

Guia de preparação AI-103

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

5 capítulos por domínio	**32** questões comentadas	**70%** nota de corte
6 provas completas no curso		

ANTES DE COMEÇAR

Como usar este guia

Este guia foi feito para ser estudado junto com as 6 provas completas do curso AI-103: 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 120 minutos, no formato do exame. Referência de aprovação: 70%.

Domínio (capítulo)	Peso	Neste guia
1. Implementar soluções de IA generativa e agênticas	30-35%	18 conceitos · 14 pegadinhas
2. Implementar soluções de análise de texto	10-15%	16 conceitos · 14 pegadinhas
3. Implementar soluções de extração de informações	10-15%	18 conceitos · 14 pegadinhas
4. Implementar soluções de visão computacional	10-15%	17 conceitos · 13 pegadinhas
5. Planejar e gerenciar uma solução de IA do Azure	25-30%	18 conceitos · 14 pegadinhas

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

Estude um pouco todo dia, de graça

Instagram @strecht.simulados

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

Seguir no Instagram →

YouTube Strecht Simulados

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

Inscrever-se no canal →

DIAGNÓSTICO

Você já sabe isso?

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

D1 · Implementar soluções de IA generativa e agênticas

Uma plataforma de streaming quer processar pedidos de suporte assim: um agente com modelo pequeno classifica o pedido; reembolsos simples seguem para um executor com regras de negócio em código; disputas de direitos autorais seguem para um agente com modelo de raciocínio. Como você deve implementar o roteamento no workflow do Microsoft Agent Framework?

A. Usar arestas switch-case que levam cada mensagem ao executor conforme a classificação
B. Usar arestas de fan-out que enviam cada pedido a cada um dos executores ao mesmo tempo
C. Usar a orquestração Sequential para que cada pedido passe pelos três executores
D. Usar uma implantação do model router para também executar as regras de reembolso

D2 · Implementar soluções de IA generativa e agênticas

Um agente de reembolsos de uma seguradora chama a função solicitar_aprovacao_gerente, que depende de um gerente aprovar no sistema de alçadas, o que pode levar horas. Nos testes, a execução expira antes de a aplicação devolver a saída da função. Como você deve redesenhar a chamada?

A. Devolver de imediato um status pendente e consultar a aprovação em um fluxo separado
B. Aumentar o timeout da execução do agente para 24 horas na definição da ferramenta de função
C. Trocar a função por uma ferramenta MCP com require_approval igual a never
D. Aguardar a aprovação dentro da função até o gerente decidir, sem limite de tempo

D3 · Implementar soluções de IA generativa e agênticas

Uma empresa de mídia vai gerar embeddings de 3 milhões de sinopses com um modelo text-embedding-3-large implantado no seu recurso do Foundry. A equipe busca a menor latência e compatibilidade total com bibliotecas OpenAI que já usa. Qual SDK e endpoint a documentação do Foundry recomenda para esse caso?

A. O SDK do Agent Framework com FoundryChatClient e um agente dedicado
B. O SDK de uma Foundry Tool específica, como o do Azure Language
C. O SDK do Foundry apontado para o endpoint do projeto em /api/projects
D. O SDK da OpenAI apontado para o endpoint /openai/v1 do recurso

D4 · Implementar soluções de análise de texto

Um hospital transcreve ditados de laudos de radiologia com o reconhecimento de fala padrão do Azure Speech, mas termos como espondilolistese e nomes de medicamentos saem errados com frequência. A equipe tem listas de termos e frases típicas dos laudos, além de algumas horas de áudio com transcrição revisada. O que você deve fazer?

A. Usar SSML com phoneme para ensinar a pronúncia dos termos ao modelo
B. Treinar um modelo de custom speech e comparar o WER com o modelo base
C. Ativar a diarização para separar o médico de outros falantes da sala
D. Trocar para batch transcription, que tem vocabulário médico ampliado

D5 · Implementar soluções de análise de texto

Uma empresa de energia eólica grava relatórios de campo falados por técnicos, com ruído de vento e jargões como nomes de turbinas e códigos de falha. A transcrição com o modelo base tem taxa de erro de palavras (WER) alta, principalmente pelo ruído. Os áudios são transcritos em lote todas as noites. O que você deve fazer para melhorar a precisão com o menor custo de hospedagem?

A. Adicionar uma phrase list com os jargões nas requisições de transcrição em lote
B. Criar uma voz neural personalizada (custom voice) com as gravações dos técnicos
C. Treinar custom speech com áudios e transcrições humanas e usá-lo no lote sem endpoint hospedado
D. Implantar o modelo de custom speech em um endpoint dedicado e chamar a API de transcrição rápida

D6 · Implementar soluções de análise de texto

Uma rede de farmácias usa texto em fala neural para ler orientações de uso de medicamentos no app. Nomes como dipirona e cetoprofeno e algumas siglas são pronunciados de forma errada pela voz escolhida. A equipe quer corrigir a pronúncia desses termos sem trocar a voz. O que você deve usar?

A. Speech translation para converter os termos em outro idioma
B. SSML com phoneme ou um lexicon personalizado para esses termos
C. Fast transcription para validar a pronúncia de cada termo
D. Custom speech treinado com gravações dos farmacêuticos lendo os termos

D7 · Implementar soluções de extração de informações

Uma fintech quer preparar milhares de contratos e lâminas de produtos, com tabelas, gráficos e anotações manuscritas, para um agente de RAG. Precisa de uma representação limpa em Markdown, com descrição das figuras, um resumo de cada documento e chunks prontos para embeddings, sem definir campos. Qual analisador você deve usar?

A. prebuilt-imageSearch do Content Understanding
B. prebuilt-invoice do Content Understanding
C. prebuilt-read do Content Understanding com OCR ativado
D. prebuilt-documentSearch do Content Understanding

D8 · Implementar soluções de extração de informações

A startup Base Legal criou uma base de conhecimento do Foundry IQ no Azure AI Search e quer conectá-la a um prompt agent do Foundry Agent Service com autenticação sem chaves e acesso só de leitura aos índices. Como o agente deve se conectar à base de conhecimento e qual permissão você deve conceder?

A. Pela ferramenta File Search, copiando os índices para um vector store e dando Storage Blob Data Reader
B. Pela ferramenta MCP com a chave de administração do Azure AI Search gravada nas instruções do agente
C. Pela ferramenta OpenAPI apontando para o portal do Azure AI Search e dando Search Service Contributor
D. Pela ferramenta MCP com uma conexão do projeto, dando Search Index Data Reader à identidade do projeto

D9 · Implementar soluções de extração de informações

Uma empresa de energia tem milhares de laudos técnicos digitalizados em PDF (imagens escaneadas) em um contêiner do Blob Storage. Ao indexá-los no Azure AI Search com um skillset, o campo de conteúdo fica vazio. Você precisa extrair o texto das páginas escaneadas durante a indexação. O que você deve configurar?

A. Definir imageAction como generateNormalizedImages no indexador e incluir a skill de OCR
B. Definir dataToExtract como storageMetadata no indexador e incluir a skill Key Phrase Extraction
C. Definir parsingMode como jsonArray no indexador e incluir a skill Text Split no skillset
D. Habilitar o ranqueador semântico no índice e incluir a skill de detecção de idioma

D10 · Implementar soluções de visão computacional

Uma transportadora gerou com o Sora 2 um vídeo promocional aprovado pela diretoria, mas quer trocar a cor do caminhão e o céu para pôr do sol, mantendo a estrutura, o movimento e o enquadramento da cena original. Como você deve fazer essa edição?

A. Usar input_reference com um quadro do vídeo e pedir as mudanças em um novo prompt
B. Enviar o vídeo à API de edição de imagens com uma máscara transparente no caminhão
C. Criar um novo vídeo informando remix_video_id do vídeo aprovado e o ajuste no prompt
D. Criar um novo vídeo do zero com um prompt mais detalhado e a mesma semente aleatória

D11 · Implementar soluções de visão computacional

Uma fabricante de embalagens tem vídeos longos de treinamento e quer dividi-los automaticamente em trechos por procedimento (setup da máquina, troca de bobina, limpeza), gerando para cada trecho uma descrição e um campo com o nome do procedimento. Ela vai criar um analisador personalizado no Azure Content Understanding. O que você deve configurar?

A. returnDetails como true para usar cameraShotTimesMs como limites de cada procedimento
B. enableSegment como false e um campo do tipo extract para ler o nome de cada procedimento
C. baseAnalyzerId igual a prebuilt-document e um campo classify com os procedimentos
D. enableSegment como true e contentCategories descrevendo em linguagem natural como segmentar

D12 · Implementar soluções de visão computacional

Uma transportadora quer gerar uma legenda curta para cada uma das 25 fotos tiradas em cada inspeção de carga, enviando as fotos a um modelo GPT multimodal pela API de chat. Os testes com todas as fotos na mesma requisição falham. Qual ajuste você deve fazer?

A. Dividir as fotos em requisições de no máximo 10 imagens e pedir uma legenda por imagem
B. Mudar a implantação para Global Batch para aceitar mais imagens por requisição
C. Converter as fotos em um PDF de 25 páginas e enviar um único arquivo ao modelo
D. Reduzir a resolução das 25 fotos para 512x512 e manter todas elas na mesma requisição de chat

01

CAPÍTULO 1 · 30-35% DA PROVA

Implementar soluções de IA generativa e agênticas

É o maior domínio da prova, então preciso dominar o código: AIProjectClient com o endpoint do projeto, Responses API, Conversations, function calling, MCP, OpenAPI, A2A, memória e as orquestrações do Microsoft Agent Framework. As alternativas costumam ser nomes reais de parâmetros e itens (function_call_output, mcp_approval_response, previous_response_id, reasoning.summary), e só um encaixa no fluxo descrito. Também cai muito ajuste de geração (modelos de raciocínio não aceitam temperature), avaliadores certos para cada sintoma e observabilidade com traces. Onde mais erro: misturar o fluxo de aprovação MCP com o de function calling, escolher Handoff quando é Group Chat (e vice-versa) e trocar Tool Selection por Tool Input Accuracy.

Neste capítulo

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

Tópicos-chave

AIProjectClient e endpoint do projeto

No SDK azure-ai-projects 2.x, crie AIProjectClient com o endpoint do projeto (domínio services.ai.azure.com, caminho api/projects/nome-do-projeto) e uma credencial do Entra ID; get_openai_client() devolve o cliente compatível com OpenAI para Responses API e Conversations. Connection string era de projetos baseados em hub e não é suportada; azure-ai-inference é do conjunto antigo de SDKs.

OpenAI SDK no endpoint /openai/v1

Embeddings e máxima compatibilidade com bibliotecas OpenAI usam o OpenAI SDK apontado para o endpoint openai/v1 do recurso; o endpoint do projeto não atende embeddings. A API v1 permite o cliente OpenAI() com base_url terminando em /openai/v1/ e api_key igual a um provedor de token do Entra (get_bearer_token_provider), sem api_version mensal.

Chamada direta, prompt agent, hosted agent e agente efêmero

Sem ferramentas nem instruções persistentes: get_openai_client() + responses.create(model=implantação). Prompt agent é declarativo (instruções, modelo, ferramentas) e versionado com create_version; hosted agent roda código próprio (LangGraph, Agent Framework) em contêiner com endpoint, escala e identidade gerenciados. Definição só no código, sem recurso persistido: Responses API no endpoint do projeto (agente efêmero, por exemplo com FoundryChatClient). Cada alteração salva de um agente vira uma versão imutável (nome:versão), o que permite direcionar tráfego e fazer rollback; trocar instruções ou modelo = agents.create_version com novo PromptAgentDefinition.

Conversas na Responses API

Threads, Messages e Runs foram substituídos por Conversations, Items e Responses. Histórico mantido pelo serviço: conversations.create() e passar o id em cada responses.create (com agent_reference no extra_body). previous_response_id encadeia só a partir do turno anterior. Tarefa longa que estoura a conexão: background=true e consultar o status; store=false só evita persistir.

Function calling

Defina FunctionTool com nome, descrição e parâmetros em JSON Schema. O modelo devolve um item function_call (name, arguments, call_id); a aplicação executa e envia function_call_output com o mesmo call_id. O Foundry não executa function tools. As execuções expiram 10 minutos após a criação: trabalho demorado devolve status pendente e é consultado depois (polling).

Esquema de ferramenta confiável

Argumentos exatos exigem strict true, todos os campos em required e additionalProperties false; parâmetros errados se corrigem com descrições claras da função e de cada parâmetro, com exemplo de formato. Trate os argumentos como entrada não confiável e valide na aplicação. JSON mode não substitui o esquema da função.

Ferramenta MCP

allowed_tools limita quais ferramentas do servidor o agente pode chamar; require_approval (padrão always, configurável por ferramenta) define o que precisa de aprovação humana. Com aprovação, a resposta traz mcp_approval_request e a aplicação envia mcp_approval_response (approve e approval_request_id) junto com previous_response_id. Azure Functions viram ferramenta MCP pelo endpoint /runtime/webhooks/mcp.

Ferramenta OpenAPI

Aceita especificações 3.0 e 3.1 e autenticação anônima, por chave (em conexão do projeto do tipo Custom keys, com securitySchemes e security declarados na especificação) ou por identidade gerenciada (só se o destino aceita tokens do Entra). operationId é obrigatório em cada operação e usa só letras, hífen e sublinhado. Chave nunca vai na especificação, no código ou nas instruções.

Ferramentas de conhecimento e execução

Azure AI Search: índice existente via conexão do projeto (project.connections.get(nome).id), index_name, query_type padrão vector_semantic_hybrid, top_k padrão 5 e filter OData para restringir escopo. File search: arquivos enviados a um vector store. Web search: web pública atual sem recurso do Bing; Grounding with Bing Search exige conexão bing_grounding. Code Interpreter: Python em sandbox para cálculo exato e gráficos, arquivo em container_file_citation.

Memory no Foundry Agent Service

Memória de longo prazo (preview) entre sessões: tipos perfil de usuário, resumo de conversa e procedural; exige implantações de chat e de embeddings. A forma mais simples é anexar a ferramenta de busca de memória; o scope particiona o memory store, e {{$userId}} isola por usuário (com cabeçalho x-memory-user-id quando um serviço chama em nome de vários usuários). Não substitui o histórico da conversa atual.

Ferramenta A2A

Delegar a um agente externo, sem workflow formal, mantendo seu agente dono da conversa: ferramenta a2a com conexão A2A do projeto (protocolo 1.0, GA, JSON-RPC, só texto e sem streaming). a2a_preview é o protocolo 0.3, ainda em preview, e Connected Agents da API clássica não existe no Foundry atual.

Orquestrações do Microsoft Agent Framework

Sequential: estágios fixos dependentes (pipes and filters). Concurrent: especialistas independentes em paralelo sobre a mesma entrada, com agregação. Handoff: o especialista assume a conversa inteira, sem orquestrador central. Group Chat: colaboração em conversa compartilhada, incluindo maker-checker com round-robin e limite de iterações. Magentic: gerente que planeja e replaneja tarefas abertas. Agente como ferramenta (agent-as-tools): o agente principal mantém a responsabilidade e usa outros agentes como ferramentas.

Workflows híbridos e aprovação humana

Regras fixas e auditáveis ficam em executores de código; o LLM só interpreta ou redige. Roteamento por classificação usa arestas switch-case ou condicionais; fan-out envia a mesma mensagem a todos. Ação crítica: @tool com approval_mode always_require e checkpoints para pausar por horas sem perder estado. Fluxo visual de baixo código com regras: Azure Logic Apps chamando o agente.

Modelos de raciocínio

Ajuste reasoning_effort (none, minimal, low, medium, high, xhigh ou max, conforme o modelo) e o prompt; temperature, top_p, presence_penalty, frequency_penalty, logit_bias e max_tokens não são suportados. max_output_tokens (Responses) e max_completion_tokens (Chat Completions) incluem os tokens de raciocínio: status incomplete por max_output_tokens sem texto pede limite maior. Para auditar, reasoning.summary; tentar extrair o raciocínio bruto viola a política.

Engenharia de prompt e parâmetros

Temperatura vai de 0 a 2: baixa (0,2) é focada e alta (0,7) diverge; mexa em temperatura ou top_p, não nos dois. Few-shot vai em pares user e assistant após a mensagem de sistema e não altera o modelo; a ordem importa pelo viés de recência. Contra invenção no RAG: citação em linha junto de cada afirmação e uma saída como 'não encontrado'; só repetir 'não invente' não basta.

RAG, fine-tuning, destilação e prompt caching

RAG (recuperar, aumentar, gerar) traz conteúdo privado que muda, com citações; fine-tuning muda comportamento, estilo ou formato, não acrescenta conhecimento volátil; destilação usa o modelo grande como professor para treinar um menor. Prompt caching vem habilitado nos modelos suportados e exige no mínimo 1.024 tokens com os primeiros 1.024 idênticos: conteúdo fixo no início, parte variável no fim.

Avaliadores de RAG e de agente

RAG: Retrieval (contexto sem rótulos), Document Retrieval (com qrels, NDCG), Groundedness (1 a 5, LLM juiz) ou Groundedness Pro (preview, aprovado ou reprovado via Content Safety), Relevance e Response Completeness (omissões frente ao ground_truth). Agente: Task Completion, Task Adherence, Intent Resolution, Task Navigation Efficiency, Tool Call Accuracy, Tool Selection, Tool Input Accuracy, Tool Output Utilization e Tool Call Success; Tool Use Quality agrupa vários.

Observabilidade e análise de erros

Comece pelo rastreamento no servidor (Application Insights conectado) e adicione instrumentação no cliente com OpenTelemetry para ver a lógica do app. Durações dos spans execute_tool e invoke_agent mostram onde está a latência; a página Traces busca pelo response ID; a aba Monitor mostra tokens, latência e taxa de sucesso. Muitas reprovações = cluster analysis. Avaliar interações passadas = fonte azure_ai_responses com os IDs em file_content (file_id dá 400).

Macetes para memorizar

💡 Function calling

O modelo PEDE, o app EXECUTA e devolve function_call_output com o call_id.

💡 MCP

allowed_tools = o que pode; require_approval = o que precisa de visto; request chega, response volta.

Orquestrações

Fila = Sequential; todos juntos = Concurrent; passa o bastão = Handoff; mesa de debate = Group Chat; chefe que replaneja = Magentic.

Modelos de raciocínio

Sem temperatura, com esforço: reasoning_effort controla, max_output_tokens inclui o pensamento, summary mostra.

Avaliadores de ferramenta

Escolha = Selection; argumentos = Input Accuracy; uso do retorno = Output Utilization; falha técnica = Call Success.

Prompt caching

Fixo na frente, variável atrás, pelo menos 1.024 tokens iguais.

