PREPARAÇÃO PARA A CERTIFICAÇÃO · AIF-C01

AWS Certified AI Practitioner

Guia de Estudos Completo · 5 domínios da prova

Renata R Silva (Maki) @renatarsilvax é engenheira de software, criadora de conteúdo tech e 2x certificada AWS (Cloud Practitioner e AI Practitioner). Apaixonada por comunidades, ela compartilha ativamente sua rotina de estudos de engenharia de software e cloud computing nas redes sociais.

Introdução

Este guia segue o **guia oficial da prova AWS Certified AI Practitioner (AIF-C01)**, domínio por domínio, com tabelas de contraste, palavras-gatilho, armadilhas e um resumo rápido ao final de cada domínio. O conteúdo foi montado a partir de páginas oficiais da AWS (lista completa na seção **Fontes**, ao final).

Domínios da prova

Domínio	Peso
1. Fundamentos de IA e ML	20%
2. Fundamentos de IA Generativa	24%
3. Aplicações de Modelos de Fundação	28%
4. Diretrizes para IA Responsável	14%
5. Segurança, Conformidade e Governança para Soluções de IA	14%

Sobre a prova (guia oficial)

- 65 questões: **50 pontuadas** e 15 não pontuadas (não identificadas).
- Tipos: múltipla escolha (1 correta, 3 incorretas) e múltipla resposta (2 ou mais corretas entre 5 ou mais opções).
- Questões sem resposta contam como erradas; **não há penalidade por chute**.
- Resultado: nota escalonada de 100 a 1.000; **nota mínima 700**. Modelo compensatório: você não precisa passar em cada seção, só na prova toda.
- Candidato-alvo: até 6 meses de exposição a IA/ML na AWS; usa, mas não necessariamente constrói soluções de IA/ML.
- Fora do escopo: desenvolver ou programar modelos, engenharia de dados e de features, ajuste de hiperparâmetros, construir pipelines, análise matemática ou estatística de modelos, implementar protocolos de segurança ou frameworks de governança.

Domínio 1: Fundamentos de IA e ML (20%)

Tarefas oficiais: 1.1 Explicar conceitos e terminologias básicas de IA · 1.2 Identificar casos de uso práticos de IA · 1.3 Descrever o ciclo de vida de desenvolvimento de IA/ML.

1.1 Conceitos e terminologias básicas de IA

A hierarquia: IA, ML, deep learning, IA generativa

Pense em círculos aninhados: **IA ⊃ ML ⊃ deep learning ⊃ IA generativa**. São camadas, não sinônimos. Segundo a AWS, a **IA** permite que máquinas executem tarefas de resolução de problemas semelhantes às humanas; o **machine learning** é um ramo da IA que desenvolve algoritmos e modelos estatísticos para encontrar padrões em dados históricos; o **deep learning** usa redes neurais com muitas camadas; a **IA generativa** cria conteúdo novo (texto, imagem, vídeo, áudio) a partir de prompts.

Termo	O que é	Palavra-gatilho
IA (AI)	Guarda-chuva: máquinas realizando tarefas que normalmente exigem inteligência humana.	"simular inteligência humana"
ML	Sistemas que **aprendem padrões a partir de dados** em vez de seguir regras escritas à mão.	"aprende com dados históricos"
Deep learning	ML com **redes neurais** de várias camadas; forte em dados não estruturados (imagem, texto, áudio).	"redes neurais profundas"
Rede neural	Modelo inspirado no processamento do cérebro humano, base das tecnologias modernas de IA.	"camadas de neurônios"
Computer vision	Extrai informação de imagens e vídeos (detecção de objetos, rostos, moderação de conteúdo).	"analisar imagens/vídeos"
NLP	Processamento de linguagem natural: interpretar, entender e gerar linguagem humana.	"sentimento, entidades, tradução"
Modelo	O artefato treinado que recebe uma entrada e produz uma saída (predição, classe, texto).	"artefato treinado"
Algoritmo	O procedimento matemático usado *para treinar* o modelo.	"método de aprendizado"
Treinamento x inferência	Treinar = o modelo aprende dos dados. **Inferência** = usar o modelo treinado para gerar predições com dados novos.	"fase de uso em produção"
LLM	Large Language Model: modelo de linguagem de grande escala, tipo de foundation model para texto.	"gera e entende texto"
IA generativa	Cria conteúdo novo. Saída é um artefato (parágrafo, imagem, código).	"criar, redigir, gerar"
IA agêntica (agentic AI)	Sistema autônomo que age de forma independente para atingir objetivos predefinidos, com raciocínio (LLM), memória, ferramentas e dados externos.	"agir sozinho", "executar tarefas em várias etapas"

Semelhanças e diferenças entre IA, ML, GenAI, deep learning e IA agêntica

Conceito	Foco principal	Saída típica	Exemplo
IA	Simular capacidades humanas	Decisões, ações, conteúdo	Assistente de voz
ML	Aprender padrões de dados	Rótulo ou número (classe, preço)	Detecção de fraude
Deep learning	ML com redes neurais profundas	Predição em dados complexos	Reconhecer objetos em fotos
GenAI	Criar conteúdo novo	Texto, imagem, áudio, vídeo, código	Resumir um documento
IA agêntica	Perceber, raciocinar, **agir** e aprender com autonomia	Tarefas concluídas (ações)	Agente que consulta sistemas e resolve um chamado

> **Regra de bolso:** se o sistema produz um **novo artefato** (parágrafo, imagem, resumo), é IA generativa. Se produz um **rótulo ou um número** (fraude/não fraude, preço previsto), é ML tradicional, mesmo que haja uma rede neural por baixo. Se ele **decide e executa ações** em vários passos com ferramentas, é IA agêntica.

Viés, justiça e ajuste (fit)

- **Viés (bias)** nos dados ou no modelo: o modelo tem desempenho desigual entre grupos, reproduzindo ou ampliando preconceitos presentes nos dados históricos.
- **Justiça (fairness)**: tratamento equitativo entre grupos; é o objetivo oposto ao viés prejudicial.
- Atenção: "bias" também aparece no sentido estatístico (alto viés = suposições rígidas demais = **underfitting**). O contexto da pergunta diz qual dos dois sentidos vale.

Ajuste	O que acontece	Sintoma	Como corrigir
Overfitting (sobreajuste)	O modelo memoriza detalhes e ruído dos dados de treino.	Ótimo no treino, **ruim em dados novos** (alta variância)	Mais dados, simplificar o modelo, regularização, parada antecipada, dropout
Underfitting (subajuste)	O modelo é simples demais para capturar o padrão (alto viés).	**Ruim no treino e no teste**	Modelo mais complexo, melhores atributos (features), treinar por mais tempo
Bom ajuste	Captura o padrão e generaliza.	Desempenho parecido em treino e teste	Meta: equilibrar viés e variância

> **Armadilha:** treino excelente + teste ruim = overfitting. Treino ruim + teste ruim = underfitting. A AWS descreve overfitting como aprender os detalhes e o ruído dos dados de treino.

Tipos de inferência

Tipo	Quando usar	Gatilho	Opção no Amazon SageMaker AI
Em tempo real (real-time)	Cargas interativas com requisito de baixa latência; endpoint persistente.	"resposta imediata", "baixa latência"	Real-time inference (endpoint)
Em lote (batch)	Grande volume de dados offline, sem urgência e sem endpoint persistente.	"processar o dataset inteiro de uma vez"	Batch transform
Assíncrona	Payloads grandes (até 1 GB) e processamentos longos (até 1 hora); as requisições entram em fila.	"arquivos grandes", "pode demorar"	Asynchronous inference
Serverless	Sem gerenciar infraestrutura; ideal para períodos ociosos entre picos de tráfego e que toleram *cold start*.	"tráfego intermitente/imprevisível"	Serverless Inference

Armadilha: serverless não é "mais rápido": o preço é o cold start. Assíncrona não é lote: ela enfileira requisições individuais e responde quando termina; lote processa um conjunto de dados inteiro.

Tipos de dados em modelos de IA

Tipo de dado	Descrição	Exemplo
Rotulados (labeled)	Cada exemplo tem a resposta correta conhecida (ground truth). Necessário para aprendizado supervisionado.	Transações marcadas como fraude ou legítimas
Não rotulados (unlabeled)	Dados brutos sem resposta. Usados em aprendizado não supervisionado.	Milhões de textos soltos
Estruturados	Linhas e colunas com esquema fixo.	Tabela de vendas
Não estruturados	Sem esquema fixo; terreno do deep learning e da GenAI.	Texto livre, imagem, áudio, vídeo
Tabulares	Dados estruturados em tabela.	Planilha de clientes
Séries temporais	Valores ordenados no tempo.	Vendas diárias, leituras de sensores
Imagem / texto	Entradas de visão computacional e NLP.	Fotos de produtos, avaliações de clientes

Conjuntos de dados: treino (o modelo aprende), **validação** (ajuste de hiperparâmetros e escolha do modelo) e **teste** (separado, usado no fim para estimar o desempenho real). Misturar teste com treino causa **vazamento de dados (data leakage)**.

Tipos de aprendizado

Tipo	Como aprende	Exemplos de uso	Palavra-gatilho
Supervisionado	Com dados **rotulados** (entrada → saída correta conhecida).	Detecção de fraude, previsão de preço, classificação de spam	"exemplos históricos rotulados"
Não supervisionado	Encontra estrutura em dados **sem rótulos**.	Segmentação de clientes (agrupamento), detecção de anomalias	"sem rótulos", "agrupar itens semelhantes"
Semissupervisionado	Poucos dados rotulados + muitos não rotulados.	Rotular é caro, mas há alguns exemplos	"poucos dados rotulados"
Por reforço (RL)	Um agente age em um ambiente e recebe **recompensas** por passos corretos.	Jogos, robótica, alinhamento de LLMs com preferência humana (RLHF)	"agente", "recompensa", "tentativa e erro"
Autossupervisionado	O modelo gera seus próprios rótulos a partir do dado (ex.: prever o próximo token).	Pré-treinamento de LLMs/FMs com texto bruto	"sem rótulos humanos" em FMs

> **Armadilha:** "dados não rotulados" pode ser não supervisionado *ou* autossupervisionado (pré-treino de FMs). Se a pergunta fala em agente, recompensa e política, é reforço, nunca supervisionado.

1.2 Casos de uso práticos de IA

Quando a IA/ML agrega valor

- **Apoiar a decisão humana**: recomendações, previsões, triagem.
- **Escalabilidade**: tratar volumes que seriam inviáveis manualmente.
- **Automação** de tarefas repetitivas (transcrição, moderação de imagens, classificação de documentos).
- O padrão é complexo demais para regras escritas à mão, existem dados representativos e algum erro é tolerável.

Quando a IA/ML não é apropriada

- Quando é preciso um **resultado específico e determinístico**, não uma predição (ex.: aplicar 10% de desconto se o pedido passar de um valor fixo: basta uma regra).
- Quando a **análise custo-benefício** não compensa (custo de dados, treino, operação maior que o ganho).
- Quando não há dados suficientes ou representativos.
- Quando é exigida lógica 100% explicável e auditável (ex.: cálculo regulatório).

> **Armadilha:** em perguntas com regra fixa e simples, a resposta certa costuma ser "não precisa de ML".

Escolhendo a técnica de ML

Técnica	Pergunta que responde	Saída	Exemplo
Regressão	Quanto? (valor numérico contínuo)	Número	Preço de imóvel, demanda do próximo mês
Classificação	Qual categoria?	Classe/rótulo	Fraude ou não, spam ou não
Agrupamento (clustering)	Quais itens se parecem? (sem rótulos)	Grupos	Segmentação de clientes
Detecção de anomalias	O que foge do padrão?	Itens atípicos	Transação incomum

Aplicações reais

- **Visão computacional**: detecção de objetos, moderação de conteúdo.
- **NLP**: análise de sentimento, extração de entidades, tradução.
- **Reconhecimento de fala**: transcrição de chamadas.
- **Sistemas de recomendação**: produtos e conteúdo personalizados.
- **Detecção de fraude** e **previsão (forecasting)** de séries temporais.
- **Bases de conhecimento (knowledge bases)**: respostas fundamentadas em documentos da empresa.
- **IA agêntica**: agentes que executam tarefas em várias etapas.

