

Guia de preparação AI-901

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

2
capítulos por domínio

14
questões comentadas

70%
nota de corte

6
provas completas no curso

ANTES DE COMEÇAR

Como usar este guia

Este guia foi feito para ser estudado junto com as 6 provas completas do curso AI-901: o guia explica o que cai e onde estão as armadilhas; as provas treinam o ritmo e mostram o que você ainda não domina.

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

Mapa do exame

Cada prova do curso tem 50 questões e 45 minutos, no formato do exame. Referência de aprovação: 70%.

Domínio (capítulo)	Peso	Neste guia
1. Identificar os conceitos e os recursos de IA	40-45%	19 conceitos · 13 pegadinhas
2. Implementar soluções de IA usando o Microsoft Foundry	55-60%	18 conceitos · 14 pegadinhas

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

Estude um pouco todo dia, de graça

Instagram @strecht.simulados
Um post de prova por dia: questão comentada, pegadinhas e o "não confunda" de cada certificação.
Seguir no Instagram →

YouTube Strecht Simulados
Shorts diários com pegadinhas de prova e, aos sábados, um simulado comentado.
Inscrever-se no canal →

DIAGNÓSTICO

Você já sabe isso?

Responda às 6 questões sem consultar nada. As respostas, com a explicação e o capítulo para revisar, estão no fim do guia. Comece o estudo pelos capítulos em que errou.

D1 · Identificar os conceitos e os recursos de IA

Uma equipe ajustou um modelo com seus exemplos (fine-tuning). O que é necessário antes de usá-lo no aplicativo?

A. Desativar os filtros de conteúdo
B. Implantar o modelo ajustado
C. Treinar outro modelo do zero
D. Apagar o modelo de base

D2 · Identificar os conceitos e os recursos de IA

Dois aplicativos usam o mesmo modelo, e a empresa quer controlar a capacidade de cada um separadamente. Qual abordagem atende?

A. Usar a mesma chave e trocar a temperatura por app
B. Usar uma implantação e dividir o prompt ao meio entre os apps
C. Implantar o modelo em um notebook de cada equipe
D. Criar duas implantações do modelo com cotas próprias

D3 · Identificar os conceitos e os recursos de IA

Um assistente não sabe responder sobre uma lei aprovada no mês passado. Qual é a explicação mais provável?

A. Modelos de linguagem não respondem sobre leis
B. O conhecimento do modelo tem uma data de corte
C. A temperatura impede respostas sobre leis novas
D. O modelo esquece tudo depois de cada resposta

D4 · Implementar soluções de IA usando o Microsoft Foundry

O interpretador de código de um agente gerou um gráfico como arquivo. Como o aplicativo cliente pode mostrá-lo ao usuário?

A. Implantando um modelo de imagem para recriá-lo
B. Baixando o arquivo gerado pela resposta do agente
C. Convertendo o gráfico em áudio antes de exibir
D. Pedindo ao usuário que desenhe o gráfico de novo

D5 · Implementar soluções de IA usando o Microsoft Foundry

Uma empresa quer que seu agente do Foundry use ferramentas expostas por um servidor remoto que segue um protocolo aberto para ferramentas de IA. O que configurar?

A. A detecção de idioma do Azure Language
B. A geração de imagens do catálogo
C. A transcrição em lote do Azure Speech no projeto
D. A ferramenta MCP (Model Context Protocol)

D6 · Implementar soluções de IA usando o Microsoft Foundry

Como evitar surpresas com o custo de um projeto de IA no Azure?

A. Desligar as métricas de uso do projeto
B. Aumentar as cotas das implantações do projeto
C. Usar o maior modelo do catálogo em tudo
D. Configurar orçamentos e alertas de custo

01

CAPÍTULO 1 · 40-45% DA PROVA

Identificar os conceitos e os recursos de IA

Este domínio vale de 40 a 45% da prova e mistura três blocos: os seis princípios de IA responsável, o funcionamento e a configuração dos modelos generativos (tokens, janela de contexto, parâmetros, tipos de implantação) e a identificação da carga de trabalho certa para cada cenário (texto, fala, visão, extração de informações, IA generativa e agentes). Quase tudo vem como um caso curto de empresa, e a resposta é o nome do princípio, do parâmetro ou da carga de trabalho. Os erros mais comuns estão na troca de princípios parecidos, principalmente privacidade e segurança contra confiabilidade e segurança, e em distratores que oferecem mexer na temperatura para resolver qualquer problema. Também é preciso separar bem classificar de detectar, reconhecer fala de sintetizar fala e Global Standard de Data Zone.

Neste capítulo

19 tópicos-chave · 13 pegadinhas · 7 macetes · 14 gatilhos · 4 questões comentadas

Tópicos-chave

Seis princípios de IA responsável da Microsoft

Imparcialidade, confiabilidade e segurança, privacidade e segurança, inclusão, transparência e responsabilidade. Popularidade, velocidade e lucratividade aparecem como distratores e não são princípios. A prova descreve uma decisão ou um problema e pede o princípio envolvido.

Imparcialidade

Evitar tratamento desigual entre grupos. O viés costuma vir de dados históricos (que carregam decisões passadas) ou de dados de treino pouco representativos; região, temperatura ou autenticação não causam viés. A prática cobrada é medir o desempenho por grupo (gênero, idade, região), porque a acurácia média esconde diferenças, e testar com prompts variados para reduzir estereótipos.

Confiabilidade e segurança

O sistema deve funcionar de forma consistente e segura, com testes rigorosos (inclusive em condições inesperadas) e planos para falhas. Exemplos cobrados: encaminhar a conversa a um atendente humano quando a IA não consegue resolver, revisão humana em áreas críticas e monitorar a qualidade das respostas em produção para detectar piora.

Privacidade e segurança

Proteger dados pessoais, controlar o acesso e respeitar o consentimento e as leis de privacidade. Inclui minimizar dados pessoais nos prompts, proteger os registros e, em RAG, filtrar os documentos recuperados pelas permissões de cada usuário (filtros de segurança no índice). A Microsoft informa que prompts, respostas e dados de ajuste fino do cliente não treinam os modelos de base sem permissão.

Inclusão, transparência e responsabilidade

Inclusão: atender idiomas, níveis de leitura, deficiências e contextos diferentes desde o projeto, envolvendo pessoas diversas nos testes (legendas ao vivo e linguagem simples são exemplos). Transparência: deixar claro que a pessoa interage com IA, explicar fatores de decisões, limitações e indicar conteúdo criado com IA; as notas de transparência descrevem como os sistemas funcionam e onde podem falhar. Responsabilidade: quem projeta e implanta responde pelo sistema, com governança, aprovação, auditoria, pessoa no processo em decisões de alto impacto e avaliação de impacto.

Como funcionam os modelos generativos

Modelos baseados em transformadores são pré-treinados em grandes volumes de texto, representam o texto como tokens e geram a saída prevendo, a cada passo, o próximo token mais provável dado o contexto. O mecanismo de atenção calcula o quanto cada token influencia os outros. Depois do pré-treinamento, ajustes com exemplos e feedback humano fazem o modelo de chat seguir instruções, que é a diferença para o modelo de base. Por padrão o modelo não consulta a internet.

Token e cobrança

Token é um pedaço de texto, como uma palavra ou parte dela. Limites de contexto e custos são medidos em tokens; na cobrança por uso, tokens de entrada e de saída têm preços próprios, então prompts longos e respostas longas encarecem.