RAG x fine-tuning

Saber coisa nova = RAG; jeito de responder = fine-tuning; grande ensina pequeno = destilação.

Tabela de memorização: não confunda

Quando é...	...e quando é
function_call_output Chegou function_call de uma function tool: a aplicação executa a função e devolve o resultado com o mesmo call_id.	**mcp_approval_response** Chegou mcp_approval_request de uma ferramenta MCP com require_approval: devolve approve e approval_request_id com previous_response_id.
conversations.create (conversation) Histórico persistente mantido pelo serviço por dias, retomável em outro dispositivo.	**previous_response_id** Encadear rapidamente a nova resposta à anterior ou continuar após uma aprovação MCP.
Handoff A triagem passa a conversa inteira ao especialista, que assume a tarefa, sem orquestrador central.	**Agente como ferramenta** O agente principal mantém a responsabilidade pela resposta e chama especialistas como ferramentas.
Concurrent Perspectivas independentes sobre a mesma entrada ao mesmo tempo, com agregação e menor latência total.	**Sequential** Estágios fixos em que cada um depende da saída do anterior.
Group Chat (maker-checker) Redator e revisor iterando até aprovar, com limite de rodadas.	**Magentic** Problema aberto sem plano definido, com gerente que planeja e replaneja dinamicamente.
Groundedness Detectar invenção fora do contexto recuperado; precisa de response e context, sem gabarito.	**Response Completeness** Detectar omissão frente à resposta esperada; precisa de ground_truth.

Tool Selection O agente escolhe a ferramenta errada.	**Tool Input Accuracy** A ferramenta está certa, mas os parâmetros vêm em formato inválido ou incompletos.

Números que caem

Item	Valor
Expiração das execuções com function calling	**10 minutos após a criação**
Prompt caching: tamanho mínimo elegível	**1.024 tokens, com os primeiros 1.024 idênticos**
Ferramenta Azure AI Search: top_k padrão	**5**
Code Interpreter: sessão	**ativa por até 1 hora, com 30 minutos de ociosidade**
Memory: limites por store	**até 100 escopos por store e 10.000 memórias por escopo**
Temperatura: faixa	**0 a 2 (0,2 focado; 0,7 divergente)**
Groundedness: escala	**nota de 1 a 5**
Painel de monitoramento: sinais de alerta	**latência acima de 10 segundos; taxa de sucesso abaixo de 95%**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Responder a um mcp_approval_request com function_call_output ou com uma mensagem de usuário dizendo 'aprovado'.
Como não cair: Aprovação MCP só avança com mcp_approval_response (approve e approval_request_id) + previous_response_id.

⚠ **Alerta de prova:** Resolver a aprovação trocando require_approval para never.
Como não cair: Isso elimina o controle humano exigido. Para restringir ferramentas use allowed_tools; para revisar, mantenha require_approval.

⚠ **Alerta de prova:** Esperar que o Agent Service execute a function tool sozinho.
Como não cair: Function tools rodam no cliente: execute e devolva function_call_output com o call_id.

⚠ **Alerta de prova:** Aumentar o timeout da execução ou bloquear a função até o gerente aprovar.
Como não cair: A execução expira em 10 minutos e não há timeout de 24 horas a configurar. Devolva status pendente e consulte depois.

⚠ **Alerta de prova:** Manter temperature=0 e top_p num modelo de raciocínio para estabilizar as respostas.
Como não cair: Modelos de raciocínio rejeitam esses parâmetros (e penalties, logit_bias, max_tokens). Ajuste reasoning_effort e as instruções.

⚠ **Alerta de prova:** Pedir no prompt 'mostre todo o raciocínio interno' ou subir reasoning_effort para vê-lo.
Como não cair: reasoning_effort controla quanto pensa, não expõe nada. Use reasoning.summary; extrair o bruto pode levar a limitação ou suspensão.

⚠ **Alerta de prova:** max_output_tokens só conta o texto visível, então resposta incomplete sem texto é bug.
Como não cair: O limite inclui os tokens de raciocínio. Aumente max_output_tokens.

⚠ **Alerta de prova:** Gerar embeddings pelo endpoint do projeto ou por connection string de hub.
Como não cair: Embeddings = OpenAI SDK no /openai/v1 do recurso.

⚠ **Alerta de prova:** Guardar histórico em vector store, nas instruções ou criando versão nova do agente a cada mensagem.
Como não cair: Histórico = Conversations. Versões são configuração imutável e vector store é para busca em arquivos.

⚠ **Alerta de prova:** Copiar o acervo para um File search quando já existe índice no Azure AI Search.
Como não cair: Índice existente = ferramenta Azure AI Search pela conexão do projeto, sem duplicar documentos.

⚠ **Alerta de prova:** Reduzir top_k para 1 ou pôr o código do cliente no prompt para isolar documentos por cliente ou região.
Como não cair: Escopo fixo de recuperação = parâmetro filter da ferramenta Azure AI Search.

⚠ **Alerta de prova:** Usar Connected Agents ou a2a_preview para delegar a um agente parceiro em GA.
Como não cair: Connected Agents saiu do Foundry atual e a2a_preview é o protocolo 0.3. Use a ferramenta a2a (A2A 1.0).

⚠ **Alerta de prova:** Fazer fine-tuning a cada mudança do regulamento para responder com a fonte.
Como não cair: Conteúdo que muda e precisa de citação = RAG. Fine-tuning muda comportamento e estilo.

⚠ **Alerta de prova:** Colocar regra regulatória nas instruções com temperature 0 ou numa votação entre agentes.
Como não cair: Decisão determinística e reproduzível = executor de código no workflow; o LLM só redige.

Como cai na prova

📝 **Como cai:** Trecho de fluxo de código (item retornado pelo agente) e pergunta qual o próximo passo da aplicação.
Como resolver: Identifique o tipo do item: function_call pede function_call_output com call_id; mcp_approval_request pede mcp_approval_response com previous_response_id; status queued ou tarefa longa pede polling. Descarte alternativas que recriam o agente ou desligam a aprovação.

📝 **Como cai:** Descrição de colaboração entre agentes e pergunta qual padrão de orquestração.
Como resolver: Pergunte: a ordem é fixa? (Sequential) As análises são independentes e simultâneas? (Concurrent) Alguém assume a conversa? (Handoff) Há ciclo de crítica com limite? (Group Chat) O plano é aberto? (Magentic)

📝 **Como cai:** Sintoma de qualidade (inventou, omitiu, ferramenta errada, parâmetro errado) e pergunta qual avaliador.
Como resolver: Mapeie o sintoma: inventou fora do contexto = Groundedness; omitiu do gabarito = Response Completeness; tem qrels = Document Retrieval; ferramenta errada = Tool Selection; parâmetro errado = Tool Input Accuracy; não entendeu o pedido = Intent Resolution.

📝 **Como cai:** Modelo de raciocínio com custo, latência ou erro de parâmetro.
Como resolver: Erro de parâmetro = remova temperature e top_p. Lento em tarefa simples = reasoning_effort none ou minimal. Teto de custo = max_output_tokens (Responses) ou max_completion_tokens (Chat). Resposta incomplete sem texto = aumente o limite.

📝 **Como cai:** Escolha de ferramenta para o agente (índice existente, arquivo da conversa, web atual, cálculo, API REST, outro agente).
Como resolver: Índice existente = Azure AI Search; arquivo enviado = File search; web atual sem recurso Bing = Web search; cálculo e gráfico = Code Interpreter; API descrita por OpenAPI = OpenAPI; lógica que o app executa = function; outro agente = a2a.

📝 **Como cai:** Escolha TRÊS recursos para um agente que recupera normas, consulta sistema interno e lembra preferências.
Como resolver: Some uma peça por requisito: ferramenta Azure AI Search (recuperação), function tool (sistema que o app executa) e busca de memória (preferências entre sessões).

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
resposta traz function_call com call_id e o agente não conclui	**Executar a função e enviar function_call_output com o mesmo call_id**
require_approval always e item mcp_approval_request	**mcp_approval_response com approve e approval_request_id + previous_response_id**
servidor MCP com dezenas de ferramentas, agente só pode usar algumas	**allowed_tools**
agente não lembra a mensagem anterior	**conversations.create e conversation=id em cada responses.create**
pedido de vários minutos, front-end encerra a conexão	**background=true e polling do status**
agente já escrito em LangGraph, sem reescrever	**Hosted agent em contêiner**
definição do agente versionada no código, sem recurso no Foundry	**Responses API no endpoint do projeto (agente efêmero)**
triagem passa a conversa inteira ao especialista	**Handoff**
redator e revisor em rodadas até aprovar, com limite	**Group Chat (maker-checker) com limite de iterações**
três análises independentes do mesmo documento ao mesmo tempo	**Concurrent com agregação**
CSV enviado, cálculo exato e gráfico para baixar	**Code Interpreter (container_file_citation)**
API com especificação OpenAPI, sem código cliente	**Ferramenta OpenAPI (identidade gerenciada ou conexão do projeto com chave)**
erro de parâmetro não suportado em modelo GPT-5 de raciocínio	**Remover temperature e top_p; usar reasoning_effort**
resposta inventa dado ausente dos trechos recuperados	**Groundedness (avaliar) e citação em linha + 'não encontrado' (prompt)**

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

- function_call pede function_call_output (call_id); mcp_approval_request pede mcp_approval_response (previous_response_id).
- Modelos de raciocínio: sem temperature e top_p; reasoning_effort ajusta, max_output_tokens inclui o raciocínio, reasoning.summary explica.
- Handoff = especialista assume a conversa; Group Chat = ciclo redator-revisor com limite; Concurrent = paralelo e independente.
- Endpoint do projeto (api/projects/nome) para agentes e Responses API; /openai/v1 do recurso para embeddings.
- Regra determinística vai em código (executor), nunca nas instruções do modelo.

Essenciais para a prova (checklist)

- ☐ Sei criar o cliente do projeto com AIProjectClient, endpoint do projeto e credencial Entra, e quando usar o OpenAI SDK em /openai/v1.
- ☐ Sei quando usar chamada direta, prompt agent, hosted agent e agente efêmero.
- ☐ Sei completar o ciclo de function calling e por que a execução expira em 10 minutos.
- ☐ Sei configurar allowed_tools e require_approval e responder a uma solicitação de aprovação MCP.
- ☐ Sei configurar a ferramenta OpenAPI com identidade gerenciada ou chave em conexão do projeto.
- ☐ Sei manter histórico com Conversations e isolar memória por usuário com scope {{$userId}}.
- ☐ Sei escolher entre Sequential, Concurrent, Handoff, Group Chat, Magentic e agente como ferramenta.
- ☐ Sei quais parâmetros os modelos de raciocínio não aceitam e como limitar custo e ver o raciocínio.
- ☐ Sei associar cada sintoma ao avaliador de RAG ou de agente correto.
- ☐ Sei localizar latência por span, investigar por response ID e usar cluster analysis.

Questões de revisão do capítulo

R1.1 · Implementar soluções de IA generativa e agênticas

Analistas de uma geradora de energia enviam ao agente planilhas CSV com medições horárias de 30 usinas e pedem cálculo de fator de capacidade, comparações entre meses e um gráfico para o relatório. As respostas atuais trazem erros aritméticos porque o modelo calcula sem executar código. Qual ferramenta você deve adicionar ao agente?

A. O File Search, que indexa as planilhas em um vector store para recuperação
B. O Code Interpreter, que executa Python em ambiente isolado e pode gerar gráficos
C. A ferramenta de pesquisa na web, para buscar fatores de capacidade publicados
D. A ferramenta Azure AI Search, com as planilhas indexadas em um índice híbrido próprio

R1.2 · Implementar soluções de IA generativa e agênticas

O assistente de RAG de uma corretora de seguros recebe trechos de apólices no prompt, mas às vezes inventa respostas quando a informação não está nos trechos. Sem trocar o modelo, quais instruções de prompt reduzem esse problema segundo as boas práticas de engenharia de prompt?

A. Pedir que o modelo deixe os trechos de lado e responda pelas condições gerais que conhece
B. Pedir citações em linha para cada afirmação e responder 'não encontrado' se faltar informação
C. Pedir que o modelo use o próprio conhecimento quando os trechos parecerem incompletos
D. Pedir respostas mais longas e detalhadas para que o modelo explique melhor cada cláusula

R1.3 · Implementar soluções de IA generativa e agênticas

Uma varejista quer gerar, a partir do app de backoffice em Python, resumos de avaliações de produtos com um modelo implantado no projeto do Foundry. Não há ferramentas, instruções persistentes nem histórico de conversa, e a equipe não quer criar nem versionar um agente para isso. Qual abordagem você deve usar no código?

A. Criar um prompt agent com create_version e chamá-lo por agent_reference
B. Criar um hosted agent em contêiner e invocá-lo pelo protocolo Responses
C. Obter o cliente com get_openai_client e chamar responses.create com o modelo
D. Criar uma conversa com conversations.create antes de cada resumo pedido

R1.4 · Implementar soluções de IA generativa e agênticas

Uma operadora logística quer uma única implantação que escolha, a cada requisição, entre vários LLMs para equilibrar custo e qualidade. Por compliance, só três modelos aprovados podem processar os dados, e a solução deve continuar respondendo se um deles tiver falhas transitórias. Como você deve configurar o model router?

A. Implantação personalizada com o modo Quality e sem subconjunto definido
B. Implantação rápida padrão, que roteia entre os modelos suportados da região
C. Três implantações separadas com balanceamento por um gateway de API
D. Implantação personalizada com um subconjunto de pelo menos dois modelos aprovados

Respostas comentadas

R1.1. B) O Code Interpreter, que executa Python em ambiente isolado e pode gerar gráficos

Por quê: Item do guia 'Projetar fluxos de trabalho, fluxos aprimorados por ferramentas e pipelines de raciocínio em várias etapas': ferramentas de execução de código dão precisão às etapas numéricas que o modelo sozinho erra.

✗ A) Incorreta. O File Search recupera trechos de texto; não executa cálculos sobre as medições.
✗ C) Incorreta. Valores publicados na web não substituem os cálculos sobre os dados da própria empresa.
✗ D) Incorreta. Indexar planilhas permite pesquisá-las, mas o modelo continuaria calculando sem executar código.

> **Dica de prova:** Cálculo e gráfico sobre arquivos = Code Interpreter.

R1.2. B) Pedir citações em linha para cada afirmação e responder 'não encontrado' se faltar informação

Por quê: Entre as boas práticas: fornecer dados de fundamentação, especificar a estrutura da saída com citações (em linha funcionam melhor que no final) e dar ao modelo uma saída quando ele não souber a resposta.

✗ A) Incorreta. Descartar os trechos elimina a fundamentação que o RAG deveria fornecer.
✗ C) Incorreta. Usar conhecimento próprio é exatamente o que gera respostas sem fundamento nos documentos.
✗ D) Incorreta. Respostas mais longas aumentam a chance de conteúdo não sustentado pelos trechos.

> **Dica de prova:** Contra invenção em RAG: citação em linha + 'não encontrado'.

R1.3. C) Obter o cliente com get_openai_client e chamar responses.create com o modelo

Por quê: Para implantar e consumir modelos no Foundry, comece por uma chamada direta ao modelo quando não houver ferramentas ou orquestração; agentes entram quando são necessárias instruções persistentes, ferramentas ou versionamento.

✗ A) Incorreta. Prompt agents servem quando há instruções, ferramentas e versionamento; aqui a equipe não quer criar um agente.
✗ B) Incorreta. Hosted agents executam código próprio de agente em contêiner; é complexidade desnecessária para uma chamada simples ao modelo.
✗ D) Incorreta. Conversas guardam histórico multiturno; resumos independentes não precisam delas.

> **Dica de prova:** Só enviar prompt ao modelo? project.get_openai_client() + responses.create(model=implantação).

R1.4. D) Implantação personalizada com um subconjunto de pelo menos dois modelos aprovados

Por quê: O model router é um modelo treinado que escolhe o LLM por requisição, com os modos Balanced (padrão), Cost e Quality. Ele honra os limites de zona de dados, tem a janela de contexto limitada pelo menor modelo e decide o roteamento só pelo texto, mesmo quando há imagens.

✗ A) Incorreta. O modo Quality prioriza a qualidade, mas sem subconjunto o roteamento inclui modelos não aprovados.
✗ B) Incorreta. A implantação padrão pode rotear para modelos suportados fora da lista aprovada, violando a restrição de compliance.
✗ C) Incorreta. Isso exige construir a lógica de roteamento por conta própria, que é o que o model router evita.

> **Dica de prova:** Router com compliance = subconjunto de modelos (dois ou mais para failover).

02

CAPÍTULO 2 · 10-15% DA PROVA

Implementar soluções de análise de texto

Apesar do nome, este domínio junta texto, fala e tradução. Preciso saber o que continua como capacidade principal do Azure Language (PII, detecção de idioma, NER, custom NER, Text analytics for health) e o que é legado (sentimento, frases-chave, resumo, entity linking), porque para projeto novo de sentimento e tom a resposta é LLM com structured outputs. Na fala, cai a escolha entre tempo real, fast transcription, batch transcription, custom speech, SSML, custom voice, Voice Live e Live Interpreter; na tradução, Translator 2026-06-06 com NMT ou LLM, tradução adaptativa, Custom Translator e tradução de documentos. Onde mais erro: trocar custom speech (reconhecimento) por custom voice ou SSML (síntese), e esquecer que a tradução por LLM não funciona com private endpoint.

Neste capítulo

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

Tópicos-chave

Azure Language: capacidades principais e legadas

Principais: detecção de PII, detecção de idioma, NER pré-configurado e personalizado (custom NER) e Text analytics for health. Legadas, mantidas para implementações existentes: CLU, classificação personalizada, frases-chave, sentimento e mineração de opiniões, resumo e entity linking. Sentimento, mineração de opiniões e resumo têm aposentadoria em 31/03/2029; entity linking em 01/09/2028.

Tipos de detecção de PII

Text PII: strings avulsas, síncrono, para prompts e logs antes do LLM. Conversation PII: transcrições e chats com turnos, enviados como job assíncrono para manter o contexto entre turnos. Document PII: arquivos nativos (PDF, DOCX) em pipelines baseados em armazenamento, preservando a formatação e devolvendo metadados JSON das ocultações. piiCategories limita categorias; characterMask (padrão) e entityMask definem a redação.

Text analytics for health

Extrai entidades médicas (medicamentos, dosagens, diagnósticos), relações entre elas e asserções como negação ('paciente nega febre'), sem treinar modelo. Custom NER exigiria rotular e treinar e não traz relações nem asserções.

Custom NER

Extrai entidades próprias do domínio (código de subestação, número de máquina) a partir de exemplos rotulados; é capacidade principal do Azure Language e oferece contêineres Docker para rodar on-premises. NER pré-configurado só reconhece categorias predefinidas; classificação personalizada atribui classe ao documento inteiro e é legada.

