


# Guia de preparação
# Google Cloud Digital Leader

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

| | | |
|---|---|---|
| **6** capítulos por domínio | **36** questões comentadas | **70%** nota de corte |
| **6** provas completas no curso | | |

ANTES DE COMEÇAR

# Como usar este guia

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

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

## Mapa do exame

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

| Domínio (capítulo) | Peso | Neste guia |
|---|---|---|
| 1. Confiança e Segurança com o Google Cloud | 18% | 19 conceitos · 13 pegadinhas |
| 2. Escalando com as Operações do Google Cloud | 10% | 16 conceitos · 13 pegadinhas |
| 3. Explorando a Transformação de Dados com o Google Cloud | 18% | 20 conceitos · 13 pegadinhas |
| 4. Inovando com a Inteligência Artificial do Google Cloud | 18% | 17 conceitos · 13 pegadinhas |
| 5. Modernizar a Infraestrutura e as Aplicações com o Google Cloud | 18% | 18 conceitos · 14 pegadinhas |
| 6. Transformação Digital com o Google Cloud | 18% | 19 conceitos · 12 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 · Confiança e Segurança com o Google Cloud

**Uma empresa quer a garantia de que o provedor passa por auditorias independentes regularmente. Qual evidência pode solicitar?**

A. Promessas verbais do vendedor da conta
B. Certificações e relatórios de auditoria
C. Anúncios em redes sociais do provedor
D. Depoimentos de clientes sem verificação

D2 · Confiança e Segurança com o Google Cloud

**Uma empresa quer registrar quem alterou uma configuração crítica em um projeto do Google Cloud. Qual recurso fornece essa informação?**

A. Classe Coldline do Storage
B. Relatório de faturamento
C. Logs de auditoria do Cloud
D. Cloud Translation API

D3 · Confiança e Segurança com o Google Cloud

**Qual é a importância da integridade dos dados?**

A. Garante acesso rápido aos dados
B. Garante baixo custo de armazenamento
C. Garante que ninguém veja os dados
D. Garante dados sem alteração indevida

D4 · Escalando com as Operações do Google Cloud

**Uma empresa quer impedir que desenvolvedores criem recursos em regiões não autorizadas. Qual recurso de governança ajuda?**

A. Relatórios de faturamento
B. Cloud CDN
C. APIs pré-treinadas
D. Políticas da organização

D5 · Escalando com as Operações do Google Cloud

**Uma equipe de SRE define que um serviço deve estar disponível 99,9% do tempo. Como se chama essa meta interna?**

A. Política de senhas da empresa
B. Relatório de custos mensal
C. Contrato de licença de software
D. Objetivo de nível de serviço (SLO)

D6 · Escalando com as Operações do Google Cloud

**Na hierarquia de recursos do Google Cloud, qual é o nível mais alto, que representa a empresa?**

A. Recurso individual
B. Projeto do Google Cloud
C. Pasta de departamento
D. Nó de organização

D7 · Explorando a Transformação de Dados com o Google Cloud

**Uma empresa guarda milhões de arquivos com padrão de acesso imprevisível e não quer escolher a classe de armazenamento de cada um. Qual opção do Cloud Storage reduz o custo de forma automática?**

A. Réplica em outra região
B. Autoclass no bucket
C. Classe Archive fixa
D. Classe Standard fixa

D8 · Explorando a Transformação de Dados com o Google Cloud

**Uma empresa guarda fotos e vídeos enviados por clientes, que são dados não estruturados. Qual produto é o mais adequado para armazená-los?**

A. BigQuery ML
B. Cloud Storage
C. Cloud SQL
D. Spanner

D9 · Explorando a Transformação de Dados com o Google Cloud

**Qual é a diferença entre dados estruturados e não estruturados?**

A. Não estruturados sempre estão em tabelas
B. Estruturados são só imagens e vídeos
C. Estruturados seguem um esquema definido
D. Não há diferença prática entre eles

D10 · Inovando com a Inteligência Artificial do Google Cloud

**Qual é a relação entre inteligência artificial e machine learning?**

A. São conceitos sem relação
B. IA é um subconjunto do ML
C. ML é um tipo de banco de dados
D. ML é um subconjunto da IA

D11 · Inovando com a Inteligência Artificial do Google Cloud

**Uma empresa quer um modelo que diferencie seu serviço e tem dados exclusivos e equipe experiente. Qual caminho oferece mais controle?**

A. Usar só APIs genéricas prontas
B. Treinar um modelo no Agent Platform
C. Comprar relatórios prontos de mercado
D. Criar regras fixas em uma planilha

D12 · Inovando com a Inteligência Artificial do Google Cloud

**Uma equipe de analistas que conhece SQL quer criar um modelo de previsão de vendas usando os dados que já estão no BigQuery. Qual recurso usar?**

A. BigQuery ML
B. Cloud Translation API
C. Cloud Armor
D. Speech-to-Text API

01

CAPÍTULO 1 · 18% DA PROVA

# Confiança e Segurança com o Google Cloud

Neste domínio eu preciso dominar a base conceitual de segurança (tríade confidencialidade, integridade e disponibilidade; autenticação, autorização e auditoria; menor privilégio; zero trust) e saber qual produto do Google Cloud resolve cada problema: IAM, Cloud Audit Logs, Cloud KMS com CMEK, Confidential Computing, Cloud Armor, Security Command Center, Google Security Operations, Google Threat Intelligence, Sensitive Data Protection e Model Armor. Também caem os estados dos dados, residência e soberania, e as fontes de confiança (Compliance Resource Center, Compliance Reports Manager, relatórios de transparência, princípios de privacidade). As alternativas erradas costumam ser outro pilar da mesma família, como integridade no lugar de confidencialidade ou autorização no lugar de autenticação. Onde mais erro: os três estados dos dados e os produtos de segurança com nomes novos.

**Neste capítulo**

19 tópicos-chave · 13 pegadinhas · 6 macetes · 14 gatilhos · 4 questões comentadas

## Tópicos-chave

### Tríade de segurança

Confidencialidade garante que só autorizados vejam (criptografia e IAM). Integridade garante dados sem alteração indevida (hashes, assinaturas, controles de acesso). Disponibilidade garante acesso quando preciso.

### Autenticação, autorização e auditoria

Autenticação confirma quem é. Autorização define o que pode fazer, como só leitura, por papéis do IAM. Auditoria registra quem fez o quê e quando.

### Cloud IAM e menor privilégio

O IAM controla quem pode fazer o quê em cada recurso por meio de papéis predefinidos ou personalizados. Menor privilégio é dar só as permissões necessárias; dar papel de proprietário a todos ou usar conta compartilhada viola o princípio. No desligamento de funcionário, revogar os acessos.

### Verificação em duas etapas (2SV)

Exige um segundo fator, como app ou chave de segurança, e protege mesmo se a senha vazar. Não dispensa senhas fortes, treinamento nem IAM.

### Cloud Audit Logs

Registram atividades administrativas e de acesso a dados: respondem quem alterou uma configuração crítica e quem acessou um recurso e quando.

### Três estados dos dados

Em repouso (armazenados), cifrados por padrão pelo Google Cloud sem ação extra do cliente; em trânsito (circulando na rede), protegidos por TLS e criptografia da rede; em uso (na memória, durante o processamento), protegidos pelo Confidential Computing.

### Chaves gerenciadas pelo cliente (CMEK)

Com o Cloud KMS, o cliente controla as próprias chaves e pode revogá-las, reforçando o controle sobre os dados. A criptografia padrão continua existindo.

### Cloud Armor

Protege aplicações na borda com regras de WAF, bloqueio por IP e país e proteção contra DDoS, junto com a escala da rede global do Google.

### Security Command Center

Central de gestão de riscos do Google Cloud: inventário de ativos, configurações erradas, vulnerabilidades, ameaças e riscos de cargas de IA, com prioridade do que corrigir primeiro.

### Google Security Operations

Antes chamado de Chronicle: ingere a telemetria de segurança da organização em escala e acelera detecção, investigação e resposta. O Gemini no Google Security Operations resume casos, busca eventos em linguagem natural e sugere respostas.

### Google Threat Intelligence

Visão proativa das ameaças: quais grupos atacam o setor e o que priorizar, combinando a visibilidade do Google, a experiência de resposta a incidentes da Mandiant e a base colaborativa do VirusTotal.

### Sensitive Data Protection

Incorporou a Cloud DLP: descobre, classifica e desidentifica (mascara) dados sensíveis, como CPF e números de cartão, em tabelas e arquivos.

### Model Armor

Filtra prompts e respostas de aplicações de IA generativa contra ataques como injeção de prompt e vazamento de dados, com qualquer modelo, sozinho ou integrado ao Security Command Center.

