ apenas modelos aprovados pelo comitê → **Políticas IAM liberando só esses modelos, mais SCPs do AWS Organizations**

‣ quem chamou qual modelo e quando → **AWS CloudTrail**

‣ guardar prompts e respostas para auditoria → **Logs de invocação do Bedrock, criptografados com KMS**

‣ sem passar pela internet → **AWS PrivateLink e endpoints VPC**

‣ mascarar CPF e cartão nas entradas e saídas → **Filtro de informações sensíveis do Bedrock Guardrails**

⚠ Elimina a trilha de auditoria; mantenha os logs criptografados, com acesso restrito e retenção limitada.

⚠ Protege só o transporte; é preciso mascarar PII, controlar acesso e garantir que os dados não treinem modelos.

⚠ Criptografia não analisa conteúdo; use filtro de PII do Guardrails ou mascaramento antes do envio.

GABARITO

# Respostas do diagnóstico

**D1. B) Taxa de conclusão de tarefas sem transferência para um atendente** → revise o capítulo 1

Por quê: O guia v1.1 pede identificar métricas de alinhamento ao objetivo de negócio para aplicações de IA, como taxa de conclusão de tarefas, satisfação do usuário e custo por interação, e avaliar aplicações completas (RAG, agentes, fluxos), e não só o modelo.

✗ A) Incorreta. O BLEU mede semelhança textual com referências, não se a consulta foi remarcada.

✗ C) Incorreta. O tamanho das respostas afeta o custo, mas não indica se o agente cumpriu a tarefa.

✗ D) Incorreta. Perplexidade é uma métrica técnica de modelagem de linguagem e não reflete resultado de negócio.

> **Dica de prova:** Valor de negócio de um agente = tarefas concluídas, satisfação e custo por interação. BLEU e ROUGE medem texto.

**D2. B) Adotar modelo do Bedrock só para inferência, com os e-mails enviados sob controles de privacidade e criptografia** → revise o capítulo 1

Por quê: Esta alternativa é a mais apropriada porque o Amazon Bedrock oferece acesso a modelos de base em modo somente inferência, sem usar os dados do cliente para treinar modelos públicos. Além disso, a empresa pode aplicar criptografia em trânsito (HTTPS) e em repouso (usando KMS) e integrar com controles de segurança da AWS, atendendo a requisitos de privacidade e compliance enquanto se beneficia de IA generativa gerenciada.

✗ A) Incorreta. Colocar e-mails em um bucket S3 público é um grave problema de segurança, expondo dados sensíveis. Executar modelos localmente em computadores pessoais também dificulta o controle de acesso, auditoria e conformidade.

✗ C) Incorreta. Treinar um modelo do zero é caro, complexo e desnecessário para gerar rascunhos de e-mail. Também não resolve, por si só, o problema de privacidade e governança de dados, além de ignorar as vantagens de serviços gerenciados.

✗ D) Incorreta. Permitir que um provedor externo use os dados para treinar modelos públicos contraria diretamente a exigência de não usar dados sensíveis de clientes para esse fim, aumentando o risco de exposição e problemas regulatórios.

> **Dica de prova:** Quando o enunciado mencionar privacidade, dados sensíveis e necessidade de evitar treino em modelos públicos, pense em serviços gerenciados que ofereçam 'somente inferência', isolamento de dados e integração nativa com criptografia e controle de acesso (como Amazon Bedrock e IAM).

**D3. C) No prompt de sistema (system prompt) enviado em cada chamada** → revise o capítulo 1

Por quê: O prompt de sistema define persona, tom e restrições que valem para toda a conversa e acompanha cada chamada, por exemplo no campo system da API Converse. Temperatura e limite de tokens não carregam instruções, e uma base de conhecimento fornece conteúdo, não regras de comportamento.

✗ A) Incorreta. Fornece conteúdo sobre concorrentes, o oposto do desejado.

✗ B) Incorreta. Só limita o tamanho da resposta.

✗ D) Incorreta. A temperatura controla aleatoriedade, não comportamento.

> **Dica de prova:** Persona, tom e regras fixas do assistente: prompt de sistema.

**D4. C) Adotar diretrizes com revisão humana de saídas críticas, proteção de dados, transparência e canal de reporte** → revise o capítulo 2

Por quê: Esta alternativa descreve vários pilares de IA responsável: supervisão humana (revisão de resultados críticos), proteção de dados (cuidados com informações sensíveis), transparência (deixar claro quando a IA é usada) e governança (canais formais para reportar problemas). Esses elementos ajudam a reduzir riscos de vieses, violações de privacidade e uso inadequado dos modelos, alinhando a prática ao conjunto de melhores práticas de IA responsável.

✗ A) Incorreta. Transferir toda a responsabilidade para o funcionário, sem controles, não segue princípios de governança. Falta supervisão, controles de segurança, diretrizes de uso e mecanismos de transparência, criando exposição jurídica e...

✗ B) Incorreta. Restringir o uso à equipe de TI sem registrar dados de entrada e uso ignora requisitos de rastreabilidade, auditoria e prestação de contas, que são fundamentais para práticas responsáveis de IA.