Janela de contexto

É o limite de tokens que o modelo considera em uma chamada, e prompt e resposta dividem esse limite: se o prompt quase enche a janela, sobra pouco espaço para a resposta, sem aumento automático. Se a conversa ou o documento não cabe, é preciso resumir, cortar ou usar RAG para enviar só os trechos relevantes. Não confundir com capacidade (usuários simultâneos).

Data de corte e alucinação

O conhecimento do modelo tem uma data de corte, por isso ele não sabe de fatos recentes; a solução é fundamentar com dados atualizados (RAG ou pesquisa na web). Alucinação é o conteúdo incorreto gerado pelo modelo; fundamentação é a técnica que reduz o problema, com contexto relevante, instrução para usar só o contexto e admitir quando não souber, temperatura baixa e citação de fontes.

Embeddings

Uma lista de números (vetor) que representa o significado do texto: textos com significado parecido ficam próximos, e a similaridade de cosseno é uma medida comum dessa proximidade. Servem para busca semântica, agrupamento, recomendação e RAG. Um modelo de embeddings gera vetores, não respostas, imagens ou áudio.

RAG (geração aumentada por recuperação)

O aplicativo recupera trechos relevantes de documentos (por exemplo, com busca vetorial) e os envia ao modelo para fundamentar a resposta. Resolve o fato de o modelo não conhecer documentos internos, sem treinar modelo do zero. Mostrar citações aumenta a transparência e permite ao usuário verificar a resposta. Se a solução só consulta documentos, um chat com RAG é mais simples e seguro que um agente com ações.

Escolha do modelo no catálogo

Os cartões de modelo trazem descrição, modalidades de entrada e saída, contexto, benchmarks, licença e orientações de uso responsável; o catálogo também tem rankings de qualidade, custo e desempenho. Modelo multimodal processa mais de um tipo de dado (texto e imagem, texto e áudio); modelo de raciocínio dedica mais processamento a problemas complexos; modelos pequenos (SLMs, como a família Phi) atendem tarefas específicas e simples com menor custo e latência. Para agentes, o modelo precisa de suporte a chamada de ferramentas (function calling). A escolha final se faz avaliando com um conjunto de dados próprio.

Tipos de implantação

Global Standard roteia pela infraestrutura global e costuma ter mais cota e disponibilidade; Data Zone restringe o processamento a uma zona (UE ou EUA) para residência de dados; regional padrão tem menos capacidade. Padrão cobra por token; provisionada (PTU) reserva unidades de throughput para tempo real; batch processa arquivos com muitas requisições de forma assíncrona, com resultados em até 24 horas e custo menor. O playground não é implantação.

Cota, TPM e versões

A cota da assinatura é distribuída entre as implantações em tokens por minuto (TPM), o que também define as requisições por minuto. Duas implantações do mesmo modelo, com cotas próprias, separam a capacidade de dois aplicativos. Modelos têm versões com datas de aposentadoria; a implantação pode atualizar automaticamente ou ficar fixa até a troca planejada. Um modelo ajustado (fine-tuning) também precisa ser implantado antes do uso. A região afeta modelos disponíveis, latência e residência dos dados.

Parâmetros de geração

Temperatura e top_p controlam a aleatoriedade: valores menores dão respostas consistentes e factuais, maiores dão criatividade; ajuste um deles de cada vez. Máximo de tokens limita o tamanho (custo e latência); stop define sequências que encerram a resposta sem incluí-las; seed ajuda a repetir resultados (sem garantia total); penalidade de frequência reduz repetições e de presença incentiva novos assuntos; stream só muda a forma de entrega.

Mensagem do sistema e engenharia de prompt

A mensagem do sistema define persona, tom, formato de saída e restrições; a mensagem do usuário traz o pedido de cada turno. Técnicas cobradas: exemplos de entrada e saída (few-shot), instruções claras, etapas numeradas, delimitadores (como aspas triplas) para separar instrução de dados e saídas estruturadas com esquema JSON. Comece pela engenharia de prompt e pelo RAG; ajuste fino serve para padronizar estilo, formato ou tarefa com muitos exemplos.

Agentes de IA

Agentes combinam modelo de linguagem, instruções e ferramentas para buscar dados, chamar APIs e concluir tarefas em várias etapas. São indicados quando é preciso decidir o próximo passo a cada caso; tarefas sempre iguais, com passos conhecidos, ficam melhor em um fluxo fixo, e tarefas simples (traduzir, sintetizar fala) bastam com o serviço específico.

Cargas de trabalho e técnicas de análise

Texto: análise de sentimento, mineração de opinião (opinião ligada a alvos como bateria ou tela), reconhecimento de entidades, extração de frases-chave, detecção e mascaramento de PII, detecção de idioma, resumo (extrativo, abstrativo e de conversas). Fala: reconhecimento (fala em texto) e síntese (texto em fala). Visão: classificação de imagens, detecção de objetos, OCR, análise de vídeo, detecção de pessoas. Extração de informações: o Content Understanding transforma documentos, imagens, áudio e vídeo em dados estruturados com um esquema de campos.

Segurança de IA generativa

O Azure AI Content Safety analisa textos e imagens nas categorias ódio, violência, sexual e automutilação, com níveis de severidade; os Prompt Shields detectam jailbreak (ataque direto do usuário) e injeção indireta (instrução escondida em documento). A mitigação é em camadas: modelo, plataforma de segurança (filtros), mensagem do sistema e fundamentação, e experiência do usuário. O ciclo é identificar danos, medir, mitigar e operar com monitoramento; red teaming (testes adversariais) encontra os riscos.

Macetes para memorizar

Seis princípios

I-C-P-I-T-R: Imparcialidade, Confiabilidade, Privacidade, Inclusão, Transparência, Responsabilidade. Velocidade, popularidade e lucro nunca entram.

As duas seguranças

Privacidade e segurança protege DADOS; confiabilidade e segurança protege o FUNCIONAMENTO.

Temperatura

Temperatura é o botão do acaso: baixa = previsível, alta = criativa. Em qualquer outro problema, ela é distratora.

Visão

Classificar diz O QUÊ; detectar diz O QUÊ e ONDE.

Implantações

Global = mais cota; Data Zone = dado fica na zona; PTU = capacidade reservada; Batch = barato e até 24 horas.

Ciclo de IA generativa responsável

Identificar, Medir, Mitigar, Operar: primeiro mapeia o dano, por último monitora.

Prompt Shields

Jailbreak vem do usuário (direto); injeção vem escondida no documento (indireto).

Tabela de memorização: não confunda

Quando é...	...e quando é

Classificação de imagens Quando basta uma categoria (rótulo) para a imagem inteira.	**Detecção de objetos** Quando é preciso saber onde está cada item ou contá-los: devolve classe e caixa delimitadora (coordenadas) de cada objeto.
Reconhecimento de fala (fala em texto) Transcrever áudio em tempo real ou em lote, comandos de voz, legendas ao vivo.	**Síntese de fala (texto em fala)** Gerar áudio natural a partir de texto com vozes neurais, ajustáveis por SSML.
Resumo extrativo Seleciona as frases mais importantes do próprio texto.	**Resumo abstrativo** Sintetiza o conteúdo com frases novas.
Implantação Global Standard Sem exigência de residência de dados e com necessidade da maior disponibilidade e cota.	**Implantação Data Zone** Quando o processamento precisa ficar restrito a uma zona de dados, como UE ou EUA.
Provisionada (PTU) Reservar capacidade dedicada para cargas em tempo real.	**Lote (batch)** Grandes volumes assíncronos, sem resposta imediata, com resultado em até 24 horas e custo menor.
Privacidade e segurança O cenário fala de dados pessoais, consentimento, controle de acesso, registros protegidos.	**Confiabilidade e segurança** O cenário fala de funcionar de forma consistente, testes, falhas, fallback para humano, revisão humana em área crítica, monitoramento.
Modelo preditivo Escolher entre classes ou valores, como classificar e-mail como spam.	**Modelo generativo** Criar conteúdo novo, como escrever a resposta do e-mail, imagem, áudio ou código.