Serviços gerenciados de IA/ML da AWS (pré-treinados, via API)

Serviço	O que faz	Palavra-gatilho
Amazon Comprehend	NLP de *compreensão*: sentimento, entidades, frases-chave, detecção de PII. Não é generativo.	"analisar sentimento", "extrair entidades"
Amazon Transcribe	Fala para texto.	"transcrever áudio/chamadas"
Amazon Polly	Texto para fala.	"voz natural a partir de texto"
Amazon Translate	Tradução de idiomas.	"traduzir conteúdo"
Amazon Lex	Chatbots conversacionais por voz e texto.	"construir chatbot/bot de voz"
Amazon Rekognition	Análise de imagem e vídeo, detecção de objetos e rostos, moderação.	"detectar objetos em imagens"
Amazon Textract	Extrai texto e dados de documentos e formulários digitalizados.	"extrair dados de PDF/formulário"
Amazon Personalize	Recomendações personalizadas.	"recomendação de produtos"
Amazon SageMaker AI	Plataforma para construir, treinar e implantar seus próprios modelos de ML.	"dados rotulados próprios", "algoritmo customizado"

> **Padrão da prova:** se a necessidade é uma tarefa estreita (sentimento, tradução, transcrição) e não exige gerar conteúdo novo, o serviço pré-treinado costuma ser mais simples e barato que montar uma solução com um foundation model.

ML tradicional x foundation models (FMs)

Critério	ML tradicional	Foundation model
Escopo	Modelo específico para uma tarefa (ex.: classificar sentimento)	Modelo amplo e adaptável a várias tarefas por prompt
Explicabilidade	Modelos mais simples são mais fáceis de explicar	Mais difícil de interpretar
Restrições regulatórias e operacionais	Preferível quando há exigência de explicar cada decisão ou de resultado previsível	Exige controles extras (guardrails, revisão humana)
Dados	Normalmente precisa de dados rotulados próprios	Já pré-treinado; adapta com prompt, RAG ou fine-tuning
Custo de infraestrutura	Menor	Treinar do zero é caro; usar via API (Amazon Bedrock) evita isso

> **Armadilha:** "regulatório" ou "precisa explicar cada decisão" aponta para modelo tradicional/simples. "Várias tarefas", "conversa", "gerar conteúdo" aponta para FM.

1.3 Ciclo de vida de desenvolvimento de IA/ML

Componentes do pipeline de ML

Etapa	O que acontece	Ferramentas AWS (exemplos)
1. Definição do problema de negócio	Definir o problema e a métrica de sucesso antes de mexer nos dados.	-
2. Coleta e preparação de dados	Coletar, limpar, rotular e transformar; engenharia de atributos.	SageMaker AI (Data Wrangler, Ground Truth, Feature Store)
3. Seleção e treinamento do modelo	Escolher o algoritmo e treinar; ajustar hiperparâmetros.	Amazon SageMaker AI (treino, ajuste automático)
4. Avaliação	Validar com dados separados usando métricas técnicas e de negócio.	SageMaker AI, Amazon Bedrock (avaliação de modelos)
5. Implantação	Disponibilizar o modelo como serviço (tempo real, lote, assíncrono, serverless).	SageMaker AI endpoints, Amazon Bedrock
6. Monitoramento e retreinamento	Observar desempenho e drift; retreinar quando degradar.	SageMaker AI (monitoramento de modelos)

Fontes de modelos (inclusive FMs)

- **Modelos pré-treinados de código aberto** (ex.: disponíveis via Amazon SageMaker JumpStart).

- **Modelos pré-treinados de provedores** acessados por API gerenciada (Amazon Bedrock).
- **Treinar um modelo customizado** do zero com seus dados (mais caro e exige mais expertise).

Como usar o modelo em produção

Método	Descrição	Quando
API gerenciada	Você chama o modelo por API sem cuidar da infraestrutura (ex.: Amazon Bedrock).	Rapidez, baixa carga operacional
API auto-hospedada (self-hosted)	Você implanta o modelo em seus próprios endpoints/infraestrutura (ex.: SageMaker AI, com escolha de instância).	Controle total, modelo próprio ou não disponível em API gerenciada

Serviços AWS por estágio

Necessidade	Serviço
Acessar FMs prontos por API, RAG, agentes, guardrails	**Amazon Bedrock**
Construir, treinar e implantar modelos próprios; hospedar modelos abertos	**Amazon SageMaker AI** (inclui **SageMaker JumpStart**)
Assistente de IA de trabalho (criar conteúdo, pesquisar, automatizar tarefas)	**Amazon Quick**
Desenvolvimento de software com IA agêntica (specs, geração e verificação de código)	**Kiro**

Fundamentos de MLOps

MLOps é o conjunto de práticas que automatiza e simplifica fluxos e implantações de ML, unindo desenvolvimento e operações ao longo do ciclo de vida.

Conceito	Ideia
Experimentação	Testar e comparar execuções de treino, dados e hiperparâmetros.
Processos repetíveis	Controle de versão e automação para reproduzir resultados.
Sistemas escaláveis	Pipelines e infraestrutura que crescem com a demanda.
Gerenciar débito técnico	Evitar acúmulo de atalhos e dependências frágeis no código e nos pipelines de ML.
Prontidão para produção	Testes, CI/CD e governança antes de liberar um modelo.
Monitoramento do modelo	Acompanhar qualidade e deriva (drift) em produção.
Retreinamento	Atualizar o modelo quando o desempenho cair ou os dados mudarem.

Benefícios citados pela AWS: entrada mais rápida no mercado, maior produtividade com experimentação padronizada e melhor implantação, com versionamento centralizado e CI/CD.

Métricas de desempenho do modelo

Métrica	Fórmula/ideia	Use quando
Acurácia (accuracy)	Acertos / total de predições	Classes equilibradas; enganosa em dados desbalanceados
Precisão (precision)	Dos casos marcados como positivos, quantos eram realmente positivos	Falso positivo é caro (ex.: e-mail legítimo indo para spam)
Recall (sensibilidade)	De todos os positivos reais, quantos o modelo encontrou	Perder um positivo é caro (ex.: fraude, doença)
F1 score	Média harmônica de precisão e recall	Um número único que equilibra os dois
Matriz de confusão	Tabela de VP, FP, VN, FN	Origem de todas as métricas acima
AUC-ROC	Capacidade de separar classes em todos os limiares; mais perto de 1 é melhor	Comparar classificadores

> **Armadilha:** acurácia alta com classes desbalanceadas pode esconder um modelo ruim (ex.: 99% de transações são legítimas). Custo de perder o positivo = recall. Custo de alarme falso = precisão.

Métricas de negócio

- **Custo por usuário** e **custos de desenvolvimento**.
- **Feedback dos clientes** (satisfação).
- **ROI** (retorno sobre o investimento): benefício obtido em relação ao custo total.
- Métrica técnica boa não garante valor de negócio: sempre ligue o modelo ao objetivo definido na etapa 1.

📋 Resumo rápido do Domínio 1

- IA ⊃ ML ⊃ deep learning ⊃ GenAI: camadas, não sinônimos. Gera artefato novo = GenAI; rótulo ou número = ML tradicional; decide e age com ferramentas = IA agêntica.
- Supervisionado = dados rotulados. Não supervisionado = sem rótulos (agrupar, anomalias). Reforço = agente + recompensa. Autossupervisionado = rótulos gerados do próprio dado (pré-treino de FMs).
- Overfitting = bom no treino, ruim em dado novo. Underfitting = ruim em ambos.
- Treino / validação / teste separados; misturar = vazamento de dados.
- Inferência: tempo real = baixa latência; lote = dataset grande sem urgência; assíncrona = payload grande/longo; serverless = tráfego intermitente com cold start.
- Regressão = número; classificação = categoria; clustering = grupos sem rótulos.
- Comprehend (NLP), Transcribe (fala para texto), Polly (texto para fala), Translate, Lex (chatbot), Rekognition (imagem), Textract (documentos), Personalize (recomendação).
- Pipeline: problema → dados → treino → avaliação → implantação → monitoramento e retreino. MLOps = automação, versionamento, monitoramento.
- Fontes de modelo: abertos pré-treinados, via API gerenciada, ou treinar customizado. Produção: API gerenciada (Bedrock) ou auto-hospedada (SageMaker AI).
- Precisão = menos falsos positivos. Recall = menos falsos negativos. F1 = equilíbrio. Negócio = ROI, custo por usuário, feedback.

> **Armadilhas:** (1) regra fixa e simples não precisa de ML. (2) acurácia engana em dados desbalanceados. (3) serviço pré-treinado estreito costuma vencer o FM para tarefa única. (4) exigência de explicabilidade/regulação favorece modelo mais simples.

Domínio 2: Fundamentos de IA Generativa (24%)

Tarefas oficiais: 2.1 Explicar os conceitos básicos de IA generativa (GenAI) · 2.2 Compreender capacidades e limitações da GenAI para resolver problemas de negócio · 2.3 Descrever a infraestrutura e as tecnologias da AWS para construir aplicações de GenAI.

2.1 Conceitos básicos de IA generativa

Termos fundamentais

Termo	Definição	Gatilho
Foundation model (FM)	Grande rede de deep learning treinada em conjuntos de dados enormes e amplos; realiza tarefas gerais e é adaptada (prompt, RAG, fine-tuning) em vez de treinada do zero.	"modelo de propósito geral"
LLM	FM de linguagem, baseado em **transformers**, que prevê o próximo token.	"gera texto"
Token	Unidade básica que o modelo processa (parte de palavra). Preço, janela de contexto e latência são medidos em tokens.	"cobrança por token"
Janela de contexto	Máximo de tokens que o modelo considera de uma vez. Maior janela permite documentos longos, mas aumenta custo e latência.	"documento muito longo"
Chunking	Dividir documentos grandes em pedaços (chunks) menores antes de gerar embeddings, para recuperar só os trechos relevantes.	"quebrar documentos para RAG"
Embeddings / vetores	Representações numéricas (vetores) que capturam o significado; conteúdos parecidos ficam próximos no espaço vetorial.	"busca semântica", "similaridade"
Banco de dados vetorial	Armazena embeddings e faz busca por similaridade (ex.: Amazon OpenSearch Service, Amazon Aurora, Amazon RDS).	"armazenar vetores"
Engenharia de prompt	Guiar o modelo para a saída desejada escolhendo formato, frases e palavras do prompt, sem alterar os pesos.	"melhorar a resposta sem treinar"
Modelo multimodal	Aceita e/ou produz mais de uma modalidade (texto, imagem, áudio, vídeo).	"texto + imagem"
Modelo de difusão	Gera imagens partindo de ruído aleatório e removendo o ruído iterativamente (ex.: Stable Diffusion).	"texto para imagem"
Transformer	Arquitetura dos LLMs modernos; a **autoatenção (self-attention)** pesa a importância de cada token em relação aos demais.	"arquitetura dos LLMs"

> **Armadilha:** chunking e embeddings são etapas de preparação para RAG e busca. O modelo de difusão gera imagem; o LLM baseado em transformer gera texto. Embeddings não são o texto gerado, são vetores.

Técnicas de prompt

Técnica	Descrição
Zero-shot	Só a instrução, sem exemplos.
Few-shot	Poucos exemplos no prompt para orientar formato e comportamento.
Chain-of-thought (CoT)	Divide uma pergunta complexa em passos intermediários de raciocínio, em vez de responder direto.
Modelos de prompt (templates)	Prompts reutilizáveis com variáveis.
Prompt negativo	Diz ao modelo o que não incluir (comum em geração de imagem).

> Nenhuma técnica de prompt altera os pesos do modelo. É a opção mais barata e rápida e não exige dados de treino. Risco de segurança: **prompt injection** (entrada maliciosa tentando sobrescrever as instruções).

Customizar um FM: prompt, RAG ou fine-tuning

Abordagem	Altera pesos?	Resolve	Custo/complexidade	Dados
Engenharia de prompt	Não	Ajuste rápido de comportamento	Menor	Nenhum
RAG	Não	Fatos atuais ou privados; reduz alucinação sem retreinar	Médio (vetor + recuperação)	Seus documentos (sem rótulos)
Fine-tuning	**Sim**	Estilo, formato, tarefa ou jargão específico	Maior (treino)	Pares de exemplo rotulados

RAG (Retrieval Augmented Generation): o modelo consulta uma base de conhecimento autoritativa fora dos dados de treino antes de responder. Passos: (1) converter dados externos em vetores num banco vetorial; (2) transformar a pergunta em vetor e recuperar o que é relevante; (3) aumentar o prompt com o conteúdo recuperado; (4) manter os dados externos atualizados. Benefícios: custo menor que retreinar, informação atual, confiança (citação de fontes) e controle sobre as fontes. No Amazon Bedrock, o recurso gerenciado é o **Knowledge Bases**.