✗ D) Incorreta. Adotar uma proibição total não caracteriza IA responsável; é uma medida extrema que impede benefícios de inovação em vez de gerir riscos com políticas, monitoramento e controles adequados.

> **Dica de prova:** Quando a questão falar em 'IA responsável', procure por elementos como: supervisão humana, transparência, privacidade e segurança de dados, mitigação de vieses, governança (políticas, auditoria, registro) e mecanismos de contestação ou reporte de problemas.

**D5. D) Adotar moderação de conteúdo e filtros de segurança, com revisão de prompts e respostas para achar vieses** → revise o capítulo 2

Por quê: Esta alternativa é a mais alinhada com diretrizes de IA responsável porque combina mecanismos técnicos (moderação de conteúdo e filtros de segurança) com governança prática (revisão de prompts e respostas para identificar vieses). Essa abordagem reconhece que o provedor oferece salvaguardas, mas que a organização continua responsável por supervisionar, testar e ajustar o uso do modelo no seu contexto específico.

✗ A) Incorreta. Aumentar parâmetros pode melhorar capacidade do modelo, mas não garante redução de viés. Sem testes, filtros e políticas, o modelo pode continuar ou até amplificar vieses presentes nos dados de treinamento.

✗ B) Incorreta. Confiar somente nas configurações padrão ignora o princípio de responsabilidade compartilhada. Mesmo com salvaguardas do provedor, a organização precisa configurar, testar e monitorar o uso do modelo de acordo com o seu domínio...

✗ C) Incorreta. Remover referências sensíveis dos registros de auditoria dificulta identificar e medir vieses. Para IA responsável, é importante ter métricas e logs que permitam detectar se certos grupos estão sendo tratados de forma desigual.

> **Dica de prova:** Em questões de IA responsável, procure opções que falem de: testes contínuos, monitoramento, mitigação de viés, transparência, revisão humana e uso de mecanismos de segurança (filtros, moderação, políticas). Desconfie de respostas que insinuam que a tecnologia sozinha "resolve tudo" sem supervisão ou governança.

**D6. D) Testar com dados diversos e usar guardrails para monitorar e bloquear respostas inadequadas** → revise o capítulo 2

Por quê: Esta alternativa está correta porque combina duas práticas essenciais de IA responsável: testes com dados diversos (para identificar vieses e comportamentos problemáticos em diferentes grupos e contextos) e o uso de filtros de segurança/guardrails (para bloquear ou mitigar, em tempo de execução, respostas ofensivas, discriminatórias ou que violem políticas de uso). Essa abordagem é proativa, mensurável e alinhada a princípios de justiça, segurança e responsabilidade.

✗ A) Incorreta. Delega toda a responsabilidade ao provedor de nuvem, ignorando a necessidade de governança interna, monitoramento contínuo e avaliação de risco específica do contexto de uso da organização.

✗ B) Incorreta. Reduzir o tamanho do modelo, por si só, não resolve vieses ou conteúdo ofensivo; sem curadoria de dados, testes e controles de segurança, o problema pode persistir.

✗ C) Incorreta. Embora o feedback dos usuários seja importante, não registrar incidentes impede análise de causa raiz, melhoria contínua e rastreabilidade, contrariando boas práticas de governança e accountability em IA responsável.

> **Dica de prova:** Sempre que a questão falar em reduzir riscos de vieses, discriminação ou conteúdo nocivo em IA, procure por palavras-chave como 'testes com dados diversos', 'monitoramento contínuo', 'guardrails/filtros de segurança', 'governança' e 'registro de incidentes'.

**D7. B) Amazon Textract, com análise de formulários e tabelas** → revise o capítulo 3

Por quê: Para documentos como notas fiscais, formulários e extratos, o Amazon Textract vai além do OCR simples: entende formulários (chave e valor) e tabelas e entrega dados estruturados para sistemas como ERPs.

✗ A) Incorreta. O Polly converte texto em fala; não extrai dados de documentos.

✗ C) Incorreta. O Rekognition identifica objetos, cenas, rostos e conteúdo impróprio em imagens; não estrutura campos de documentos.

✗ D) Incorreta. O Comprehend analisa texto já extraído (sentimento, entidades); sozinho, não lê a imagem digitalizada nem as tabelas.

> **Dica de prova:** Extrair campos e tabelas de documentos digitalizados = Textract.

**D8. B) Um agente que interage com um ambiente, toma ações sequenciais, recebe recompensas ou penalidades e ajusta sua política para maximizar a recompensa acumulada ao longo do tempo.** → revise o capítulo 3

Por quê: Esta alternativa descreve corretamente aprendizado por reforço: existe um agente que interage com um ambiente, escolhe ações, observa estados e recebe recompensas ou penalidades. O objetivo é aprender uma política que maximize a recompensa acumulada ao longo do tempo. Diferente do aprendizado supervisionado, não há um rótulo 'correto' imediato para cada estado-ação, e diferente do não supervisionado, há um sinal explícito de desempenho (recompensa) guiando o aprendizado sequencial.