### Confiança zero e Identity-Aware Proxy

O IAP aplica o modelo de confiança zero do Google (BeyondCorp): cada acesso é verificado por identidade, dispositivo e contexto, sem depender de VPN. O perímetro tradicional confia em quem está na rede interna.

### Residência e soberania de dados

Exigências de manter dados no país ou região e controlar o acesso de suporte. O Google Cloud atende com escolha de regiões, políticas de localização de recursos, criptografia e soluções de soberania.

### Fontes de conformidade

Compliance Resource Center reúne informações de conformidade por setor e região. Compliance Reports Manager oferece certificações e relatórios de auditorias independentes, como ISO 27001 e SOC 2. Auditorias de terceiros validam os controles de forma independente.

### Princípios de confiança e privacidade

O cliente é dono dos seus dados e os controla; o Google não usa esses dados para publicidade nem os vende. Transparência informa como os dados são acessados (Access Transparency) e os relatórios de transparência mostram como o Google lida com pedidos de governos.

### Infraestrutura própria do Google

O Google projeta servidores, chips de segurança e rede, o que dá controle da cadeia de hardware e software e segurança em cada camada (defesa em profundidade).

### Ameaças comuns

Phishing usa mensagens falsas para roubar credenciais; ransomware cifra dados e pede resgate; DDoS sobrecarrega com tráfego de muitas origens; configurações erradas de permissões são causa comum de vazamentos na nuvem.

## Macetes para memorizar

**Tríade CID**

Confidencialidade = ninguém vê; Integridade = ninguém mexe; Disponibilidade = está lá quando preciso.

**Controle de acesso**

Autenticação = Quem é? Autorização = O que pode? Auditoria = O que fez?

**Estados dos dados**

Repouso = no disco (padrão); trânsito = na rede (TLS); uso = na memória (Confidential Computing).

**Produtos com Armor**

Cloud Armor protege a aplicação web; Model Armor protege o modelo de IA.

**Nomes antigos**

Chronicle virou Google Security Operations; Cloud DLP virou Sensitive Data Protection.

**Compliance**

Resource Center = consultar a documentação; Reports Manager = baixar o relatório para o auditor.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **Autenticação**<br>Provar quem é (senha, 2SV). | **Autorização**<br>Definir o que pode fazer (papéis do IAM, só leitura). |
| **Integridade**<br>Dados sem alteração indevida. | **Confidencialidade**<br>Só pessoas autorizadas veem os dados. |
| **Criptografia em trânsito**<br>Dados circulando na rede, como do navegador para a aplicação. | **Confidential Computing**<br>Dados em uso, cifrados na memória durante o processamento. |
| **Criptografia padrão**<br>Proteção automática em repouso sem ação do cliente. | **CMEK**<br>Cliente quer controlar e poder revogar as próprias chaves. |
| **Security Command Center**<br>Postura: configurações erradas, vulnerabilidades e riscos nos projetos do Google Cloud. | **Google Security Operations**<br>Ingerir telemetria da organização inteira para detectar, investigar e responder a ameaças. |
| **Google Threat Intelligence**<br>Saber quem ataca o setor e priorizar alertas. | **Cloud Armor**<br>Bloquear ataques web, países e DDoS na aplicação. |
| **Compliance Resource Center**<br>Documentação de conformidade por setor e região para o regulador. | **Compliance Reports Manager**<br>Baixar certificações e relatórios de auditoria independente para os auditores. |

## Números que caem

| Item | Valor |
|---|---|
| Estados dos dados | **3 (em repouso, em trânsito, em uso)** |
| Normas citadas em auditorias independentes | **ISO 27001 e SOC 2** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Marcar integridade quando a pergunta é sobre só autorizados verem os dados.
**Como não cair:** Ver é confidencialidade; não alterar é integridade; acessar quando preciso é disponibilidade.

⚠ **Alerta de prova:** Marcar autenticação quando o funcionário deve só ler, sem alterar.
**Como não cair:** Autenticação só comprova identidade; definir o que pode fazer é autorização pelo IAM.

⚠ **Alerta de prova:** Dizer que autorizar registra ações já realizadas.
**Como não cair:** Registrar ações é auditoria (Cloud Audit Logs).

⚠ **Alerta de prova:** Escolher criptografia em trânsito para proteger dados na memória durante o processamento.
**Como não cair:** Em trânsito protege a rede; em uso é Confidential Computing. Backup em outra região também não protege dados em uso.

⚠ **Alerta de prova:** Marcar em uso para dados enviados do navegador à aplicação.
**Como não cair:** Durante o envio, os dados estão em trânsito; armazenados, em repouso.

⚠ **Alerta de prova:** Achar que usar CMEK acaba com a criptografia padrão ou com o IAM.
**Como não cair:** A criptografia continua e o controle de acesso segue necessário; CMEK só aumenta o controle do cliente.

⚠ **Alerta de prova:** Escolher Cloud Armor com regras WAF, Sensitive Data Protection ou Cloud Audit Logs para saber quais grupos atacam o setor.
**Como não cair:** Cloud Armor bloqueia ataques, Sensitive Data Protection descobre e mascara dados sensíveis e Audit Logs registram atividades do projeto; inteligência sobre quem ataca é Google Threat Intelligence.

⚠ **Alerta de prova:** Proteger um chatbot contra injeção de prompt com Cloud NAT, CDN ou DNS.
**Como não cair:** Filtrar prompts e respostas de LLMs é Model Armor.

⚠ **Alerta de prova:** Achar que, na nuvem, o cliente não tem responsabilidades ou não precisa de controle de acesso.
**Como não cair:** A nuvem segue responsabilidade compartilhada: parte fica com o provedor, mas dados e acessos continuam do cliente.

⚠ **Alerta de prova:** Marcar perímetro tradicional para acesso por identidade e contexto sem VPN.
**Como não cair:** O perímetro confia na rede interna; sem VPN e verificando cada acesso é confiança zero (IAP).

⚠ **Alerta de prova:** Para residência na Europa, replicar dados para todos os continentes ou remover criptografia.
**Como não cair:** Residência exige escolha de regiões e controles de localização; replicar para todo lugar viola a exigência.

⚠ **Alerta de prova:** Confundir DDoS com phishing ou ransomware.
**Como não cair:** DDoS = tráfego de muitas origens; roubo de senha por e-mail = phishing; cifrar arquivos com resgate = ransomware.

⚠ **Alerta de prova:** Aceitar promessas do vendedor ou depoimentos como evidência de conformidade.
**Como não cair:** Evidência são certificações e relatórios de auditoria de terceiros (Compliance Reports Manager).

## Como cai na prova

📝 **Como cai:** Definição de pilar ou etapa com as outras três da mesma família como distratores.
**Como resolver:** Ache o verbo: ver (confidencialidade), alterar (integridade), acessar quando preciso (disponibilidade); provar (autenticação), permitir (autorização), registrar (auditoria).

📝 **Como cai:** Cenário de dados em determinado momento: enviado, armazenado ou processado.
**Como resolver:** Enviado = em trânsito; armazenado = em repouso; processado na memória = em uso e Confidential Computing.

📝 **Como cai:** Cenário de ameaça pede o produto de segurança.
**Como resolver:** Ataque web ou DDoS leva a Cloud Armor; risco de configuração leva a Security Command Center; detecção e resposta leva a Security Operations; quem ataca leva a Threat Intelligence; dado sensível leva a Sensitive Data Protection; prompt malicioso leva a Model Armor.

📝 **Como cai:** Regulador ou auditor pede comprovação.
**Como resolver:** Certificações e relatórios independentes (Compliance Reports Manager); documentação por setor (Compliance Resource Center); pedidos de governos (relatórios de transparência).

📝 **Como cai:** Múltipla resposta sobre responsabilidades do cliente ou proteção de contas.
**Como resolver:** Do cliente: identidades, acessos e dados. Proteger contas: 2SV e menor privilégio. Descarte tudo que é físico ou de rede do Google.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| quem pode fazer o quê, permissões mínimas | **Cloud IAM com papéis e menor privilégio** |
| quem alterou a configuração, quem acessou e quando | **Cloud Audit Logs** |
| controlar e revogar as próprias chaves | **CMEK com Cloud KMS** |
| cifrado inclusive na memória durante o processamento | **Confidential Computing** |
| bloquear países, ataques web, milhões de requisições | **Cloud Armor e proteção DDoS da rede global** |
| configurações erradas, vulnerabilidades em todos os projetos, priorizar correção | **Security Command Center** |
| telemetria de toda a organização, detectar, investigar e responder | **Google Security Operations** |
| Mandiant, VirusTotal, grupos que visam o setor | **Google Threat Intelligence** |
| achar e mascarar CPF e cartão | **Sensitive Data Protection** |
| prompts maliciosos, injeção de prompt em chatbot | **Model Armor** |
| acesso sem VPN por identidade, dispositivo e contexto | **Confiança zero com Identity-Aware Proxy** |
| dados não podem sair do país, acesso de suporte controlado | **Residência e soberania de dados** |
| pedidos de governos para acessar dados | **Relatórios de transparência do Google** |
| proteger contas contra roubo de senha | **2SV + IAM com menor privilégio** |

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

- Dados em repouso são cifrados por padrão, sem ação do cliente.
- O Google não usa dados de clientes do Google Cloud para publicidade e o cliente é dono dos dados.
- Zero trust = verificar identidade e contexto em cada acesso, sem VPN.
- Configuração errada de permissões é causa comum de vazamento: revisar com menor privilégio.