No Amazon Bedrock, os métodos de customização incluem **supervised fine-tuning** (dados rotulados, ajusta os parâmetros), **reinforcement fine-tuning** (funções de recompensa em vez de pares rotulados) e **distillation** (transfere conhecimento de um modelo professor maior para um modelo aluno menor, mais rápido e barato). Pré-treinamento contínuo (continued pre-training) é o conceito de adaptar o modelo a um domínio com dados não rotulados.

Pergunta da prova	Resposta
"Nossos dados mudam com frequência"	RAG (fine-tuning exigiria retreinar sempre)
"Modelo precisa conhecer documentos internos"	RAG
"Adotar um tom/formato específico de forma permanente"	Fine-tuning
"Ajuste rápido, sem dados de treino"	Engenharia de prompt
"Modelo menor, mais barato, mantendo a qualidade de um maior"	Distillation

Parâmetros de inferência

Parâmetro	O que controla	Valor baixo	Valor alto
Temperature	Forma da distribuição de probabilidade do próximo token	Respostas mais determinísticas e focadas	Respostas mais aleatórias e criativas
Top K	Número fixo dos candidatos mais prováveis considerados	Menos candidatos	Mais candidatos, permite saídas menos prováveis
Top P	Fração da probabilidade acumulada considerada (tamanho do conjunto varia por token)	Conjunto menor	Conjunto maior, mais diversidade
Response length (max tokens)	Limita o número de tokens da resposta	Resposta curta	Resposta longa
Stop sequences	Sequências que fazem o modelo parar de gerar	-	-
Penalidades	Penalizam repetição, frequência ou tipos de tokens	-	-

> **Armadilha:** Top K é um número fixo de candidatos; Top P é um limiar de probabilidade acumulada. Para respostas consistentes e factuais use temperature baixa. Para criatividade, temperature alta.

Casos de uso de GenAI

Caso de uso	Exemplo
Geração de imagem, vídeo e áudio	Criar imagens a partir de texto
Resumo	Resumir contratos, reuniões, tickets
Assistentes de IA	Perguntas e respostas sobre dados da empresa
Tradução	Conteúdo multilíngue
Geração de código	Sugestões e geração de código
Agentes de atendimento ao cliente	Chatbots fundamentados em documentos (RAG)
Busca	Busca semântica com embeddings
Motores de recomendação	Conteúdo e produtos personalizados

Ciclo de vida de um FM

Etapa	O que acontece
1. Seleção de dados	Escolher e preparar dados amplos e de qualidade
2. Seleção do modelo	Escolher arquitetura/FM conforme tarefa, custo e restrições
3. Pré-treinamento	Treinar em grandes volumes de dados (aprendizado autossupervisionado)
4. Fine-tuning	Especializar para uma tarefa ou domínio
5. Avaliação	Medir com métricas automáticas, avaliação humana e métricas de negócio
6. Implantação	Disponibilizar o modelo (API gerenciada ou hospedagem própria)
7. Feedback	Coletar retorno de usuários e monitorar para melhorar o modelo

Preço baseado em tokens

- Modelos cobram separadamente por tokens de **entrada** (prompt, contexto) e de **saída** (resposta), em geral por milhão de tokens. O preço varia por provedor/modelo, Região e modalidade.
- **Efeito no custo:** prompts longos, muito contexto recuperado (RAG) e respostas longas aumentam o custo. Reduzir o tamanho do prompt e limitar max tokens reduz custo.
- **Efeito no desempenho:** mais tokens aumentam a latência. Modelos maiores costumam ser mais capazes, porém mais caros e mais lentos.

Engenharia de contexto (context engineering)

É o projeto do **conteúdo que entra na janela de contexto** do FM: instruções, histórico de conversa, documentos recuperados (RAG), resultados de ferramentas e memória. Vai além de escrever um bom prompt: decide *o que* o modelo vê, em que ordem e com quanto volume, equilibrando qualidade, custo (tokens) e latência.

Conceitos de IA agêntica

Conceito	Ideia
Agente	Sistema autônomo que age de forma independente para atingir objetivos predefinidos. Ciclo: perceber, raciocinar (LLM), agir e aprender.
Uso de ferramentas (tool usage)	O agente chama APIs, funções e sistemas externos para executar ações.
Memória	Mantém continuidade da tarefa e do contexto (curto e longo prazo).
Orquestração de fluxo (workflow)	Coordena as etapas, decisões e ferramentas da tarefa.
Sistemas multiagente	Vários agentes especializados coordenados, muitas vezes sob um agente supervisor; mais escalável e flexível para aplicações complexas.
Padrões de comunicação multiagente	Formas como os agentes trocam informação e delegam tarefas (ex.: supervisor que distribui para agentes especialistas).
MCP (Model Context Protocol)	Protocolo padrão para conectar agentes a sistemas, ferramentas e dados externos. Em plataformas como o Amazon Bedrock AgentCore, o gateway conecta agentes a servidores MCP, APIs e funções Lambda.

> **Armadilha:** GenAI "responde"; IA agêntica "age". Em perguntas sobre conectar agentes a sistemas externos de forma padronizada, a palavra-chave é MCP.

2.2 Capacidades e limitações da GenAI para negócios

Vantagens x desvantagens

Vantagens	Desvantagens
Adaptabilidade a muitas tarefas	**Alucinações**: respostas plausíveis, porém falsas
Responsividade e rapidez para gerar resultados	**Interpretabilidade** baixa (caixa-preta)
Capacidade conversacional	**Imprecisão** e possível viés
Geração de conteúdo em escala	**Não determinismo**: o mesmo prompt pode gerar saídas diferentes
Tempo de valor menor que construir modelos do zero	Custo e latência em escala; incerteza legal/IP sobre dados e saídas

> **Mitigar alucinação:** RAG (fundamentar em fontes), temperature menor, revisão humana e guardrails (como o Guardrails for Amazon Bedrock).

Fatores para escolher um modelo

Fator	O que avaliar
Tipo de modelo e modalidades	Texto, imagem, multimodal, embeddings
Requisitos de desempenho e capacidades	Qualidade na tarefa, tamanho da janela de contexto
Restrições e conformidade	Residência de dados, regulação, licença, privacidade
Custo	Preço por token, necessidade de throughput reservado
Latência	Tempo de resposta exigido pela aplicação
Complexidade do modelo	Modelos maiores = mais capazes, porém mais caros e lentos; avalie um modelo menor se atender

Valor de negócio e métricas

Métrica	O que mostra
Desempenho entre domínios	Quão bem o modelo atua em diferentes tipos de tarefa
ROI	Retorno sobre o investimento
Eficiência	Tempo e custo economizados para atingir o resultado
Taxa de conversão	Quantos visitantes realizam a ação desejada
Receita média por usuário (ARPU)	Receita gerada por usuário
Acurácia	Correção das respostas
Valor do tempo de vida do cliente (CLV)	Receita total esperada de um cliente

> **Armadilha:** métrica de qualidade do modelo (acurácia) não é métrica de negócio. Para medir impacto, use conversão, ARPU, CLV, ROI e eficiência.

2.3 Infraestrutura e tecnologias AWS para GenAI

Serviços e recursos

Serviço	Papel	Gatilho
Amazon Bedrock	Plataforma gerenciada para construir apps e agentes de GenAI com FMs de vários provedores por API, sem gerenciar infraestrutura. Inclui Knowledge Bases, Guardrails, agentes, customização e avaliação de modelos. A AWS afirma que não armazena nem usa seus dados para treinar modelos.	"acessar FMs por API", "sem infraestrutura"
Amazon SageMaker AI	Construir, treinar e implantar seus próprios modelos, com controle total.	"treinar", "controle da infraestrutura"
Amazon SageMaker JumpStart	Catálogo de modelos pré-treinados e abertos para implantar (inclusive pela interface do Studio).	"implantar modelo aberto rapidamente"
Amazon Quick	Assistente de IA para o trabalho: cria documentos e dashboards, pesquisa, automatiza fluxos e se integra a aplicações corporativas.	"assistente de trabalho"
Kiro	Sistema de engenharia agêntico: desenvolvimento orientado a specs, agentes que planejam, constroem e verificam código.	"desenvolvimento com agentes"
Strands Agents	Kit para construir agentes (framework suportado pelo Amazon Bedrock AgentCore).	"construir agentes em código"
Amazon Bedrock AgentCore	Plataforma para colocar agentes em produção com segurança: runtime, gateway, identidade, observabilidade e avaliação; qualquer framework e qualquer modelo.	"agentes em produção em escala"
Amazon Nova	Família de FMs da Amazon.	"modelos da própria Amazon"

Dimensão	Amazon Bedrock	Amazon SageMaker AI
Melhor para	FMs, apps de GenAI, rapidez	Modelos customizados e ML tradicional, controle total
Infraestrutura	Gerenciada, sem servidores para você operar	Você escolhe instâncias e endpoints
Origem do modelo	FMs de provedores via API	Seu modelo ou modelos do JumpStart
Gatilho	"chatbot", "gerar texto/imagem", "RAG", "agentes"	"dados rotulados próprios", "algoritmo customizado"

> **Armadilha:** JumpStart fica dentro do SageMaker AI, mas implanta FMs. Se o cenário exige controle de instância ou modelo fora do Amazon Bedrock, ele pode ser a resposta; se quer "FM rápido e gerenciado", é Bedrock.

Vantagens de usar serviços de GenAI da AWS

- **Acessibilidade e menor barreira de entrada**: usar FMs por API sem montar infraestrutura.
- **Eficiência e economia**: não treinar do zero; pagar pelo uso.
- **Velocidade de chegada ao mercado**.
- Capacidade de atender objetivos de negócio com recursos como RAG, agentes e guardrails.

Benefícios da infraestrutura AWS para GenAI

- **Segurança**: criptografia em trânsito e em repouso; os dados não são usados para treinar modelos no Amazon Bedrock.
- **Conformidade**: o Amazon Bedrock cita aderência a padrões como ISO, SOC, CSA STAR Nível 2, GDPR, FedRAMP High e elegibilidade HIPAA.
- **Responsabilidade e segurança (safety)**: guardrails para filtrar conteúdo nocivo e informações sensíveis.

Trade-offs de custo

Opção	Como cobra	Quando
On-Demand	Por token, sem compromisso	Uso variável ou inicial
Batch inference	Processamento em lote, preço menor que o on-demand para modelos selecionados (a AWS cita 50% de desconto)	Sem urgência
Provisioned Throughput	Por hora, com compromisso (1 ou 6 meses), capacidade reservada	Volume alto e consistente; modelos customizados
Modelos customizados	Treino cobrado por tokens processados (tokens do corpus x épocas) + armazenamento mensal do modelo + inferência	Quando prompt e RAG não bastam

- **Responsividade, disponibilidade e redundância**: mais capacidade reservada e redundância entre Regiões aumentam custo.
- **Desempenho**: modelos maiores e throughput dedicado custam mais.
- **Cobertura regional**: modelos e recursos variam por Região; o preço também varia por Região.

> **Armadilha:** Provisioned Throughput é compromisso por tempo (hora/1 ou 6 meses), não "pague por token". On-Demand é o inverso: sem compromisso e por token.

📋 Resumo rápido do Domínio 2

- FM = modelo amplo pré-treinado, adaptado por prompt/RAG/fine-tuning. LLM = FM de texto baseado em transformer (autoatenção).
- Token = unidade de processamento e de preço. Janela de contexto = máximo de tokens vistos de uma vez.
- Embeddings = vetores de significado, guardados em banco vetorial; chunking = dividir documentos antes de embutir.
- Difusão = ruído para imagem. Multimodal = mais de uma modalidade.
- Zero-shot sem exemplos; few-shot com exemplos; CoT passo a passo. Nenhuma muda pesos.
- Fatos atuais/privados = RAG. Estilo/formato permanente = fine-tuning. Ajuste rápido sem dados = prompt. Modelo menor e barato = distillation.
- Temperature baixa = determinístico; alta = criativo. Top K = nº fixo de candidatos; Top P = probabilidade acumulada.
- Ciclo do FM: dados → modelo → pré-treino → fine-tuning → avaliação → implantação → feedback.
- Preço por token: entrada e saída cobradas separadamente; prompts e respostas longos custam mais e aumentam latência.
- Engenharia de contexto = decidir o que entra na janela de contexto. Agentes = ferramentas, memória, orquestração, multiagente; MCP conecta agentes a sistemas externos.
- Vantagens: adaptabilidade, conversa, geração de conteúdo. Limites: alucinação, não determinismo, baixa interpretabilidade, imprecisão.
- Métricas de negócio: ROI, conversão, ARPU, CLV, eficiência, desempenho entre domínios.
- Bedrock = FMs gerenciados por API. SageMaker AI = construir/treinar/implantar. JumpStart = modelos abertos. AgentCore = agentes em produção. Quick = assistente de trabalho. Kiro = dev agêntico.
- Custo: On-Demand (por token), Batch (menor preço), Provisioned Throughput (compromisso, capacidade reservada).