Números que caem

Item	Valor
Peso do domínio na prova	**40 a 45% (cerca de 21 de 50 questões)**
Princípios de IA responsável da Microsoft	**6**
Categorias do Content Safety	**4: ódio, violência, sexual e automutilação, com níveis de severidade**
Prazo de retorno da implantação em lote (batch)	**até 24 horas**
Etapas do ciclo de IA generativa responsável	**4: identificar, medir, mitigar e operar**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Aumentar a temperatura resolve o problema (consistência, formato, regras, alucinação, viés, falta de dados).
Como não cair: Temperatura só controla a aleatoriedade. Aumentá-la aumenta a variação; para respostas consistentes, diminua. Ela não traz documentos, não separa capacidade, não bloqueia conteúdo nem guarda histórico.

⚠ **Alerta de prova:** Escolher confiabilidade e segurança porque o enunciado fala em proteger dados.
Como não cair: Dados pessoais, consentimento e acesso são privacidade e segurança; confiabilidade e segurança é sobre funcionar de forma consistente e segura.

⚠ **Alerta de prova:** Nas alternativas de princípios, marcar a ação que parece boa mesmo que apoie outro princípio.
Como não cair: Testar com pessoas de idades diferentes é inclusão; restringir acesso é privacidade; medir por grupo é imparcialidade; avisar que o conteúdo foi gerado por IA é transparência; citar fontes é transparência, não impede acesso indevido.

⚠ **Alerta de prova:** Medir só a acurácia média para provar que o modelo é justo.
Como não cair: A média esconde diferenças; a imparcialidade exige comparar o desempenho entre grupos.

⚠ **Alerta de prova:** Chamar a geração de informação inventada de fundamentação.
Como não cair: O problema é alucinação; fundamentação é a técnica que o reduz.

⚠ **Alerta de prova:** Escolher o maior modelo de raciocínio para garantir qualidade em tarefa simples e volumosa.
Como não cair: Custo e latência altos sem necessidade; um modelo pequeno ou mini, avaliado para a tarefa, costuma bastar.

⚠ **Alerta de prova:** Achar que a resposta tem a janela inteira reservada ou que ela dobra para caber.
Como não cair: Entrada e saída dividem a janela; reduza o contexto enviado para reservar tokens à saída.

⚠ **Alerta de prova:** Usar um modelo de embeddings para responder perguntas ou descrever imagens.
Como não cair: Embeddings geram vetores para busca e similaridade; respostas exigem modelo de chat, e imagens exigem modelo com visão.

⚠ **Alerta de prova:** Usar reconhecimento facial com identificação para contar pessoas em um local.
Como não cair: Detectar e contar pessoas é diferente de identificá-las; a identificação facial tem acesso restrito por riscos de imprecisão e uso indevido.

⚠ **Alerta de prova:** Dar a um assistente de dúvidas um agente com acesso de escrita a sistemas.
Como não cair: Se só precisa consultar documentos, um chat com RAG é suficiente, mais simples e seguro; permissões além do necessário aumentam o risco.

⚠ **Alerta de prova:** Escolher Data Zone ou regional quando o cenário não exige residência de dados.
Como não cair: Sem exigência de região e com foco em disponibilidade e cota, Global Standard.

⚠ **Alerta de prova:** Aumentar réplicas ou cota para deixar o texto mais consistente.
Como não cair: Réplicas e cota afetam capacidade, não a variação do texto; o limite de TPM controla quantos tokens por minuto a implantação aceita.

⚠ **Alerta de prova:** Achar que o modelo copia trechos da internet em tempo real ou busca respostas prontas.
Como não cair: O modelo prevê o próximo token com base nos padrões aprendidos; regras manuais descrevem sistema de regras e respostas fixas descrevem FAQ.

Como cai na prova

📝 **Como cai:** Cenário de empresa com uma decisão de projeto e a pergunta: qual princípio de IA responsável essa decisão apoia?
Como resolver: Identifique a palavra-chave da ação (dados, idiomas, explicar, quem responde, grupos, falha) e associe ao princípio. Descarte alternativas que descrevem outro princípio, mesmo sendo boas práticas.

📝 **Como cai:** Cenário de negócio pedindo a carga de trabalho ou técnica: notas fiscais, fotos de produtos, ligações, prateleiras, comentários de clientes.
Como resolver: Pergunte qual é a entrada (texto, áudio, imagem, documento) e qual é a saída (categoria, posição, campos, áudio, texto novo). Campos estruturados = extração de informações; posição = detecção de objetos; opinião = sentimento.

📝 **Como cai:** Problema de comportamento do modelo (varia demais, respostas longas, formato errado, inventa) e pergunta qual parâmetro ou técnica ajustar.
Como resolver: Varia demais: temperatura menor. Longa e cara: máximo de tokens. Formato: few-shot ou saídas estruturadas com esquema JSON. Inventa: fundamentação/RAG e instrução para usar só o contexto. Parar em um ponto: stop.

📝 **Como cai:** Requisito de capacidade, custo ou residência e pergunta qual tipo de implantação escolher.
Como resolver: Residência de dados: Data Zone. Máxima cota sem restrição: Global Standard. Tempo real com capacidade reservada: PTU. Grande volume sem pressa e mais barato: batch.

📝 **Como cai:** Questão de múltipla resposta (Escolha DUAS) sobre princípios ou componentes.
Como resolver: Marque só os itens que pertencem à lista oficial ou ao conceito; distratores costumam ser palavras de negócio (velocidade, lucratividade) ou ações que pertencem a outro conceito.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
dados de clientes, consentimento, controlar acesso	**Privacidade e segurança**
encaminhar a humano, testes em condições inesperadas, monitorar piora em produção	**Confiabilidade e segurança**
vários idiomas, linguagem simples, acessibilidade, legendas	**Inclusão**
avisar que é IA, explicar a decisão, nota de transparência, conteúdo sintético	**Transparência**
governança, auditoria, aprovação, quem responde, avaliação de impacto	**Responsabilidade**
dados históricos, resultado desigual entre grupos, estereótipos	**Imparcialidade (viés nos dados)**
fato recente que o assistente não conhece	**Data de corte; RAG ou pesquisa na web**
responder com base em documentos internos	**RAG (fundamentação)**
texto oculto em documento mandando ignorar regras	**Injeção de prompt; Prompt Shields**
escolha automática do modelo por prompt, equilibrando custo e qualidade	**Roteador de modelos (model router)**
número, data e total de notas, recibos, formulários, atributos de fotos, campos de ligações ou vídeos	**Extração de informações (Content Understanding)**
onde está cada item na foto, contar produtos	**Detecção de objetos**
opinião sobre bateria, tela, atributo específico	**Mineração de opinião**
mascarar nomes, telefones, documentos	**Detecção de PII**

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

- Os seis princípios e que privacidade e segurança é diferente de confiabilidade e segurança.
- Temperatura só mexe na aleatoriedade; em problemas de formato, memória, capacidade, viés ou acesso ela é distratora.
- Prompt e resposta dividem a janela de contexto, e cobrança é por tokens de entrada e de saída.
- Prompts, respostas e dados de ajuste fino do cliente não são usados para treinar os modelos de base no Foundry sem permissão.
- Comece por engenharia de prompt e RAG; ajuste fino é para padronizar estilo, formato ou tarefa com muitos exemplos.