✗ A) Incorreta. Este é um exemplo clássico de aprendizado supervisionado, pois há dados rotulados (gato/cão) e o objetivo é aprender um mapeamento entrada->rótulo.

✗ C) Incorreta. Este é um exemplo típico de aprendizado não supervisionado, usando clustering para encontrar grupos em dados sem rótulos pré-definidos.

✗ D) Incorreta. Embora envolva sinais de erro (por exemplo, predição da próxima palavra), este cenário se enquadra principalmente em aprendizado supervisionado/sequencial (modelagem de linguagem), não em reforço, pois não há agente tomando...

> **Dica de prova:** Associe mentalmente: 'supervisionado = rótulos explícitos', 'não supervisionado = descobrir estrutura sem rótulos', 'reforço = agente + ambiente + ações + recompensas ao longo do tempo'. Sempre que aparecer 'política', 'episódios' ou 'recompensa acumulada', pense em aprendizado por reforço.

**D9. C) Ajustar hiperparâmetros e comparar versões com dados separados do treino** → revise o capítulo 3

Por quê: A etapa de validação é usada para ajustar hiperparâmetros (por exemplo, taxa de aprendizado, profundidade de árvore, número de camadas, etc.) e comparar diferentes versões de modelos. Para isso, utiliza-se um conjunto de dados separado do treino, permitindo avaliar qual configuração funciona melhor sem ainda olhar para o conjunto de teste final.

✗ A) Incorreta. Descreve um processo de engenharia de dados / coleta de dados, que pode ocorrer antes ou durante o ciclo de desenvolvimento, mas não é o objetivo da etapa de validação em si.

✗ B) Incorreta. Refere-se diretamente à fase de treino, na qual o modelo ajusta seus parâmetros para minimizar a função de perda, e não à validação.

✗ D) Incorreta. Descreve o objetivo do conjunto de teste, que é estimar o desempenho final do modelo em dados nunca vistos, não a etapa de validação.

> **Dica de prova:** Lembre-se da tríade clássica: treino (aprender parâmetros), validação (ajustar hiperparâmetros e escolher o melhor modelo) e teste (avaliar desempenho final). Se a pergunta falar em "dados nunca vistos para estimar o desempenho em produção", pense em conjunto de teste;

**D10. A) Kiro, com desenvolvimento orientado por especificações** → revise o capítulo 4

Por quê: O guia v1.1 incluiu o Kiro entre as tecnologias para desenvolver aplicações de IA generativa e entre os serviços por etapa do ciclo de IA/ML. Seu diferencial é o desenvolvimento orientado por especificações, que dá estrutura e rastreabilidade ao código gerado por agentes.

✗ B) Incorreta. O Step Functions orquestra fluxos de trabalho em máquinas de estado; não é um ambiente de desenvolvimento com agentes.

✗ C) Incorreta. O Prompt Management versiona e testa prompts; não transforma pedidos em requisitos, design e tarefas.

✗ D) Incorreta. O SageMaker Canvas cria modelos de ML sem código para analistas; não gera especificações de software.

> **Dica de prova:** Agentes de código + specs (requisitos, design e tarefas) = Kiro.

**D11. B) Reduzir o tamanho dos trechos (chunks), com alguma sobreposição entre eles** → revise o capítulo 4

Por quê: Chunking é dividir documentos em trechos antes de gerar embeddings. Trechos grandes demais diluem a relevância e encarecem o prompt; pequenos demais perdem contexto. As bases de conhecimento do Bedrock oferecem chunking padrão (tamanho em tokens com porcentagem de sobreposição), hierárquico e semântico.

✗ A) Incorreta. A temperatura afeta a aleatoriedade da geração; não melhora a precisão da recuperação.

✗ C) Incorreta. Enviar o manual inteiro aumenta ainda mais o custo e pode exceder a janela de contexto.

✗ D) Incorreta. Um modelo de imagens não gera embeddings de texto para a busca.

> **Dica de prova:** RAG trazendo muito texto irrelevante = revise o chunking.

**D12. D) Amazon Nova Micro, modelo só de texto e de menor latência** → revise o capítulo 4

Por quê: Critérios de seleção de modelos no guia: custo, modalidade, latência, suporte a idiomas, tamanho e complexidade do modelo, personalização e tamanho de entrada e saída. Para tarefas simples de texto em grande volume, um modelo menor e só de texto costuma equilibrar melhor custo e latência; valide com testes no seu caso de uso.

✗ A) Incorreta. O Nova Pro atende tarefas multimodais mais complexas, com custo e latência maiores do que o necessário aqui.

✗ B) Incorreta. O Nova Reel gera vídeos; não se aplica à classificação de mensagens.

✗ C) Incorreta. O Nova Canvas gera imagens; não é indicado para classificar texto.

> **Dica de prova:** Tarefa simples + muito volume + só texto = modelo pequeno e rápido. Não pague por capacidade que não usa.

## Agora, treine no formato real

Faça as 6 provas completas do curso AIF-C01: 390 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 Amazon Web Services (AWS). As questões são inéditas e não reproduzem a prova real.