> **Armadilhas:** (1) RAG muda o que o modelo sabe, fine-tuning muda como ele se comporta. (2) temperature não corrige fatos, só variação. (3) Top K (fixo) x Top P (acumulado). (4) métrica técnica não é métrica de negócio. (5) JumpStart está no SageMaker AI, mas implanta FMs.

Domínio 3: Aplicações de Modelos de Fundação (28%)

Tarefas oficiais: 3.1 considerações de design para aplicações que usam FMs · 3.2 técnicas eficazes de engenharia de prompts · 3.3 processo de treinamento e ajuste fino (fine-tuning) de FMs · 3.4 métodos para avaliar o desempenho de FMs.

> **Este é o maior domínio da prova.** A regra de ouro: a AWS premia a opção **mais gerenciada** e **mais barata que resolve** o problema. A escada de personalização, do mais barato ao mais caro, é: **engenharia de prompts → RAG → fine-tuning → continued pre-training → pre-training do zero**.

3.1 Design de aplicações com modelos de fundação

Critérios de seleção de um FM

Critério	Pergunta a fazer	Palavra-gatilho
Custo	Preço por token (entrada e saída) ou por hora (Provisioned Throughput)?	orçamento, barato
Modalidade	O modelo recebe/gera texto, imagem, áudio, vídeo ou embeddings?	imagem, multimodal, embeddings
Latência	Resposta quase em tempo real ou pode ser demorada?	tempo real, resposta rápida
Multilíngue	Suporta os idiomas dos usuários?	vários idiomas
Tamanho do modelo	Modelos maiores tendem a ser mais capazes, porém mais lentos e caros; menores são mais rápidos e baratos.	leve, rápido
Complexidade do modelo	A tarefa exige raciocínio complexo ou é simples (classificar, resumir)?	tarefa simples vs complexa
Customização	O modelo aceita fine-tuning, continued pre-training ou distillation no Amazon Bedrock?	adaptar ao domínio
Tamanho de entrada/saída	A janela de contexto comporta o prompt e a resposta esperada?	documento longo, contexto
Prompt caching	Há um contexto longo e repetido entre requisições? O cache reduz latência e custo de tokens de entrada.	contexto repetido, mesmo documento

> **Amazon Bedrock** oferece FMs de vários provedores por uma **API única**, sem gerenciar infraestrutura, com segurança e privacidade. Os modelos **Amazon Nova** estão no escopo da prova. Para **comparar modelos** antes de escolher, use o **Amazon Bedrock Model Evaluation** (seção 3.4).

Parâmetros de inferência

Para cada token, o modelo calcula uma distribuição de probabilidade e amostra dela. Os parâmetros ajustam essa distribuição ou limitam a resposta. Valores padrão e faixas dependem do modelo.

Parâmetro	Valor baixo	Valor alto
Temperature	Respostas mais **determinísticas** e previsíveis (fatos, extração, código).	Respostas mais **aleatórias e criativas** (brainstorming, texto criativo).
Top K	Considera poucos candidatos (os K tokens mais prováveis).	Considera mais candidatos, permitindo saídas menos prováveis.
Top P	Considera pequena fatia da probabilidade acumulada (ex.: 0.7).	Considera fatia maior (ex.: 0.9), mais diversidade.
Tamanho da resposta (max tokens)	Limita o número de tokens gerados (reduz custo e latência).	Permite respostas mais longas.
Stop sequences	Sequências de caracteres que interrompem a geração.	-
Penalidades	Penalizam repetição, frequência ou tamanho em excesso.	-

> **Armadilha:** temperature **não** muda o conhecimento do modelo nem corrige alucinação; só altera a aleatoriedade. Para respostas **consistentes e factuais**, use temperature **baixa**. O tamanho da entrada + saída tem limite (janela de contexto) e cada token é cobrado.

RAG (Retrieval Augmented Generation)

RAG usa informações de fontes de dados para melhorar a relevância e a precisão das respostas geradas, **sem retreinar** o modelo e sem alterar seus pesos. Fluxo: (1) os documentos são divididos em **chunks**; (2) um **modelo de embeddings** converte cada chunk em um **vetor**; (3) os vetores ficam em um **vector database**; (4) na consulta, a pergunta também vira vetor e a **busca por similaridade** recupera os trechos mais próximos; (5) os trechos são adicionados ao prompt e o FM gera a resposta, podendo incluir **citações** das fontes.

- **Aplicações de negócio:** chatbot de suporte baseado em manuais, assistente de RH/TI sobre documentos internos, busca semântica em base de conhecimento, respostas com dados atualizados e privados.
- **Amazon Bedrock Knowledge Bases:** RAG gerenciado. Conecta fontes de dados, cuida de ingestão, embeddings e recuperação. Hoje oferece *Managed Knowledge Base* (a AWS gerencia ingestão, indexação, armazenamento e recuperação) e *Customer-managed* (você escolhe e administra o vector store).
- **Embeddings:** representações numéricas (vetores) que capturam o significado; textos semelhantes ficam próximos. O Amazon Bedrock oferece modelos de embeddings (ex.: Amazon Titan Text Embeddings, Cohere Embed). A dimensão do vetor deve coincidir com a do índice.

Serviços AWS para armazenar embeddings (vector databases)

Serviço	Observação
Amazon OpenSearch Service (incl. OpenSearch Serverless)	Busca vetorial e híbrida. Único (com Serverless) que suporta vetores binários nas Knowledge Bases.
Amazon Aurora (PostgreSQL) e **Amazon RDS for PostgreSQL**	Extensão **pgvector**. Ideal se você já usa banco relacional.
Amazon Neptune (Neptune Analytics)	Banco de grafos; suporta **GraphRAG** (relações entre entidades).
Amazon S3 Vectors	Armazenamento de vetores no S3, custo menor, indicado para consultas pouco frequentes.
Amazon DocumentDB	Também aparece na lista de serviços do escopo (banco de documentos, compatível com MongoDB).
Terceiros	As Knowledge Bases também suportam Pinecone, Redis Enterprise Cloud e MongoDB Atlas.

> **Palavras-gatilho do RAG:** "responder com base nos documentos da empresa", "reduzir alucinações com dados próprios", "dados atualizados sem retreinar", "citar a fonte". **Fine-tuning não é a resposta** quando o dado muda com frequência.

Custo das abordagens de personalização

Abordagem	O que faz	Custo / esforço	Quando usar
Prompt engineering / **in-context learning**	Exemplos e instruções dentro do prompt; pesos não mudam.	**Mais baixo**; sem treino.	Primeira tentativa, tarefas simples.
RAG	Injeta dados externos no prompt; pesos não mudam.	Baixo a médio (índice vetorial + tokens maiores).	Dados privados ou que mudam.
Fine-tuning	Treina com dados **rotulados** e ajusta pesos.	Médio a alto (treino por tokens × épocas, armazenamento do modelo, Provisioned Throughput).	Estilo, formato ou tarefa específica.
Continued pre-training	Continua o pré-treino com dados **não rotulados** do domínio.	Alto.	Adaptar ao vocabulário de um domínio inteiro.
Model distillation	Transfere conhecimento de um modelo **teacher** grande para um **student** menor, mais rápido e barato.	Custo de gerar/treinar uma vez; **inferência mais barata** depois.	Manter qualidade com menor custo e latência em produção.
Pre-training do zero	Treina um FM inteiro.	**Mais alto** (enormes dados e computação).	Raro; só para organizações com recursos extremos.

> **Armadilha:** no Amazon Bedrock, um modelo **customizado** só pode ser usado se você comprar **Provisioned Throughput**. O treino é cobrado por tokens processados (tokens do corpus × épocas) e o armazenamento do modelo é cobrado por mês.

Bedrock: modos de preço e outros recursos de aplicação

Recurso	Função	Palavra-gatilho
On-demand	Paga por tokens processados, sem compromisso.	tráfego imprevisível, testes
Provisioned Throughput	Capacidade dedicada por **Model Units**, cobrança **por hora**; compromissos: sem compromisso, **1 mês** ou **6 meses** (mais longo, mais desconto). Obrigatório para modelo customizado.	carga alta e constante, modelo customizado
Batch inference	Processa grandes volumes de forma assíncrona, sem tempo real.	offline, grande volume
Prompt caching	Reaproveita prefixos de prompt repetidos para reduzir latência e custo de tokens de entrada; é melhor-esforço (não garante acerto). Só em inferência on-demand, não no batch.	mesmo documento em várias perguntas
Guardrails	Salvaguardas configuráveis em entrada e saída (detalhe abaixo).	bloquear tópico, mascarar PII

Agentes de IA

Um **agente** é um FM capaz de entender o pedido do usuário, **quebrar a tarefa em etapas**, coletar informações em conversa, **chamar APIs/sistemas** da empresa e consultar bases de conhecimento para cumprir a tarefa. No **Agents for Amazon Bedrock**, os componentes principais são **action groups** (ações/APIs que o agente pode executar) e a associação opcional com uma **knowledge base**; é possível acompanhar o raciocínio por **traces**. Exemplos de negócio: processar pedidos de sinistro de seguro, fazer reservas de viagem, automatizar atendimento.

> **Atualização do escopo:** o guia oficial lista **Amazon Bedrock AgentCore** e **Strands Agents** (SDK de código aberto para construir agentes) como serviços no escopo. A documentação indica que o Amazon Bedrock Agents (agora "Agents Classic") não aceita novos cadastros, e orienta agentes definidos em código com Strands Agents e AgentCore. Para a prova, o conceito vale: **agente = FM + planejamento em etapas + ferramentas/APIs + memória**.

Preciso de...	Use
Responder com base em documentos	**Knowledge Bases** (RAG)
Executar ações em sistemas (API, Lambda), multi-etapas	**Agentes** (action groups)
Controlar/filtrar o que entra e sai do modelo	**Guardrails**
Comparar modelos ou medir qualidade	**Model Evaluation**
Reutilizar e versionar prompts	**Prompt Management**

Amazon Bedrock Guardrails (controle do que o modelo diz)

Salvaguarda	O que faz
Content filters	Detectam conteúdo nocivo em texto/imagem em categorias: Hate, Insults, Sexual, Violence, Misconduct e Prompt Attack (ataques de prompt). A intensidade é configurável.
Denied topics	Bloqueia tópicos indesejados definidos por você.
Word filters	Bloqueia palavras/frases (ex.: palavrões, nomes de concorrentes).
Sensitive information filters	Bloqueiam ou **mascaram PII** (ex.: CPF/SSN, endereço) e aceitam padrões regex.
Contextual grounding checks	Detectam **alucinações**: resposta não embasada na fonte ou irrelevante à pergunta (típico em RAG).
Automated Reasoning checks	Validam respostas contra regras lógicas.

- Aplicam-se a **entradas e saídas**; podem ser usados em chamadas ao FM ou isoladamente pela API **ApplyGuardrail**.
- Têm rascunho (working draft), janela de teste e **versões**.
- Pergunta "como impedir que o modelo fale de X / redigir PII / filtrar toxicidade" → **Guardrails**.

3.2 Engenharia de prompts

Engenharia de prompts é otimizar o texto de entrada de um LLM para obter as respostas desejadas. A qualidade do prompt impacta a qualidade da resposta. Não altera o modelo; é a forma **mais barata e rápida** de melhorar resultados.

Conceitos e componentes de um prompt

Componente	Descrição
Instrução (task)	O que o modelo deve fazer (ex.: "Resuma em uma frase").
Contexto	Informação de apoio sobre o domínio ou a situação, que guia o uso da entrada.
Dados de entrada (input)	O texto/dado sobre o qual o modelo atua.
Exemplos (demonstrações)	Pares entrada/saída que calibram o formato esperado.
Indicador de saída / formato	Como a resposta deve vir (JSON, lista, tom, tamanho).
Negative prompt	Diz o que o modelo **não** deve incluir/gerar (muito usado em geração de imagens).