Essenciais para a prova (checklist)

☐ Sei associar cada um dos seis princípios de IA responsável a uma ação concreta de projeto.

☐ Sei explicar tokens, janela de contexto, data de corte, atenção e a diferença entre modelo de base e de chat.

☐ Sei quando usar embeddings, RAG, ajuste fino e roteador de modelos.

☐ Sei o efeito de temperatura, top_p, máximo de tokens, stop, seed, penalidades e stream.

☐ Sei escolher entre Global Standard, Data Zone, regional, PTU e batch.

☐ Sei separar classificação de imagens, detecção de objetos, OCR e detecção de pessoas.

☐ Sei diferenciar sentimento, mineração de opinião, entidades, frases-chave, PII, idioma e os tipos de resumo.

☐ Sei reconhecer jailbreak, injeção de prompt e as camadas de mitigação com Content Safety e Prompt Shields.

☐ Sei quando um agente é melhor que um fluxo fixo e quando um chat com RAG basta.

Questões de revisão do capítulo

R1.1 · Identificar os conceitos e os recursos de IA

Uma empresa quer que cada prompt seja enviado automaticamente ao modelo mais adequado, equilibrando custo e qualidade. Qual recurso do Foundry atende?

A. A transcrição em lote do Azure Speech
B. A detecção de idioma do Azure Language
C. O filtro de conteúdo da implantação
D. O roteador de modelos (model router)

R1.2 · Identificar os conceitos e os recursos de IA

Um app precisa descrever fotos enviadas pelos usuários e responder perguntas sobre elas. Qual modelo escolher no catálogo?

A. Um modelo de tradução de texto
B. Um modelo de chat com suporte a imagens
C. Um modelo de síntese de fala
D. Um modelo só de embeddings de texto em português

R1.3 · Identificar os conceitos e os recursos de IA

Para reduzir riscos de um app de IA generativa, a Microsoft recomenda combinar várias camadas de proteção. Qual combinação reflete essa ideia?

A. Uso de um único filtro, sem testes nem monitoramento posterior
B. Modelo, filtros, mensagem do sistema e experiência do usuário
C. Remoção dos avisos de IA para não confundir os usuários
D. Temperatura alta e respostas longas para dar mais liberdade

R1.4 · Identificar os conceitos e os recursos de IA

As respostas de um assistente estão variando demais para a mesma pergunta, e a empresa quer respostas mais consistentes. Qual parâmetro ajustar?

A. Remover a mensagem do sistema
B. Diminuir a temperatura
C. Aumentar o número de réplicas da implantação
D. Aumentar a temperatura

Respostas comentadas

R1.1. D) O roteador de modelos (model router)

Por quê: O roteador de modelos analisa cada prompt e encaminha para um modelo menor ou maior, conforme a complexidade.

✗ A) Incorreta. Processa áudio gravado.
✗ B) Incorreta. Identifica o idioma.
✗ C) Incorreta. Bloqueia conteúdo nocivo.

> **Dica de prova:** Escolha automática de modelo: model router.

R1.2. B) Um modelo de chat com suporte a imagens

Por quê: Os cartões de modelo indicam as modalidades de entrada e saída aceitas, o que ajuda a escolher o modelo certo.

✗ A) Incorreta. Não interpreta imagens.
✗ C) Incorreta. Gera áudio.
✗ D) Incorreta. Não interpreta imagens.

> **Dica de prova:** Perguntas sobre fotos: modelo com visão.

R1.3. B) Modelo, filtros, mensagem do sistema e experiência do usuário

Por quê: As camadas incluem o modelo, a plataforma de segurança (filtros), as instruções e a fundamentação, e a experiência do usuário.

✗ A) Incorreta. Uma camada só é insuficiente.
✗ C) Incorreta. Prejudica a transparência.
✗ D) Incorreta. Aumenta o risco.

> **Dica de prova:** Mitigação em camadas.

R1.4. B) Diminuir a temperatura

Por quê: Temperatura e top_p controlam a aleatoriedade; valores menores geram respostas mais previsíveis, úteis em tarefas factuais.

✗ A) Incorreta. Tende a deixar o comportamento menos controlado.
✗ C) Incorreta. Afeta capacidade, não a variação do texto.
✗ D) Incorreta. Aumenta a variação das respostas.

> **Dica de prova:** Mais consistência: temperatura menor.

02

CAPÍTULO 2 · 55-60% DA PROVA

Implementar soluções de IA usando o Microsoft Foundry

Este é o domínio mais pesado, de 55 a 60% da prova, e cobra a parte prática: implantar modelos no portal do Foundry, chamar chat pelo SDK em Python, montar agentes com ferramentas e usar as Foundry Tools (Language, Speech, Vision, Content Understanding) e modelos multimodais e de imagem. As perguntas trazem um sintoma de aplicativo (erro 404, 429, resposta cortada, agente que esquece) ou uma necessidade e pedem a configuração, o parâmetro ou a ferramenta certa. Os erros mais comuns estão em detalhes de fluxo: quem executa a função, como manter o histórico, qual configuração de fala afeta entrada ou saída e quando o Content Understanding faz mais do que o necessário. Também é preciso dominar os métodos de campo dos analisadores (extrair, gerar, classificar).

Neste capítulo

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

Tópicos-chave

Recurso e projeto do Foundry

O recurso do Foundry reúne modelos, agentes e as Foundry Tools (Speech, Language, Vision, Content Understanding) sob o mesmo endpoint e controle de acesso. O projeto fica dentro do recurso e reúne implantações, agentes e arquivos de uma solução. Conexões guardam com segurança o acesso a recursos externos, como Azure AI Search e armazenamento.

Implantar e usar um modelo

Escolha o modelo no catálogo (Microsoft, OpenAI, Meta, Mistral e outros), crie uma implantação no projeto e use o endpoint e o nome da implantação no aplicativo; não precisa de VM com GPU, Kubernetes, Docker nem treinar. O nome da implantação pode ser diferente do nome do modelo, e usar o nome errado gera 404. Modelos de texto e de imagem são implantações diferentes.

SDK do Foundry em Python

Pacotes azure-ai-projects (cliente do projeto) e azure-identity (credenciais como DefaultAzureCredential). O cliente do projeto recebe o endpoint do projeto e uma credencial, expõe operações sobre implantações, conexões, conjuntos de dados e agentes, e pode criar um cliente OpenAI autenticado para chat. Endpoint e configurações ficam em variável de ambiente ou arquivo .env.

Autenticação

O recomendado é autenticação sem chave com Microsoft Entra ID (DefaultAzureCredential, identidades gerenciadas) e RBAC; a identidade precisa de uma função no recurso ou projeto, senão há erro de acesso. Chave de API, quando usada, fica em cofre (Azure Key Vault) ou variável de ambiente, com rotação, nunca no código, no HTML ou no prompt.