## Essenciais para a prova (checklist)

- ☐ Sei diferenciar confidencialidade, integridade e disponibilidade.
- ☐ Sei diferenciar autenticação, autorização e auditoria.
- ☐ Sei os três estados dos dados e a proteção de cada um.
- ☐ Sei quando usar CMEK em vez da criptografia padrão.
- ☐ Sei escolher entre Cloud Armor, Security Command Center, Google Security Operations e Google Threat Intelligence.
- ☐ Sei para que servem Sensitive Data Protection e Model Armor.
- ☐ Sei explicar confiança zero e o papel do Identity-Aware Proxy.
- ☐ Sei onde buscar evidências de conformidade e transparência.
- ☐ Sei explicar residência e soberania de dados com exemplos.

## Questões de revisão do capítulo

R1.1 · Confiança e Segurança com o Google Cloud

**Qual é a importância da disponibilidade na tríade de segurança?**

A. Garante que os custos sejam baixos
B. Garante que os dados não mudem
C. Garante acesso quando preciso
D. Garante que os dados sejam secretos

R1.2 · Confiança e Segurança com o Google Cloud

**Qual é uma vantagem da criptografia padrão dos dados armazenados no Google Cloud?**

A. Eliminação dos riscos existentes
B. Acesso livre para qualquer pessoa
C. Proteção sem ação extra do cliente
D. Dispensa de controle de acesso

R1.3 · Confiança e Segurança com o Google Cloud

**Qual é a diferença entre autenticação e autorização?**

A. Autenticar define permissões; autorizar prova identidade
B. Autenticar prova quem é; autorizar define o que pode
C. Autorizar registra ações já realizadas
D. São a mesma etapa com nomes diferentes

R1.4 · Confiança e Segurança com o Google Cloud

**Uma empresa quer ter controle total sobre as chaves de criptografia e poder revogá-las. Qual opção é a mais adequada?**

A. Nenhuma criptografia nos dados
B. Chaves gerenciadas pelo cliente
C. Chaves padrão só do provedor
D. Senha anotada em papel

## Respostas comentadas

### R1.1. C) Garante acesso quando preciso

Por quê: Confidencialidade, integridade e disponibilidade são pilares da segurança.

✗ A) Incorreta. Não é pilar de segurança.
✗ B) Incorreta. Isso é integridade.
✗ D) Incorreta. Isso é confidencialidade.

> **Dica de prova:** Disponibilidade: acesso quando preciso.

### R1.2. C) Proteção sem ação extra do cliente

Por quê: O Google Cloud cifra dados em repouso por padrão.

✗ A) Incorreta. Riscos continuam existindo.
✗ B) Incorreta. É o contrário.
✗ D) Incorreta. O IAM continua necessário.

> **Dica de prova:** Criptografia padrão: proteção automática.

### R1.3. B) Autenticar prova quem é; autorizar define o que pode

Por quê: Autenticação, autorização e auditoria são pilares do controle de acesso.

✗ A) Incorreta. É o contrário.
✗ C) Incorreta. Isso é auditoria.
✗ D) Incorreta. São etapas distintas.

> **Dica de prova:** Autenticar: quem é; autorizar: o que pode.

### R1.4. B) Chaves gerenciadas pelo cliente

Por quê: Chaves gerenciadas pelo cliente (CMEK) aumentam o controle.

✗ A) Incorreta. Reduz a segurança.

✗ C) Incorreta. Dão menos controle ao cliente.

✗ D) Incorreta. Não é gestão de chaves.

> **Dica de prova:** Controle das chaves: gerenciadas pelo cliente.

02

CAPÍTULO 2 · 10% DA PROVA

# Escalando com as Operações do Google Cloud

Este é o domínio de menor peso, mas é cheio de pares fáceis de trocar. Eu preciso dominar a gestão de custos (FinOps com pessoas, processos e tecnologia; orçamentos com alertas, cotas, rótulos; CapEx para OpEx e TCO), a hierarquia de recursos com herança de políticas, as políticas da organização, e a confiabilidade no estilo SRE: SLI, SLO, SLA, quatro sinais de ouro, redundância entre zonas e regiões e backups. Também cai qual ferramenta do Google Cloud Observability usar: Monitoring, Logging, Trace, Profiler ou Error Reporting. Onde mais erro: SLO x SLA, cotas x orçamentos e Cloud Trace x Cloud Profiler.

**Neste capítulo**
16 tópicos-chave · 13 pegadinhas · 6 macetes · 14 gatilhos · 4 questões comentadas

## Tópicos-chave

### FinOps e governança financeira

Controlar custos de forma duradoura envolve pessoas, processos e tecnologia: uma equipe responsável (por exemplo, FinOps), processos de revisão e ferramentas como orçamentos, relatórios e cotas. Engenharia, finanças e negócio trabalham juntos; aprovar centralmente cada recurso cria gargalo.

### Etapas da gestão de custos

Visibilidade (primeiro passo, com relatórios do Cloud Billing), responsabilidade (recursos com dono) e otimização contínua (revisar gastos, não manter tudo no tamanho máximo).

### Orçamentos e alertas

Avisam quando o gasto de um projeto atinge um percentual do valor planejado, como 80% no mês. Servem para avisar; quem limita o consumo são as cotas.

### Cotas de recursos

Limitam o consumo por projeto e evitam erros como criar centenas de VMs sem querer. Não obtêm capacidade nem protegem dados.

### Rótulos (labels)

Etiquetas nos recursos que atribuem custos a equipes, projetos e ambientes.

### Hierarquia de recursos

Organização (topo, representa a empresa), pastas (departamentos), projetos e recursos. Políticas aplicadas em uma pasta são herdadas pelos projetos abaixo dela.

### Políticas da organização

Aplicam restrições herdadas pela hierarquia, como impedir a criação de recursos em regiões não autorizadas. Relatórios de faturamento só mostram custos, sem restringir.

### CapEx para OpEx e TCO

A nuvem troca investimento de capital (comprar servidores a cada ciclo) por despesa operacional baseada no uso, que acompanha a demanda e fica mais variável. O TCO compara todos os custos, diretos e indiretos (hardware, energia, espaço, pessoal, manutenção), e não só a fatura mensal.

### SLI, SLO e SLA

SLI é a medida do desempenho do serviço; SLO é a meta interna de confiabilidade, como 99,9% de disponibilidade; SLA é o compromisso contratual, com consequências por falhas. SLOs costumam ser mais rigorosos que os SLAs.

### SRE

Engenharia de confiabilidade de sites: aplica engenharia de software às operações, com SLIs, SLOs, SLAs, automação e monitoramento, buscando sistemas confiáveis.

### Quatro sinais de ouro

Latência, tráfego, erros e saturação: medem desempenho e confiabilidade de um serviço voltado ao usuário, acompanhados com o Cloud Monitoring.

### Google Cloud Observability

Reúne Cloud Monitoring (métricas, painéis, verificações de disponibilidade e alertas), Cloud Logging (guarda e analisa logs), Cloud Trace (rastreamento distribuído e gargalos de latência), Cloud Profiler (consumo de CPU e memória do código) e Error Reporting (agrupa e conta exceções e alerta sobre novas).

### DevOps e excelência operacional

CI/CD (integração e entrega contínuas) entrega software com mais frequência e menos erros. Excelência operacional combina automatizar implantações, monitorar métricas e registros e definir responsabilidades claras.

### Tolerância a falhas e redundância

Tolerância a falhas é continuar funcionando com falhas, não eliminar a chance de falhar. Para resistir à queda de uma zona, distribuir em várias zonas; para resistir à perda de uma região, implantar em várias regiões.

### Backups e Backup and DR Service

Backups isolados e testados permitem restaurar dados e retomar a operação após ransomware; o Backup and DR Service protege e restaura cargas de trabalho. Backups recuperam dados, mas não mantêm o serviço no ar durante a falha.

### Dynamic Workload Scheduler e Spot VMs

O Dynamic Workload Scheduler obtém GPUs e TPUs por períodos definidos, sem reservas longas, assim que houver capacidade. Spot VMs reduzem custo para cargas tolerantes a interrupções.