Técnicas

Técnica	Como é	Quando usar
Zero-shot	Só a instrução, nenhum exemplo.	Tarefas simples e comuns.
Single-shot (one-shot)	Um exemplo.	Formato simples a demonstrar.
Few-shot	Vários exemplos no prompt (também chamado *in-context learning*).	Calibrar formato/estilo sem treinar.
Chain-of-thought (CoT)	Pede ao modelo para raciocinar **passo a passo** antes da resposta (ex.: "pense passo a passo").	Problemas de lógica, matemática, várias etapas.
Prompt template	Receita com partes fixas e campos substituíveis (ex.: `{{texto}}`).	Reutilizar o mesmo padrão em escala.

> **Boas práticas:** ser **específico e conciso**; dar contexto claro; definir o formato de saída; usar exemplos quando preciso; **experimentar e iterar** (descobrir o que funciona por tentativa); usar delimitadores/múltiplos comentários ou seções para organizar; combinar com **guardrails** para segurança. Para reduzir alucinações: otimizar o prompt, usar RAG, ou trocar de modelo.

Riscos e limitações

Risco	Descrição
Exposição (prompt leaking)	O modelo revela dados sensíveis ou o próprio prompt/instruções do sistema.
Poisoning (envenenamento)	Dados ou prompts maliciosos introduzidos para distorcer o comportamento do modelo.
Hijacking / **prompt injection**	Instruções maliciosas embutidas na entrada sequestram o objetivo original do prompt.
Jailbreaking	Tentativa de contornar as proteções de segurança do modelo para obter conteúdo proibido.

- Mitigações: **Guardrails** (inclui filtro de Prompt Attack), validação de entrada, menor privilégio para ferramentas, não colocar segredos no prompt.
- Limitação: o modelo é **estocástico** (a resposta pode variar) e **não lembra** interações anteriores a menos que o histórico seja enviado no prompt.

Amazon Bedrock Prompt Management

Permite **criar, editar, salvar e reutilizar prompts**. Conceitos: **variáveis** (placeholders preenchidos no teste ou na execução), **variantes** do prompt (mensagem, modelo ou configuração de inferência diferentes, comparáveis lado a lado), **prompt builder** (interface visual no console) e **versões** (snapshots salvos para implantar de forma controlada). Integra-se à aplicação na inferência ou por um nó de prompt no **Amazon Bedrock Flows**. Também permite ativar prompt caching.

> **Armadilha:** "versionar e gerenciar prompts de forma centralizada" → **Prompt Management**. Não confunda com **Model Cards** (documentação de modelo) nem com **Guardrails** (segurança).

3.3 Treinamento e fine-tuning de FMs

Etapas e métodos

Etapa / método	Dados	O que faz
Pre-training	Enormes volumes de dados **não rotulados** (aprendizado autossupervisionado)	Cria o FM de base; o mais caro.
Continued pre-training (continuous)	Dados **não rotulados** do domínio	Continua o pré-treino para o modelo aprender o vocabulário/conhecimento de um domínio.
Fine-tuning (supervisionado)	Dados **rotulados** (pares entrada/saída)	Ajusta os pesos para uma tarefa específica. É uma forma de **transfer learning**.
Instruction tuning	Pares instrução/resposta	Ensina o modelo a seguir instruções.
Reinforcement fine-tuning (Bedrock)	Prompts + **funções de recompensa** (via AWS Lambda)	Aprende com notas de feedback em vez de pares rotulados.
RLHF	Feedback humano (preferências/rankings)	Alinha o modelo às preferências humanas (utilidade, segurança); treina-se um modelo de recompensa.
Distillation	Prompts do seu caso de uso (respostas geradas pelo teacher)	Transfere conhecimento do modelo grande (teacher) para o menor (student).

> **Fine-tuning vs continued pre-training:** rotulados e tarefa específica → **fine-tuning**. Muito texto bruto do domínio, sem rótulo → **continued pre-training**. **Transfer learning** = reaproveitar o que um modelo pré-treinado já aprendeu em uma nova tarefa.

Preparação dos dados para fine-tuning

Aspecto	O que garantir
Curadoria (data curation)	Dados limpos, sem duplicatas e ruído, de boa qualidade. Qualidade importa mais que volume.
Governança	Privacidade, licenças, conformidade, proveniência e controle de acesso aos dados.
Tamanho	Quantidade suficiente de exemplos para a tarefa; muito pouco não ensina, e o modelo pode sofrer overfitting.
Rotulagem	Rótulos corretos e consistentes (ex.: com **SageMaker Ground Truth**, humanos no ciclo).
Representatividade	Dados cobrem a diversidade real de casos e grupos, evitando viés.
Formato	No Bedrock, dados de treino em arquivo (JSON Lines) no **Amazon S3**.
RLHF	Exige dados de preferência humana, usados para treinar o modelo de recompensa.

> **Armadilha: "Garbage in, garbage out"**: dados ruidosos, enviesados ou não representativos degradam o fine-tuning. Fine-tuning com dados sensíveis exige governança.

Amazon SageMaker AI e serviços relacionados

O **Amazon SageMaker AI** cobre o ciclo completo de ML (preparar, construir, treinar, implantar, monitorar). O **Amazon SageMaker JumpStart** oferece modelos pré-treinados e FMs que podem ser **implantados e ajustados** no SageMaker.

Preciso de...	Use
Acesso gerenciado a FMs por API, sem infraestrutura	**Amazon Bedrock**
Treinar/construir seu próprio modelo, ML tradicional, controle total	**Amazon SageMaker AI**
Implantar/ajustar rápido um modelo pré-treinado/open source em instância própria	**SageMaker JumpStart**
Rotular dados com humanos no ciclo	**SageMaker Ground Truth**
Preparar/transformar dados visualmente	**SageMaker Data Wrangler** (também AWS Glue DataBrew)
Detectar viés e explicar previsões	**SageMaker Clarify**
Detectar drift em produção	**SageMaker Model Monitor**
Orquestrar o fluxo de ML (CI/CD)	**SageMaker Pipelines**
Documentar uso pretendido e limitações do modelo	**SageMaker Model Cards**
ML sem código para analistas	**SageMaker Canvas**

Armadilha: Clarify = viés e explicabilidade (antes e depois do treino). **Model Monitor** = mudança ao longo do tempo em produção. A documentação indica que o Clarify não está mais aberto a novos clientes, mas segue como conceito de prova. Em dúvida entre Bedrock e SageMaker, "foundation model pronto + app GenAI rápido" aponta para **Bedrock**; "treinar algoritmo/modelo próprio" aponta para **SageMaker**.

Assistentes e outros serviços de IA da AWS (para contraste)

Serviço	Função / palavra-gatilho
Amazon Q Business	Assistente de IA generativa sobre dados corporativos, com respostas que respeitam permissões e trazem citações; conecta a S3, SharePoint, Salesforce, etc. A documentação atual informa que não está mais aberto a novos clientes e cita o **Amazon Quick** como alternativa; Quick está no escopo da prova.
Amazon Q Developer	Assistente voltado ao desenvolvimento de software (sugestões de código, transformação). Conceito de assistente de código.
Kiro	Ferramenta de desenvolvimento de software com IA, listada no escopo (categoria Developer Tools).
Amazon Comprehend	NLP: sentimento, entidades, idioma, PII em texto.
Amazon Rekognition	Análise de imagem e vídeo (objetos, rostos, moderação).
Amazon Textract	Extrai texto, tabelas e formulários de documentos digitalizados.
Amazon Transcribe	Fala para texto.
Amazon Polly	Texto para fala.
Amazon Translate	Tradução automática.
Amazon Lex	Chatbots conversacionais por voz e texto.
Amazon Personalize	Recomendações personalizadas.
AWS Transform	Serviço no escopo (modernização/transformação de cargas), citar apenas como nome.

> Esses serviços de IA **pré-treinados** resolvem tarefas específicas sem treinar modelo. Quando o cenário é **geração aberta de texto/imagem**, pense em **Bedrock**.

3.4 Avaliação do desempenho de FMs

Abordagens de avaliação

Abordagem	Descrição
Avaliação humana (human-in-the-loop)	Pessoas avaliam utilidade, coerência, correção e tom. Padrão-ouro, porém lenta e cara. No Bedrock, a equipe pode ser de funcionários ou especialistas do setor.
Benchmark datasets	Conjuntos de dados padronizados com respostas de referência para comparar modelos de forma objetiva.
Avaliação automática (programática)	O Bedrock calcula métricas com datasets integrados ou seus próprios prompts.
LLM-as-a-judge	Um segundo LLM (modelo juiz) pontua as respostas e explica cada nota. Métricas integradas do Bedrock: Correctness, Completeness, Faithfulness, Helpfulness, Logical coherence, Relevance, Following instructions, Professional style and tone, Harmfulness, Stereotyping, Refusal. Aceita métricas customizadas.
Avaliação de RAG	Avalia knowledge bases (recuperação e geração) com dataset de perguntas e **ground truth**.

> **Amazon Bedrock Model Evaluation** (Bedrock evaluations): avalia modelos e knowledge bases do Bedrock e também modelos/fontes RAG fora do Bedrock. Tipos de job: **programático**, com **trabalhadores humanos**, com **modelo juiz** e **RAG**. Suporta FMs, modelos customizados e importados, prompt routers e modelos com Provisioned Throughput. Serve para **comparar modelos antes de escolher**.

Métricas

Métrica	Mede	Uso típico	Limitação
ROUGE (Recall-Oriented Understudy for Gisting Evaluation)	Sobreposição de n-gramas/sequências (foco em **recall**) com um texto de referência	**Resumo** automático	Superfície do texto; ignora sinônimos e semântica
BLEU (Bilingual Evaluation Understudy)	Sobreposição de n-gramas (foco em **precisão**) com a referência	**Tradução** automática	Penaliza paráfrases válidas
BERTScore	**Similaridade semântica** usando embeddings	Quando as palavras diferem mas o sentido deve ser o mesmo	Mais computação; depende do modelo de embeddings
Perplexity	Quão "surpreso" o modelo fica com o texto; menor é melhor	Fluência de modelos de linguagem	Não mede factualidade nem utilidade
LLM-as-a-judge	Notas de um LLM em critérios (correção, coerência, relevância...)	Avaliação em escala com custo menor que a humana	Depende da qualidade do juiz

> **Armadilha: ROUGE** = resumo/recall; **BLEU** = tradução/precisão; **BERTScore** = significado (semântica). Todos comparam com uma **referência**. Nenhum substitui a avaliação humana para qualidade subjetiva.

- **Métricas de classificação** (quando a saída é certo/errado): **accuracy**, **precision** (custo de falso positivo alto), **recall** (custo de falso negativo alto, ex.: fraude), **F1** (equilíbrio) e **AUC-ROC**.

Avaliar se o FM atende aos objetivos de negócio

Métrica de negócio	Exemplo
Produtividade	Tempo economizado por tarefa, volume de tickets resolvidos por agente.
Engajamento do usuário	Tempo de uso, retorno de usuários, interações por sessão.
Taxa de conclusão de tarefas (task completion rate)	Percentual de pedidos resolvidos sem intervenção humana.
Satisfação do usuário	CSAT, NPS, avaliações positivas das respostas.
Custo por interação	Custo de tokens e infraestrutura por conversa atendida.
Outros	Taxa de conversão, receita média por usuário, valor do tempo de vida do cliente, eficiência (custo e tempo para atingir o desempenho-alvo). **Testes A/B** comparam versões de prompt/modelo contra um KPI.

Avaliar aplicações construídas com FMs (RAG, agentes, workflows)

- **RAG:** avaliar **recuperação** (os documentos certos foram trazidos? relevância do contexto) e **geração** (a resposta é fiel ao contexto? responde à pergunta? correção e completude). Use a avaliação de RAG do Bedrock e o **contextual grounding check** do Guardrails em produção.
- **Agentes:** avaliar se concluem a tarefa, se escolhem as ações/ferramentas certas, o número de etapas, a latência e o custo; use **traces** para inspecionar o raciocínio.
- **Workflows** (ex.: Amazon Bedrock Flows): avaliar ponta a ponta e por etapa (prompt, recuperação, ferramenta).
- Combine métricas de qualidade com **latência, taxa de erro e custo por interação**.

> **Armadilha:** qualidade do modelo (ROUGE, BERTScore) **não é** o mesmo que **valor de negócio** (satisfação, conclusão de tarefa, custo por interação). A pergunta "o FM atinge o objetivo de negócio?" pede métricas de negócio.