Sentimento e tom em projeto novo

Use um modelo do Foundry com prompt de classificação e structured outputs (campos enum para rótulos próprios como irônico, irritado ou ansioso). O sentimento do Azure Language é legado e não identifica tom como ironia; Prompt Shields detecta ataques, não tom.

Structured outputs

Garantem aderência ao esquema: json_schema com strict true, todos os campos em required (para simular opcional, união com null) e additionalProperties false. Na Chat Completions vai em response_format; na Responses API em text.format; em Python, responses.parse com uma classe Pydantic em text_format e leitura em output_parsed. JSON mode (json_object) só garante JSON válido; chamadas de função paralelas não são suportadas (parallel_tool_calls false).

Personalizar saídas para tarefas de domínio

A escada é engenharia de prompt e few-shot, depois RAG se faltar contexto e fine-tuning (SFT com pares em JSONL) quando é preciso mudar estilo e estrutura com consistência. Formato garantido vem do esquema estrito; estilo e terminologia vêm de instruções com exemplos. Para avaliar o modelo ajustado de forma barata e temporária, implantação Developer.

Azure Translator 2026-06-06

Permite escolher por solicitação entre NMT (deploymentName general) e uma implantação de LLM (deploymentName com o nome da implantação), com controles tone e gender e adaptação por pares de referência só na tradução por LLM. O NMT é cobrado por caracteres e o LLM por tokens; reserve o LLM para tom e contexto e mantenha o volume no NMT. A tradução por LLM não funciona quando o Translator usa private endpoint.

Tradução adaptativa, Custom Translator e dicionário dinâmico

Tradução personalizada adaptativa aplica algumas centenas de pares alinhados nas requisições, ideal para terminologia que muda rápido, sem treino. Custom Translator treina um NMT dedicado com corpus grande e estável (deploymentName com o categoryID), funciona com private endpoint, e cada atualização exige retreino e reimplantação (pode levar cerca de 48 horas). Dicionário dinâmico (mstrans:dictionary) exige inglês na origem ou no destino e só é seguro para substantivos compostos.

Tradução de documentos

Preserva formatação e layout de arquivos. Síncrona: um arquivo por requisição, sem conta de armazenamento, com glossário opcional e arquivo devolvido na resposta. Assíncrona em lote: vários arquivos, com contêineres de origem e destino no Blob Storage. A tradução de texto trabalha com strings e não preserva o arquivo.

Modos de fala para texto

Tempo real para microfone ou fluxo ao vivo; fast transcription para arquivos individuais com resposta síncrona e latência previsível; batch transcription para grandes volumes assíncronos no Blob Storage, com diarização e identificação de idioma opcionais. Phrase lists não estão disponíveis na transcrição em lote.

Custom speech

Melhora o reconhecimento (STT): texto do domínio ajusta o vocabulário, e áudio com transcrição humana ajusta condições acústicas como ruído. Avalie com WER comparando com transcrição humana. Endpoints hospedados são cobrados; na transcrição em lote o modelo personalizado é usado sem endpoint. Personalização leve em tempo de execução = lista de frases.

Texto para fala: SSML e custom voice

SSML ajusta a síntese: pronúncia com phoneme ou lexicon personalizado, pausas, estilos e várias vozes, sem trocar a voz. Custom voice cria voz sintética de uma pessoa real; é recurso de acesso limitado por elegibilidade e exige gravação da declaração de consentimento do locutor antes do treino. SSML não clona timbre.

Voice Live API

Agente de voz em tempo real sem orquestrar manualmente STT, LLM e TTS: baixa latência de fala para fala, supressão de ruído, cancelamento de eco, detecção natural de fim de turno, voz personalizada, avatares e chamadas de função. É compatível com os eventos da Realtime API, conecta por WebSocket a modelos como gpt-realtime ou a um agente do Foundry Agent Service e aceita listas de frases e custom speech; cobrança em níveis Pro, Standard e Lite.

Tradução de fala

A API padrão exige especificar o idioma de origem. O Live Interpreter identifica continuamente o idioma, aceita troca de idioma no meio da sessão e entrega fala traduzida com voz natural que preserva o estilo. Uma chamada devolve até dois idiomas de destino; a partir do terceiro há cobrança de tradução de texto. O resultado pode ser texto, fala sintetizada ou ambos; tradução de vídeo é para arquivos gravados.

Raciocínio multimodal sobre áudio

Para avaliar como algo foi dito (hesitação, tom, interrupções), envie o áudio a um modelo com entrada de áudio (como gpt-audio, gpt-4o-audio-preview ou gpt-4o-mini-audio-preview) na Chat Completions, em uma parte input_audio com dados base64 e formato wav; o parâmetro modalities pede saída em texto e, se quiser, áudio. A transcrição descarta essas pistas da voz.

Macetes para memorizar

💡 PII

String = Text; diálogo = Conversation; arquivo = Document.

💡 Fala para texto

Ao vivo = tempo real; um arquivo já = fast; montanha de arquivos = batch.

💡 Custom speech x custom voice

Speech ESCUTA melhor; Voice FALA com a voz de alguém.

💡 Translator 2026-06-06

general = NMT; nome da implantação = LLM; tom só com LLM; LLM não entra por private endpoint.

💡 Tradução de fala

Até dois destinos inclusos na chamada; do terceiro em diante, cobrança de tradução de texto.

💡 Azure Language

Fica: PII, idioma, NER, custom NER, saúde. Sai: sentimento, frases-chave, resumo, entity linking.

Tabela de memorização: não confunda

Quando é...	...e quando é

Text PII String avulsa (anotação, prompt, log) mascarada de forma síncrona antes do LLM.	**Conversation PII** Transcrição com turnos e interlocutores, como job assíncrono, mantendo contexto entre turnos; arquivo PDF ou DOCX é Document PII.
Fast transcription Um arquivo por vez, resposta síncrona e rápida com latência previsível.	**Batch transcription** Milhares de gravações no Blob, assíncrono, sem pressa, com diarização e opção de custom speech sem endpoint.
Custom speech O reconhecimento erra jargões ou sofre com ruído (fala para texto).	**SSML / custom voice** A voz sintética pronuncia errado (SSML phoneme ou lexicon) ou precisa ser a voz de uma pessoa (custom voice).
Live Interpreter Idioma de entrada desconhecido ou trocando no meio da fala, com voz natural.	**Tradução de fala padrão** Idioma de origem conhecido e fixado no início da sessão.
Tradução adaptativa Poucos pares, terminologia que muda toda semana, sem treinar nem reimplantar.	**Custom Translator** Corpus grande e estável (dezenas de milhares de pares) ou recurso acessível só por private endpoint.
Tradução de documentos síncrona Um arquivo por vez, sem conta de armazenamento, arquivo devolvido na resposta.	**Tradução de documentos em lote** Milhares de arquivos em processo noturno, com contêineres no Blob Storage.
JSON mode Só precisa de JSON válido, sem esquema fixo.	**Structured outputs (json_schema strict)** Campos obrigatórios e sem extras, validados contra esquema ou classe Pydantic.

Números que caem

Item	Valor
Sentimento, mineração de opiniões e resumo do Azure Language: aposentadoria	**31/03/2029**
Entity linking: aposentadoria	**01/09/2028**
Tradução de fala: idiomas de destino por chamada	**até dois; o terceiro gera cobrança de tradução de texto**
Custom Translator: retreino e reimplantação	**pode levar até cerca de 48 horas**
Voice Live: localidades de reconhecimento	**mais de 140**
Translator: versão de API com escolha entre NMT e LLM	**2026-06-06**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Usar sentimento e mineração de opiniões do Azure Language em projeto novo de tom e ironia.
Como não cair: São legados (aposentadoria 31/03/2029) e não identificam ironia. Use LLM com structured outputs.

⚠ **Alerta de prova:** Usar o NER pré-configurado para ocultar CPF, nomes e telefones.
Como não cair: A capacidade feita para detectar e redigir dados pessoais é a detecção de PII.

⚠ **Alerta de prova:** Enviar cada fala da transcrição isoladamente para a Text PII.
Como não cair: Perde o contexto entre turnos. Conversation PII como job assíncrono.

⚠ **Alerta de prova:** JSON mode com o esquema descrito no prompt garante os campos.
Como não cair: JSON mode só garante JSON válido. Use structured outputs com strict (text.format na Responses API).

⚠ **Alerta de prova:** response_format na Responses API.
Como não cair: response_format é da Chat Completions; na Responses API o esquema vai em text.format.

⚠ **Alerta de prova:** deploymentName general com tone formal.
Como não cair: general é o NMT; tone e gender só valem com deploymentName apontando para a implantação do LLM.

⚠ **Alerta de prova:** Tradução por LLM com tom formal num Translator acessível só por private endpoint.
Como não cair: LLM não funciona com private endpoint. Terminologia com corpus grande = Custom Translator (NMT).

⚠ **Alerta de prova:** Phrase list na transcrição em lote para jargões e ruído.
Como não cair: Phrase lists não existem no lote e não resolvem ruído. Treine custom speech com áudio e use no lote sem endpoint.

⚠ **Alerta de prova:** Hospedar o custom speech em endpoint dedicado para um trabalho noturno em lote.
Como não cair: Endpoint hospedado gera custo; a transcrição em lote usa o modelo personalizado sem endpoint.

⚠ **Alerta de prova:** Corrigir pronúncia da voz sintetizada com custom speech.
Como não cair: Custom speech é reconhecimento. Pronúncia = SSML com phoneme ou lexicon.

⚠ **Alerta de prova:** Imitar o timbre de um dublador com SSML ou treinar custom speech com as gravações dele.
Como não cair: Voz de pessoa real = custom voice, com acesso limitado aprovado e consentimento gravado.

⚠ **Alerta de prova:** Encadear Speech SDK, LLM e síntese ou usar fast transcription para um agente de voz ao vivo.
Como não cair: Agente de voz em tempo real sem orquestrar componentes = Voice Live API.

⚠ **Alerta de prova:** Cada idioma de destino da tradução de fala exige sessão separada.
Como não cair: Uma chamada devolve até dois destinos; o terceiro gera cobrança de tradução de texto.

⚠ **Alerta de prova:** Transcrever e mandar o texto ao LLM para avaliar hesitação e tom.
Como não cair: A transcrição descarta as pistas da voz. input_audio num modelo com entrada de áudio.

Como cai na prova

📝 **Como cai:** Cenário de análise de texto e alternativas que misturam recursos principais e legados do Azure Language.
Como resolver: Descarte legados (frases-chave, sentimento, resumo, entity linking) quando o enunciado fala em projeto novo. PII para dados pessoais, saúde para clínico, custom NER para entidades próprias; tom e rótulos próprios = LLM com structured outputs.

📝 **Como cai:** Problema de fala: erro de reconhecimento, pronúncia, voz da marca ou agente de voz.
Como resolver: Primeiro decida a direção: reconhecimento (fala para texto) ou síntese (texto para fala). Reconhecimento: phrase list ou custom speech. Síntese: SSML ou custom voice. Conversa ao vivo sem orquestrar: Voice Live.

📝 **Como cai:** Transcrição com volume, latência e custo definidos.
Como resolver: Um arquivo e resposta já = fast transcription. Grande volume sem pressa = batch transcription (com custom speech sem endpoint se houver jargão). Ao vivo = tempo real ou Voice Live.

📝 **Como cai:** Tradução de texto com tom, terminologia e restrição de rede.
Como resolver: Tom ou gênero = LLM no Translator 2026-06-06 (deploymentName = implantação). Private endpoint elimina LLM: Custom Translator. Poucos pares e mudança frequente = adaptativa. Arquivos com layout = tradução de documentos (síncrona ou em lote).

📝 **Como cai:** Garantia de formato JSON para extração ou classificação.
Como resolver: Procure json_schema com strict, todos os campos em required e additionalProperties false, no parâmetro certo da API (text.format na Responses, response_format na Chat Completions). Regex, temperatura 0 e few-shot não garantem.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
ocultar dados pessoais mantendo o PDF ou DOCX original	**Document PII**
mascarar CPF e nomes em texto antes do LLM, resposta síncrona	**Text PII do Azure Language**
medicamento, dosagem, relação entre eles e negação de sintoma	**Text analytics for health**
entidades próprias, exemplos rotulados, on-premises	**Custom NER em contêiner Docker**
sentimento e tom em JSON fixo, projeto novo	**Modelo do Foundry + structured outputs com enum**
objeto tipado em Python com a Responses API	**responses.parse com Pydantic em text_format**
termos técnicos saem errados na transcrição	**Custom speech (medir WER); leve = lista de frases**
voz sintetizada pronuncia nome de medicamento errado	**SSML com phoneme ou lexicon**
agente de voz por telefone, ruído, eco, fim de turno natural	**Voice Live API**
palestrante troca de idioma, não informar idioma, voz natural ao vivo	**Live Interpreter**
milhares de gravações no Blob, transcrever à noite separando falantes	**Batch transcription com diarização**
terminologia muda toda semana, algumas centenas de pares	**Tradução personalizada adaptativa**
um DOCX por vez com glossário, sem conta de armazenamento	**Tradução de documentos síncrona**
analisar hesitação no próprio áudio, sem transcrever	**input_audio (base64, wav) em modelo de áudio**

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

- Projeto novo de sentimento e tom = LLM + structured outputs; sentimento do Azure Language é legado.
- PII: Text para string, Conversation para turnos, Document para arquivo com formatação.
- Custom speech melhora o reconhecimento; SSML e custom voice mexem na síntese.
- Tradução por LLM não funciona com private endpoint; idioma desconhecido ao vivo = Live Interpreter.

Essenciais para a prova (checklist)

- ☐ Sei listar as capacidades principais e as legadas do Azure Language.
- ☐ Sei escolher entre Text, Conversation e Document PII.
- ☐ Sei quando usar Text analytics for health e custom NER.
- ☐ Sei garantir esquema com structured outputs na Chat Completions e na Responses API.
- ☐ Sei configurar o Translator 2026-06-06 com NMT ou LLM e quais controles só o LLM tem.
- ☐ Sei escolher entre tradução adaptativa, Custom Translator, dicionário dinâmico e tradução de documentos.
- ☐ Sei escolher entre tempo real, fast transcription e batch transcription.
- ☐ Sei melhorar o reconhecimento com phrase list e custom speech e medir com WER.
- ☐ Sei quando usar Voice Live, Live Interpreter e input_audio.

Questões de revisão do capítulo

R2.1 · Implementar soluções de análise de texto

Um e-commerce brasileiro vai expandir para a Espanha e precisa traduzir descrições de produtos do português para o espanhol com tom formal, usando o Azure Translator em Foundry Tools na versão de API 2026-06-06 e a tradução feita por um modelo GPT implantado. Como você deve configurar cada item de targets na solicitação?

A. language es, deploymentName com a implantação do LLM e tone formal
B. language es, deploymentName general e tone formal
C. language es, deploymentName com a implantação do LLM e textType html
D. language es, deploymentName com o categoryID do Custom Translator e tone formal

R2.2 · Implementar soluções de análise de texto

Um banco gera resumos de relatórios de prevenção à lavagem de dinheiro com um modelo do Foundry. Mesmo com instruções detalhadas e exemplos few-shot, os resumos variam no uso da terminologia interna e na estrutura exigida pelo compliance. A equipe tem 3.000 pares de relatório e resumo aprovados. Qual abordagem você deve usar para personalizar a saída?

A. Aumento da temperatura para gerar resumos mais variados
B. RAG com os resumos aprovados indexados no Azure AI Search como exemplos
C. Summarization do Azure Language com resumo abstrativo
D. Fine-tuning supervisionado (SFT) com os pares aprovados em JSONL

R2.3 · Implementar soluções de análise de texto

Uma produtora transmite ao vivo um festival em que os apresentadores alternam entre português, espanhol e inglês sem aviso. Ela quer oferecer à audiência, em tempo real, áudio traduzido com voz natural que preserve o estilo do apresentador, sem precisar definir o idioma de entrada. Qual recurso do Azure Speech você deve usar?

A. A tradução de vídeo para os arquivos gravados após o evento
B. A API de transcrição rápida com diarização habilitada
C. A transcrição em lote com identificação de idioma
D. O Live Interpreter da tradução de fala em tempo real

R2.4 · Implementar soluções de análise de texto

Antes de enviar anotações de sinistros a um LLM para resumo, uma seguradora precisa detectar e mascarar CPF, nomes, telefones e endereços no texto, recebendo a versão já redigida de forma síncrona. Qual recurso você deve usar?

A. Extração de frases-chave do Azure Language, excluindo as frases com dados pessoais
B. Detecção de idioma do Azure Language, roteando textos em português para revisão
C. Detecção de PII de texto do Azure Language, usando o texto redigido retornado
D. Reconhecimento de entidades nomeadas pré-configurado, removendo pessoas citadas

Respostas comentadas

R2.1. A) language es, deploymentName com a implantação do LLM e tone formal

Por quê: Traduzir texto com o Azure Translator ou com fluxos de LLM: a versão 2026-06-06 permite escolher por solicitação entre NMT (general) e uma implantação de LLM, com controles de tom e gênero e adaptação de domínio. A tradução por LLM não funciona quando o Translator usa private endpoint.

✗ B) Incorreta. general é o nome reservado do sistema de tradução neural (NMT); os controles de tom são da tradução por LLM.

✗ C) Incorreta. textType indica se o texto é plain ou html; sem tone formal, o requisito de tom não é atendido.

✗ D) Incorreta. deploymentName com o categoryID usa o modelo de tradução neural personalizado treinado no Custom Translator; os controles tone e gender valem para a tradução por LLM.

> **Dica de prova:** Translator 2026-06-06: deploymentName = LLM para usar tone e gender; general = NMT.

R2.2. D) Fine-tuning supervisionado (SFT) com os pares aprovados em JSONL

Por quê: Personalizar saídas para tarefas de domínio, como sumarização de compliance, segue uma escada: engenharia de prompt e few-shot, depois RAG se faltar contexto e fine-tuning quando é preciso mudar comportamento ou formato com consistência.

✗ A) Incorreta. Temperatura maior aumenta a variação, o oposto do que o compliance precisa.

✗ B) Incorreta. RAG acrescenta conhecimento ao contexto, mas a lacuna aqui é de estilo e estrutura, e não de informação faltante.

✗ C) Incorreta. Summarization do Azure Language é recurso legado e não aprende a terminologia e a estrutura internas do banco.

> **Dica de prova:** Prompt e few-shot não bastaram para o estilo e há milhares de exemplos aprovados? Fine-tuning (SFT).

R2.3. D) O Live Interpreter da tradução de fala em tempo real

Por quê: Item do guia 'Traduzir fala para outros idiomas usando modelos de linguagem e Ferramentas Foundry': a tradução de fala do Azure Speech oferece a API padrão e o Live Interpreter para cenários conversacionais com idioma de entrada não especificado.