## Macetes para memorizar

### 💡 SLI, SLO, SLA

I de Indicador (mede), O de Objetivo (meta), A de Acordo (contrato).

### 💡 Hierarquia

O-P-P-R: Organização, Pastas, Projetos, Recursos — a política desce como cascata.

### 💡 Sinais de ouro

LTES: Latência, Tráfego, Erros, Saturação.

### 💡 Observability

Monitoring = painel; Logging = diário; Trace = GPS da requisição; Profiler = raio-x do código; Error Reporting = caixa de erros.

### 💡 Controle de custo

Orçamento avisa, cota trava, rótulo separa, política proíbe.

### 💡 FinOps

Custo sob controle = Pessoas + Processos + Tecnologia (PPT).

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **SLO**<br>Meta interna, como 99,9% de disponibilidade, definida pela equipe de SRE. | **SLA**<br>Compromisso contratual com o cliente, com consequência por falhas. |
| **SLI**<br>A medida do desempenho do serviço. | **SLO**<br>A meta que essa medida precisa atingir. |
| **Orçamento com alerta**<br>Ser avisado quando o gasto chegar a um percentual do planejado. | **Cotas de recursos**<br>Limitar o consumo e evitar criar recursos demais por erro. |
| **Cloud Trace**<br>Descobrir em qual serviço a requisição perde mais tempo (latência entre serviços). | **Cloud Profiler**<br>Ver onde o código consome CPU e memória. |
| **Cloud Monitoring**<br>Métricas de CPU, latência e disponibilidade, painéis e alertas por limite. | **Error Reporting**<br>Agrupar, contar e alertar sobre novas exceções no código. |
| **Várias zonas**<br>Resistir à queda de uma zona. | **Várias regiões**<br>Resistir à perda de uma região inteira. |
| **Políticas da organização**<br>Restringir o que pode ser criado, como regiões não autorizadas. | **Rótulos**<br>Separar e atribuir custos por equipe, projeto ou ambiente. |

## Números que caem

| Item | Valor |
|---|---|
| Exemplo de SLO de disponibilidade | **99,9% do tempo** |
| Exemplo de alerta de orçamento | **80% do valor planejado no mês** |
| Níveis da hierarquia de recursos | **4 (organização, pastas, projetos, recursos)** |
| Sinais de ouro do monitoramento | **4 (latência, tráfego, erros, saturação)** |
| Peso do domínio na prova | **cerca de 10%** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Marcar SLO como contratual e SLA como meta interna.
**Como não cair:** É o contrário: SLA = contrato; SLO = meta interna, geralmente mais rigorosa.

⚠ **Alerta de prova:** Definir SLI como meta interna ou contrato com multa.
**Como não cair:** Meta é SLO, contrato é SLA; SLI é só a medida.

⚠ **Alerta de prova:** Achar que uma ferramenta de faturamento sozinha controla custos.
**Como não cair:** Ferramentas sem processos e responsáveis não mudam o comportamento: pessoas, processos e tecnologia.

⚠ **Alerta de prova:** Usar cotas para obter GPUs por alguns dias.
**Como não cair:** Cotas limitam o consumo, mas não obtêm capacidade; para isso, Dynamic Workload Scheduler.

⚠ **Alerta de prova:** Usar escalonamento automático ou balanceamento para recuperar dados após ransomware.
**Como não cair:** Eles ajustam capacidade e distribuem tráfego; recuperar dados exige backups isolados e testados.

⚠ **Alerta de prova:** Achar que backups semanais mantêm um serviço crítico no ar na queda de uma zona.
**Como não cair:** Backups recuperam, mas não mantêm o serviço; distribua em várias zonas.

⚠ **Alerta de prova:** Implantar em uma única zona para resistir à perda de uma região.
**Como não cair:** Perda de região exige implantação em várias regiões.

⚠ **Alerta de prova:** Marcar projeto ou pasta como topo da hierarquia.
**Como não cair:** O topo é o nó de organização; pastas ficam abaixo dela e projetos abaixo das pastas.

⚠ **Alerta de prova:** Aplicar a mesma política em cada VM separadamente.
**Como não cair:** Aplique na pasta do departamento e deixe a herança levar aos projetos.

⚠ **Alerta de prova:** Achar que a nuvem muda de OpEx para CapEx ou torna a despesa fixa.
**Como não cair:** A nuvem vai de CapEx para OpEx e torna a despesa mais variável, acompanhando o uso.

⚠ **Alerta de prova:** Avaliar o custo da nuvem pela fatura mensal isolada ou pelo preço de uma VM.
**Como não cair:** TCO inclui custos evitados de hardware, energia, espaço, pessoal e manutenção.

⚠ **Alerta de prova:** Escolher Cloud Profiler ou Error Reporting para achar o serviço lento numa cadeia de microsserviços.
**Como não cair:** O caminho da requisição e a latência entre serviços são do Cloud Trace.

⚠ **Alerta de prova:** Achar que tolerância a falhas é eliminar a chance de falhar.
**Como não cair:** Falhas sempre podem ocorrer; tolerância é continuar funcionando com elas, usando redundância e distribuição.

## Como cai na prova

📝 **Como cai:** Cenário de custo pede o recurso de governança.
**Como resolver:** Avisar (orçamento com alerta), limitar consumo (cota), separar por equipe (rótulo), proibir região (política da organização), primeiro passo (visibilidade).

📝 **Como cai:** Definição de SLI, SLO ou SLA com os outros dois como distratores.
**Como resolver:** Medida é SLI, meta interna é SLO, contrato é SLA.

📝 **Como cai:** Cenário de diagnóstico pede a ferramenta do Google Cloud Observability.
**Como resolver:** Lentidão entre serviços (Trace), CPU e memória do código (Profiler), exceções (Error Reporting), métricas, painéis e alertas (Monitoring), logs (Logging).

📝 **Como cai:** Cenário de resiliência: queda de zona, perda de região ou ransomware.
**Como resolver:** Zona leva a várias zonas; região leva a várias regiões; dados cifrados por ransomware levam a backups isolados e testados.

📝 **Como cai:** Pergunta financeira sobre o modelo de pagamento da nuvem.
**Como resolver:** Pagar pelo uso = CapEx para OpEx, custos acompanham o uso real; comparar tudo = TCO com custos evitados.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| avisar quando o gasto chegar a 80% do planejado | **Orçamento com alerta de limite** |
| evitar criar centenas de VMs por erro | **Cotas de recursos** |
| quanto cada equipe gasta, etiquetas | **Rótulos (labels)** |
| impedir recursos em regiões não autorizadas | **Políticas da organização** |

| | |
|---|---|
| mesma política para todos os projetos do departamento | **Aplicar na pasta (herança)** |
| nível mais alto que representa a empresa | **Nó de organização** |
| parar de comprar servidores e pagar mensalmente pelo uso | **CapEx para OpEx** |
| comparar custo total incluindo energia, pessoal e manutenção | **TCO** |
| meta de 99,9% definida pela equipe | **SLO** |
| latência, tráfego, erros e saturação | **Quatro sinais de ouro** |
| em qual serviço a requisição perde tempo | **Cloud Trace** |
| exceções novas agrupadas e contadas | **Error Reporting** |
| painéis de CPU e disponibilidade com alerta por limite | **Cloud Monitoring** |
| GPUs por poucos dias sem reserva longa | **Dynamic Workload Scheduler** |

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

- SLA é contrato; SLO é meta interna; SLI é a medida.
- Políticas aplicadas na pasta são herdadas pelos projetos abaixo.
- Orçamento com alerta avisa quando o gasto chega ao percentual definido; quem limita o consumo são as cotas.
- Backups recuperam dados; redundância entre zonas e regiões mantém o serviço no ar.

## Essenciais para a prova (checklist)

- ☐ Sei explicar FinOps como pessoas, processos e tecnologia.
- ☐ Sei diferenciar orçamentos, cotas, rótulos e políticas da organização.
- ☐ Sei a hierarquia de recursos e como funciona a herança de políticas.
- ☐ Sei explicar CapEx para OpEx e o que entra no TCO.
- ☐ Sei diferenciar SLI, SLO e SLA.
- ☐ Sei os quatro sinais de ouro.
- ☐ Sei escolher entre Cloud Monitoring, Logging, Trace, Profiler e Error Reporting.
- ☐ Sei escolher entre várias zonas, várias regiões e backups conforme a falha.
- ☐ Sei para que serve o Dynamic Workload Scheduler.