📋 Resumo rápido do Domínio 3

- **Escada de personalização**: prompt engineering → RAG → fine-tuning → continued pre-training → pre-training. Do mais barato ao mais caro; comece pelo mais simples.
- **Seleção de FM**: custo, modalidade, latência, multilíngue, tamanho, complexidade, customização, tamanho de entrada/saída, prompt caching.
- **Inferência**: temperature baixa = determinístico; alta = criativo. Top K/Top P limitam candidatos. Max tokens, stop sequences e penalidades controlam o tamanho.
- **RAG** = chunks + embeddings + vector DB + busca por similaridade; sem retreinar. **Knowledge Bases** = RAG gerenciado. Vetores: OpenSearch, Aurora/RDS PostgreSQL (pgvector), Neptune, S3 Vectors.
- **Agentes** = FM + etapas + APIs (action groups). **Guardrails** = filtros de conteúdo, tópicos negados, palavras, PII, grounding, Automated Reasoning.
- **Preço Bedrock**: on-demand (tokens) vs Provisioned Throughput (por hora, 1 ou 6 meses; obrigatório para modelo customizado). Batch para offline. Prompt caching reduz custo/latência.
- **Prompts**: zero-shot, single-shot, few-shot, chain-of-thought, templates, negative prompts. Riscos: exposição, poisoning, hijacking, jailbreaking. **Prompt Management** = variáveis, variantes, versões.
- **Treinamento**: pre-training (não rotulado), continued pre-training (não rotulado do domínio), fine-tuning (rotulado), RLHF (feedback humano), distillation (teacher → student). Dados: curadoria, governança, tamanho, rotulagem, representatividade.
- **SageMaker** = ciclo de ML completo (Ground Truth, Data Wrangler, Clarify, Model Monitor, Pipelines, Model Cards, JumpStart, Canvas). **Bedrock** = FMs gerenciados por API.
- **Avaliação**: humana, benchmark, Bedrock Model Evaluation, LLM-as-a-judge. ROUGE (resumo), BLEU (tradução), BERTScore (semântica). Métricas de negócio: conclusão de tarefa, satisfação, custo por interação, produtividade, engajamento.

> **Armadilhas:** (1) usar fine-tuning quando o problema é dado que muda (use RAG). (2) achar que temperature corrige alucinação. (3) esquecer o Provisioned Throughput para modelo customizado. (4) trocar ROUGE e BLEU (resumo vs tradução). (5) confundir Clarify (viés/explicação) com Model Monitor (drift). (6) tratar métricas de qualidade como prova de valor de negócio.

Domínio 4: Diretrizes para IA Responsável (14%)

Tarefas oficiais: 4.1 explicar o desenvolvimento de sistemas de IA responsáveis · 4.2 reconhecer a importância de modelos transparentes e explicáveis.

4.1 Desenvolvimento de sistemas de IA responsáveis

Dimensões de IA responsável na AWS

A AWS organiza a IA responsável em oito dimensões centrais. A prova pode listar várias e pedir que você classifique um cenário.

Dimensão	Ideia central (AWS)	Exemplo de cenário
Fairness (equidade)	Considerar os impactos sobre diferentes grupos de partes interessadas.	Modelo de crédito aprova menos um grupo demográfico.
Explainability	Entender e avaliar as saídas do sistema.	Explicar ao cliente por que o empréstimo foi negado.
Privacy and Security	Obter, usar e proteger dados e modelos de forma adequada.	Dados pessoais vazando em respostas.
Safety	Evitar saídas nocivas e o uso indevido.	Chatbot gera conteúdo tóxico ou perigoso.
Controllability	Ter mecanismos para monitorar e direcionar o comportamento do sistema.	Humano pode intervir, corrigir ou desligar o sistema.
Veracity and Robustness	Saídas corretas, mesmo com entradas inesperadas ou adversariais.	Alucinações; resposta quebra com entrada atípica.
Governance	Incorporar boas práticas na cadeia de suprimentos de IA, incluindo provedores e implantadores.	Políticas, auditoria, aprovação antes de produção.
Transparency	Permitir que as partes interessadas façam escolhas informadas sobre o uso do sistema.	Avisar o usuário que está falando com uma IA e documentar limitações.

Características de IA responsável: termos da prova

- **Bias** (viés): resultados sistematicamente desiguais para certos grupos, quase sempre originados nos dados.
- **Fairness** e **inclusivity**: o sistema funciona bem para populações diversas, sem excluir ninguém.
- **Robustness**: desempenho confiável sob entradas variadas ou adversariais.
- **Safety**: evitar dano. **Veracity**: a saída é verdadeira (o oposto de alucinação).

Ferramentas para promover IA responsável

Preocupação	Ferramenta AWS	Observação
Conteúdo nocivo, tópicos proibidos, PII, alucinação em tempo de execução	**Amazon Bedrock Guardrails**	Filtra entradas e respostas. Veja detalhes abaixo.
Viés em dados ou modelo; explicar previsões	**Amazon SageMaker Clarify**	Viés pré e pós-treinamento; atribuição de features (valores de Shapley).
Mudança em produção (qualidade, viés, atribuição)	**Amazon SageMaker Model Monitor**	Monitora qualidade de dados, qualidade do modelo, bias drift e feature attribution drift.
Revisão humana de previsões incertas	**Amazon Augmented AI (Amazon A2I)**	Fluxos de revisão humana de previsões de ML, com vários revisores.
Rotulagem e feedback humano no ciclo de ML	**SageMaker Ground Truth**	Aplica feedback humano ao longo do ciclo.
Comparar e avaliar modelos de fundação	**Amazon Bedrock Evaluations** (Model Evaluation)	Avaliação automática, com juiz LLM ou com revisores humanos.
Documentar uso pretendido, risco e treino	**SageMaker Model Cards**	Também **AWS AI Service Cards** (do lado da AWS).
Visibilidade e controle sobre modelos de ML	**ML Governance do SageMaker**	Ferramentas de governança do SageMaker, como os Model Cards e o Model Registry.

Bedrock Guardrails: o que filtra (verificado na documentação). Content filters (Hate, Insults, Sexual, Violence, Misconduct e Prompt Attack, com força configurável); **Denied topics**; **Word filters** (palavras e frases personalizadas, ex.: nomes de concorrentes); **Sensitive information filters** (bloqueiam ou mascaram PII, com regex personalizado); **Contextual grounding checks** (detectam alucinação em respostas não fundamentadas na fonte ou irrelevantes, útil em RAG); **Automated Reasoning checks** (validam respostas contra regras lógicas). Avaliam **entradas e saídas**, podem ser usados com a API `ApplyGuardrail` sem invocar um modelo, e você configura a mensagem devolvida ao violar um filtro.

Seleção responsável de modelos: sustentabilidade

- Treinar e hospedar modelos grandes consome muita energia e computação. Prefira o **modelo menor adequado à tarefa** (right-sizing), evite retreinar sem necessidade e use customização leve em vez de treinar do zero.
- Use hardware eficiente (ex.: AWS Trainium e AWS Inferentia) e serviços gerenciados/serverless para reduzir desperdício.
- O **AWS Well-Architected Generative AI Lens** inclui o pilar Sustainability: minimizar recursos computacionais em treinamento, customização, hospedagem, processamento e armazenamento de dados.

Armadilha: "treinar um modelo maior" raramente é a resposta em perguntas de sustentabilidade ou custo.

Riscos legais de IA generativa

Risco	Descrição	Mitigação
Violação de propriedade intelectual	Dados de treino ou saídas que infringem direitos autorais ou licenças.	Dados licenciados ou próprios, revisão das licenças dos modelos, revisão humana.
Saídas enviesadas	Discriminação de grupos, com risco regulatório e reputacional.	Dados diversos, Clarify, auditorias humanas, análise por subgrupo.
Perda de confiança do cliente	Erros públicos, vazamentos ou conteúdo impróprio.	Guardrails, transparência, testes antes de publicar.
Risco ao usuário final	Decisões erradas em saúde, finanças, jurídico.	Humano no circuito (A2I), avisos, limites de escopo.
Alucinações	Respostas plausíveis, porém falsas.	RAG, grounding checks, validação de saída, revisão humana.

Características de bons conjuntos de dados

- **Inclusivos e diversos**: representam os grupos afetados.
- **Fontes curadas**: origem conhecida, licenciada e de qualidade.
- **Balanceados**: classes e grupos sem sub-representação grave.
- **Rótulos de qualidade**: rótulos errados ou enviesados contaminam o modelo.

Efeitos de viés (bias) e variância (variance)

Conceito	O que é	Efeito
Viés alto (high bias)	Modelo simples demais, suposições rígidas.	**Underfitting**: erra no treino e no teste.
Variância alta (high variance)	Modelo sensível demais aos detalhes do treino.	**Overfitting**: ótimo no treino, ruim em dados novos.
Viés nos dados (sentido de equidade)	Dados que sub-representam ou distorcem grupos.	Inexatidão e resultados desiguais entre grupos demográficos.

> **Armadilha:** "bias" tem dois sentidos na prova: **bias/variance** (erro do modelo, underfitting) e **viés/equidade** (discriminação de grupos). Leia o contexto.

Detectar e monitorar viés, confiabilidade (trustworthiness) e veracidade

- **Analisar a qualidade dos rótulos** antes de treinar.
- **Análise por subgrupo**: comparar métricas entre grupos demográficos, não só a média global.
- **Auditorias humanas** e avaliação humana (Bedrock Evaluations, Ground Truth, A2I).
- **Clarify** (viés em dados e modelo) e **Model Monitor** (deriva em produção).
- **Avaliação de veracidade**: comparar respostas a fontes confiáveis, grounding checks e métricas de avaliação de modelo.

4.2 Modelos transparentes e explicáveis

	Transparente e explicável	Não transparente (caixa-preta)
Exemplos	Regressão linear, árvores de decisão; modelos open source com dados e licença documentados.	Redes profundas e grandes modelos de fundação.
Vantagem	Dá para entender por que decidiu; mais fácil auditar.	Costuma ter desempenho maior em dados complexos e não estruturados.
Risco	Pode ter menor desempenho em problemas complexos.	Difícil explicar, auditar e depurar.

Ferramentas e práticas de transparência

- **SageMaker Model Cards**: documentam em um só lugar uso pretendido e não pretendido, classificação de risco (unknown, low, medium, high), detalhes de treino, métricas e resultados de avaliação; geram versões imutáveis das edições e podem ser exportadas em PDF.
- **AWS AI Service Cards**: documentação da AWS sobre uso pretendido, limitações, decisões de design de IA responsável e boas práticas de otimização de seus serviços e modelos. Diferença: Service Cards descrevem serviços/modelos *da AWS*; Model Cards documentam *seus* modelos.
- **Amazon Bedrock Evaluations**: compara modelos com avaliação automática, juiz LLM ou revisores humanos, para escolher com base em evidências.
- **Modelos open source, dados e licenciamento**: transparência sobre origem, termos de uso e dados de treino.
- **SageMaker Clarify**: explica previsões por importância de features (valores de Shapley) e gera relatórios para equipes de risco e reguladores.

> Nota de disponibilidade: a documentação atual informa que Clarify e Model Monitor não estão abertos a novos clientes (clientes existentes seguem normalmente). Para a prova, os conceitos e a associação "viés/explicabilidade e deriva" continuam valendo.

Trade-off entre segurança, transparência e desempenho

- Modelos mais **interpretáveis** tendem a sacrificar desempenho; modelos de alto desempenho tendem a ser menos explicáveis. Em contextos regulados (crédito, saúde, RH), explicabilidade pode pesar mais que alguns pontos de acurácia.
- Segurança e transparência também se tensionam: expor demais o funcionamento ou os dados pode facilitar ataques ou vazar informação sensível. Avalie **interpretabilidade e desempenho** em conjunto e escolha conforme o risco do caso de uso.

Design centrado no ser humano para IA explicável

- **Mecanismos de feedback do usuário**: polegar para cima/baixo, relato de erros, revisão humana (A2I).
- **Transparência nas decisões da IA**: informar que é IA, mostrar fontes, explicar fatores, indicar incerteza.
- Envolver usuários e partes interessadas durante todo o ciclo de vida, não só no fim.