Chamada de chat

Obrigatórios: nome do modelo ou da implantação e a lista de mensagens; temperatura e máximo de tokens são opcionais. Cada mensagem é um dicionário com role e content; papéis system (instruções), user (pessoa), assistant (respostas do modelo) e tool (resultados de ferramentas). O texto da resposta está em choices[0].message.content; o campo usage informa tokens do prompt, da resposta e o total.

Motivo de término e erros comuns

Motivo stop = término natural; length = máximo de tokens atingido (aumente o limite ou peça respostas mais curtas); content_filter = o filtro de conteúdo bloqueou a saída (a resposta pode vir vazia), e o app deve tratar com mensagem adequada, sem tentar contornar a política. Erro 404 = recurso ou implantação não encontrado; 401 = credencial incorreta; 429 = limite de taxa ou cota (novas tentativas com espera crescente, distribuir carga ou aumentar o TPM).

Histórico de conversa

Modelos de chat não guardam estado entre chamadas: o aplicativo reenvia as mensagens anteriores (sistema, usuário e assistente) com a nova pergunta, guardando o histórico por sessão de cada usuário. Quando o histórico cresce, resuma ou corte as mensagens mais antigas. Pares user e assistant de exemplo ensinam formato e estilo; stream mostra a resposta em partes; n gera várias escolhas com custo proporcional.

Agentes no Foundry

Base do agente: implantação do modelo, nome e instruções (escopo, tom, formato, idioma, recusas); ferramentas são opcionais. Testa-se no playground de agentes, que mostra respostas e uso de ferramentas, e o portal oferece código de exemplo. As mensagens ficam em uma conversa (thread) e cada execução (run) aplica instruções e ferramentas; a resposta só fica completa ao fim da execução. Agentes ficam no projeto e podem ser chamados pelo ID por vários apps; conversas ficam guardadas até serem excluídas pelo SDK ou API.

Ferramentas de conhecimento

Pesquisa de arquivos (file search): divide os arquivos em partes, gera embeddings e guarda em um índice vetorial; as anotações permitem mostrar citações. Azure AI Search: consulta índices existentes com busca por palavra-chave, vetorial ou híbrida, via conexão do projeto. Pesquisa na web: fundamenta em fontes atuais e citáveis, contornando a data de corte.

Ferramentas de ação e interpretador de código

Chamada de função: o modelo recebe nome, descrição e esquema JSON dos parâmetros, decide chamar e devolve nome e argumentos; o aplicativo cliente executa e devolve a saída ao agente. OpenAPI permite chamar APIs (autenticação anônima, chave ou identidade gerenciada); MCP descobre e chama ferramentas de servidores compatíveis. Interpretador de código escreve e executa Python em ambiente isolado para cálculos e gráficos, usa arquivos anexados e gera arquivos para download.

Segurança, rastreamento e avaliação

Agentes seguros seguem menor privilégio, exigem aprovação humana antes de ações sensíveis e mantêm rastreabilidade. O rastreamento (tracing) registra prompts, chamadas de ferramentas, respostas e tempos. Avaliadores: qualidade (fundamentação, relevância, coerência, fluência), risco e segurança (violência, ódio, sexual, automutilação) e de agentes (aderência à tarefa, resolução de intenção, precisão nas chamadas de ferramentas). Soluções multiagente usam agentes conectados com um orquestrador.

Filtros de conteúdo

Classificam ódio, violência, sexual e automutilação por severidade e atuam sobre prompts (entrada) e respostas (saída), de texto e de imagem; filtros personalizados definem limites por implantação. As saídas de agentes também passam pelos filtros, e Prompt Shields protegem contra injeção. Bloquear na entrada evita processamento e respostas indevidas.

Azure Language

Sentimento com rótulo e confiança de 0 a 1 por documento e por frase (textos mistos podem receber rótulo misto); mineração de opinião; NER (texto, categoria e confiança); NER personalizado treinado com exemplos rotulados; vinculação de entidades (desambigua termos); frases-chave; PII com escolha de categorias e versão para conversas; resumo (extrativo com número de frases, abstrativo com tamanho aproximado, de conversas por aspecto); detecção de idioma com confiança. Aceita lista de documentos por chamada e tem playground no portal.

Azure Speech: reconhecimento

A configuração recebe endpoint ou região e chave ou credencial do Entra ID, além de idioma (por exemplo pt-BR) ou identificação automática entre idiomas candidatos. A configuração de áudio define a entrada: microfone, arquivo ou fluxo.
Reconhecimento único para após uma frase; contínuo processa até o fim. Evento recognizing traz parciais, recognized traz o resultado final. Lista de frases e fala personalizada melhoram termos específicos; há filtro de palavras ofensivas, carimbos de tempo por palavra para legendas e transcrição em lote para arquivos em armazenamento.

Azure Speech: síntese e tradução

Escolha a voz neural pelo nome (por exemplo pt-BR-FranciscaNeural) e ajuste voz, estilo (alegre, calmo), ritmo, tom, pausas e pronúncia com SSML. A configuração de áudio define a saída (alto-falante, arquivo ou fluxo em memória) e o formato (8 ou 16 kHz para telefonia, MP3 para web). A tradução de fala aceita vários idiomas de destino ao vivo. Voz neural personalizada exige cadastro, aprovação e consentimento do locutor.

Modelos multimodais com áudio e imagem

Imagens vão no conteúdo da mensagem junto com o texto, por URL acessível ou em base64; vários modelos aceitam mais de uma imagem; o nível de detalhe baixo consome menos tokens e o alto ajuda em textos pequenos. Áudio vai codificado (por exemplo base64) na mensagem com o formato (wav ou mp3), e o modelo pode responder em texto ou áudio. APIs de áudio em tempo real usam conexão contínua (WebSocket ou WebRTC) com baixa latência.

Visão e geração de imagens

A Análise de Imagem do Azure Vision devolve legendas, tags, objetos e OCR, com coordenadas precisas; modelos multimodais são melhores para descrições e perguntas abertas, e os valores lidos devem ser conferidos. Modelos de imagem (família gpt-image) geram e editam imagens: parâmetros de tamanho (quadrado, paisagem, retrato), qualidade e n; edição usa máscara e prompt; prompts detalhados (sujeito, estilo, cena, iluminação, paleta) melhoram o resultado; imagens geradas levam credenciais de conteúdo (C2PA). Há também modelos de geração de vídeo.

Content Understanding

Usa analisadores (pré-criados ou personalizados) que combinam tipo de conteúdo e esquema de campos, para documentos, imagens, áudio e vídeo, com modelos generativos implantados no Foundry. Campos podem ser extraídos (valor presente), gerados (resumo, tema) ou classificados (entre opções fixas). O resultado é JSON com conteúdo em Markdown (inclusive tabelas e marcas de seleção), campos, confiança e fundamentação (página e posição). A análise é assíncrona: envia o ID do analisador e o arquivo ou URL e consulta o status até concluir. Vídeo vem em segmentos com início e fim; áudio pode ter diarização (identificação de locutores).

Macetes para memorizar

Agente mínimo

Modelo + Nome + Instruções; ferramenta é acessório.

Chamada de função

O modelo pede, o app faz: o modelo devolve nome e argumentos, quem executa é o cliente.

Códigos HTTP

401 = quem é você? 404 = não achei a implantação. 429 = calma, limite de taxa.

Motivos de término

stop = acabou bem; length = faltou token; content_filter = bloqueado pelo filtro.

Campos do Content Understanding

Extrair COPIA o que está lá; Gerar CRIA (resumo); Classificar ESCOLHE da lista.