## Questões de revisão do capítulo

R2.1 · Escalando com as Operações do Google Cloud

**Uma equipe de IA precisa de várias GPUs por alguns dias para um treinamento e quer evitar reservas longas, recebendo a capacidade assim que ela estiver disponível. Qual recurso do Google Cloud atende?**

A. Rótulos aplicados aos buckets
B. Dynamic Workload Scheduler
C. Cotas de recursos por projeto
D. Cache do Cloud CDN

R2.2 · Escalando com as Operações do Google Cloud

**O que é um indicador de nível de serviço (SLI)?**

A. Uma medida do desempenho do serviço
B. Um contrato com multa por falhas
C. Um relatório de custos da nuvem
D. Uma meta interna de confiabilidade

R2.3 · Escalando com as Operações do Google Cloud

**O que a engenharia de confiabilidade de sites (SRE) busca?**

A. Sistemas confiáveis com automação
B. Trabalho manual em cada tarefa diária
C. Eliminação do monitoramento
D. Mudanças sem qualquer controle

R2.4 · Escalando com as Operações do Google Cloud

**Uma empresa quer entender por que migrar para a nuvem pode reduzir o custo total, mesmo que a fatura mensal pareça alta. O que deve considerar?**

A. O valor da fatura mensal isolado
B. O preço de uma única VM
C. O custo do café da equipe
D. Custos evitados de hardware

## Respostas comentadas

### R2.1. B) Dynamic Workload Scheduler

Por quê: Controlar o consumo envolve cotas, alertas de orçamento, relatórios do Cloud Billing, Spot VMs e o Dynamic Workload Scheduler, que obtém GPUs e TPUs por períodos definidos sem reservas longas.

✗ A) Incorreta. Rótulos organizam custos, sem alocar GPUs.
✗ C) Incorreta. Cotas limitam o consumo, mas não obtêm capacidade.
✗ D) Incorreta. Faz cache de conteúdo, sem relação com GPUs.

> **Dica de prova:** GPUs por poucos dias, sem reserva longa: Dynamic Workload Scheduler.

### R2.2. A) Uma medida do desempenho do serviço

Por quê: SLI mede, SLO define a meta e SLA é o compromisso contratual.

✗ B) Incorreta. Isso é SLA.
✗ C) Incorreta. Não é SLI.
✗ D) Incorreta. Isso é SLO.

> **Dica de prova:** Medida do serviço: SLI.

### R2.3. A) Sistemas confiáveis com automação

Por quê: SRE aplica engenharia de software às operações, com SLIs, SLOs, SLAs e automação.

✗ B) Incorreta. É o contrário.
✗ C) Incorreta. Monitoramento é essencial.
✗ D) Incorreta. É o contrário.

> **Dica de prova:** SRE: confiabilidade com automação.

### R2.4. D) Custos evitados de hardware

Por quê: O TCO compara todos os custos, incluindo hardware, energia, espaço, pessoal e manutenção evitados, e não só a fatura mensal da nuvem.

✗ A) Incorreta. Ignora custos indiretos.
✗ B) Incorreta. Ignora outros custos.
✗ C) Incorreta. Não é relevante.

> **Dica de prova:** Custo total: incluir custos evitados.

03

CAPÍTULO 3 · 18% DA PROVA

# Explorando a Transformação de Dados com o Google Cloud

Aqui eu preciso escolher o armazenamento e o banco certos para cada tipo de dado e acesso: Cloud Storage e suas classes, Cloud SQL, Spanner, AlloyDB, Firestore, Bigtable e BigQuery. Também caem os pipelines (Pub/Sub, Dataflow, Managed Service for Apache Spark), a diferença entre data lake e data warehouse, o BI com Looker e conceitos de valor dos dados, como a cadeia de suprimentos de dados, dados de primeira, segunda e terceira parte, governança e democratização. As questões são de cenário com quatro produtos lado a lado, e a resposta sai de duas pistas: o tipo de dado (relacional, documento, objeto, série temporal) e o padrão de acesso (frequência, escala, tempo real). Onde mais erro: classes Nearline x Coldline x Archive, Cloud SQL x Spanner e Firestore x Bigtable.

**Neste capítulo**

20 tópicos-chave · 13 pegadinhas · 6 macetes · 14 gatilhos · 4 questões comentadas

## Tópicos-chave

### Dados estruturados x não estruturados

Estruturados seguem um esquema definido (tabelas). Não estruturados são fotos, vídeos, textos, contratos e posts; a nuvem permite processá-los em escala junto com os estruturados.

### Data lake

Guarda grandes volumes de dados brutos, estruturados e não estruturados, sem definir esquema agora, para análises futuras ainda não definidas. No Google Cloud, fica no Cloud Storage.

### Data warehouse

Consolida dados organizados de várias fontes para análises rápidas, BI e relatórios padronizados; elimina silos e dá visão única. É otimizado para análise, enquanto o banco transacional atende operações da aplicação.

### BigQuery

Data warehouse sem servidor do Google Cloud: consultas com SQL padrão, escala para petabytes, separa armazenamento e processamento e escala sem gerenciar infraestrutura. É o destino natural para migrar um warehouse local.

### BigQuery Omni e BigQuery sharing

O BigQuery Omni analisa dados guardados em outras nuvens sem copiá-los (análise multinuvem, evita silos). O BigQuery sharing facilita o compartilhamento seguro de dados entre parceiros.

### Classes do Cloud Storage

Standard para acesso frequente (diário); Nearline para cerca de uma vez por mês; Coldline para cerca de uma vez por trimestre; Archive para acesso anual ou menos e retenção longa com o menor custo, como auditoria por sete anos.

### Autoclass

Transiciona objetos automaticamente entre Standard, Nearline, Coldline e Archive conforme o acesso, sem cobrança de recuperação nem de exclusão antecipada. Ideal para padrão de acesso imprevisível.

### Gestão do ciclo de vida

Regras que movem objetos para classes mais baratas por idade ou condição. Não confundir com controle de versões (guarda versões antigas) nem assinatura de URL (acesso temporário).

### Cloud SQL

Banco relacional gerenciado para cargas tradicionais (MySQL, PostgreSQL), regional e com escala limitada. Destino para migrar o banco local sem mudar a aplicação.

### Spanner

Banco relacional com consistência forte, escala horizontal global e alta disponibilidade entre regiões e continentes. Indicado para reservas globais e instituições financeiras.

### AlloyDB para PostgreSQL

Opção gerenciada de alto desempenho compatível com PostgreSQL; caminho de modernização, por exemplo de um Oracle. Relacionais no Google Cloud: Cloud SQL, Spanner e AlloyDB.

### Firestore

Banco NoSQL de documentos para aplicativos web e móveis, com esquema flexível e sincronização em tempo real entre dispositivos, como perfis de jogadores.

### Bigtable

Banco NoSQL de colunas largas para grandes volumes com latência muito baixa: séries temporais, IoT, bilhões de leituras de sensores por dia.

### NoSQL

Esquema flexível e escala horizontal; inclui documentos, chave-valor e colunas largas. Junções entre tabelas são típicas dos relacionais.

### Pub/Sub, Dataflow e Managed Service for Apache Spark

Pub/Sub transporta mensagens de forma assíncrona e desacoplada, ingerindo eventos. Dataflow processa dados em lote e em fluxo. Managed Service for Apache Spark (antes Dataproc) executa jobs Spark existentes, com clusters gerenciados ou sem servidor, sem reescrever código.

### Database Migration Service

Migra bancos para serviços gerenciados como Cloud SQL e AlloyDB com mínimo de indisponibilidade, replicando as mudanças até a virada. O Storage Transfer Service transfere objetos e arquivos, não bancos em uso.

### Looker

Plataforma de BI integrada ao BigQuery: painéis em tempo real, alertas, autoatendimento para as áreas de negócio e definições de métricas consistentes (camada semântica). Data Studio também visualiza e compartilha dados.

### Cadeia de suprimentos de dados

Gênese (origem), coleta, processamento (limpar e transformar), armazenamento, análise e ativação, quando os insights geram ação, como disparar ofertas personalizadas.

### Dados de primeira, segunda e terceira parte

Primeira parte: coletados pela própria empresa. Segunda parte: dados primários de um parceiro, compartilhados diretamente sob acordo. Terceiros: comprados ou obtidos de fontes externas, como clima, e enriquecem a análise sem substituir os internos.

### Governança e democratização de dados

Governança define regras de qualidade, acesso, privacidade e ciclo de vida, garantindo qualidade e conformidade. Democratizar é dar acesso seguro a quem precisa, não publicar tudo nem restringir à TI. O primeiro passo para gerar valor é reunir dados confiáveis e acessíveis.

## Macetes para memorizar

### 💡 Classes do Cloud Storage

Standard = sempre; Nearline = mês (N de Next month); Coldline = trimestre (frio); Archive = ano (arquivo morto).

### 💡 Bancos relacionais

SQL regional = Cloud SQL; SQL planetário = Spanner; PostgreSQL turbinado = AlloyDB.

### 💡 NoSQL

Firestore = app no celular (documento); Bigtable = big volume de sensor (colunas largas).