📋 Resumo rápido do Domínio 4

- 8 dimensões AWS: Fairness, Explainability, Privacy and Security, Safety, Controllability, Veracity and Robustness, Governance, Transparency.
- Bedrock Guardrails = filtro em tempo de execução (conteúdo nocivo, tópicos, palavras, PII, grounding, Automated Reasoning). Clarify = viés + "por quê". Model Monitor = deriva em produção. A2I = revisão humana. Model Cards = documentação.
- Dados bons: diversos, curados, balanceados, com rótulos de qualidade. Use análise por subgrupo e auditorias humanas.
- Viés alto = underfitting; variância alta = overfitting.
- Riscos legais de GenAI: IP, saídas enviesadas, perda de confiança, risco ao usuário final, alucinações.
- Interpretabilidade versus desempenho: simples = explicável; profundo = poderoso e opaco.
- Sustentabilidade: modelo do tamanho certo, hardware eficiente, evitar retreino.

> **Armadilhas:** (1) Clarify, Guardrails e A2I não são sinônimos. (2) Model Cards documentam seus modelos; AI Service Cards documentam serviços da AWS. (3) "Bias" pode ser viés de equidade ou bias/variance. (4) Modelo maior nem sempre é a resposta.

Domínio 5: Segurança, Conformidade e Governança para Soluções de IA (14%)

Tarefas oficiais: 5.1 explicar métodos para proteger sistemas de IA · 5.2 reconhecer regulamentações e requisitos de governança e conformidade para sistemas de IA.

5.1 Proteger sistemas de IA

Serviços e recursos de segurança aplicados a IA

Recurso	Papel em soluções de IA
AWS IAM (roles, policies, permissions)	Controle de acesso com **privilégio mínimo**: quem pode invocar modelos, acessar dados de treino e alterar configurações. No Bedrock, é possível restringir ações a modelos e recursos específicos.
Criptografia (AWS KMS)	Em repouso: dados de treino e validação no S3, modelos customizados, agentes, ingestão de bases de conhecimento e avaliações podem usar chaves KMS. Em trânsito: TLS (o Bedrock exige TLS 1.2 e recomenda 1.3).
Amazon Macie	Descobre e ajuda a proteger dados sensíveis (como PII) armazenados no Amazon S3, útil antes de usar esses dados em treino, fine-tuning ou RAG.
AWS PrivateLink e **Amazon VPC**	Endpoint de interface (VPC endpoint) para acessar o Bedrock de forma privada, sem passar pela internet pública; VPC Flow Logs monitoram o tráfego.
AWS Secrets Manager	Guarda credenciais e chaves de API usadas por aplicações e agentes (e vector stores de terceiros).
Amazon Bedrock Guardrails	Filtra entradas e saídas: conteúdo nocivo, ataques de prompt, tópicos negados, PII, alucinação.
Amazon Bedrock AgentCore Identity	Gerencia identidade, acesso e autenticação de agentes, compatível com provedores de identidade existentes (ex.: Amazon Cognito, Okta).
Policy in AgentCore	Controle determinístico: define quais ferramentas o agente pode usar, quais ações e sob que condições, interceptando cada chamada de ferramenta via AgentCore Gateway.

> **Armadilha:** o Bedrock é gerenciado, mas **IAM, chaves KMS, configuração de rede e de Guardrails continuam sendo do cliente**.

Modelo de responsabilidade compartilhada para IA

	AWS	Cliente
Escopo	Segurança *da* nuvem: infraestrutura global, hardware, serviço gerenciado e plataforma de modelos.	Segurança *na* nuvem: IAM, classificação e proteção de dados, escolhas de criptografia, rede, configuração de Guardrails e uso responsável.
No Bedrock	Opera a infraestrutura e mantém as contas de implantação dos modelos; os provedores de modelo não têm acesso a prompts e respostas dos clientes.	Gerencia seus dados, permissões, logs, chaves e a forma como a aplicação usa o modelo.

Origem dos dados, citação de fontes e catalogação

- **Source citation**: indicar de onde veio a informação (ex.: citações em respostas RAG) para o usuário verificar.
- **Data lineage**: rastrear a origem e as transformações dos dados até o modelo. **Data cataloging**: inventário organizado e pesquisável de conjuntos de dados (ex.: AWS Glue Data Catalog).
- **Model Cards**: documentam treino, métricas, uso pretendido e risco do modelo, apoiando auditoria.

Boas práticas de engenharia de dados segura

- **Qualidade de dados**: completude, consistência, precisão e atualidade antes do treino.
- **Tecnologias de preservação de privacidade** (privacy-enhancing technologies): anonimização, mascaramento, pseudonimização, minimização de dados.
- **Controle de acesso aos dados**: privilégio mínimo, segregação de ambientes (ex.: AWS Lake Formation em lagos de dados).
- **Integridade dos dados**: evitar adulteração e envenenamento de dados (data poisoning), com validação e versionamento.

Considerações de segurança e privacidade

Tema	Ameaça ou requisito	Mitigação
Segurança de aplicação, proteção de infraestrutura	Aplicação e endpoints expostos.	IAM, VPC, PrivateLink, WAF no front-end; modelo de responsabilidade compartilhada.
Detecção de ameaças, gerenciamento de vulnerabilidades	Atividade suspeita; falhas conhecidas em software e imagens.	Monitoramento e achados de segurança; **Amazon Inspector** para varredura de vulnerabilidades.
Prompt injection	Entrada maliciosa tenta sobrescrever as instruções do sistema.	Filtro Prompt Attack dos Guardrails, validação de entradas, privilégio mínimo para agentes.
Criptografia em repouso e em trânsito	Interceptação ou acesso indevido.	KMS (repouso) e TLS (trânsito).
Vazamento de dados (data leakage)	Modelo revela PII ou dados confidenciais.	Filtros de informação sensível, minimização de dados, Macie, IAM.
Filtragem e validação de saída; toxicidade	Respostas nocivas, tóxicas ou fora do escopo.	Guardrails, validação programática da saída, revisão humana.
Trilha de auditoria e logs de interações de IA	Provar quem chamou o quê e o que foi perguntado e respondido.	**CloudTrail** (quem e quando) + **model invocation logging** do Bedrock (conteúdo, para S3 e/ou CloudWatch Logs).

> **Invocation logging:** vem **desativado por padrão**. Quando ativado, registra requisições, respostas e metadados das chamadas Converse, ConverseStream, InvokeModel e InvokeModelWithResponseStream, com destino Amazon S3 e/ou CloudWatch Logs (mesma conta e Região).

Necessidade	Resposta
Impedir saída nociva em tempo real	Bedrock Guardrails (prevenção)
Registrar o que foi perguntado e respondido	Model invocation logging (registro)
Registrar quem chamou qual API e quando	AWS CloudTrail (auditoria)
Métricas, alarmes e logs operacionais	Amazon CloudWatch (monitoramento)

Detecção de alucinação e técnicas de grounding

- **RAG grounding**: ancorar a resposta em documentos recuperados (Amazon Bedrock Knowledge Bases).
- **Contextual grounding checks** e **Automated Reasoning checks** (Guardrails): detectam respostas não fundamentadas na fonte ou que violam regras lógicas.
- **Output validation**: validar formato e fatos por regras ou por outro modelo (juiz).
- **Confidence scoring**: usar a confiança para encaminhar casos incertos à revisão humana (A2I).

5.2 Governança e conformidade

Serviços AWS para governança e conformidade

Serviço	Para que serve	Pergunta que responde
AWS Artifact	Portal de autoatendimento para relatórios de conformidade da AWS e acordos.	"Onde baixo a prova de que a AWS atende ao padrão X?"
AWS CloudTrail	Registra chamadas de API e atividade na conta.	Quem fez o quê e quando?
AWS Config	Avalia e registra configurações de recursos e conformidade com regras.	Como o recurso está configurado e o que mudou?
Amazon Inspector	Varredura automatizada de vulnerabilidades.	Há vulnerabilidades conhecidas?
AWS Trusted Advisor	Verificações e recomendações de boas práticas (inclui segurança e custo).	Estou seguindo boas práticas?
Amazon CloudWatch	Métricas, logs e alarmes.	Como está a operação?
AWS Well-Architected Tool e **Generative AI Lens**	Revisão de cargas de trabalho; o Lens estende o framework para GenAI (excelência operacional, segurança, confiabilidade, desempenho, custo, sustentabilidade).	Minha carga de GenAI segue boas práticas?

> AWS Audit Manager (auditoria contínua) aparece em material de estudo, mas não consta da lista oficial de serviços no escopo desta prova.

> **Armadilha: CloudTrail** = quem chamou qual API. **Config** = estado e mudanças de configuração. **Artifact** = relatórios de conformidade da AWS. Não são intercambiáveis.

Estratégias de governança de dados

Tema	O que fazer
Ciclo de vida dos dados	Definir coleta, uso, armazenamento, arquivamento e descarte (ex.: S3 e S3 Glacier por camadas).
Logging	Registrar acesso e uso dos dados e das interações com modelos.
Residência de dados	Escolher Regiões AWS para manter dados dentro da jurisdição exigida. Atenção a inferência entre Regiões nos recursos que a oferecem.
Monitoramento e observação	Acompanhar qualidade, deriva, acesso e anomalias de forma contínua.
Retenção	Manter dados e logs pelo tempo exigido por política ou lei, e não além.

Processos de governança

- **Políticas** claras de uso aceitável de IA, dados e modelos; **cadência de revisão** periódica e **estratégias de revisão** (incluindo revisão humana).
- **Transparência**: padrões de documentação e divulgação (Model Cards, AI Service Cards).
- **Treinamento de equipes** em uso responsável, segurança e conformidade.
- **Frameworks de governança**, como o **Generative AI Security Scoping Matrix**.
- Conformidade herdada: serviços AWS possuem certificações como ISO e SOC (relatórios no AWS Artifact), mas configurar seu uso de forma conforme é responsabilidade do cliente.

Generative AI Security Scoping Matrix

Escopo	Descrição	Propriedade
1: Consumer app	Uso de serviço público de GenAI de terceiros (ex.: chat em site público).	Menor
2: Enterprise app	Aplicação corporativa de terceiros com recursos de IA embutidos e termos empresariais.	
3: Pre-trained models	Aplicação própria sobre modelo de fundação de terceiros via API (ex.: chatbot com RAG).	
4: Fine-tuned models	Modelo de fundação refinado com dados do negócio.	
5: Self-trained models	Modelo treinado do zero com dados próprios ou adquiridos.	Maior

Disciplinas de segurança aplicadas em todos os escopos: **Governance and Compliance**, **Legal and Privacy**, **Risk Management**, **Controls** e **Resilience**. Quanto maior o escopo, maior a sua responsabilidade.

Governança de IA no SageMaker

- **SageMaker Model Cards**: documentação de governança, integradas ao **Model Registry** (versões de modelos e fluxo de aprovação).
- **Clarify** gera relatórios de governança de modelo para equipes de risco e reguladores.

📋 Resumo rápido do Domínio 5

- Pilha padrão de segurança: IAM (privilégio mínimo) + KMS (criptografia) + PrivateLink/VPC (tráfego privado) + CloudTrail (auditoria) + Macie (dados sensíveis no S3) + Guardrails (filtros).
- Prevenir = Guardrails. Registrar conteúdo = invocation logging (desligado por padrão). Registrar quem chamou = CloudTrail.
- Responsabilidade compartilhada: AWS protege a nuvem; você protege IAM, dados, chaves, rede e configuração de Guardrails, mesmo no Bedrock.
- Prompt injection: filtro Prompt Attack, validação de entrada, privilégio mínimo para agentes (AgentCore Identity e Policy).
- Alucinação: RAG grounding, grounding checks, validação de saída, confiança e revisão humana.
- Conformidade: Artifact (relatórios), Config (configurações), CloudTrail (API), Inspector (vulnerabilidades), Trusted Advisor (boas práticas).
- Governança de dados: ciclo de vida, residência (Regiões), retenção, logging, linhagem e catálogo.
- Security Scoping Matrix: 5 escopos (consumer, enterprise, pre-trained, fine-tuned, self-trained) e 5 disciplinas.

> **Armadilhas:** (1) "A AWS cuida de tudo no Bedrock" é falso. (2) Guardrails previne; logging registra. (3) CloudTrail não mostra o conteúdo dos prompts. (4) Artifact fornece relatórios da AWS, não audita a sua configuração (isso é Config).

Revisão Final

Use este capítulo na véspera da prova: pares confundíveis dos 5 domínios, armadilhas que custam pontos e um autoteste para explicar em voz alta.