✗ A) Incorreta. A tradução de vídeo trabalha sobre arquivos já gravados, não sobre a transmissão ao vivo.

✗ B) Incorreta. A transcrição rápida devolve texto de arquivos e separa falantes, mas não traduz nem sintetiza fala.

✗ C) Incorreta. A transcrição em lote processa áudio gravado e não entrega tradução falada em tempo real.

> **Dica de prova:** Idioma de entrada desconhecido e voz natural ao vivo = Live Interpreter.

R2.4. C) Detecção de PII de texto do Azure Language, usando o texto redigido retornado

Por quê: A detecção de PII é uma capacidade principal (core) do Azure Language, com três tipos: Text PII (síncrono, para prompts e logs), Conversation PII (transcrições com turnos) e Document PII (arquivos nativos como PDF e DOCX, preservando a estrutura).

✗ A) Incorreta. Frases-chave resumem conceitos principais e não detectam nem mascaram dados pessoais.

✗ B) Incorreta. Detectar o idioma não identifica nem oculta informações pessoais.

✗ D) Incorreta. O NER pré-configurado categoriza entidades genéricas; a capacidade feita para detectar e redigir dados pessoais é a detecção de PII.

Dica de prova: Mascarar dado pessoal antes do LLM = Azure Language PII (texto).

03

CAPÍTULO 3 · 10-15% DA PROVA

Implementar soluções de extração de informações

Aqui o foco é o pipeline de recuperação e a extração de documentos: indexadores e skillsets do Azure AI Search (OCR, Document Layout, Content Understanding, Custom Web API, embeddings e vectorizer), configuração de busca híbrida, semântica e vetorial, conexão do índice ou da base do Foundry IQ ao agente, e analisadores do Azure Content Understanding e modelos do Document Intelligence. As alternativas trazem nomes de propriedades reais (imageAction, markdownHeaderDepth, vectorFilterMode, projectionMode, estimateFieldSourceAndConfidence) e só uma resolve o sintoma. Onde mais erro: esquecer o imageAction no PDF escaneado, usar vectorizer com modelo diferente da skill, deixar orderby em consulta híbrida e confundir prebuilt-documentSearch com prebuilt-layout ou prebuilt-read.

Neste capítulo

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

Tópicos-chave

Busca híbrida com RRF

Combina texto completo (search) e vetores (vectorQueries) numa só consulta, mesclados por Reciprocal Rank Fusion: o texto acerta códigos exatos e a parte vetorial acerta linguagem natural. Filtros e facetas funcionam, mas orderby sobrescreve a relevância. Com ranker semântico por cima, use k igual a 50 nas consultas vetoriais.

Ranker semântico

Reordena os 50 primeiros resultados pela compreensão da pergunta e devolve legendas e respostas extraídas literalmente do índice (não cria texto). Exige configuração semântica com título e campos de conteúdo e palavras-chave em ordem de prioridade; cada documento contribui com até 2.048 tokens e os campos de baixa prioridade são truncados. O título tem limite de 128 tokens; @search.rerankerScore vai de 0 a 4; usa queryType semantic e só campos de texto.

HNSW e exhaustive KNN

HNSW é aproximado, fica em memória, consome a cota do índice vetorial e se ajusta por m, efConstruction e efSearch. Exhaustive KNN é exato, por força bruta, para conjuntos pequenos ou para medir revocação. Para comparar sem novo índice, rode a consulta com exhaustive true no campo HNSW; um campo criado como exhaustiveKnn não aceita consulta HNSW.

vectorFilterMode

Define em que etapa o filtro entra na busca vetorial. preFilter aplica o filtro durante a travessia do HNSW e evita sumir resultados com filtro muito seletivo e k pequeno. postFilter filtra depois e pode gerar falsos negativos; strictPostFilter (preview) filtra após o top-k global e tem o maior risco.

Vetorização integrada

Na indexação, o skillset fragmenta (Text Split ou Content Understanding) e gera embeddings com uma skill como Azure OpenAI Embedding; na consulta, um vectorizer definido no índice e ligado ao campo pelo vector profile vetoriza a pergunta. Os dois precisam usar o mesmo modelo de embeddings; mesma dimensão com modelos diferentes não falha, mas devolve resultados irrelevantes. Agende o indexador para reprocessar chamadas limitadas pela cota do modelo.

OCR na ingestão para RAG

PDF escaneado sem camada de texto deixa o campo content vazio. Defina imageAction como generateNormalizedImages no indexador (pré-requisito das skills de imagem; a extração de imagens é cobrada) e use a skill OCR, que usa o Read do Azure Vision e reconhece texto impresso e manuscrito, com Text Merge para gerar o merged_text.

Skill Document Layout

Usa o modelo de layout do Document Intelligence v4.0. outputFormat markdown com markdownHeaderDepth (padrão h6) fragmenta por seção de título, incorporando níveis mais profundos à seção do nível escolhido; outputFormat text com chunkingProperties fragmenta por tamanho. markdownHeaderDepth só vale com markdown, e chunkingProperties só com text. Acima de 20 documentos por indexador por dia exige recurso do Foundry anexado para cobrança.

Skill Azure Content Understanding

GA na API 2026-04-01 do Azure AI Search e recomendada para skillsets novos: entrega tabelas em Markdown, trata tabela que atravessa páginas como unidade e faz a fragmentação (chunks entre páginas), dispensando Text Split. É cobrada pelo preço do Content Understanding, sem os 20 documentos gratuitos por dia, e não é indicada para documentos que levam mais de cinco minutos.

Tipos de skills e Custom Web API

Skills internas (OCR, Text Split, embeddings, GenAI Prompt, Content Understanding), personalizadas (Custom Web API e AML) e utilitárias (Conditional, Shaper, Text Merge, Text Split). Regra ou modelo próprio (por exemplo numa Azure Function) entra pela Custom Web API skill, com o contrato values e recordId. GenAI Prompt usa LLM e Custom Entity Lookup só procura termos de uma lista.

Index projections

Controlam a indexação um-para-muitos: cada chunk vira documento de pesquisa próprio repetindo campos do pai (título, data). projectionMode skipIndexingParentDocuments não indexa o pai inteiro; includeIndexingParentDocuments, o padrão, indexa também o pai. Knowledge store grava enriquecimentos no Storage e não cria documentos no índice.

Busca multimodal

Verbalizar imagens com a GenAI Prompt skill favorece RAG por texto e citações; consulta enviando uma foto exige embeddings multimodais das imagens e o vectorizer equivalente no índice. Muitas soluções combinam os dois caminhos.

Índice para a ferramenta Azure AI Search do agente

O índice precisa de campos de texto pesquisáveis e recuperáveis, campos vetoriais e um campo recuperável com a URL de origem (e, se quiser, título) para citações com link. O parâmetro filter da ferramenta restringe escopo por região ou cliente; top_k e query_type não isolam documentos.

Foundry IQ conectado ao agente

A base de conhecimento do Foundry IQ é exposta ao agente como ferramenta MCP (knowledge_base_retrieve) por uma conexão do projeto (RemoteTool) com a identidade gerenciada do projeto, que precisa de Search Index Data Reader no serviço de busca (403 indica falta dela). Criar a conexão exige Foundry Project Manager. Para conteúdo com ACL, encaminhe o token do usuário no cabeçalho x-ms-query-source-authorization.

Analisadores predefinidos do Content Understanding

De RAG: prebuilt-documentSearch, imageSearch, audioSearch e videoSearch geram Markdown, resumo, descrição de figuras e conteúdo pronto para indexação, sem esquema de campos. Utilitários: prebuilt-read (OCR sem modelo generativo), prebuilt-layout (layout e tabelas sem LLM), prebuilt-documentFieldSchema (propõe esquema) e prebuilt-digitalParse. De domínio: prebuilt-invoice, prebuilt-contract; compostos: prebuilt-procurement, tax.us e mortgage.us.

Campos, confiança e formato no Content Understanding

Valor literal com página, posição e confiança = método extract + estimateFieldSourceAndConfidence (base para aprovação automática e revisão humana focada). A configuração tableFormat markdown troca tabelas em HTML por Markdown; chartFormat cuida de gráficos e annotationFormat só aceita markdown.

Implantações de modelo do Content Understanding

O serviço exige uma implantação de chat completion e uma de embeddings associadas ao recurso. Defina padrões com PATCH em contentunderstanding/defaults e sobrescreva por requisição com modelDeployments. Os tokens são cobrados pela implantação conectada; o Content Understanding cobra só seus próprios medidores.

Classificação e segmentação de lotes mistos

Um PDF com vários documentos é separado e roteado com contentCategories (até 200 categorias com descrição, incluindo other) apontando um analyzerId por tipo e enableSegment true; com false, o arquivo inteiro recebe uma única categoria. Não exige dados de treino, e as APIs dedicadas de classificador foram descontinuadas. O agentic mode não faz esse roteamento.

Document Intelligence: modelos personalizados e prebuilt

Custom neural lida com a mesma informação em layouts variados (estruturados e semiestruturados) e é o ponto de partida; custom template exige layout fixo; composed models com classificador tratam vários tipos. Formulários padronizados com alta precisão e baixa latência ficam no prebuilt de faturas do Document Intelligence (extração determinística); contratos variados que exigem inferência ficam no Content Understanding.

Macetes para memorizar

PDF escaneado

Normaliza, lê, junta: generateNormalizedImages, OCR, Text Merge.

Vetores

Mesmo modelo nos dois lados: skill embute, vectorizer pergunta.

Ranker semântico

Só os 50 primeiros, ordem dos campos importa, nada de orderby.

Filtro vetorial

Filtro apertado? Filtre antes (preFilter).

Analisadores do Content Understanding

Search = RAG; read = só OCR; layout = estrutura sem LLM; invoice e contract = campos de domínio.

Lote misto

Separar e rotear = categorias com analyzerId + other + enableSegment ligado.

Document Intelligence

Layout muda = neural; layout fixo = template.

Tabela de memorização: não confunda

Quando é...	...e quando é
preFilter Filtro muito seletivo com k pequeno, quando resultados relevantes estão sumindo.	**strictPostFilter** Filtrar depois do top-k global, aceitando o maior risco de falsos negativos (preview).
Skill Document Layout Fragmentar por hierarquia de títulos (markdown + markdownHeaderDepth) em skillset existente.	**Skill Azure Content Understanding** Tabelas que atravessam páginas, Markdown para RAG, chunks entre páginas e dispensa de Text Split.
Skill de embedding Gerar vetores dos chunks durante a indexação.	**Vectorizer** Converter a pergunta em vetor no momento da consulta, com o mesmo modelo da skill.
skipIndexingParentDocuments Só os chunks como documentos de pesquisa, com campos do pai repetidos.	**includeIndexingParentDocuments** Indexar também o documento pai inteiro (padrão), misturando formatos no índice.
prebuilt-documentSearch Markdown limpo com figuras descritas, resumo e chunks para RAG, sem esquema.	**prebuilt-layout / prebuilt-read** Extrair layout ou só texto (OCR) sem modelo generativo, sem descrições nem resumo.
Custom neural Mesmos campos em layouts que variam muito entre emissores.	**Custom template** Formulário de layout estático e consistente.
Document Intelligence prebuilt de faturas Documentos padronizados com alta precisão e baixa latência.	**Content Understanding prebuilt-contract** Contratos com cláusulas variadas que pedem campos inferidos e raciocínio, sem dados rotulados.

Números que caem

Item	Valor
Ranker semântico: resultados reclassificados	**50 primeiros**
Ranker semântico: tokens por documento	**até 2.048 tokens resumidos**
Ranker semântico: limite do título	**128 tokens**
@search.rerankerScore	**0 a 4**
Content Understanding: categorias de classificação	**até 200, incluindo other**
Skill Azure Content Understanding: limite indicado	**documentos de até cinco minutos de processamento; GA na API 2026-04-01**
Document Layout: uso sem recurso anexado	**até 20 documentos por indexador por dia**

markdownHeaderDepth padrão	**h6**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Habilitar ranker semântico, detecção de idioma ou parsingMode json para ler PDF escaneado.
Como não cair: Sem imageAction generateNormalizedImages e skill OCR (com Text Merge) o texto da imagem nunca entra no índice.

⚠ **Alerta de prova:** imageAction none com Image Analysis para gerar descrições das imagens.
Como não cair: Com none nenhuma imagem é extraída, e descrição não substitui a transcrição do texto.

⚠ **Alerta de prova:** Vectorizer com outro modelo de mesma dimensão funciona igual.
Como não cair: A consulta não falha, mas os vetores ficam em espaços diferentes. Skill e vectorizer = mesmo modelo.

⚠ **Alerta de prova:** Aumentar top acima de 50 para o ranker semântico reclassificar mais resultados.
Como não cair: Só os 50 primeiros seguem para a reclassificação; ajuste a ordem dos campos na configuração semântica.

⚠ **Alerta de prova:** Colocar o corpo longo no campo de título ou convertê-lo em vetor para entrar na reclassificação.
Como não cair: Título tem limite de 128 tokens e o ranker usa só texto. Ponha o corpo em prioridade mais alta nos campos de conteúdo.

⚠ **Alerta de prova:** Manter orderby por data numa consulta híbrida com ranker semântico.
Como não cair: orderby sobrescreve a relevância. Remova-o e use k igual a 50.

⚠ **Alerta de prova:** Recriar o campo como exhaustiveKnn para medir a revocação do HNSW.
Como não cair: Exige reindexar e o campo não aceita HNSW. Use exhaustive true na consulta.

⚠ **Alerta de prova:** strictPostFilter ou remover o filtro quando filtros seletivos zeram resultados.
Como não cair: preFilter aplica o filtro durante a travessia do HNSW.

⚠ **Alerta de prova:** markdownHeaderDepth com outputFormat text, ou chunkingProperties com markdown.
Como não cair: markdownHeaderDepth só com markdown; chunkingProperties só com text.

⚠ **Alerta de prova:** Usar GenAI Prompt ou Shaper para aplicar uma regra proprietária que já existe numa Azure Function.
Como não cair: GenAI Prompt torna a regra probabilística e Shaper só monta tipos. Lógica própria = Custom Web API skill.

⚠ **Alerta de prova:** Conectar o Foundry IQ com chave de administração do Search nas instruções ou dar Foundry Agent Consumer.
Como não cair: Ferramenta MCP + conexão do projeto + Search Index Data Reader na identidade gerenciada do projeto.

⚠ **Alerta de prova:** Citação sem link se resolve com top_k maior ou query_type semantic.
Como não cair: Falta um campo recuperável com a URL de origem no índice.

⚠ **Alerta de prova:** Usar a API dedicada de classificador ou enableSegment false para separar documentos de um PDF.
Como não cair: As APIs de classificador foram descontinuadas e false classifica o arquivo inteiro. contentCategories com analyzerId, categoria other e enableSegment true.

⚠ **Alerta de prova:** prebuilt-read extrai campos de fatura, ou agentic mode serve para faturas padronizadas.
Como não cair: prebuilt-read é só OCR; o agentic mode (preview) tem custo e latência maiores e não suporta extract. Faturas padronizadas com alta precisão e baixa latência = modelo prebuilt de faturas do Document Intelligence.

Como cai na prova

📝 **Como cai:** Sintoma no índice (conteúdo vazio, tabela quebrada, resultados vetoriais ruins, ranking ruim) e quatro configurações.
Como resolver: Localize a etapa: extração (imageAction e OCR), enriquecimento (Document Layout, Content Understanding, Custom Web API), vetorização (skill e vectorizer com o mesmo modelo) ou consulta (híbrida, semântico, filtro). Corrija na etapa onde o problema nasce.

📝 **Como cai:** Escolha DUAS configurações de consulta para melhor relevância com códigos e linguagem natural.
Como resolver: Par clássico: consulta híbrida com RRF + ranker semântico. Descarte vetorial pura, só texto com perfil de pontuação e exhaustive KNN.

📝 **Como cai:** Qual analisador do Content Understanding usar para RAG, fatura, contrato ou lote misto.
Como resolver: Sem esquema e para RAG = prebuilt-documentSearch (vídeo: prebuilt-videoSearch). Campos de domínio = prebuilt-invoice ou prebuilt-contract. Lote misto = contentCategories + enableSegment true (ou analisador composto como prebuilt-procurement).

📝 **Como cai:** Agente conectado a índice ou Foundry IQ falhando (403, sem link, documentos de outro escopo).
Como resolver: 403 = função de dados na identidade gerenciada (Search Index Data Reader para Foundry IQ). Sem link = campo URL recuperável. Escopo = filter na ferramenta.

📝 **Como cai:** Configuração de propriedades da Document Layout ou da index projection.
Como resolver: Confira a dependência: markdownHeaderDepth só com markdown; chunkingProperties só com text; padrão do header é h6; projectionMode padrão inclui o pai, skip exclui.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
PDF escaneado indexado com campo content vazio	**imageAction generateNormalizedImages + skill OCR + Text Merge**
códigos exatos e perguntas em linguagem natural na mesma base	**Busca híbrida (RRF) + ranker semântico**
medir revocação do HNSW sem criar outro índice	**exhaustive true na consulta**
filtro muito seletivo, k pequeno, poucos ou nenhum resultado	**vectorFilterMode preFilter**
resultados vetoriais irrelevantes depois de configurar o vectorizer	**Vectorizer com o mesmo modelo da skill de embedding**
tabela que atravessa páginas vira texto corrido	**Skill Azure Content Understanding**
chunk por seção de título até h3	**Document Layout com outputFormat markdown e markdownHeaderDepth h3**
regra ou modelo próprio numa Azure Function durante a indexação	**Custom Web API skill (values e recordId)**
cada chunk como documento, sem indexar o pai	**Index projection com skipIndexingParentDocuments**
buscar enviando a foto de um produto	**Embeddings multimodais + vectorizer de imagem**
PDF com figuras para RAG, Markdown, resumo, sem esquema	**prebuilt-documentSearch**

valor literal com página, posição e confiança para revisão humana	**Método extract + estimateFieldSourceAndConfidence**
analisadores falham porque nenhuma implantação foi associada	**PATCH em contentunderstanding/defaults (chat + embeddings)**
mesmos campos em layouts muito variados, dezenas de exemplos rotulados	**Custom neural do Document Intelligence**

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

- PDF escaneado = imageAction generateNormalizedImages + OCR + Text Merge.
- Skill de embedding e vectorizer usam o mesmo modelo.
- Híbrida + semântico: sem orderby, k = 50, ordem dos campos na configuração semântica.
- Valor literal com confiança = extract + estimateFieldSourceAndConfidence; RAG sem esquema = prebuilt-documentSearch.

Essenciais para a prova (checklist)

☐ Sei configurar OCR na ingestão de PDFs escaneados.