### 💡 Pipeline

Pub/Sub recebe, Dataflow transforma, BigQuery analisa, Looker mostra.

**Cadeia de dados**

Gênese, Coleta, Processa, Armazena, Analisa, Ativa: nasce, pega, limpa, guarda, entende, age.

**Origem dos dados**

1ª parte = eu coletei; 2ª parte = meu parceiro me deu; 3ª parte = comprei de fora.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **Data lake**<br>Dados brutos, sem esquema, para exploração e ciência de dados futura. | **Data warehouse**<br>Dados organizados para relatórios padronizados, consultas rápidas e visão única. |
| **Cloud SQL**<br>Relacional tradicional em uma única região, migrando MySQL sem mudar a aplicação. | **Spanner**<br>Relacional com consistência forte e escala horizontal entre regiões e continentes. |
| **Firestore**<br>Documentos para app web ou móvel, com sincronização em tempo real. | **Bigtable**<br>Bilhões de leituras e escritas com latência muito baixa, IoT e séries temporais. |
| **Nearline**<br>Acesso cerca de uma vez por mês. | **Coldline**<br>Acesso cerca de uma vez por trimestre (Archive fica para anual ou menos). |
| **Autoclass**<br>Acesso imprevisível; a classe é ajustada automaticamente conforme o padrão de acesso. | **Regras de ciclo de vida**<br>Mover objetos para classes mais baratas por idade ou condição definida. |
| **Pub/Sub**<br>Ingerir e transportar mensagens e eventos de forma assíncrona. | **Dataflow**<br>Processar e transformar os dados em fluxo ou em lote. |
| **Database Migration Service**<br>Migrar banco em uso para Cloud SQL ou AlloyDB com pouca indisponibilidade. | **Storage Transfer Service**<br>Transferir objetos e arquivos entre armazenamentos. |

## Números que caem

| Item | Valor |
|---|---|
| Standard | **acesso frequente (diário)** |
| Nearline | **acesso cerca de uma vez por mês** |
| Coldline | **acesso cerca de uma vez por trimestre** |
| Archive | **acesso anual ou menos** |

| | |
|---|---|
| Etapas da cadeia de suprimentos de dados | **6 (gênese, coleta, processamento, armazenamento, análise, ativação)** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Escolher Nearline para acesso trimestral ou Coldline para acesso mensal.
**Como não cair:** Nearline = mensal; Coldline = trimestral; Archive = anual ou menos; Standard = frequente.

⚠ **Alerta de prova:** Fixar a classe Archive para arquivos de acesso imprevisível.
**Como não cair:** Archive encarece as leituras dos que voltam a ser acessados e Standard fixa mantém custo alto nos pouco usados; acesso imprevisível é Autoclass.

⚠ **Alerta de prova:** Achar que réplica em outra região reduz o custo de armazenamento.
**Como não cair:** Réplica aumenta a disponibilidade, mas não ajusta a classe ao uso.

⚠ **Alerta de prova:** Escolher Cloud SQL para relacional global com consistência forte.
**Como não cair:** Cloud SQL é relacional, mas regional e com escala limitada; global é Spanner.

⚠ **Alerta de prova:** Marcar Bigtable como banco relacional.
**Como não cair:** Bigtable é NoSQL de colunas largas e Firestore é NoSQL de documentos; relacionais são Cloud SQL, Spanner e AlloyDB.

⚠ **Alerta de prova:** Escolher Firestore para telemetria de bilhões de leituras.
**Como não cair:** Firestore é voltado a documentos de aplicativos; volume massivo com baixa latência é Bigtable.

⚠ **Alerta de prova:** Guardar dados de auditoria de sete anos no BigQuery ou Cloud SQL.
**Como não cair:** Dados parados com acesso raro e retenção longa vão para Cloud Storage Archive.

⚠ **Alerta de prova:** Usar Storage Transfer Service ou Cloud Interconnect para migrar um banco com pouca indisponibilidade.
**Como não cair:** O Interconnect só cria a conexão de rede e o Storage Transfer move arquivos; migrar o banco é o Database Migration Service.

⚠ **Alerta de prova:** Achar que Pub/Sub executa jobs Spark ou que Cloud Run functions roda Spark.
**Como não cair:** Jobs Spark existentes sem reescrever vão para o Managed Service for Apache Spark.

⚠ **Alerta de prova:** Confundir dados de segunda parte com dados de terceiros.
**Como não cair:** Parceiro com acordo direto = segunda parte; dados agregados e vendidos por fontes sem relação direta = terceiros.

⚠ **Alerta de prova:** Na cadeia de dados, chamar de coleta ou armazenamento a etapa de usar o insight para agir.
**Como não cair:** Usar insights para disparar ação é ativação; limpar e transformar é processamento; a origem é a gênese.

⚠ **Alerta de prova:** Achar que democratizar dados é publicar tudo na Internet.
**Como não cair:** Democratizar é acesso amplo com governança e segurança, para quem precisa.

⚠ **Alerta de prova:** Achar que o warehouse registra vendas em tempo real e o transacional guarda dados antigos.
**Como não cair:** É o contrário: o transacional registra as operações; o warehouse é otimizado para análise.

## Como cai na prova

📝 **Como cai:** Cenário descreve frequência de acesso a arquivos e pede a classe do Cloud Storage.
**Como resolver:** Traduza a frequência: diária (Standard), mensal (Nearline), trimestral (Coldline), anual ou auditoria (Archive), imprevisível (Autoclass).

📝 **Como cai:** Cenário pede o banco: quatro bancos nas alternativas.
**Como resolver:** Primeiro decida relacional ou NoSQL; depois a escala: relacional regional (Cloud SQL) ou global (Spanner); documentos de app (Firestore) ou volume massivo de baixa latência (Bigtable); análise (BigQuery).

📝 **Como cai:** Cenário de tempo real: painel contínuo, fraude, saldo atualizado.
**Como resolver:** Ingestão é Pub/Sub, processamento é Dataflow, destino analítico é BigQuery e visualização é Looker.

📝 **Como cai:** Múltipla resposta sobre BigQuery, Looker ou pipelines.
**Como resolver:** BigQuery: sem servidor e SQL padrão. Looker: métricas consistentes e autoatendimento. Pipelines: Pub/Sub e Dataflow. Descarte DDoS (Cloud Armor) e documentos (Firestore).

📝 **Como cai:** Conceito de valor dos dados: etapa da cadeia, tipo de dado de parceiro, governança.
**Como resolver:** Localize o verbo do enunciado: agir com insight (ativação), limpar (processamento); quem coletou o dado (primeira, segunda ou terceira parte).

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| SQL em petabytes sem gerenciar servidores, destino do warehouse local | **BigQuery** |
| fotos e vídeos de clientes, dados não estruturados | **Cloud Storage** |
| dados brutos sem esquema para uso futuro, clickstream | **Data lake no Cloud Storage** |
| acesso imprevisível, não quer escolher a classe | **Autoclass** |
| retenção longa, auditoria, quase nunca acessado | **Archive** |
| MySQL local para gerenciado sem mudar a aplicação, estoque em uma região | **Cloud SQL** |
| relacional global, consistência forte entre continentes | **Spanner** |
| app móvel, documentos, sincronização em tempo real | **Firestore** |
| IoT, séries temporais, bilhões de leituras com baixa latência | **Bigtable** |
| milhões de eventos por segundo processados em tempo real | **Pub/Sub + Dataflow** |
| jobs Apache Spark sem reescrever | **Managed Service for Apache Spark** |
| migrar banco com mínimo de indisponibilidade até a virada | **Database Migration Service** |
| painéis de autoatendimento, métricas consistentes, sem depender da TI | **Looker** |
| fraude no momento da compra, decisão com dados atuais | **Análise em fluxo em tempo real** |

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

- Relacionais: Cloud SQL, Spanner e AlloyDB. NoSQL: Firestore e Bigtable.
- Autoclass não cobra recuperação nem exclusão antecipada.
- Managed Service for Apache Spark é o nome atual do Dataproc.
- Warehouse = análise; transacional = operações; lake = dados brutos.

## Essenciais para a prova (checklist)

- ☐ Sei escolher a classe do Cloud Storage pela frequência de acesso.

- ☐ Sei quando usar Autoclass e quando usar regras de ciclo de vida.
- ☐ Sei diferenciar Cloud SQL, Spanner, AlloyDB, Firestore e Bigtable por cenário.
- ☐ Sei diferenciar data lake, data warehouse e banco transacional.
- ☐ Sei montar um pipeline de streaming com Pub/Sub, Dataflow e BigQuery.
- ☐ Sei quando usar Database Migration Service e Managed Service for Apache Spark.
- ☐ Sei os benefícios do Looker e do BigQuery Omni.
- ☐ Sei as seis etapas da cadeia de suprimentos de dados.
- ☐ Sei diferenciar dados de primeira, segunda e terceira parte.