Eventos de fala

recognizING = ainda falando (parcial); recognizED = terminou (final).

Pacotes do SDK

azure-ai-projects abre o projeto; azure-identity prova quem você é.

Configuração de áudio

No reconhecimento ela é a ENTRADA (microfone, arquivo); na síntese ela é a SAÍDA (alto-falante, arquivo).

Tabela de memorização: não confunda

Quando é...	...e quando é
Conversão de fala em texto do Azure Speech Quando só a transcrição é necessária.	**Content Understanding para áudio** Quando é preciso extrair campos (motivo, produto, resolução, satisfação) ou resumos por locutor da gravação.
Reconhecimento único Comandos curtos: para após uma frase.	**Reconhecimento contínuo** Áudio longo ou ditado: processa até o fim, emitindo eventos a cada trecho.
Lista de frases Melhorar o reconhecimento de alguns nomes e termos específicos.	**Modelo de fala personalizado** Casos complexos de domínio, treinando com textos e áudios com transcrições.
Pesquisa de arquivos Responder com base em documentos enviados, com citações.	**Interpretador de código** Executar Python para cálculos, análise de dados e gráficos.
Pesquisa de arquivos ou AI Search Conhecimento em documentos e índices.	**Função ou ferramenta OpenAPI** Dados em tempo real ou ações em sistemas, chamando uma API.
Detecção de objetos do Azure Vision Quando precisa de coordenadas precisas de cada objeto.	**Modelo multimodal com visão** Descrições, perguntas abertas sobre fotos, gráficos, diagramas, capturas de tela e manuscritos.

Campo classificar (classify) Escolher um valor de uma lista fixa, como urgente ou normal, mesmo que a palavra não apareça no texto.	**Campo gerar (generate)** Criar valor a partir do conteúdo, como resumo ou tema; extrair (extract) só serve para valores presentes no arquivo.

Números que caem

Item	Valor
Peso do domínio na prova	**55 a 60% (cerca de 29 de 50 questões)**
Pontuação de confiança da análise de sentimento	**entre 0 e 1 para cada rótulo; o rótulo principal é o de maior confiança**
Formato de áudio para telefonia na síntese	**8 ou 16 kHz (MP3 para web)**
Categorias dos filtros de conteúdo	**4: ódio, violência, sexual e automutilação**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Usar o nome do modelo no código e estranhar o erro 404.
Como não cair: O código usa o nome da implantação criada no projeto, que pode ser diferente do nome do modelo.

⚠ **Alerta de prova:** Ler o 429 como palavra proibida, credencial errada ou recurso inexistente.
Como não cair: 429 é limite de taxa ou cota; 401 é credencial; 404 é não encontrado; filtro de conteúdo gera outro tipo de resposta.

⚠ **Alerta de prova:** Achar que o modelo de linguagem, o portal ou o usuário executa a função chamada.
Como não cair: O modelo só devolve nome e argumentos; o aplicativo cliente executa a função e devolve o resultado ao agente.

⚠ **Alerta de prova:** Enviar o código-fonte da função ou senhas para o modelo poder usá-la.
Como não cair: O modelo precisa apenas do nome, da descrição e do esquema dos parâmetros.

⚠ **Alerta de prova:** Trocar a mensagem do sistema, aumentar a temperatura ou criar implantação nova para o chat lembrar o assunto.
Como não cair: O modelo não guarda estado: envie a lista de mensagens anteriores, guardada por sessão de cada usuário.

⚠ **Alerta de prova:** Criar um agente novo ou apagar a conversa a cada pergunta.
Como não cair: Reutilize a mesma conversa (thread) e crie uma execução (run) para processar a nova mensagem.

⚠ **Alerta de prova:** Aplicar filtros só à saída para bloquear prompts nocivos.
Como não cair: Filtros na entrada barram o prompt antes do processamento; os filtros valem também para saídas de agentes.

⚠ **Alerta de prova:** Trocar a voz neural para melhorar a transcrição.
Como não cair: Voz neural afeta só a síntese (saída); para reconhecimento use idioma pt-BR, lista de frases ou fala personalizada.

⚠ **Alerta de prova:** Usar Content Understanding com vários campos quando só se quer a transcrição.
Como não cair: Só transcrever é Azure Speech; Content Understanding faz mais do que o necessário.

⚠ **Alerta de prova:** Criar um analisador novo para cada fatura ou imagem.
Como não cair: Analisadores são reutilizáveis: crie uma vez e envie cada arquivo para análise, acompanhando as operações.

⚠ **Alerta de prova:** Usar o rótulo único do documento para uma mensagem com elogio e reclamação.
Como não cair: Use o sentimento por frase; o rótulo do documento mistura as opiniões.

⚠ **Alerta de prova:** Enviar só o nome do arquivo de imagem ou áudio no prompt.
Como não cair: O modelo não acessa o arquivo: envie a imagem por URL ou base64, e o áudio codificado na própria mensagem.

⚠ **Alerta de prova:** Aumentar o máximo de tokens, reduzir a temperatura ou criar outra implantação para mostrar o texto enquanto o modelo gera.
Como não cair: Use streaming: o app recebe e exibe os trechos conforme são gerados, o que melhora a percepção de velocidade; mais tokens só alongam a resposta.

⚠ **Alerta de prova:** Ignorar a confiança e gravar todos os campos extraídos, ou descartar o lote inteiro.
Como não cair: Encaminhe para revisão humana os campos com confiança baixa e automatize o que é seguro.

Como cai na prova

📝 **Como cai:** Sintoma de aplicativo em produção (erro HTTP, resposta cortada, bloqueio, chat sem memória) e pergunta sobre a causa.
Como resolver: Mapeie o sintoma direto: 404 implantação, 401 credencial, 429 cota, length tokens, content_filter filtro bloqueou, sem memória = histórico não reenviado. Elimine alternativas sobre temperatura, idioma do prompt ou tamanho da resposta, que não geram esses erros.

📝 **Como cai:** Necessidade de um agente e pergunta qual ferramenta adicionar.
Como resolver: Documentos enviados: pesquisa de arquivos. Índice existente: Azure AI Search. Dados atuais da web: pesquisa na web. Cálculo ou gráfico: interpretador de código. Sistema ou API em tempo real: função ou OpenAPI. Ferramentas publicadas em servidores: MCP.

📝 **Como cai:** Trecho de código Python ou requisito do SDK (o que passar ao cliente, onde ler a resposta, quais pacotes).
Como resolver: Cliente do projeto = endpoint do projeto + credencial (DefaultAzureCredential). Chamada de chat = implantação + mensagens. Texto em choices[0].message.content; consumo em usage. Pacotes azure-ai-projects e azure-identity.

📝 **Como cai:** Aplicativo de fala com problema ou requisito (frase cortada, termos técnicos, voz, formato para telefonia, legendas).
Como resolver: Defina se o problema é de reconhecimento (entrada) ou de síntese (saída). Entrada: contínuo, idioma, lista de frases, carimbos de tempo. Saída: voz neural, SSML, formato de áudio.

📝 **Como cai:** Extração de dados de documentos, imagens, áudio ou vídeo e pergunta sobre analisador, campos ou resultado.
Como resolver: Content Understanding com analisador reutilizável e esquema de campos; escolha o método do campo (extrair, gerar, classificar); o resultado é JSON com conteúdo em Markdown, confiança e origem; baixa confiança vai para revisão humana.