☐ Sei montar busca híbrida com ranker semântico e por que orderby prejudica.

☐ Sei os limites do ranker semântico e como a ordem dos campos afeta o resultado.

☐ Sei medir revocação do HNSW e escolher o vectorFilterMode certo.

☐ Sei configurar skill de embedding e vectorizer de forma consistente.

☐ Sei escolher entre Document Layout e a skill Content Understanding e configurar markdownHeaderDepth.

☐ Sei quando usar Custom Web API skill e index projections com skipIndexingParentDocuments.

☐ Sei conectar índice e Foundry IQ ao agente com as permissões e campos certos.

☐ Sei escolher o analisador predefinido do Content Understanding para RAG, fatura, contrato e lote misto.

☐ Sei escolher entre custom neural, custom template e prebuilt do Document Intelligence.

Questões de revisão do capítulo

R3.1 · Implementar soluções de extração de informações

Um arquivo público estadual indexa PDFs de processos antigos que misturam texto digital e imagens escaneadas de despachos manuscritos. As buscas não encontram o conteúdo das imagens. No indexador do Blob Storage e no conjunto de habilidades, o que você deve configurar para incluir esse texto no índice?

A. dataToExtract contentAndMetadata e habilidade Key Phrase Extraction sobre o conteúdo
B. imageAction generateNormalizedImages, habilidade OCR e Text Merge para gerar o merged_text
C. parsingMode json e habilidade Language Detection sobre as imagens de cada página
D. imageAction none e habilidade Image Analysis para gerar descrições das imagens

R3.2 · Implementar soluções de extração de informações

Uma fintech usa um campo vetorial HNSW no Azure AI Search para buscar cláusulas contratuais semelhantes. Antes de ajustar efSearch, a equipe quer medir a revocação da busca aproximada comparando os resultados com os vizinhos mais próximos exatos, sem criar outro índice. O que você deve fazer?

A. Habilitar o semantic ranker e usar o rerankerScore como verdade de referência
B. Recriar o campo vetorial com exhaustiveKnn e consultar usando HNSW
C. Aplicar quantização binária ao campo e comparar com a versão sem compressão
D. Executar as consultas de teste com o parâmetro exhaustive true no campo HNSW

R3.3 · Implementar soluções de extração de informações

Uma fabricante de compressores indexa manuais técnicos em PDF no Azure AI Search para RAG. Muitas tabelas de especificação atravessam várias páginas e hoje viram texto corrido, perdendo informação. A equipe quer tabelas em Markdown, tabelas entre páginas tratadas como uma unidade e fragmentos que possam atravessar páginas. Qual habilidade você deve usar no conjunto de habilidades?

A. Habilidade Document Layout com outputFormat text e fragmentação por caracteres e sobreposição
B. Habilidade Text Split com páginas de 2.000 caracteres e sobreposição de 200
C. Habilidade OCR combinada com Text Merge sobre as imagens normalizadas do PDF
D. Habilidade Azure Content Understanding, que também faz a fragmentação do conteúdo

R3.4 · Implementar soluções de extração de informações

Um banco indexou manuais de produtos no Azure AI Search com uma skill AzureOpenAIEmbedding que usa text-embedding-ada-002. Ao adicionar a vetorização no momento da consulta, um desenvolvedor configurou o vectorizer do índice com uma implantação de text-embedding-3-small, de mesma dimensão. As consultas não falham, mas os resultados vetoriais ficaram irrelevantes. O que você deve corrigir?

A. Apontar o vectorizer para uma implantação do mesmo modelo usado pela skill
B. Habilitar o semantic ranker para reordenar os resultados da consulta vetorial
C. Trocar o algoritmo do campo vetorial de HNSW para exhaustive KNN
D. Reduzir a dimensão do campo vetorial com truncateDimension

Respostas comentadas

R3.1. B) imageAction generateNormalizedImages, habilidade OCR e Text Merge para gerar o merged_text

Por quê: Na ingestão para RAG com OCR, extrair imagens embutidas exige imageAction diferente de none. A habilidade OCR usa o Read da Azure Vision e reconhece texto impresso e manuscrito; o resultado pode ser mesclado ao conteúdo para busca e vetorização.

✗ A) Incorreta. Sem extrair imagens e aplicar OCR, o texto escaneado continua fora do índice; frases-chave não leem imagens.
✗ C) Incorreta. O modo json serve para blobs JSON, e a detecção de idioma não extrai texto de imagens.
✗ D) Incorreta. Com imageAction none nenhuma imagem é extraída, e descrições de imagem não substituem a transcrição do despacho.

> **Dica de prova:** Texto em imagem dentro do PDF = generateNormalizedImages + OCR + Text Merge.

R3.2. D) Executar as consultas de teste com o parâmetro exhaustive true no campo HNSW

Por quê: Configurar busca vetorial: HNSW é aproximada, fica em memória, consome a cota do índice vetorial e é ajustável por m, efConstruction e efSearch; exhaustive KNN é exata, por força bruta, indicada para conjuntos pequenos ou para medir revocação.

✗ A) Incorreta. O rerankerScore mede relevância semântica de texto, e não os vizinhos vetoriais exatos.
✗ B) Incorreta. Campos indexados para exhaustiveKnn não suportam consultas HNSW, e recriar o campo exige reindexação.
✗ C) Incorreta. A quantização reduz armazenamento e pode diminuir a precisão; não fornece a verdade de referência.

> **Dica de prova:** Medir a revocação do HNSW = mesma consulta com exhaustive true.

R3.3. D) Habilidade Azure Content Understanding, que também faz a fragmentação do conteúdo

Por quê: A habilidade Azure Content Understanding é GA na API 2026-04-01 do Azure AI Search, é cobrada pelo preço do Content Understanding (sem os 20 documentos gratuitos por dia) e não é indicada para documentos que exigem mais de cinco minutos de processamento.

✗ A) Incorreta. Com outputFormat text, a Document Layout gera tabelas e figuras como texto simples, o que pode causar perda de informação.

✗ B) Incorreta. A Text Split divide o texto por tamanho, sem entender layout nem tabelas.

✗ C) Incorreta. O OCR extrai texto de imagens; não reconstrói a estrutura das tabelas nem as une entre páginas.

> **Dica de prova:** Tabela que cruza páginas e Markdown para RAG = habilidade Content Understanding.

R3.4. A) Apontar o vectorizer para uma implantação do mesmo modelo usado pela skill

Por quê: Na vetorização integrada, chunking e embeddings acontecem na indexação (skillset), e o vectorizer, definido no índice e ligado ao campo por um vector profile, vetoriza a consulta. O assistente Import data já cria o vectorizer com o mesmo modelo usado na indexação.

✗ B) Incorreta. O semantic ranker reordena os 50 primeiros resultados com base em texto; não corrige vetores de consulta incompatíveis.

✗ C) Incorreta. Exhaustive KNN muda a estratégia de busca, mas continuaria comparando vetores de espaços diferentes.

✗ D) Incorreta. Truncar dimensões é técnica de compressão para modelos que a suportam; não resolve a diferença entre os modelos.

> **Dica de prova:** Skill de embedding e vectorizer = o MESMO modelo de embeddings.

04

CAPÍTULO 4 · 10-15% DA PROVA

Implementar soluções de visão computacional

Neste domínio preciso de três blocos: gerar e editar imagens e vídeos (gpt-image e Sora 2), entender imagens e vídeos (modelos de chat com visão e Azure Content Understanding) e aplicar IA responsável a conteúdo visual (Content Safety, Prompt Shields, procedência C2PA). A prova cobra parâmetros exatos: detail low ou high, background transparent com PNG, máscara transparente do mesmo tamanho, input_fidelity, remix_video_id, enableSegment e contentCategories. Onde mais erro: achar que serviços com aposentadoria anunciada (Image Analysis 4.0, Custom Vision) ainda são a resposta para projeto novo, usar o método extract em imagens e tentar barrar injeção de prompt em imagem com blocklist ou moderação de dano.

Neste capítulo

17 tópicos-chave · 13 pegadinhas · 6 macetes · 13 gatilhos · 4 questões comentadas

Tópicos-chave

Modelos de chat com visão

Recebem partes text e image_url (URL ou base64 em data URL), com até 10 imagens por chamada e 20 MB por imagem. Várias imagens na mesma mensagem permitem análise conjunta (por exemplo, resumo único dos danos). Para legenda curta ou detalhada, descreva no prompt o formato e o tamanho desejados.

Parâmetro detail

Aceita low, high ou auto (padrão, o modelo decide pelo tamanho). low processa uma versão de 512x512, mais rápida e barata, mas perde detalhes finos; high analisa em alta resolução para texto miúdo, gravações pequenas e etiquetas, consumindo mais tokens. Recortar a região de interesse também ajuda.

Alt-text e descrição estendida

Para seguir um guia de acessibilidade próprio, use modelo de chat com visão com o guia nas instruções e saída estruturada com os dois campos. Instruções distintas por tipo de imagem e detail low em fotos e high em gráficos equilibram custo e precisão. OCR só lê texto visível e não descreve gráficos.

Perguntas fundamentadas em evidência visual

Envie as fotos na mesma requisição, com detail high quando houver detalhes pequenos, e instrua o modelo a usar só o que está visível e a dizer quando a informação não aparece. Regras de marca (logotipo, cores, área de proteção) seguem o mesmo padrão: regras nas instruções, saída estruturada e revisão humana.

Controles de geração do gpt-image

quality (low, medium, high), size, n (1 a 10), output_format (png, padrão, ou jpeg), output_compression (0 a 100, só JPEG) e background (auto ou transparent, que exige PNG). Os modelos GPT-image devolvem imagens em base64 (b64_json); response_format não é suportado e WEBP não é suportado no Azure.

Edição e inpainting

Use o endpoint de edição (edits) com a imagem original, a máscara e o prompt. A máscara é PNG, com as mesmas dimensões da imagem e pixels totalmente transparentes onde o modelo pode alterar; áreas opacas são preservadas. A API aceita até 16 imagens de entrada (PNG ou JPG, menores que 50 MB).

input_fidelity

Definido como high, preserva estilo e características da imagem de entrada (rostos, logotipo, detalhes do produto) em edições sutis. Não é suportado no gpt-image-1-mini; quality, n e compressão não substituem esse controle.

Sora 2 (preview)

Usa a API v1 e gera vídeos de forma assíncrona: videos.create devolve um job (status queued), a aplicação consulta o status até concluir e baixa o MP4 com download_content. Suporta texto para vídeo, imagem para vídeo e vídeo gerado para vídeo (remix), com áudio.

input_reference e remix_video_id

input_reference ancora o primeiro quadro numa imagem, que deve ter o mesmo tamanho do vídeo pedido; imagens com rostos humanos são rejeitadas. remix_video_id parte de um vídeo já concluído e aplica um ajuste pontual no prompt, preservando estrutura, movimento e enquadramento.

Azure Content Understanding: analisadores e campos

Analisadores personalizados derivam de prebuilt-document, prebuilt-image, prebuilt-audio ou prebuilt-video e definem um fieldSchema em que a descrição funciona como mini-prompt. Métodos: extract (valor literal, só documentos), generate (valor produzido pelo modelo, como descrição) e classify (categoria fixa com enum). Para fotos: classify para tipo e generate para descrição.

Análise de vídeo no Content Understanding

Duas etapas: extração de conteúdo (transcrição WebVTT, quadros-chave, detecção de shots) e extração de campos com segmentação. enableSegment true com contentCategories descrevendo em linguagem natural como dividir cria capítulos lógicos; com false o vídeo inteiro é um segmento. cameraShotTimesMs marca só cortes de câmera. A amostragem é de cerca de 1 quadro por segundo em 512x512.

Agentic mode e opções de documento

config.workflow agentic (preview) substitui o antigo pro mode (API 2025-05-01-preview, aposentada) para campos que exigem raciocínio entre partes do documento, com custo e latência maiores e sem suporte a extract. enableFigureDescription gera texto alternativo de figuras; enableFigureAnalysis extrai dados de gráficos; enableBarcode lê códigos; enableFormula extrai fórmulas em LaTeX.

Moderação de imagens no Content Safety

A API Analyze Image classifica Hate, Sexual, Violence e Self-harm com severidade na escala 0, 2, 4 e 6; há também análise multimodal de texto e imagem (preview). Blocklists só fazem correspondência exata de texto e não analisam imagens; groundedness e Prompt Shields não moderam conteúdo visual.

Custom categories (rapid)

Para um padrão novo (meme ofensivo, logotipo proibido, símbolo de concorrente), crie um incidente com algumas amostras de texto ou imagem e comece a barrar em pouco tempo, sem treinar modelo. Combina correspondência de imagem e avaliação por LLM. A versão standard (texto, com treinamento) foi aposentada em 1º de setembro de 2026.

Injeção indireta por texto em imagem

Trate o texto de imagens de terceiros como não confiável: extraia com OCR e envie como documento ao Prompt Shields (um userPrompt e até cinco documents, com attackDetected em cada um) ou use o controle Indirect attack nos guardrails. Reduzir detail, converter para cinza ou usar Analyze Image não detectam o ataque.

Procedência de conteúdo (C2PA)

Imagens geradas no Azure OpenAI já saem com Content Credentials (C2PA) assinadas e marca d'água invisível, sem configurar nada. A verificação usa o site de detecção de procedência do Foundry, a Content Provenance Detection API ou ferramentas C2PA. Recortes e conversões podem remover os sinais, e a procedência não prova veracidade nem autoria; EXIF não é assinado.

Detecção de objetos com modelo próprio

Para projeto novo e de longo prazo com imagens rotuladas e caixas delimitadoras, use AutoML do Azure Machine Learning. Custom Vision e Image Analysis 4.0 têm aposentadoria em 25/09/2028, e a personalização de modelos do Image Analysis 4.0 já foi aposentada em 31/03/2025. Sem rótulos, campos de componente e dano saem de um analisador prebuilt-image do Content Understanding.

Macetes para memorizar

Máscara de inpainting

PNG, mesmo tamanho, buraco transparente onde vai mudar.

Transparência

Transparente só vive em PNG; JPEG não tem alfa.

Sora 2

Cria, espera, baixa. Foto no começo = input_reference; mexer no pronto = remix.

Métodos de campo no Content Understanding

extract copia (só documento); classify escolhe da lista; generate escreve.

detail

Lupa = high; olhada rápida = low; deixa o modelo decidir = auto.

Injeção em imagem

Texto na imagem é documento suspeito: OCR primeiro, Prompt Shields depois.

Tabela de memorização: não confunda

Quando é...	...e quando é
detail low Volume alto e pouco detalhe (fotos gerais, alt-text de fotos), priorizando custo e velocidade.	**detail high** Texto miúdo, números de etiqueta, chassi gravado, gráficos com dados, aceitando mais tokens.
remix_video_id Ajustar um vídeo já gerado e aprovado mantendo cena, movimento e enquadramento.	**input_reference** Começar um vídeo novo exatamente a partir de uma foto como primeiro quadro.
Analyze Image (categorias de dano) Classificar violência, sexual, ódio (inclusive símbolos de ódio) e automutilação com severidade.	**Custom categories (rapid)** Barrar rapidamente um padrão específico novo a partir de poucas amostras, sem treino.

Método classify Campo com categorias fixas (tipo de dano, estado da gôndola) definido por enum.	**Método generate** Campo de texto livre produzido pelo modelo, como descrição ou resumo.
enableSegment true + contentCategories Dividir o vídeo por lógica de negócio: notícia, procedimento, assunto da aula.	**cameraShotTimesMs (returnDetails)** Saber onde ocorrem os cortes de câmera, que não coincidem com capítulos lógicos.
AutoML do Azure Machine Learning Detector de objetos próprio, treinado com imagens rotuladas, para longo prazo.	**Analisador prebuilt-image do Content Understanding** Extrair campos estruturados de fotos (componente, dano, severidade) sem rotular imagens.

Números que caem

Item	Valor
Imagens por requisição de chat com visão	**até 10 (20 MB por imagem)**
detail low: resolução processada	**512x512**
gpt-image: n	**1 a 10 imagens**
output_compression	**0 a 100, só JPEG**
Edição de imagens: entradas	**até 16 imagens PNG ou JPG, menores que 50 MB**
Content Understanding em vídeo: amostragem	**cerca de 1 quadro por segundo, redimensionado para 512x512**
Severidade de imagens no Content Safety	**0, 2, 4 e 6**
Prompt Shields: documentos por chamada	**até cinco documents**
Custom categories (standard): aposentadoria	**1º de setembro de 2026**
Custom Vision e Image Analysis 4.0: aposentadoria	**25/09/2028**
Personalização de modelos do Image Analysis 4.0	**aposentada em 31/03/2025**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** background transparent com output_format jpeg.
Como não cair: JPEG não tem canal alfa. Transparente exige PNG; output_compression só vale para JPEG.

⚠ **Alerta de prova:** Usar response_format url ou pedir saída WEBP num modelo gpt-image.
Como não cair: GPT-image devolve base64 e gera PNG ou JPEG; response_format e WEBP não são suportados.

⚠ **Alerta de prova:** Máscara com pixels pretos ou opacos na área a editar, em JPEG ou em tamanho menor.
Como não cair: Máscara = PNG, mesmas dimensões, transparente onde vai mudar; opaco é o que se preserva.

⚠ **Alerta de prova:** Definir input_fidelity high mantendo o gpt-image-1-mini.
Como não cair: O gpt-image-1-mini não suporta input_fidelity. Troque para um modelo gpt-image que suporte.

⚠ **Alerta de prova:** Gerar o vídeo de novo com o mesmo prompt e o ajuste no texto, ou usar input_reference com um quadro.
Como não cair: Nova geração muda a cena e input_reference só ancora o primeiro quadro. Ajuste pontual = remix pelo id.

⚠ **Alerta de prova:** Esperar o MP4 na resposta de videos.create ou ler b64_json.
Como não cair: Vídeo é job assíncrono: consultar o status até concluir e baixar; b64_json é das imagens.

⚠ **Alerta de prova:** Enviar 25 fotos na mesma chamada reduzindo a resolução ou mudando o tipo de implantação.
Como não cair: O limite é de 10 imagens por requisição de chat; divida em lotes.

⚠ **Alerta de prova:** Usar o método extract em analisador de imagem ou vídeo.
Como não cair: extract é só para documentos; em imagem use classify e generate.

⚠ **Alerta de prova:** Usar os tempos de shot (cameraShotTimesMs) como limites de capítulos.
Como não cair: Cortes de câmera não são capítulos lógicos. Use enableSegment true + contentCategories.

⚠ **Alerta de prova:** Bloquear injeção escrita na imagem com blocklist, Analyze Image ou detail low.
Como não cair: Blocklist pega só a frase exata e Analyze Image vê dano visual. OCR + Prompt Shields para documentos.