## Questões de revisão do capítulo

R3.1 · Explorando a Transformação de Dados com o Google Cloud

**O que significa democratizar os dados em uma empresa?**

A. Restringir dados à equipe de TI
B. Publicar todos os dados na Internet
C. Apagar dados sem uso aparente
D. Dar acesso seguro a quem precisa

R3.2 · Explorando a Transformação de Dados com o Google Cloud

**Uma empresa quer guardar dados de clickstream brutos para uso futuro em ciência de dados, sem definir um esquema agora. Qual solução é a mais adequada?**

A. Data lake no Cloud Storage
B. Planilha compartilhada do time
C. Tabela relacional com esquema fixo
D. Banco de documentos em app móvel

R3.3 · Explorando a Transformação de Dados com o Google Cloud

**Uma empresa quer migrar seu data warehouse local para o Google Cloud e aproveitar análises sem servidor. Qual produto é o destino natural?**

A. Cloud Run
B. Firestore
C. BigQuery
D. Cloud CDN

R3.4 · Explorando a Transformação de Dados com o Google Cloud

**Uma rede de lojas quer ver as vendas em painéis atualizados em tempo real, a partir dos dados no BigQuery. Qual combinação atende?**

A. Firestore com relatórios impressos
B. Pub/Sub com e-mails diários
C. Cloud Storage com planilhas locais
D. BigQuery com painéis no Looker

## Respostas comentadas

### R3.1. D) Dar acesso seguro a quem precisa

Por quê: Democratização combina acesso amplo com governança e segurança.

✗ A) Incorreta. É o contrário.
✗ B) Incorreta. É risco de segurança.
✗ C) Incorreta. Não é o conceito.

> **Dica de prova:** Democratizar: acesso seguro a quem precisa.

### R3.2. A) Data lake no Cloud Storage

Por quê: Data lakes em Cloud Storage guardam dados brutos para análises futuras.

✗ B) Incorreta. Não escala.
✗ C) Incorreta. Exige esquema definido.
✗ D) Incorreta. Não é o propósito.

> **Dica de prova:** Dados brutos sem esquema: data lake.

### R3.3. C) BigQuery

Por quê: BigQuery é o data warehouse gerenciado do Google Cloud.

✗ A) Incorreta. Executa contêineres.
✗ B) Incorreta. Banco de documentos.
✗ D) Incorreta. Entrega conteúdo.

> **Dica de prova:** Destino do warehouse: BigQuery.

### R3.4. D) BigQuery com painéis no Looker

Por quê: Looker sobre BigQuery oferece relatórios e painéis em tempo real.

✗ A) Incorreta. Não oferece painéis.
✗ B) Incorreta. Não é visualização em tempo real.
✗ C) Incorreta. Não oferece tempo real.

> **Dica de prova:** Painéis em tempo real: BigQuery e Looker.

04

CAPÍTULO 4 · 18% DA PROVA

# Inovando com a Inteligência Artificial do Google Cloud

Aqui eu preciso separar IA, ML, IA generativa e IA agêntica, e saber escolher o caminho certo de solução: API pré-treinada, AutoML, BigQuery ML, modelo personalizado ou agentes prontos. A prova cobra muito o nome atual das ofertas: Gemini Enterprise Agent Platform (antes Vertex AI), Agent Studio, ADK, Agent Runtime, Gemini Enterprise, AI Hypercomputer e Cloud TPU. Também caem qualidade de dados (seis dimensões), IA responsável (vieses, explicabilidade) e os tipos de problema que o ML resolve. O enunciado quase sempre dá a pista em prazo, esforço, dados próprios e experiência da equipe. Onde mais erro: confundir as APIs entre si (Speech-to-Text x Text-to-Speech) e Gemini Enterprise (para funcionários) com Gemini Enterprise Agent Platform (para desenvolvedores).

**Neste capítulo**

17 tópicos-chave · 13 pegadinhas · 6 macetes · 14 gatilhos · 4 questões comentadas

## Tópicos-chave

### IA, ML e BI

IA são máquinas fazendo tarefas cognitivas; ML é um subconjunto da IA que aprende padrões com dados para prever e automatizar decisões. BI e análise tradicional descrevem e visualizam o que já aconteceu.

### IA generativa

Cria conteúdo novo, como textos e imagens, a partir de instruções em linguagem natural. Resumir contratos, responder perguntas e redigir e-mails são tarefas atendidas por modelos de fundação como o Gemini.

### IA agêntica

Agentes que planejam, usam ferramentas e executam tarefas de várias etapas rumo a um objetivo, com humanos cuidando das exceções e decisões sensíveis. Remodela atendimento, vendas, operações, pesquisa e produtividade.

### O que o ML resolve

Encontrar padrões em muitos dados, previsão com base em histórico, classificação, recomendação e detecção de anomalias. Substitui ou simplifica sistemas de regras manuais que ficam complexos demais, como regras antifraude desatualizadas.

### APIs pré-treinadas

Vision (texto, objetos, rótulos, logotipos e rostos em imagens), Natural Language (sentimento, entidades e sintaxe), Cloud Translation, Speech-to-Text (áudio para texto) e Text-to-Speech (texto para áudio). São o caminho mais rápido e de menor esforço para casos comuns e genéricos.

### Document AI

Solução pronta que extrai campos, como nome e CPF, de formulários e documentos digitalizados sem treinar modelos do zero.

### AutoML no Agent Platform

Treina modelos personalizados com pouco código usando dados rotulados próprios, mesmo com pouca experiência em ML. É a escolha quando as APIs pré-treinadas não reconhecem os produtos específicos da empresa.

### BigQuery ML

Cria e usa modelos de ML com SQL padrão sobre os dados que já estão no BigQuery, sem mover os dados e sem exigir Python. Ideal para analistas que dominam SQL.

### Modelo personalizado

Treinamento próprio no Agent Platform: máximo de controle e diferenciação, com maior esforço e especialização. Sem dados suficientes nem equipe, é alto esforço com resultado incerto.

### Gemini Enterprise Agent Platform

Antes chamado de Vertex AI: plataforma completa de IA para desenvolvedores, com mais de 200 modelos no Model Garden (Gemini, de terceiros e abertos), ajuste, avaliação, implantação, monitoramento e governança de modelos e agentes.

### Agent Studio, ADK e Agent Runtime

Agent Studio (antes Vertex AI Studio) é a interface visual para testar prompts, comparar respostas e ajustar o Gemini com dados próprios. ADK (Agent Development Kit) é o framework aberto para criar agentes em código. Agent Runtime (antes Vertex AI Agent Engine) executa os agentes em ambiente gerenciado.

### Gemini Enterprise

Plataforma agêntica para os funcionários: busca corporativa em apps como SharePoint e Jira, assistente de IA, agentes prontos do Google e de parceiros e criação de agentes sem código, com governança central.

### AI Hypercomputer e Cloud TPU

O AI Hypercomputer integra hardware otimizado (GPUs NVIDIA e Cloud TPUs), software aberto (PyTorch, JAX, TensorFlow) e consumo flexível (Spot VMs, reservas e Dynamic Workload Scheduler). A Cloud TPU é o acelerador projetado pelo próprio Google para treinar modelos muito grandes.

### Benefícios das ofertas de IA do Google Cloud

Infraestrutura de IA de ponta, nuvem de dados pronta para IA, modelos próprios sofisticados, plataforma completa para desenvolvedores e agentes e aplicações de IA prontos, que geram valor em semanas sem equipe de ML.

### Qualidade dos dados

Dados ruins geram previsões ruins, e o modelo não corrige erros de origem nem a quantidade substitui a qualidade. Dimensões: completude, unicidade, atualidade, validade, exatidão e consistência; registro duplicado viola a unicidade.

### IA responsável

Justiça, transparência e explicabilidade, privacidade e responsabilidade. Dados que representam só parte dos clientes geram viés e resultados injustos; explicar por que um crédito foi negado exige IA explicável.

### Escolha da solução de IA

Considerações estratégicas: velocidade de implementação, esforço de desenvolvimento, potencial de diferenciação, experiência técnica exigida e liberdade de escolha. Caminho comum: começar com APIs e evoluir para algo personalizado.

## Macetes para memorizar

### 💡 Escada de esforço da IA

API pronta (rápido) → AutoML (meus dados, pouco código) → modelo próprio (diferenciação máxima).

### 💡 Speech x Text

Leia da esquerda para a direita: Speech-to-Text vira texto; Text-to-Speech vira fala.

**Nomes atuais**

Vertex AI virou Gemini Enterprise Agent Platform; Vertex AI Studio virou Agent Studio; Agent Engine virou Agent Runtime.

**Gemini Enterprise x Agent Platform**

Enterprise = equipe toda usa; Agent Platform = desenvolvedor constrói.