Tabela de contraste: pares confundíveis

Par	Distinção	Domínio
IA vs ML vs Deep Learning vs GenAI	Aninhados: IA ⊃ ML ⊃ Deep Learning ⊃ GenAI. Saída é rótulo ou número: ML tradicional. Cria conteúdo novo: GenAI.	1, 2
Supervisionado vs não supervisionado vs por reforço	Com rótulos, prevê. Sem rótulos, encontra estrutura (clusters). Agente com recompensa e penalidade aprende política.	1
Overfitting vs underfitting	Overfitting (variância alta): ótimo no treino, ruim em dados novos. Underfitting (viés alto): ruim nos dois.	1, 4
Precision vs recall	Precision: evitar falsos positivos. Recall: evitar falsos negativos (ex.: fraude, doença).	1, 3
Prompt engineering vs RAG vs fine-tuning	Prompt: sem dados, mais barato. RAG: fatos atuais ou privados, sem mudar pesos. Fine-tuning: estilo, formato ou domínio, com dados rotulados e pesos alterados.	2, 3
Temperature vs top-p vs top-k	Temperature: aleatoriedade geral (0 = determinístico). Top-p: conjunto dinâmico por probabilidade acumulada. Top-k: número fixo de candidatos.	2
Amazon SageMaker vs Amazon Bedrock	SageMaker: construir, treinar e implantar modelos com controle total. Bedrock: usar modelos de fundação via API, gerenciado, sem infraestrutura.	3
Bedrock Knowledge Bases vs Agents	Knowledge Bases: RAG gerenciado. Agents: executam tarefas em várias etapas e chamam APIs.	3
Amazon Q Business vs Q Developer	Q Business: assistente sobre dados da empresa. Q Developer: assistente para código.	3
BLEU/ROUGE vs BERTScore	Sobreposição superficial de n-gramas vs similaridade semântica por embeddings.	3
Clarify vs Model Monitor vs Guardrails	Clarify: viés e explicabilidade (dados e modelo). Model Monitor: deriva em produção. Guardrails: filtro de entradas e saídas em tempo de execução.	4
A2I vs Ground Truth	A2I: revisão humana de previsões em produção. Ground Truth: rotulagem de dados de treino com humanos.	3, 4
Guardrails vs Bedrock Evaluations	Guardrails: proteção em tempo de execução. Evaluations: comparação e avaliação de modelos antes de escolher.	3, 4
Model Cards vs AI Service Cards	Model Cards (SageMaker): documentam seus modelos. AI Service Cards: transparência da AWS sobre seus serviços e modelos.	4
Fairness vs explainability vs transparency	Fairness: impacto sobre grupos. Explainability: entender e avaliar saídas. Transparency: stakeholders decidirem com informação.	4
Modelo interpretável vs caixa-preta	Interpretável: explicável, menos desempenho. Caixa-preta: alto desempenho, difícil de explicar.	4

Prevenção vs registro	Guardrails previne. Invocation logging registra conteúdo. CloudTrail registra quem chamou a API.	5
CloudTrail vs Config vs CloudWatch	CloudTrail: chamadas de API. Config: configuração e mudanças dos recursos. CloudWatch: métricas, logs e alarmes.	5
AWS Artifact vs AWS Config	Artifact: relatórios de conformidade da AWS. Config: conformidade da sua configuração.	5
Macie vs Inspector	Macie: dados sensíveis no S3. Inspector: vulnerabilidades de software.	5
KMS vs TLS	KMS: chaves para criptografia em repouso. TLS: criptografia em trânsito.	5
IAM vs PrivateLink	IAM: quem pode fazer o quê. PrivateLink: caminho de rede privado, sem internet pública.	5
Responsabilidade AWS vs cliente	AWS: segurança da nuvem. Cliente: IAM, dados, chaves, rede e Guardrails, mesmo em serviço gerenciado.	5

10 armadilhas que custam pontos

1. **ML nem sempre é a resposta.** Regra simples e determinística não precisa de ML.
2. **RAG vs fine-tuning.** "Dados mudam com frequência" ou "usar documentos internos": RAG. "Tom ou formato permanente": fine-tuning.
3. **Mais gerenciado costuma vencer.** Pedido genérico de modelo de fundação via API: Bedrock. Treino próprio, algoritmo customizado: SageMaker.
4. **Clarify, Guardrails e A2I não são sinônimos.** "IA responsável" não é uma resposta única.
5. **Guardrails previne; logging registra.** Invocation logging é desligado por padrão e não bloqueia nada.
6. **CloudTrail não é Config.** Quem chamou a API vs como o recurso está configurado.
7. **"A AWS cuida de tudo" é quase sempre errado.** IAM, dados, chaves, rede e Guardrails são do cliente.
8. **Dois sentidos de bias.** Bias/variance (erro de modelo) vs viés de equidade. Leia o contexto.
9. **Modelo maior não é sustentável.** Right-sizing, hardware eficiente e evitar retreino são as respostas de sustentabilidade e custo.
10. **Alucinação não é resolvida só mudando o modelo.** Pense em RAG, grounding checks, validação de saída e humano no circuito.

> **Armadilha:** em perguntas com duas respostas plausíveis, escolha a que atende exatamente ao verbo do enunciado: *prevenir*, *registrar*, *explicar*, *monitorar*, *documentar*. Não há penalidade por chute: nunca deixe em branco.

Autoteste: explique em voz alta

1. Qual é a diferença entre aprendizado supervisionado, não supervisionado e por reforço? Dê um exemplo de cada.
2. Em que cenário RAG é a escolha certa e em qual o fine-tuning é melhor? Por que RAG não altera os pesos?

3. Quando escolher o Amazon Bedrock e quando o Amazon SageMaker? Por que o JumpStart confunde essa escolha?
4. O que temperature, top-p e top-k controlam? Qual valor de temperature gera respostas mais previsíveis?
5. Explique overfitting e underfitting em termos de viés e variância, e como cada um aparece nas métricas de treino e teste.
6. Diga em uma frase o que fazem Clarify, Model Monitor, Guardrails e A2I.
7. Qual a diferença entre SageMaker Model Cards e AWS AI Service Cards? E entre Guardrails e Bedrock Evaluations?
8. Por que "treinar um modelo maior" pode ser a resposta errada em perguntas de sustentabilidade? Que alternativas existem?
9. Dê três responsabilidades que continuam do cliente ao usar o Bedrock e diga como IAM, KMS e PrivateLink se complementam.
10. Como você provaria a um auditor quem chamou um modelo, o que foi perguntado e que a AWS tem certificação ISO ou SOC? Cite o serviço de cada evidência.

> Respostas de referência para o item 10: CloudTrail (quem e quando), model invocation logging do Bedrock (conteúdo) e AWS Artifact (relatórios de conformidade da AWS).

Fontes

Conteúdo consultado em **22/11/2026**, em páginas públicas do **site oficial da AWS** (aws.amazon.com e docs.aws.amazon.com):

- Guia oficial da prova AIF-C01: https://docs.aws.amazon.com/aws-certification/latest/ai-practitioner-01/ai-practitioner-01.html
- Guia oficial, Domínio 1: https://docs.aws.amazon.com/aws-certification/latest/ai-practitioner-01/ai-practitioner-01-domain1.html
- Guia oficial, Domínio 2: https://docs.aws.amazon.com/aws-certification/latest/ai-practitioner-01/ai-practitioner-01-domain2.html
- Guia oficial, Domínio 3: https://docs.aws.amazon.com/aws-certification/latest/ai-practitioner-01/ai-practitioner-01-domain3.html
- Guia oficial, Domínio 4: https://docs.aws.amazon.com/aws-certification/latest/ai-practitioner-01/ai-practitioner-01-domain4.html
- Guia oficial, Domínio 5: https://docs.aws.amazon.com/aws-certification/latest/ai-practitioner-01/ai-practitioner-01-domain5.html
- Serviços no escopo: https://docs.aws.amazon.com/aws-certification/latest/ai-practitioner-01/aif-01-in-scope-services.html
- What is Artificial Intelligence?: https://aws.amazon.com/what-is/artificial-intelligence/
- What is Machine Learning?: https://aws.amazon.com/what-is/machine-learning/
- Deploy models for inference (SageMaker AI): https://docs.aws.amazon.com/sagemaker/latest/dg/deploy-model.html
- What is Prompt Engineering?: https://aws.amazon.com/what-is/prompt-engineering/
- What are Foundation Models?: https://aws.amazon.com/what-is/foundation-models/
- What is RAG?: https://aws.amazon.com/what-is/retrieval-augmented-generation/
- Inference parameters (Amazon Bedrock): https://docs.aws.amazon.com/bedrock/latest/userguide/inference-parameters.html
- What is MLOps?: https://aws.amazon.com/what-is/mlops/
- Amazon Bedrock pricing: https://aws.amazon.com/bedrock/pricing/
- Amazon Bedrock: https://aws.amazon.com/bedrock/
- Customize your model (Amazon Bedrock): https://docs.aws.amazon.com/bedrock/latest/userguide/custom-models.html
- What is Agentic AI?: https://aws.amazon.com/what-is/agentic-ai/
- Amazon Bedrock AgentCore: https://aws.amazon.com/bedrock/agentcore/
- Amazon Quick: https://aws.amazon.com/quick/
- Amazon Bedrock Guardrails: https://docs.aws.amazon.com/bedrock/latest/userguide/guardrails.html
- Amazon Bedrock evaluations: https://docs.aws.amazon.com/bedrock/latest/userguide/evaluation.html
- Amazon Bedrock Prompt management: https://docs.aws.amazon.com/bedrock/latest/userguide/prompt-management.html
- Amazon Bedrock Knowledge Bases: https://docs.aws.amazon.com/bedrock/latest/userguide/knowledge-base.html

- Amazon Bedrock Provisioned Throughput: https://docs.aws.amazon.com/bedrock/latest/userguide/prov-throughput.html
- Amazon Bedrock prompt caching: https://docs.aws.amazon.com/bedrock/latest/userguide/prompt-caching.html
- Amazon Bedrock Agents: https://docs.aws.amazon.com/bedrock/latest/userguide/agents.html
- Amazon Bedrock prompt engineering concepts: https://docs.aws.amazon.com/bedrock/latest/userguide/prompt-engineering-guidelines.html
- Amazon Q Business: What is: https://docs.aws.amazon.com/amazonq/latest/qbusiness-ug/what-is.html
- Amazon Bedrock Knowledge Bases vector stores: https://docs.aws.amazon.com/bedrock/latest/userguide/knowledge-base-setup.html
- Amazon Bedrock evaluation metrics (judge): https://docs.aws.amazon.com/bedrock/latest/userguide/model-evaluation-metrics.html
- SageMaker Clarify: https://docs.aws.amazon.com/sagemaker/latest/dg/clarify-configure-processing-jobs.html
- SageMaker JumpStart foundation models: https://docs.aws.amazon.com/sagemaker/latest/dg/jumpstart-foundation-models-use.html
- Responsible AI on AWS: https://aws.amazon.com/ai/responsible-ai/
- AWS AI Service Cards (resources): https://aws.amazon.com/ai/responsible-ai/resources/
- Amazon SageMaker Model Cards: https://docs.aws.amazon.com/sagemaker/latest/dg/model-cards.html
- SageMaker Model Monitor: https://docs.aws.amazon.com/sagemaker/latest/dg/model-monitor.html
- Amazon Augmented AI: https://aws.amazon.com/augmented-ai/
- Bedrock model invocation logging: https://docs.aws.amazon.com/bedrock/latest/userguide/model-invocation-logging.html
- Bedrock data protection: https://docs.aws.amazon.com/bedrock/latest/userguide/data-protection.html
- Bedrock data encryption: https://docs.aws.amazon.com/bedrock/latest/userguide/data-encryption.html
- Bedrock VPC and PrivateLink: https://docs.aws.amazon.com/bedrock/latest/userguide/usingVPC.html
- Generative AI Security Scoping Matrix: https://aws.amazon.com/ai/security/generative-ai-scoping-matrix/
- Well-Architected Generative AI Lens: https://docs.aws.amazon.com/wellarchitected/latest/generative-ai-lens/generative-ai-lens.html
- Amazon Bedrock AgentCore overview: https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/what-is-bedrock-agentcore.html

> **Transparência:** o **AWS Skill Builder** exige login e não foi acessado diretamente; revise o conteúdo por lá. Algumas descrições curtas foram escritas com base na documentação padrão da AWS. Números e nomes de serviços mudam com o tempo: confirme nas páginas oficiais antes da prova.