⚠ **Alerta de prova:** É preciso ligar um parâmetro de marca d'água ou gravar EXIF para provar que a imagem é de IA.
Como não cair: As Content Credentials C2PA já são aplicadas automaticamente; EXIF pode ser alterado sem rastro.

⚠ **Alerta de prova:** Custom Vision ou Image Analysis 4.0 para um projeto novo de longo prazo.
Como não cair: Ambos têm aposentadoria anunciada (25/09/2028). Detector próprio = AutoML; campos sem rótulo = Content Understanding.

⚠ **Alerta de prova:** Recriar o analisador no pro mode para cálculos entre partes do contrato.
Como não cair: O pro mode era da API 2025-05-01-preview, aposentada. Use config.workflow agentic.

Como cai na prova

📝 **Como cai:** Requisito de geração ou edição de imagem e quatro combinações de parâmetros.
Como resolver: Cheque cada dependência: transparent exige PNG; output_compression só JPEG; máscara PNG do mesmo tamanho e transparente na área editada; input_fidelity indisponível no gpt-image-1-mini. A combinação sem conflito é a certa.

📝 **Como cai:** Ajuste em vídeo gerado ou criação a partir de foto com Sora 2.
Como resolver: Vídeo pronto a ajustar = remix_video_id. Foto como primeiro quadro = input_reference no tamanho exato. Depois de criar, sempre consulta de status e download; nada volta na mesma resposta.

📝 **Como cai:** Analisador personalizado do Content Understanding: qual configuração para vídeo, foto ou documento.
Como resolver: Vídeo em capítulos = enableSegment true + contentCategories. Foto = prebuilt-image com classify e generate. Documento com raciocínio = agentic. Figuras com alt-text = enableFigureDescription.

📝 **Como cai:** Moderação e segurança de imagens enviadas por usuários ou terceiros.
Como resolver: Dano visual com severidade = Analyze Image. Padrão novo com amostras = custom categories (rapid). Instrução maliciosa escrita = OCR + Prompt Shields. Procedência = C2PA. Blocklist nunca analisa imagem.

📝 **Como cai:** Modelo multimodal errando leitura ou gerando descrições fora do padrão.
Como resolver: Detalhe pequeno = detail high. Formato próprio = instruções com o guia + saída estruturada. Mais de 10 imagens = dividir em lotes.

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
número pequeno, etiqueta ou gravação que o modelo não lê	**detail high na parte image_url**
várias fotos analisadas em conjunto com um resumo único	**Várias partes image_url na mesma mensagem (até 10)**
ícone ou acessório com fundo transparente	**background transparent + output_format png**
trocar só uma região da foto mantendo o resto	**Endpoint edits + máscara PNG transparente do mesmo tamanho + prompt**
rostos ou logotipo saem diferentes após a edição	**input_fidelity high (fora do gpt-image-1-mini)**
vídeo aprovado, mudar só a cor de algo	**Remix do Sora 2 (remix_video_id)**
vídeo deve começar exatamente com uma foto	**input_reference no tamanho exato do vídeo**
dividir vídeos em capítulos por notícia, procedimento ou assunto	**enableSegment true + contentCategories**
texto alternativo de figuras em PDF no Markdown do analisador	**enableFigureDescription**
foto enviada por usuário com conteúdo sexual, violento ou símbolo de ódio	**Analyze Image do Content Safety (severidade 0, 2, 4, 6)**
meme ou logotipo proibido novo, poucas amostras, começar em horas	**Custom categories (rapid)**
'ignore as instruções anteriores' escrito numa imagem	**OCR + Prompt Shields para documentos**
verificar que a imagem foi gerada por IA	**Content Credentials C2PA + marca d'água invisível, Content Provenance Detection API**

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

- Fundo transparente = background transparent + PNG; máscara = PNG do mesmo tamanho, transparente onde edita.
- Sora 2 é assíncrono: create, status, download; ajuste pontual = remix pelo id.
- Content Understanding: extract só em documento; vídeo em capítulos = enableSegment true + contentCategories.
- Texto malicioso em imagem = OCR + Prompt Shields; procedência = C2PA automática.

Essenciais para a prova (checklist)

- ☐ Sei quando usar detail low, high e auto e o limite de imagens por chamada.
- ☐ Sei configurar geração com fundo transparente e quais formatos o Azure aceita.
- ☐ Sei preparar a máscara de inpainting e quando usar input_fidelity.
- ☐ Sei o fluxo assíncrono do Sora 2 e a diferença entre input_reference e remix_video_id.
- ☐ Sei definir campos com extract, classify e generate e qual método não vale para imagem.
- ☐ Sei segmentar vídeos com enableSegment e contentCategories.
- ☐ Sei moderar imagens com Analyze Image e custom categories (rapid).
- ☐ Sei mitigar injeção indireta escrita em imagens.
- ☐ Sei como a procedência C2PA é aplicada, verificada e o que pode removê-la.
- ☐ Sei por que AutoML é a escolha para detector de objetos novo de longo prazo.

Questões de revisão do capítulo

R4.1 · Implementar soluções de visão computacional

Uma universidade quer trocar o fundo das fotos dos professores publicadas no site por um fundo padrão da instituição, com autorização de cada um, usando a API de edição de imagens. Nos testes, os rostos saem levemente diferentes dos originais. A implantação usa o modelo gpt-image-1-mini. O que você deve fazer?

A. Manter o gpt-image-1-mini e definir input_fidelity como high na solicitação de edição
B. Definir output_format como jpeg e output_compression como 100
C. Usar um modelo gpt-image com suporte a input_fidelity e defini-lo como high
D. Aumentar quality para high e manter o gpt-image-1-mini na edição

R4.2 · Implementar soluções de visão computacional

Uma universidade gerou com Sora 2 um vídeo institucional de oito segundos e gostou das cenas, das transições e do enquadramento. Agora quer apenas trocar a cor das faixas usadas pelos formandos de azul para verde, sem refazer o restante. Qual abordagem você deve usar?

A. Editar o vídeo com a API de imagens, aplicando uma máscara a cada quadro
B. Gerar um vídeo novo com o mesmo prompt e a cor verde no texto da descrição
C. Gerar um vídeo novo usando o primeiro quadro do original como referência
D. Remixar o vídeo concluído pelo id, com um prompt que peça só essa mudança

R4.3 · Implementar soluções de visão computacional

Uma agência de publicidade gera centenas de peças por dia para uma marca de bebidas e precisa verificar, antes da publicação, se cada imagem respeita o manual da marca (logotipo sem distorção, cores oficiais e área de proteção), devolvendo um JSON com aprovado ou reprovado e os motivos para revisão humana. Qual abordagem você deve usar?

A. Analisar cada imagem na API Analyze Image do Content Safety e reprovar as de severidade média ou alta
B. Gerar novamente cada peça com quality high para garantir a fidelidade do logotipo da marca
C. Criar uma blocklist com os nomes das cores proibidas e aplicá-la ao texto de cada peça
D. Enviar cada imagem a um modelo multimodal com as regras do manual nas instruções e saída estruturada

R4.4 · Implementar soluções de visão computacional

Um marketplace de artigos usados permite que vendedores publiquem fotos dos produtos. A empresa quer verificar automaticamente cada foto antes da publicação e rejeitar imagens com conteúdo sexual ou violento acima de um nível de severidade definido pela política da plataforma. O que você deve usar?

A. A API de análise de imagem do Azure AI Content Safety
B. Uma blocklist personalizada com termos sexuais e violentos
C. A detecção de groundedness do Azure AI Content Safety
D. O Prompt Shields do Azure AI Content Safety para documentos

Respostas comentadas

R4.1. C) Usar um modelo gpt-image com suporte a input_fidelity e defini-lo como high

Por quê: Selecionar os controles de geração e edição adequados inclui conhecer os parâmetros de cada modelo: quality, size, n, output_format, background, mask e input_fidelity, este indisponível no gpt-image-1-mini.

✗ A) Incorreta. O gpt-image-1-mini não suporta input_fidelity.
✗ B) Incorreta. Formato e compressão de saída não alteram a preservação dos rostos.
✗ D) Incorreta. quality afeta o nível de detalhe e a velocidade, mas não é o controle de fidelidade aos traços da imagem de entrada.

> **Dica de prova:** Preservar rosto na edição = input_fidelity high (indisponível no gpt-image-1-mini).

R4.2. D) Remixar o vídeo concluído pelo id, com um prompt que peça só essa mudança

Por quê: Fluxos para editar vídeos gerados usam o remix do Sora 2 (preview): referencia-se o vídeo concluído e um prompt com a alteração, mantendo o restante da composição.

✗ A) Incorreta. A API de edição de imagens trabalha com imagens estáticas; aplicar máscaras quadro a quadro não é o fluxo de edição de vídeo do Sora.

✗ B) Incorreta. Uma nova geração do zero pode mudar cenas, transições e enquadramento que a universidade quer manter.

✗ C) Incorreta. A imagem de referência só ancora o primeiro quadro; o restante do vídeo seria gerado de novo.

> **Dica de prova:** Ajuste pontual em vídeo gerado = remix pelo id, uma mudança por vez.

R4.3. D) Enviar cada imagem a um modelo multimodal com as regras do manual nas instruções e saída estruturada

Por quê: Item do guia 'Impor regras de política visual, como ... manter os requisitos de uso da marca': regras de marca são verificadas com modelos multimodais orientados por instruções e saída estruturada, seguidos de revisão humana.

✗ A) Incorreta. A Analyze Image detecta categorias de dano, não o cumprimento de regras de marca.

✗ B) Incorreta. Gerar de novo não verifica nada e pode introduzir novas violações.

✗ C) Incorreta. Blocklists filtram termos em texto; não avaliam cores nem logotipos em imagens.

> **Dica de prova:** Regra de marca em imagem = modelo multimodal com as regras nas instruções + saída estruturada.

R4.4. A) A API de análise de imagem do Azure AI Content Safety

Por quê: Implementar filtros para classificar conteúdo visual inseguro ou não permitido: o Azure AI Content Safety analisa imagens nas quatro categorias de dano; para texto e imagem juntos, há a análise multimodal (preview).

✗ B) Incorreta. Blocklists fazem correspondência exata de texto e não analisam imagens.

✗ C) Incorreta. A detecção de groundedness verifica se uma resposta em texto está fundamentada nas fontes; não modera imagens.

✗ D) Incorreta. Prompt Shields detecta ataques de prompt em entradas de usuário e documentos; não classifica imagens por categoria de dano.

> **Dica de prova:** Moderar foto enviada por usuário = Content Safety, análise de imagem (severidade 0, 2, 4 ou 6).

05

CAPÍTULO 5 · 25-30% DA PROVA

Planejar e gerenciar uma solução de IA do Azure

Aqui eu preciso decidir a arquitetura certa no Foundry: tipo de implantação (Global, Data Zone, Standard, Provisioned, Batch, Developer), cota e limites de taxa, segurança sem chaves com Entra ID, rede privada e RBAC, e governança com guardrails, avaliações, red teaming e rastreamento. Quase todo enunciado traz uma restrição decisiva (dados só na UE, menor custo, sem chaves, sem mudar código, sem gerenciar IPs) e quatro opções plausíveis; quem acha a restrição acerta. Onde mais erro: achar que outro projeto ou recurso cria cota nova, confundir avaliação contínua com agendada, escolher o ponto de intervenção errado do guardrail (User input em vez de Tool response) e trocar as funções renomeadas (Foundry User x Foundry Agent Consumer). Também preciso separar Memory, Foundry IQ e File Search, que caem juntos como alternativas.

Neste capítulo

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

Tópicos-chave

Tipos de implantação e residência de dados

Os dados em repouso ficam sempre na geografia designada; o que muda é onde a inferência é processada. Global pode processar em qualquer região do Azure; Data Zone processa dentro da zona (US, EU ou APAC, e a zona EU segue o EU Data Boundary); Standard e Regional Provisioned processam na geografia escolhida. Não existe zona de dados Brasil: processar só no Brasil é Standard.

Global Batch e Data Zone Batch

Para trabalhos grandes, assíncronos e sem urgência: requisições em arquivo JSONL, meta de conclusão de 24 horas, cerca de 50% do custo do Global Standard e cota própria de tokens enfileirados, que não disputa a cota das aplicações online. Nunca serve para chat, assistente interativo ou para absorver 429 de tráfego em tempo real.

Provisioned (PTU)

Capacidade reservada, cobrada por hora, para volume alto e constante com latência previsível. É dimensionada em PTUs, não em TPM de cota: aumentar capacidade significa comprar PTUs. Reservas do Azure são compromissos mensais ou anuais. A métrica Provisioned Utilization mostra a saturação (100% = limitação e 429).

Spillover

Encaminha o excesso de uma implantação provisionada para uma implantação Standard (pay-as-you-go) do mesmo modelo e versão; o que a PTU atende fica no custo por hora e o que transborda é cobrado por token. Vale para tudo pela propriedade spilloverDeploymentName da implantação (sem mudar código) ou por requisição com o cabeçalho x-ms-spillover-deployment; o cabeçalho x-ms-spillover-from-deployment identifica o desvio.

Tipo Developer

Serve para avaliar modelos ajustados por fine-tuning: cobrança por token, vida útil fixa de 24 horas e sem SLA. Não é para produção nem para lote.

Cota e limites de taxa (TPM e RPM)

A cota é por assinatura + região + modelo + tipo de implantação; a soma das implantações não pode passar dela, e outro projeto ou outro recurso na mesma assinatura e região não cria cota nova. O TPM atribuído à implantação define seus limites de TPM e RPM. O RPM é avaliado em janelas curtas (1 a 10 segundos) e o TPM é estimado na chegada pela soma de prompt + max_tokens, por isso rajadas e max_tokens superdimensionado geram 429 com métricas baixas.

AI Gateway no Foundry Control Plane

Integrado ao Azure API Management, fica entre os clientes e as implantações e impõe, por projeto, limite de TPM e cota total de tokens por período. TPM estourado devolve 429; cota do período esgotada devolve 403. Alertas e orçamentos só avisam; o gateway bloqueia.

versionUpgradeOption

OnceNewDefaultVersionAvailable atualiza em até duas semanas após uma nova versão virar padrão; OnceCurrentVersionExpired só troca na data de aposentadoria (o valor nulo equivale a ele); NoAutoUpgrade nunca atualiza e a implantação para de funcionar na aposentadoria. A atualização automática de versão só é suportada em implantações Standard.

Model router

Modelo treinado que analisa cada prompt e o envia ao LLM mais adequado atrás de uma única implantação. Modos: Balanced (padrão, faixa de qualidade de cerca de 1% a 2%), Cost (mais economia, faixa maior) e Quality (ignora custo). O model subset restringe os modelos (compliance, contexto longo, já que a janela fica limitada pela do menor modelo); use pelo menos dois modelos para ter failover. Roteia só pelo texto e honra os limites de zona de dados.

Escolha do modelo por tarefa

LLM para tarefas amplas e complexas, SLM (como Phi-4-mini-instruct) para tarefas simples com menor custo e latência, multimodal quando há imagem ou áudio, raciocínio para problemas de várias etapas e Foundry Tools para capacidades prontas. Confira no catálogo modalidades e tool calling: os Phi-4 listados não oferecem tool calling. Offline, no dispositivo e sem custo por token = Foundry Local com modelo pequeno.

Standard agent setup x Basic setup

O Standard setup usa recursos do cliente: Azure Cosmos DB for NoSQL para conversas, Azure Storage para arquivos e Azure AI Search para vector stores, com chaves gerenciadas pelo cliente e base para rede privada. O Basic setup guarda tudo no armazenamento gerenciado da plataforma, sem esse controle. Redis e SQL Database não fazem parte.

Opções de rede do Agent Service

Combina modelo de saída (pública, BYO virtual network ou managed virtual network) com entrada (pública ou private endpoint), definido na criação do recurso. Managed VNet isola sem gerenciar IPs (ideal com faixas sobrepostas); o modo Allow only approved outbound limita a saída a destinos aprovados. Na BYO, a sub-rede de agentes é delegada a Microsoft.App/environments, exclusiva de um recurso, com endereços privados e na mesma região; os private endpoints ficam em outra sub-rede.

Autenticação sem chave

Use Microsoft Entra ID (tokens com escopo ai.azure.com) com identidade gerenciada e RBAC; em produção prefira ManagedIdentityCredential, que é determinística, em vez de DefaultAzureCredential, que percorre uma cadeia. Depois de migrar os clientes, defina disableLocalAuth=true (também no Bicep) e atribua a Azure Policy que exige key access disabled. Key Vault e rotação de chaves não impedem o uso de chaves.

RBAC do Foundry

Funções renomeadas: Azure AI User virou Foundry User (menor privilégio para quem cria e testa agentes, e a função da identidade gerenciada do projeto no recurso); Foundry Agent Consumer só invoca agentes; Foundry Project Manager gerencia o projeto e atribui Foundry User; Foundry Owner e Foundry Account Owner são altamente privilegiadas. Azure AI Developer é de workspaces do Azure Machine Learning e hubs, e funções Cognitive Services não se aplicam ao Foundry (exceto Usages Reader). Ferramenta Azure AI Search sem chave: a identidade gerenciada do recurso do Foundry recebe Search Index Data Contributor e Search Service Contributor no serviço de pesquisa.

Guardrails e controles

Guardrail é uma coleção nomeada de controles (risco, ponto de intervenção e ação) aplicada a modelos e agentes. Pontos: User input, Tool call e Tool response (ambos em preview e só para agentes) e Output. Ações: Annotate (só para modelos, registra sem bloquear) e Annotate and block. O guardrail do agente substitui por completo o do modelo, e o padrão Default.V2 é somente leitura.

Limiares de severidade dos filtros de dano

Hate, Sexual, Violence e Self-harm classificam conteúdo em safe, low, medium e high; o padrão filtra medium e high. Limite Low sinaliza low e acima (filtra mais); High só pega o mais severo (mais permissivo). Para mudar, crie um guardrail personalizado; Off só para clientes aprovados. Blocklists complementam com termos exatos, sem mudar o limiar.

Avaliação contínua, agendada e red teaming

Avaliação contínua pontua uma amostra do tráfego real (EvaluationRule com ContinuousEvaluationRuleAction e evento RESPONSE_COMPLETED; max_hourly_runs padrão 100; a identidade do projeto precisa de Foundry User). Avaliação agendada roda um conjunto de teste fixo em agenda para detectar drift (Schedule com RecurrenceTrigger). O AI Red Teaming Agent usa PyRIT para simular ataques e mede o Attack Success Rate; local cobre riscos do modelo e a nuvem cobre riscos de agente, como vazamento e ações proibidas.