📝 **Como cai:** Escolha entre serviço específico e modelo multimodal para imagem ou áudio.
Como resolver: Coordenadas precisas ou OCR em escala: Azure Vision ou Content Understanding. Perguntas abertas sobre a imagem ou prompt falado: modelo multimodal, enviando a mídia no conteúdo da mensagem.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
erro 404 na chamada do modelo	**Nome da implantação errado ou inexistente**
erro 429	**Limite de taxa ou cota: retentativa com espera crescente ou mais TPM**

resposta cortada no meio	**Motivo length: máximo de tokens atingido**
chatbot esquece a pergunta anterior	**Reenviar o histórico de mensagens (role e content)**
agente consulta pedidos em tempo real em uma API	**Chamada de função ou ferramenta OpenAPI**
agente calcula, analisa planilha ou gera gráfico	**Interpretador de código**
agente responde com base em manuais enviados e cita fontes	**Pesquisa de arquivos (file search)**
índice existente com busca híbrida	**Ferramenta Azure AI Search via conexão do projeto**
ferramentas de servidores compatíveis sem integração específica	**MCP (Model Context Protocol)**
ver etapas, chamadas de ferramentas e tempos do agente	**Rastreamento (tracing)**
voz alegre ou calma, pausas, pronúncia	**SSML**
nomes de produtos mal transcritos	**Lista de frases (ou fala personalizada)**
quem falou cada trecho da reunião	**Diarização (identificação de locutores)**
trocar o fundo de uma foto de produto mantendo o produto	**Edição de imagem com prompt e máscara**

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

- No código vai o nome da implantação, não o nome do modelo.
- Modelo de chat não tem memória: o app reenvia o histórico; no serviço de agentes, a conversa (thread) guarda as mensagens e a execução (run) processa.
- Na chamada de função, quem executa é o aplicativo cliente.
- Autenticação recomendada: Entra ID sem chave com RBAC; chave só em Key Vault.
- Content Understanding: analisador reutilizável, campos extrair/gerar/classificar, resultado com confiança e fundamentação.

Essenciais para a prova (checklist)

☐ Sei os passos para usar um modelo: recurso e projeto do Foundry, implantação, endpoint e nome da implantação.

☐ Sei montar a chamada de chat em Python e onde ler o texto, o motivo de término e o uso de tokens.

☐ Sei interpretar os erros 401, 404 e 429 e os motivos stop e length.

☐ Sei criar e testar um agente no portal e usá-lo pelo SDK com conversa e execução.

- ☐ Sei escolher entre pesquisa de arquivos, Azure AI Search, pesquisa na web, interpretador de código, funções, OpenAPI e MCP.
- ☐ Sei usar as técnicas do Azure Language, incluindo sentimento por frase, PII por categoria e resumo extrativo com número de frases.
- ☐ Sei configurar reconhecimento e síntese no Azure Speech: idioma, contínuo, eventos, lista de frases, voz neural, SSML e formato de saída.
- ☐ Sei enviar imagem e áudio a um modelo multimodal e gerar ou editar imagens com tamanho, qualidade, n e máscara.
- ☐ Sei definir um analisador do Content Understanding e interpretar o resultado assíncrono em JSON.
- ☐ Sei onde ficam filtros de conteúdo, avaliadores, rastreamento, métricas e orçamentos de custo.

Questões de revisão do capítulo

R2.1 · Implementar soluções de IA usando o Microsoft Foundry

A análise de sentimento devolveu, para um comentário, positivo 0,10, neutro 0,15 e negativo 0,75. Como interpretar?

A. O comentário está em outro idioma
B. O comentário é certamente positivo
C. O serviço não conseguiu analisar o texto
D. O comentário é provavelmente negativo

R2.2 · Implementar soluções de IA usando o Microsoft Foundry

Um app de chat implantado no Foundry deve bloquear respostas com conteúdo de ódio ou violência. O que configurar?

A. Mais tokens máximos por resposta
B. Uma voz neural diferente
C. Filtros de conteúdo na implantação
D. Uma temperatura igual a zero no modelo

R2.3 · Implementar soluções de IA usando o Microsoft Foundry

Em conversas muito longas, o histórico enviado passa do limite de contexto do modelo. O que o app pode fazer?

A. Resumir ou cortar as mensagens mais antigas
B. Trocar a voz neural usada pelo aplicativo
C. Aumentar a temperatura do modelo implantado
D. Enviar o histórico duplicado para reforçar

R2.4 · Implementar soluções de IA usando o Microsoft Foundry

Durante um evento, a fala do palestrante em português deve aparecer em inglês e espanhol ao mesmo tempo. Isso é possível com o Azure Speech?

A. Sim, com vários idiomas de destino
B. Não, a tradução de fala só aceita inglês
C. Sim, mas só gravando e traduzindo depois
D. Não, cada sessão traduz só para um idioma

Respostas comentadas

R2.1. D) O comentário é provavelmente negativo

Por quê: O serviço devolve pontuações de confiança entre 0 e 1 para cada rótulo, e o rótulo principal é o de maior confiança.

✗ A) Incorreta. Não é o que os números indicam.
✗ B) Incorreta. Tem a menor confiança.
✗ C) Incorreta. Houve resultado.

> **Dica de prova:** Maior pontuação define o sentimento.

R2.2. C) Filtros de conteúdo na implantação

Por quê: Os filtros de conteúdo classificam ódio, violência, sexual e automutilação por severidade e permitem bloquear prompts e respostas.

✗ A) Incorreta. Não bloqueia conteúdo nocivo.
✗ B) Incorreta. Não tem relação com texto.
✗ D) Incorreta. Não bloqueia conteúdo nocivo.

> **Dica de prova:** Bloquear conteúdo nocivo: filtros de conteúdo.

R2.3. A) Resumir ou cortar as mensagens mais antigas

Por quê: Estratégias comuns: janela das últimas mensagens, resumo das antigas e recuperação de trechos relevantes quando necessário.

✗ B) Incorreta. Não tem relação.
✗ C) Incorreta. Não reduz o histórico.
✗ D) Incorreta. Aumenta o problema.

> **Dica de prova:** Histórico grande: resumir ou cortar o antigo.

R2.4. A) Sim, com vários idiomas de destino

Por quê: Na tradução de fala, é possível definir vários idiomas de destino e receber o texto traduzido de cada um em tempo real.

✗ B) Incorreta. Aceita muitos idiomas.

✗ C) Incorreta. Também funciona ao vivo.

✗ D) Incorreta. Aceita vários destinos.

> **Dica de prova:** Vários destinos ao vivo: tradução de fala.

DIA DA PROVA

Estratégia de prova

O AI-901 substituiu o AI-900 e cobra fundamentos com viés prático: além de conceitos de IA responsável e de modelos, mais da metade da prova pede para implementar soluções no Microsoft Foundry, inclusive lendo trechos de código Python. Os enunciados são cenários curtos de empresa com um objetivo ou um sintoma, e a resposta é o serviço, o parâmetro, a ferramenta ou o princípio que resolve exatamente aquilo. Os distratores costumam ser serviços de outra modalidade (síntese de fala para um problema de imagem, por exemplo) ou ajustes que não têm relação com o problema, como temperatura, região ou nome do projeto. Há questões de múltipla resposta que pedem duas escolhas.

Como responder

✓ Gerencie o tempo: são cerca de 50 questões em 45 minutos (a Microsoft não divulga o número exato, que fica entre 40 e 60), menos de um minuto por item; não trave em nenhuma.

✓ Leia primeiro a pergunta final e depois o cenário, marcando a entrada (texto, áudio, imagem, documento) e a saída desejada.