Rastreamento e CI/CD

Conectar um Application Insights ao projeto habilita o rastreamento no servidor, sem mudar código, nas convenções do OpenTelemetry; consultar os traces exige Log Analytics Reader. No pipeline, a ação microsoft/ai-agent-evals (azure-ai-project-endpoint, deployment-name, data-path, agent-ids no formato nome:versão), após azure/login com OIDC, compara versões com intervalos de confiança e significância. Para hosted agents, azd pipeline config gera o workflow com provision + deploy.

Macetes para memorizar

💡 Residência de dados

Global vai para o mundo; Data Zone fica no bloco (US, EU, APAC); Standard fica em casa.

💡 Política de versão

Expired = fico até expirar e pulo; NoAuto = fico até morrer; NewDefault = sigo o padrão em duas semanas.

💡 Spillover

Propriedade = para todos, sem código; cabeçalho = caso a caso.

💡 AI Gateway

429 = rápido demais (TPM); 403 = acabou o período (cota).

💡 Limiar de severidade

Low pega tudo (low, medium, high); High só pega o pior.

💡 Onde mora o conhecimento do agente

Usuário = Memory; empresa = Foundry IQ; arquivo da conversa = File Search.

💡 Tipos de avaliação

Pipeline = antes do deploy; contínua = tráfego real; agendada = mesmo teste (drift); red teaming = ataque (ASR).

Tabela de memorização: não confunda

Quando é...	...e quando é
Avaliação contínua Pontuar uma amostra das respostas reais de produção à medida que acontecem, sem montar conjunto de teste, limitando execuções por hora.	**Avaliação agendada** Reexecutar o mesmo conjunto de teste curado em agenda fixa (toda semana, todo mês) para detectar drift comparando notas.
Data Zone Standard Processamento restrito a US, EU ou APAC, pagando por token e com cota maior que a dos tipos por geografia.	**Standard (regional)** Processamento dentro de uma geografia específica sem zona de dados, como o Brasil, aceitando cota menor.
Memory Lembrar entre sessões o contexto de cada usuário (preferências, perfil, resumos de conversa) sem a aplicação manter banco próprio.	**Foundry IQ** Fundamentar vários agentes em conteúdo organizacional curado (SharePoint, Blob, OneLake) com citações e respeito às permissões do usuário; arquivo enviado na conversa é File Search.
Foundry Agent Consumer Aplicativo, portal ou service principal que só invoca agentes já publicados, sem criar ou alterar nada.	**Foundry User** Desenvolvedor que cria e testa agentes e a identidade gerenciada do projeto no recurso do Foundry.
spilloverDeploymentName (propriedade) Transbordar todo o excesso da PTU automaticamente, sem alterar o código das chamadas.	**x-ms-spillover-deployment (cabeçalho)** Ativar o spillover seletivamente, requisição a requisição, aceitando mudar o código.

Managed virtual network Isolar a saída sem gerenciar faixas de IP, sub-redes ou peering, inclusive com IPs sobrepostos.	**BYO virtual network** A equipe de rede quer controlar faixas, peering e roteamento e consegue fornecer uma sub-rede delegada dedicada.
OnceCurrentVersionExpired Manter a versão homologada enquanto estiver em suporte e migrar sozinho na aposentadoria, sem parar.	**NoAutoUpgrade** Nunca trocar a versão, aceitando que a implantação pare de funcionar na data de aposentadoria.

Números que caem

Item	Valor
Global Batch: meta de conclusão e desconto	**24 horas; cerca de 50% do custo do Global Standard**
Tipo Developer: vida útil	**24 horas, sem SLA**
OnceNewDefaultVersionAvailable: prazo para atualizar	**até duas semanas após o novo padrão**
Janela de avaliação do RPM	**1 a 10 segundos**
Avaliação contínua: max_hourly_runs padrão	**100 execuções por hora**
Model router Balanced: faixa de qualidade considerada	**cerca de 1% a 2%**
Model Availability Rate: erros considerados	**erros de servidor 500 ou mais (429 não entra)**
Traces no portal do Foundry	**até 90 dias**
Retenção padrão das respostas (response IDs)	**30 dias**
Workflows visuais do Foundry: aposentadoria	**1º de dezembro de 2026**

Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Criar a nova implantação em outro projeto ou outro recurso da mesma região para ganhar cota.
Como não cair: Cota é por assinatura + região + modelo + tipo de implantação. Só ajuda reduzir o TPM de uma implantação existente ou pedir aumento de cota.

⚠ **Alerta de prova:** Aumentar o TPM da implantação provisionada para atender o pico.
Como não cair: Provisioned é dimensionado em PTUs. Sem comprar PTU, a saída é spillover para uma Standard do mesmo modelo e versão.

⚠ **Alerta de prova:** Criar uma implantação Global Batch para receber as requisições que deram 429.
Como não cair: Batch é assíncrono com meta de 24 horas; não atende tráfego interativo.

⚠ **Alerta de prova:** Global Standard em região da UE mantém o processamento na Europa.
Como não cair: Tipos Global podem processar em qualquer região do Azure. Residência na UE com pagamento por token = Data Zone Standard.

⚠ **Alerta de prova:** Data Zone Standard garante processamento no Brasil.
Como não cair: As zonas de dados são US, EU e APAC. Para o Brasil, Standard na geografia escolhida.

⚠ **Alerta de prova:** NoAutoUpgrade é a forma segura de manter a versão sem risco.
Como não cair: NoAutoUpgrade para de funcionar na aposentadoria. Manter a versão e não cair = OnceCurrentVersionExpired.

⚠ **Alerta de prova:** Guardar as chaves no Key Vault ou rotacioná-las atende ao requisito de nenhum app usar chave.
Como não cair: Enquanto a autenticação local estiver ativa a chave funciona. Bloquear = disableLocalAuth=true + Azure Policy de key access disabled.

⚠ **Alerta de prova:** Pipeline com 403 após desativar o acesso público é falta de permissão: dar Foundry Owner ou usar chave.
Como não cair: É alcance de rede. Rode o job em runner auto-hospedado dentro da VNet (ou agente do Azure DevOps, ou jump box via Bastion).

⚠ **Alerta de prova:** O guardrail do modelo prevalece, ou os dois se combinam pelo limite mais restritivo.
Como não cair: Quando o agente tem guardrail próprio, ele substitui por completo o do modelo, inclusive em Tool call e Tool response.

⚠ **Alerta de prova:** Editar o Default.V2 ou adicionar uma blocklist para filtrar violência de severidade baixa.
Como não cair: Default.V2 é somente leitura e blocklist não muda o limiar. Crie guardrail personalizado com Violence em Low na entrada e na saída.

⚠ **Alerta de prova:** Instrução oculta em e-mail ou página lida por ferramenta se resolve com User prompt attacks em User input.
Como não cair: Conteúdo de terceiros devolvido pela ferramenta = risco Indirect attack no ponto Tool response, com Annotate and block para parar a execução.

⚠ **Alerta de prova:** Prompt Shields ou avaliadores de qualidade medem a resistência do agente antes do lançamento.
Como não cair: Prompt Shields protege em tempo de execução; Groundedness e Coherence medem qualidade. Simular ataques e medir ASR = AI Red Teaming Agent.

⚠ **Alerta de prova:** Avaliação contínua ou alerta do Azure Monitor funcionam como portão de qualidade antes de promover a versão.
Como não cair: Ambos agem depois do deploy. Portão no pipeline = microsoft/ai-agent-evals comparando nome:versão.

⚠ **Alerta de prova:** 429 com uso de tokens bem abaixo da cota pede aumento de cota ou outra implantação.
Como não cair: É rajada (RPM em janelas curtas) ou max_tokens superdimensionado. Suavize o tráfego, use backoff respeitando retry-after-ms e ajuste max_tokens.

Como cai na prova

📝 **Como cai:** Empresa com requisito de residência, padrão de tráfego e forma de pagamento pergunta qual tipo de implantação usar.
Como resolver: 1) Onde pode processar (mundo, zona, geografia). 2) Tempo real ou lote. 3) Paga por token ou reserva PTU. 4) Elimine Global se há residência e Batch se é interativo. A combinação que sobra é a resposta.

📝 **Como cai:** Erros 429 com algum detalhe nas métricas (uso abaixo da cota, PTU a 100%, rajadas).
Como resolver: PTU a 100% = Provisioned Utilization e spillover. Uso baixo = rajada ou max_tokens alto: suavize, backoff com retry-after-ms, ajuste max_tokens. Aumentar cota ou criar outra implantação raramente é a resposta.

📝 **Como cai:** Escolha DUAS ações para eliminar o uso de chaves (às vezes com credencial previsível em produção ou recursos novos já protegidos).
Como resolver: Leia o segundo requisito. Credencial previsível no app = ManagedIdentityCredential + disableLocalAuth=true. Recursos novos já devem nascer sem chave = disableLocalAuth=true no Bicep + Azure Policy de key access disabled. Descarte Key Vault, rotação de chaves e DefaultAzureCredential com chave em variável de ambiente.

📝 **Como cai:** Agente lê conteúdo externo por ferramenta e a segurança teme instruções ocultas; pergunta risco, ponto e ação.
Como resolver: Risco = Indirect attack; ponto = Tool response (onde o conteúdo chega); ação = Annotate and block se precisa parar. User input é só o prompt do usuário; Output é tarde demais.

📝 **Como cai:** Equipe quer medir qualidade ou segurança e pergunta qual recurso de avaliação usar.
Como resolver: Pergunte quando e sobre o quê: antes do deploy no pipeline (ai-agent-evals), tráfego real (contínua), conjunto fixo recorrente (agendada), ataques simulados (AI Red Teaming Agent), muitas falhas a entender (cluster analysis).

Gatilhos do enunciado

Se o enunciado trouxer...	Pense em
prompts e respostas só na União Europeia, pagar por token, cota maior	**Data Zone Standard em região da zona EU**
processar dentro da geografia do Brasil	**Standard (regional), não Data Zone**
milhões de itens, pode levar um dia, menor custo, não afetar a cota online	**Global Batch com arquivo JSONL**
PTUs esgotam no pico e não quer comprar mais	**Spillover para Standard do mesmo modelo e versão (spilloverDeploymentName)**
avaliar modelo ajustado por algumas horas, barato, sem SLA	**Tipo Developer (24 h)**
429 com métricas abaixo do TPM, chamadas no mesmo segundo ou max_tokens alto	**Rajada: fila, backoff com retry-after-ms e max_tokens próximo do real**
limitar tokens por minuto e por mês para cada projeto ou squad	**AI Gateway no Foundry Control Plane (429 e 403)**
app só chama o agente publicado, menor privilégio	**Foundry Agent Consumer**
nenhum app pode usar chave, e os novos recursos já devem nascer assim	**disableLocalAuth=true no Bicep + Azure Policy key access disabled**
isolamento total, IPs sobrepostos, não quer gerenciar sub-rede	**Managed virtual network + private endpoint de entrada**
lembrar preferências do cliente entre sessões	**Memory com a ferramenta de busca de memória**
vários agentes, SharePoint e Blob, citações, permissões por usuário	**Base de conhecimento do Foundry IQ**
mesmos casos de teste toda semana para ver degradação	**Avaliação agendada (drift)**
funcionou no playground e falhou com erro de autorização depois de publicado	**Dar as funções à identidade do agent application**

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

- Cota = assinatura + região + modelo + tipo de implantação; projeto ou recurso novo não cria cota.
- Guardrail do agente substitui por completo o do modelo; ataque vindo de ferramenta = Indirect attack em Tool response.
- Só chama o agente = Foundry Agent Consumer; cria e testa = Foundry User (antigo Azure AI User).
- Contínua = tráfego real; agendada = conjunto fixo (drift); red teaming = ataques e ASR.

Essenciais para a prova (checklist)

☐ Sei escolher entre Global, Data Zone e Standard pelo local de processamento e entre Standard, Provisioned, Batch e Developer pela carga.

☐ Sei explicar por que 429 aparece com métricas abaixo da cota e as três correções (suavizar, backoff com retry-after-ms, max_tokens).

☐ Sei configurar spillover pela propriedade e pelo cabeçalho e quando cada um vale.

☐ Sei as três políticas de atualização de versão e qual delas para de funcionar na aposentadoria.

☐ Sei bloquear chaves (disableLocalAuth e Azure Policy) e por que Key Vault e rotação não bastam.

☐ Sei a função certa para app consumidor, desenvolvedor, identidade do projeto e ferramenta Azure AI Search sem chave.

☐ Sei montar um guardrail (risco, ponto de intervenção, ação) e qual vale quando agente e modelo têm guardrails.

☐ Sei diferenciar Standard e Basic setup e listar os três recursos BYO do Standard.

☐ Sei escolher entre managed VNet e BYO VNet e os requisitos da sub-rede delegada.

☐ Sei qual mecanismo usar para avaliar antes do deploy, em produção, em agenda e com ataques simulados.

Questões de revisão do capítulo

R5.1 · Planejar e gerenciar uma solução de IA do Azure

Antes de liberar um agente que consulta saldos e faz pagamentos por meio de ferramentas, um banco quer simular automaticamente ataques adversariais, inclusive tentativas de fazer o agente vazar dados financeiros obtidos das ferramentas e executar ações proibidas, e medir a taxa de sucesso desses ataques. O que você deve usar?

A. Avaliação contínua com os avaliadores Groundedness e Relevance
B. O AI Red Teaming Agent em execução local, com categorias de risco do modelo
C. O AI Red Teaming Agent na nuvem, com as categorias de risco de agentes
D. Um guardrail com Prompt Shields para ataques na entrada do usuário

R5.2 · Planejar e gerenciar uma solução de IA do Azure

Uma universidade atende o assistente de matrículas com uma implantação Global Provisioned dimensionada para o tráfego normal. Na semana de matrículas, os picos esgotam as PTUs e as solicitações excedentes recebem 429. A instituição não quer comprar PTUs para o pico e aceita pagar por token só pelo excedente. O que você deve configurar?

A. Comprar uma reserva anual de PTUs adicional para a semana de matrículas
B. Aumentar o TPM da implantação provisionada nas configurações de cota
C. Criar uma implantação Global Batch para receber as solicitações com 429
D. Habilitar spillover para uma implantação Standard do mesmo modelo e versão

R5.3 · Planejar e gerenciar uma solução de IA do Azure

Uma autarquia federal desativou o acesso público do recurso do Foundry, do Azure Container Registry e do Application Insights, que agora só respondem por endpoints privados em uma VNet. O pipeline do GitHub Actions que executa azd up para publicar um agente hospedado passou a falhar com erros 403 nas chamadas de plano de dados. O que você deve fazer?

A. Executar o job com a função Foundry Owner atribuída à entidade de serviço do pipeline
B. Executar o azd up com uma chave de API no lugar da identidade federada do pipeline
C. Executar o job em um runner auto-hospedado dentro da VNet, que resolve o DNS privado
D. Executar o job em um runner hospedado pelo GitHub com IPs liberados no firewall do recurso

R5.4 · Planejar e gerenciar uma solução de IA do Azure

Uma indústria química validou o comportamento do seu assistente de segurança operacional com uma versão específica de um modelo Azure OpenAI, em uma implantação Standard. A equipe não quer que a versão mude quando a Microsoft definir um novo padrão, mas também não aceita que a aplicação pare de funcionar na data de aposentadoria da versão. Qual valor de versionUpgradeOption você deve configurar?

A. NoAutoUpgrade, que mantém a versão atual mesmo depois da data de aposentadoria
B. OnceNewDefaultVersionAvailable, que migra até duas semanas após um novo padrão
C. OnceCurrentVersionExpired, que só migra para a versão padrão na data de aposentadoria
D. Auto-update to default no portal, que acompanha cada nova versão padrão

Respostas comentadas

R5.1. C) O AI Red Teaming Agent na nuvem, com as categorias de risco de agentes

Por quê: Aplicar instrumentação de IA responsável inclui avaliações de segurança e red teaming. O AI Red Teaming Agent combina PyRIT com avaliadores de risco e segurança, gera um scorecard por categoria e estratégia de ataque e deve rodar em ambiente não produtivo semelhante ao de produção.

✗ A) Incorreta. Groundedness e Relevance medem qualidade; não simulam ataques nem calculam taxa de sucesso de ataques.
✗ B) Incorreta. A execução local cobre categorias de risco do modelo; as categorias específicas de agentes exigem o red teaming na nuvem.
✗ D) Incorreta. Prompt Shields bloqueia ataques em tempo de execução; não testa o agente de forma automatizada nem gera métricas de risco antes do lançamento.

> **Dica de prova:** Simular ataques e medir ASR = AI Red Teaming Agent. Riscos de agente (vazamento, ações proibidas) = na nuvem.

R5.2. D) Habilitar spillover para uma implantação Standard do mesmo modelo e versão

Por quê: Gerenciar escala e custo inclui combinar provisioned (capacidade reservada e latência previsível) com spillover para Standard, configurado na propriedade spilloverDeploymentName ou por solicitação com o cabeçalho x-ms-spillover-deployment.

✗ A) Incorreta. Reservas do Azure são compromissos mensais ou anuais de PTUs; comprar capacidade para o pico é o que a instituição quer evitar.
✗ B) Incorreta. Implantações provisionadas são dimensionadas em PTUs; aumentar a capacidade significa comprar mais PTUs, e não ajustar TPM de cota padrão.
✗ C) Incorreta. Batch é assíncrono, com meta de 24 horas, e não atende um assistente em tempo real.

> **Dica de prova:** PTU esgotou no pico e não quer comprar mais? Spillover para Standard do mesmo modelo e versão.

R5.3. C) Executar o job em um runner auto-hospedado dentro da VNet, que resolve o DNS privado

Por quê: Depois de desativar o acesso público, chamadas de plano de dados ao Foundry e o push para o ACR falham de fora da rede. O padrão documentado é CI dentro da VNet (runner auto-hospedado ou agente do Azure DevOps) ou, pontualmente, uma jump box acessada via Azure Bastion.

✗ A) Incorreta. O erro decorre da rede, não da permissão; mais privilégio não torna os endpoints alcançáveis.
✗ B) Incorreta. Trocar a autenticação não resolve a falta de alcance de rede até os endpoints privados.
✗ D) Incorreta. Com o acesso público desativado, os endpoints só são alcançados pela rede privada; regras de IP não resolvem o acesso de fora da VNet.

> **Dica de prova:** Recursos só privados = pipeline rodando dentro da VNet (runner auto-hospedado).

R5.4. C) OnceCurrentVersionExpired, que só migra para a versão padrão na data de aposentadoria

Por quê: Há três políticas: OnceNewDefaultVersionAvailable (segue o novo padrão em até duas semanas), OnceCurrentVersionExpired (só atualiza na aposentadoria) e NoAutoUpgrade (nunca atualiza e para na aposentadoria). O valor nulo equivale a OnceCurrentVersionExpired.