✓ Elimine alternativas de modalidade errada: o que gera áudio não lê imagem, o que processa texto não analisa vídeo.

✓ Desconfie de qualquer alternativa que mexa em temperatura, região, nome da implantação ou cor do portal quando o problema não é aleatoriedade, residência ou endereço.

✓ Em múltipla resposta, confira quantas opções pede o enunciado e avalie cada alternativa isoladamente.

✓ Nos trechos de código Python, procure as peças-chave: endpoint e credencial no cliente, nome da implantação e mensagens na chamada, choices e usage na resposta.

✓ Diante de duas alternativas boas, escolha a mais simples que atende ao requisito (chat com RAG antes de agente, Speech antes de Content Understanding se só precisa transcrever, modelo pequeno para tarefa simples).

✓ A nota mínima é 700 em uma escala de 1 a 1000; o domínio de Foundry pesa mais, então priorize revisá-lo.

Erros que derrubam candidatos

✗ Confundir privacidade e segurança com confiabilidade e segurança.

✗ Cair no distrator de temperatura para problemas de formato, memória, capacidade ou conteúdo nocivo.

✗ Usar o nome do modelo em vez do nome da implantação e não reconhecer o 404.

✗ Achar que o modelo executa a função ou que lembra a conversa sozinho.

✗ Trocar configurações de síntese (voz, SSML) por configurações de reconhecimento (idioma, lista de frases) e vice-versa.

✗ Escolher a solução mais poderosa (maior modelo, agente com escrita, Content Understanding) quando uma mais simples atende.

✗ Confundir classificação de imagens com detecção de objetos e extrair com classificar nos campos do analisador.

Na semana da prova

- ☐ Refaça as provas práticas e anote cada erro como par certa x errada para revisar as pegadinhas.
- ☐ Revise os seis princípios de IA responsável com um exemplo de ação para cada um.
- ☐ Revise o fluxo de código do SDK: pacotes, cliente do projeto, chamada de chat, campos da resposta e códigos de erro.
- ☐ Revise a tabela de ferramentas de agentes e os métodos de campo do Content Understanding.
- ☐ Se for fazer a prova em um idioma em que ela não estiver disponível, o guia oficial prevê pedir 30 minutos extras; confirme isso no agendamento.
- ☐ Na véspera, revise só as seções lembrar e macetes, sem conteúdo novo.

REVISÃO FINAL

Folha de cola

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

1. Identificar os conceitos e os recursos de IA

‣ dados de clientes, consentimento, controlar acesso → **Privacidade e segurança**

‣ encaminhar a humano, testes em condições inesperadas, monitorar piora em produção → **Confiabilidade e segurança**

‣ vários idiomas, linguagem simples, acessibilidade, legendas → **Inclusão**

‣ avisar que é IA, explicar a decisão, nota de transparência, conteúdo sintético → **Transparência**

‣ governança, auditoria, aprovação, quem responde, avaliação de impacto → **Responsabilidade**

⚠ Temperatura só controla a aleatoriedade. Aumentá-la aumenta a variação; para respostas consistentes, diminua. Ela não traz documentos, não separa capacidade, não bloqueia conteúdo nem guarda histórico.

⚠ Dados pessoais, consentimento e acesso são privacidade e segurança; confiabilidade e segurança é sobre funcionar de forma consistente e segura.

⚠ Testar com pessoas de idades diferentes é inclusão; restringir acesso é privacidade; medir por grupo é imparcialidade; avisar que o conteúdo foi gerado por IA é transparência; citar fontes é transparência, não impede acesso indevido.

2. Implementar soluções de IA usando o Microsoft Foundry

‣ erro 404 na chamada do modelo → **Nome da implantação errado ou inexistente**

‣ erro 429 → **Limite de taxa ou cota: retentativa com espera crescente ou mais TPM**

‣ resposta cortada no meio → **Motivo length: máximo de tokens atingido**

‣ chatbot esquece a pergunta anterior → **Reenviar o histórico de mensagens (role e content)**

‣ agente consulta pedidos em tempo real em uma API → **Chamada de função ou ferramenta OpenAPI**

⚠ O código usa o nome da implantação criada no projeto, que pode ser diferente do nome do modelo.

⚠ 429 é limite de taxa ou cota; 401 é credencial; 404 é não encontrado; filtro de conteúdo gera outro tipo de resposta.

⚠ O modelo só devolve nome e argumentos; o aplicativo cliente executa a função e devolve o resultado ao agente.

GABARITO

Respostas do diagnóstico

D1. B) Implantar o modelo ajustado → revise o capítulo 1

Por quê: Como os modelos do catálogo, o modelo ajustado precisa de uma implantação para receber chamadas.

✗ A) Incorreta. Não é necessário.
✗ C) Incorreta. Não é necessário.
✗ D) Incorreta. Não é necessário.

> **Dica de prova:** Modelo ajustado: implantar antes de usar.

D2. D) Criar duas implantações do modelo com cotas próprias → revise o capítulo 1

Por quê: Cada implantação tem nome e limite de tokens por minuto próprios, o que facilita isolar e acompanhar o consumo.

✗ A) Incorreta. Não separa a capacidade.
✗ B) Incorreta. Não separa a capacidade.
✗ C) Incorreta. Não é o mecanismo.

> **Dica de prova:** Capacidade separada por app: duas implantações.

D3. B) O conhecimento do modelo tem uma data de corte → revise o capítulo 1

Por quê: Para informações recentes, use fundamentação com dados atualizados, como RAG ou pesquisa na web.

✗ A) Incorreta. Podem responder.
✗ C) Incorreta. Não é o motivo.
✗ D) Incorreta. Não explica a falta do fato.

> **Dica de prova:** Fato recente desconhecido: data de corte.

D4. B) Baixando o arquivo gerado pela resposta do agente → revise o capítulo 2

Por quê: Arquivos criados pelo interpretador de código ficam associados à resposta e podem ser baixados pelo aplicativo.

✗ A) Incorreta. Não é necessário.
✗ C) Incorreta. Não é necessário.
✗ D) Incorreta. Não aproveita o resultado.

> **Dica de prova:** Gráfico do agente: baixar o arquivo gerado.

D5. D) A ferramenta MCP (Model Context Protocol) → revise o capítulo 2

Por quê: Com o MCP, o agente descobre e chama ferramentas publicadas em servidores compatíveis, sem integração específica para cada uma.

✗ A) Incorreta. Processa texto.
✗ B) Incorreta. Cria imagens.
✗ C) Incorreta. Processa áudio.

> **Dica de prova:** Servidor de ferramentas padrão: MCP.

D6. D) Configurar orçamentos e alertas de custo → revise o capítulo 2

Por quê: O Gerenciamento de Custos permite criar orçamentos e alertas; métricas de tokens ajudam a encontrar os maiores consumos.

✗ A) Incorreta. Esconde o consumo.
✗ B) Incorreta. Pode aumentar o gasto.
✗ C) Incorreta. Aumenta o custo.

> **Dica de prova:** Controlar custo: orçamentos e alertas.

Agora, treine no formato real

Faça as 6 provas completas do curso AI-901: 300 questões inéditas, tempo do exame, explicação de cada alternativa e link da documentação para revisar.

Ver as 6 provas completas e o cupom do mês →

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

Instagram @strecht.simulados

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

Seguir no Instagram →

YouTube Strecht Simulados

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

Inscrever-se no canal →

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