✗ A) Incorreta. Com NoAutoUpgrade a implantação nunca é atualizada automaticamente e para de funcionar quando a versão é aposentada.

✗ B) Incorreta. Essa opção atualiza a implantação em até duas semanas após a designação de uma nova versão padrão, o que contraria o requisito de manter a versão validada.

✗ D) Incorreta. A atualização automática para o padrão troca a versão quando um novo padrão é definido, justamente o que a equipe quer evitar.

Dica de prova: Manter a versão e não cair na aposentadoria = OnceCurrentVersionExpired.

DIA DA PROVA

Estratégia de prova

O AI-103 é uma prova de decisão de arquiteto e desenvolvedor no Microsoft Foundry: quase todo enunciado descreve uma empresa, um requisito e uma restrição (sem chaves, menor custo, sem mudar código, dados na UE, sem treinar modelo, projeto novo) e pede o recurso, parâmetro ou fluxo que atende exatamente. As alternativas erradas são produtos reais usados no lugar errado, recursos legados ou com aposentadoria anunciada, ou parâmetros que existem mas pertencem a outra API. O exame valoriza a plataforma atual: Foundry Agent Service com Responses API e Conversations, guardrails, Content Understanding e Azure AI Search com vetorização integrada. Detalhes de nome importam: function_call_output x mcp_approval_response, text.format x response_format, detail high x low.

Como responder

✓ Administre o tempo: conte com cerca de 50 questões em 120 minutos (a Microsoft não divulga o número exato, que fica entre 40 e 60), pouco mais de 2 minutos por item; marque para revisão as que passarem disso.

✓ A nota mínima é 700 numa escala de 1 a 1000; não trave numa questão difícil, siga e volte no fim.

✓ Sublinhe mentalmente a restrição do enunciado (sem chaves, menor custo, sem mudar código, sem treinar, projeto novo, tempo real) e use-a para eliminar alternativas antes de comparar as restantes.

✓ Em múltipla resposta (Escolha DUAS ou TRÊS), cada alternativa certa costuma cobrir um requisito diferente do enunciado; monte a resposta como um par ou trio complementar, não como opções parecidas.

✓ Desconfie de alternativas com recursos legados ou com aposentadoria anunciada (sentimento e frases-chave do Azure Language, Image Analysis 4.0, Custom Vision, Connected Agents, pro mode, workflows visuais) quando o cenário é novo ou de longo prazo.

✓ Distribua o estudo pelos pesos: IA generativa e agêntica (30 a 35%) e planejar e gerenciar (25 a 30%) somam mais da metade; visão, texto e extração ficam com 10 a 15% cada.

✓ Quando duas alternativas usam o mesmo serviço, a diferença está no parâmetro: confira dependências (PNG para transparência, markdownHeaderDepth só com markdown, tone só com LLM, extract só em documento).

✓ A prova está disponível em português do Brasil; se um termo traduzido gerar dúvida, pense no nome do recurso em inglês, que é como ele aparece no portal e no SDK.

Erros que derrubam candidatos

✗ Escolher a solução tecnicamente possível em vez da que atende à restrição do enunciado (por exemplo, Custom Translator quando o requisito é não treinar).

✗ Achar que trocar de projeto, recurso ou região sem mudar o padrão de envio resolve cota e 429.

✗ Confundir direção ou momento: custom speech (reconhecer) x SSML (sintetizar), avaliação contínua (tráfego real) x agendada (drift), User input x Tool response.

✗ Usar parâmetros de modelos sem raciocínio (temperature, top_p, max_tokens) em modelos de raciocínio.

✗ Aceitar chave em Key Vault, nas instruções ou na especificação como se fosse autenticação sem chave.

✗ Esquecer que blocklists só casam texto exato e não analisam imagens nem variações de ataque.

✗ Ignorar o 'Escolha DUAS' e marcar só uma, ou marcar duas alternativas que resolvem o mesmo requisito.

Na semana da prova

- ☐ Refaça as provas práticas e, em cada erro, anote qual restrição do enunciado você deixou passar.
- ☐ Revise as tabelas de pares que mais derrubam: tipos de implantação, orquestrações do Agent Framework, avaliadores, modos de transcrição e analisadores do Content Understanding.
- ☐ Releia os fluxos de código: function_call e function_call_output, mcp_approval_request e mcp_approval_response, conversations.create, background e polling, videos.create e download.
- ☐ Memorize a lista do que é legado ou tem aposentadoria anunciada e o substituto de cada um.
- ☐ Na véspera, revise só os blocos de lembrar e de números de cada capítulo e descanse.

REVISÃO FINAL

Folha de cola

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

1. Implementar soluções de IA generativa e agênticas

‣ resposta traz function_call com call_id e o agente não conclui → **Executar a função e enviar function_call_output com o mesmo call_id**

‣ require_approval always e item mcp_approval_request → **mcp_approval_response com approve e approval_request_id + previous_response_id**

‣ servidor MCP com dezenas de ferramentas, agente só pode usar algumas → **allowed_tools**

‣ agente não lembra a mensagem anterior → **conversations.create e conversation=id em cada responses.create**

‣ pedido de vários minutos, front-end encerra a conexão → **background=true e polling do status**

⚠ Aprovação MCP só avança com mcp_approval_response (approve e approval_request_id) + previous_response_id.

⚠ Isso elimina o controle humano exigido. Para restringir ferramentas use allowed_tools; para revisar, mantenha require_approval.

⚠ Function tools rodam no cliente: execute e devolva function_call_output com o call_id.

2. Implementar soluções de análise de texto

‣ ocultar dados pessoais mantendo o PDF ou DOCX original → **Document PII**

‣ mascarar CPF e nomes em texto antes do LLM, resposta síncrona → **Text PII do Azure Language**

‣ medicamento, dosagem, relação entre eles e negação de sintoma → **Text analytics for health**

‣ entidades próprias, exemplos rotulados, on-premises → **Custom NER em contêiner Docker**

‣ sentimento e tom em JSON fixo, projeto novo → **Modelo do Foundry + structured outputs com enum**

⚠ São legados (aposentadoria 31/03/2029) e não identificam ironia. Use LLM com structured outputs.

⚠ A capacidade feita para detectar e redigir dados pessoais é a detecção de PII.

⚠ Perde o contexto entre turnos. Conversation PII como job assíncrono.

3. Implementar soluções de extração de informações

‣ PDF escaneado indexado com campo content vazio → **imageAction generateNormalizedImages + skill OCR + Text Merge**

‣ códigos exatos e perguntas em linguagem natural na mesma base → **Busca híbrida (RRF) + ranker semântico**

‣ medir revocação do HNSW sem criar outro índice → **exhaustive true na consulta**

‣ filtro muito seletivo, k pequeno, poucos ou nenhum resultado → **vectorFilterMode preFilter**

‣ resultados vetoriais irrelevantes depois de configurar o vectorizer → **Vectorizer com o mesmo modelo da skill de embedding**

⚠ Sem imageAction generateNormalizedImages e skill OCR (com Text Merge) o texto da imagem nunca entra no índice.

⚠ Com none nenhuma imagem é extraída, e descrição não substitui a transcrição do texto.

⚠ A consulta não falha, mas os vetores ficam em espaços diferentes. Skill e vectorizer = mesmo modelo.

4. Implementar soluções de visão computacional

‣ número pequeno, etiqueta ou gravação que o modelo não lê → **detail high na parte image_url**

‣ várias fotos analisadas em conjunto com um resumo único → **Várias partes image_url na mesma mensagem (até 10)**

‣ ícone ou acessório com fundo transparente → **background transparent + output_format png**

‣ trocar só uma região da foto mantendo o resto → **Endpoint edits + máscara PNG transparente do mesmo tamanho + prompt**

‣ rostos ou logotipo saem diferentes após a edição → **input_fidelity high (fora do gpt-image-1-mini)**

⚠ JPEG não tem canal alfa. Transparente exige PNG; output_compression só vale para JPEG.

⚠ GPT-image devolve base64 e gera PNG ou JPEG; response_format e WEBP não são suportados.

⚠ Máscara = PNG, mesmas dimensões, transparente onde vai mudar; opaco é o que se preserva.

5. Planejar e gerenciar uma solução de IA do Azure

‣ prompts e respostas só na União Europeia, pagar por token, cota maior → **Data Zone Standard em região da zona EU**

‣ processar dentro da geografia do Brasil → **Standard (regional), não Data Zone**

‣ milhões de itens, pode levar um dia, menor custo, não afetar a cota online → **Global Batch com arquivo JSONL**

‣ PTUs esgotam no pico e não quer comprar mais → **Spillover para Standard do mesmo modelo e versão (spilloverDeploymentName)**

‣ avaliar modelo ajustado por algumas horas, barato, sem SLA → **Tipo Developer (24 h)**

⚠ Cota é por assinatura + região + modelo + tipo de implantação. Só ajuda reduzir o TPM de uma implantação existente ou pedir aumento de cota.

⚠ Provisioned é dimensionado em PTUs. Sem comprar PTU, a saída é spillover para uma Standard do mesmo modelo e versão.

⚠ Batch é assíncrono com meta de 24 horas; não atende tráfego interativo.

GABARITO

Respostas do diagnóstico

D1. A) Usar arestas switch-case que levam cada mensagem ao executor conforme a classificação

→ revise o capítulo 1

Por quê: Item do guia 'Orquestrar vários modelos, fluxos ou mecanismos híbridos de LLMs e de regras': workflows em grafo combinam agentes com modelos diferentes e executores de código, com roteamento condicional.

✗ B) Incorreta. Fan-out envia a mesma mensagem a vários destinos para processamento paralelo, desperdiçando chamadas.
✗ C) Incorreta. Sequential faria cada pedido passar por todas as etapas, aumentando custo e latência.
✗ D) Incorreta. O model router escolhe modelos para prompts; não executa regras de negócio determinísticas.

> **Dica de prova:** Rotear por classificação no workflow = arestas switch-case (ou condicionais).

D2. A) Devolver de imediato um status pendente e consultar a aprovação em um fluxo separado

→ revise o capítulo 1

Por quê: Salvaguardas em fluxos semiautônomos: aprovações humanas para ações com efeito colateral, validação dos argumentos das ferramentas como entrada não confiável e respeito ao tempo de vida das execuções.

✗ B) Incorreta. A documentação define que as execuções expiram 10 minutos após a criação; não há um timeout de 24 horas a configurar na ferramenta.
✗ C) Incorreta. Remover a aprovação elimina o controle humano exigido no processo de reembolso.
✗ D) Incorreta. Bloquear a função até a decisão faz a execução expirar antes de a saída ser enviada.

> **Dica de prova:** Aprovação demorada? Status pendente já + polling (a execução expira em 10 min).

D3. D) O SDK da OpenAI apontado para o endpoint /openai/v1 do recurso

→ revise o capítulo 1

Por quê: Item do guia 'Integrar fluxos de trabalho generativos em aplicativos usando SDKs e conectores do Foundry': cada SDK tem seu endpoint; o recurso do Foundry expõe tanto o endpoint do projeto quanto o /openai/v1.

✗ A) Incorreta. O Agent Framework serve para agentes e sistemas multiagentes; gerar embeddings não exige um agente.
✗ B) Incorreta. Os SDKs das Foundry Tools atendem serviços específicos, como Speech ou Language, e não modelos de embeddings da OpenAI.
✗ C) Incorreta. O SDK do Foundry é indicado para agentes, avaliações e recursos específicos do Foundry, não como escolha principal para embeddings em massa.

> **Dica de prova:** Embeddings e máxima compatibilidade OpenAI = SDK da OpenAI no /openai/v1.

D4. B) Treinar um modelo de custom speech e comparar o WER com o modelo base → revise o capítulo 2

Por quê: Modelos de fala personalizados: compare a transcrição do modelo base com uma transcrição humana; se o WER for alto, treine custom speech e implante um endpoint, dispensável só no uso com batch transcription.

✗ A) Incorreta. SSML controla a síntese de voz (texto para fala), e não o reconhecimento.

✗ C) Incorreta. A diarização identifica quem fala, mas não corrige termos técnicos reconhecidos de forma errada.

✗ D) Incorreta. Batch transcription muda o modo de processamento, e não o vocabulário reconhecido; ela pode inclusive usar um modelo personalizado.

> **Dica de prova:** Termos técnicos errados na transcrição = custom speech (avalie com WER).

D5. C) Treinar custom speech com áudios e transcrições humanas e usá-lo no lote sem endpoint hospedado → revise o capítulo 2

Por quê: Item do guia 'Integrar a fala como uma modalidade de agente, incluindo modelos de fala personalizados': o custom speech melhora o vocabulário e as condições de áudio, e o WER mostra se o treinamento trouxe ganho.

✗ A) Incorreta. Phrase lists não estão disponíveis na transcrição em lote e não resolvem o problema de ruído.

✗ B) Incorreta. Custom voice cria vozes sintéticas para text to speech; não melhora o reconhecimento de fala.

✗ D) Incorreta. Um endpoint dedicado gera custo de hospedagem desnecessário quando o uso é só em lote.

> **Dica de prova:** Ruído e jargão em lote = custom speech treinado com áudio; no lote não precisa de endpoint.

D6. B) SSML com phoneme ou um lexicon personalizado para esses termos → revise o capítulo 2

Por quê: Converter texto em fala para interações agênticas inclui ajustar a saída com SSML: pronúncia, pausas, estilos e várias vozes no mesmo documento.

✗ A) Incorreta. Traduzir os termos mudaria o conteúdo, e não a pronúncia em português.

✗ C) Incorreta. Fast transcription transcreve áudio; não corrige a pronúncia da voz sintetizada.

✗ D) Incorreta. Custom speech melhora o reconhecimento de fala (fala para texto), e não a síntese.

> **Dica de prova:** Voz pronuncia errado? SSML com phoneme ou lexicon.

D7. D) prebuilt-documentSearch do Content Understanding → revise o capítulo 3

Por quê: Produzir representações limpas e fundamentadas para agentes e RAG com Content Understanding: os analisadores de RAG (documentSearch, imageSearch, audioSearch e videoSearch) geram Markdown e saídas semânticas prontas para indexação.

✗ A) Incorreta. prebuilt-imageSearch descreve imagens avulsas; não trata documentos com várias páginas e tabelas.

✗ B) Incorreta. prebuilt-invoice extrai campos de faturas, e não uma representação completa para RAG.

✗ C) Incorreta. prebuilt-read extrai o texto (OCR), mas não descreve figuras, não resume nem prepara chunks semânticos.

> **Dica de prova:** Documento para RAG em Markdown, com resumo e figuras descritas = prebuilt-documentSearch.

D8. D) Pela ferramenta MCP com uma conexão do projeto, dando Search Index Data Reader à identidade do projeto → revise o capítulo 3

Por quê: Item do guia 'Conectar pipelines de recuperação diretamente para fluxos de trabalho e ferramentas de agente': bases de conhecimento do Foundry IQ são expostas aos agentes como ferramenta MCP; criar a conexão exige a função Foundry Project Manager.

✗ A) Incorreta. Copiar índices para um vector store abandona a recuperação agêntica e as permissões da base de conhecimento.

✗ B) Incorreta. Gravar a chave nas instruções expõe o segredo e não é autenticação sem chave.

✗ C) Incorreta. O portal não é uma API para o agente, e Search Service Contributor gerencia o serviço, sem relação com a leitura pedida.

> **Dica de prova:** Foundry IQ no agente = ferramenta MCP + conexão do projeto + Search Index Data Reader.

D9. A) Definir imageAction como generateNormalizedImages no indexador e incluir a skill de OCR → revise o capítulo 3

Por quê: Item do guia 'Configurar o fluxo de ingestão de RAG, incluindo documentos e usando o OCR': a normalização de imagens é pré-requisito para as skills de OCR e de análise de imagem, e a extração de imagens é cobrada pelo Azure AI Search.

✗ B) Incorreta. storageMetadata extrai só metadados do blob, e frases-chave dependem de um texto que ainda não existe.

✗ C) Incorreta. jsonArray serve para arquivos JSON, e dividir texto não resolve a falta de texto extraído.

✗ D) Incorreta. O ranqueador semântico atua nas consultas e não extrai texto de imagens durante a indexação.

> **Dica de prova:** PDF escaneado sem texto = imageAction + skill de OCR.

D10. C) Criar um novo vídeo informando remix_video_id do vídeo aprovado e o ajuste no prompt → revise o capítulo 4

Por quê: O Sora 2 (preview) usa a API v1, gera vídeos de forma assíncrona (criar o job, consultar o status e baixar o MP4) e suporta texto para vídeo, imagem para vídeo e vídeo gerado para vídeo, com áudio.

✗ A) Incorreta. input_reference ancora só o primeiro quadro, sem preservar o movimento do vídeo original.

✗ B) Incorreta. A API de edição de imagens trabalha com imagens, não com vídeos.

✗ D) Incorreta. Gerar do zero não garante manter a cena aprovada; o remix existe justamente para ajustes direcionados.

> **Dica de prova:** Ajustar vídeo gerado mantendo a cena = remix_video_id.

D11. D) enableSegment como true e contentCategories descrevendo em linguagem natural como segmentar → revise o capítulo 4

Por quê: A análise de vídeo do Content Understanding tem duas etapas: extração de conteúdo (transcrição WebVTT, quadros-chave, detecção de shots) e extração de campos com segmentação. Lembre a limitação de amostragem de cerca de 1 quadro por segundo, redimensionado para 512x512.

✗ A) Incorreta. Os tempos de corte de câmera vêm da detecção de shots e não correspondem aos limites lógicos de cada procedimento.

✗ B) Incorreta. enableSegment false trata o vídeo inteiro como um único segmento, e o método extract é suportado só para documentos.

✗ C) Incorreta. Analisadores de vídeo derivam de prebuilt-video, não de prebuilt-document.

> **Dica de prova:** Dividir vídeo por lógica de negócio = enableSegment + contentCategories.

D12. A) Dividir as fotos em requisições de no máximo 10 imagens e pedir uma legenda por imagem → revise o capítulo 4

Por quê: Modelos de chat com visão aceitam imagens por URL ou dados base64, com até 10 imagens por chamada. Para legendas concisas ou detalhadas, ajuste a instrução (por exemplo, uma frase de até 15 palavras ou um parágrafo descritivo) e o parâmetro detail.

✗ B) Incorreta. O tipo de implantação muda custo e latência, não o limite de imagens por chamada de chat.

✗ C) Incorreta. A API de chat com visão recebe imagens em image_url; agrupar em PDF não contorna o limite.

✗ D) Incorreta. Reduzir a resolução diminui tokens, mas não muda o número máximo de imagens por requisição.

> **Dica de prova:** Muitas imagens? Até 10 por requisição de chat.

Agora, treine no formato real

Faça as 6 provas completas do curso AI-103: 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.