**Qualidade de dados**

CUAVEC: Completude, Unicidade, Atualidade, Validade, Exatidão, Consistência.

**IA x ML x BI**

BI olha o retrovisor; ML olha o para-brisa; generativa cria; agêntica dirige.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **API pré-treinada**<br>Caso comum e genérico com necessidade de resultado rápido, sem treinar nada. | **AutoML**<br>Dados rotulados próprios (produtos específicos) e pouca experiência em ML. |
| **AutoML**<br>Personalização com baixo esforço e pouco código. | **Modelo personalizado**<br>Diferenciação máxima e controle total, com dados exclusivos e equipe experiente. |
| **BigQuery ML**<br>Analistas com SQL criando modelos sem mover os dados do warehouse. | **Agent Platform**<br>Ciclo completo de ML e agentes: escolher, ajustar, implantar e governar modelos. |
| **Gemini Enterprise**<br>Funcionários de várias áreas usando agentes prontos, busca corporativa e criação sem código. | **Gemini Enterprise Agent Platform**<br>Desenvolvedores comparando modelos, ajustando e criando agentes em código. |
| **Speech-to-Text**<br>Transformar gravações de ligações em texto. | **Text-to-Speech**<br>Ler textos em voz alta, como em app de acessibilidade. |
| **Vision API**<br>Imagens: placas, logotipos, texto em fotos. | **Natural Language API**<br>Texto: sentimento de avaliações, entidades e sintaxe. |
| **BI**<br>Explicar o passado com painéis e relatórios. | **Machine learning**<br>Aprender padrões para prever e automatizar decisões. |

## Números que caem

| Item | Valor |
|---|---|
| Modelos no Model Garden do Agent Platform | **mais de 200** |

| | |
|---|---|
| Dimensões de qualidade dos dados | **6 (completude, unicidade, atualidade, validade, exatidão, consistência)** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Achar que IA é um subconjunto do ML.
**Como não cair:** É o contrário: ML está dentro da IA.

⚠ **Alerta de prova:** Usar a API genérica mesmo quando ela não reconhece os produtos específicos.
**Como não cair:** Produtos específicos com imagens próprias e sem especialistas pedem AutoML.

⚠ **Alerta de prova:** Escolher AutoML para transcrever áudio com o menor esforço.
**Como não cair:** AutoML exige coletar e rotular dados; caso comum e rápido é API pré-treinada.

⚠ **Alerta de prova:** Escolher modelo do zero em TensorFlow para equipe com poucos especialistas.
**Como não cair:** Isso exige muito mais especialização; com dados próprios e pouca experiência, AutoML.

⚠ **Alerta de prova:** Marcar dados únicos ou diferenciação máxima como fatores que favorecem API pré-treinada.
**Como não cair:** Esses fatores favorecem AutoML ou modelo próprio; API pré-treinada é para rapidez em caso comum.

⚠ **Alerta de prova:** Confundir Speech-to-Text com Text-to-Speech.
**Como não cair:** Leia na ordem: Speech-to-Text transforma fala em texto; Text-to-Speech transforma texto em fala.

⚠ **Alerta de prova:** Usar a Cloud Translation API para resumir contratos ou ajustar modelos.
**Como não cair:** Translation só traduz; resumir, responder e redigir é Gemini no Agent Platform; ajustar é Agent Studio.

⚠ **Alerta de prova:** Treinar o modelo mesmo com dados incompletos e desatualizados, ou comprar mais processadores.
**Como não cair:** A prioridade é melhorar a qualidade dos dados antes de treinar.

⚠ **Alerta de prova:** Marcar volume do conjunto como a dimensão violada por registros duplicados.
**Como não cair:** Volume não é dimensão de qualidade; duplicidade viola a unicidade.

⚠ **Alerta de prova:** Achar que IA responsável dispensa revisão humana ou que ML garante 100% de acerto.
**Como não cair:** Modelos erram e precisam de monitoramento contínuo; IA responsável mantém supervisão humana.

⚠ **Alerta de prova:** Escolher BigQuery ML para dar agentes prontos aos funcionários.
**Como não cair:** BigQuery ML cria modelos com SQL; agentes prontos e busca corporativa para funcionários é Gemini Enterprise.

⚠ **Alerta de prova:** Marcar disco persistente, Cloud NAT ou DNS como recurso do AI Hypercomputer.
**Como não cair:** O acelerador do Google para IA é a Cloud TPU, ao lado das GPUs NVIDIA.

⚠ **Alerta de prova:** Na IA agêntica no atendimento, escolher árvore de menus fixa ou relatórios de reclamações.
**Como não cair:** Menus fixos são automação tradicional e relatórios olham o passado; agêntica resolve de ponta a ponta com humano nas exceções.

## Como cai na prova

📝 **Como cai:** Cenário dá prazo, dados e equipe e pede o caminho de IA.
**Como resolver:** Caso comum e pressa leva a API; dados próprios sem especialistas leva a AutoML; SQL e dados no BigQuery leva a BigQuery ML; diferenciação com equipe e dados exclusivos leva a modelo personalizado; valor em semanas sem equipe leva a agentes prontos.

📝 **Como cai:** Cenário de tarefa específica pede a API pré-treinada.
**Como resolver:** Identifique a entrada e a saída: imagem (Vision), texto para sentimento (Natural Language), idioma (Translation), áudio para texto (Speech-to-Text), texto para áudio (Text-to-Speech).

📝 **Como cai:** Pergunta sobre a plataforma ou produto com nome novo.
**Como resolver:** Desenvolvedor e ciclo de ML e agentes: Agent Platform; interface visual: Agent Studio; código: ADK; execução: Agent Runtime; funcionários: Gemini Enterprise; infraestrutura: AI Hypercomputer e TPU.

📝 **Como cai:** Problema de dados ou de ética no modelo.
**Como resolver:** Dados ruins: melhorar qualidade antes de treinar e nomear a dimensão; dados parciais: viés; decisão a explicar: IA explicável.

📝 **Como cai:** Múltipla resposta sobre fatores de escolha ou práticas responsáveis.
**Como resolver:** Fatores: prazo e esforço, especialização da equipe. Práticas: explicar decisões e avaliar vieses. Descarte esconder o uso de IA, usar dados sem consentimento e dispensar revisão humana.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| analistas que sabem SQL, dados já no BigQuery, sem Python | **BigQuery ML** |
| imagens próprias rotuladas, pouca experiência em ML | **AutoML no Agent Platform** |
| resultado rápido, caso comum, sem treinar | **API pré-treinada** |
| extrair nome e CPF de formulários digitalizados | **Document AI** |
| placas, logotipos, texto em fotos | **Cloud Vision API** |
| sentimento de avaliações escritas | **Natural Language API** |
| comparar Gemini com modelos de terceiros e abertos, ajustar e implantar agentes | **Gemini Enterprise Agent Platform** |
| testar prompts e ajustar o Gemini em interface visual | **Agent Studio** |
| agentes em código com framework aberto | **ADK no Agent Platform** |
| funcionários, SharePoint e Jira, agentes sem programar | **Gemini Enterprise** |
| GPUs, TPUs, software aberto e consumo flexível | **AI Hypercomputer** |
| acelerador projetado pelo Google para modelos muito grandes | **Cloud TPU** |
| explicar por que o crédito foi negado | **IA explicável e responsável** |
| transações anormais sem regras manuais | **Detecção de anomalias** |

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

- ML é subconjunto da IA; agêntica é subconjunto da generativa que planeja e age.
- Gemini Enterprise Agent Platform é o nome atual do Vertex AI.
- Qualidade de dados vem antes de treinar qualquer modelo.
- Mais personalização = mais diferenciação e mais esforço.

## Essenciais para a prova (checklist)

- ☐ Sei diferenciar IA, ML, BI, IA generativa e IA agêntica.
- ☐ Sei escolher entre API pré-treinada, AutoML, BigQuery ML e modelo personalizado.
- ☐ Sei qual API pré-treinada resolve cada tarefa de imagem, texto, idioma e áudio.
- ☐ Sei o papel do Agent Platform, Agent Studio, ADK e Agent Runtime.
- ☐ Sei diferenciar Gemini Enterprise de Gemini Enterprise Agent Platform.
- ☐ Sei o que compõe o AI Hypercomputer e o que é a Cloud TPU.
- ☐ Sei as seis dimensões de qualidade dos dados.
- ☐ Sei as práticas de IA responsável e o risco de viés.
- ☐ Sei os tipos de problema que o ML resolve: previsão, classificação, recomendação, anomalias.

## Questões de revisão do capítulo

R4.1 · Inovando com a Inteligência Artificial do Google Cloud

**Uma equipe de marketing quer criar automaticamente textos e imagens inéditos para campanhas a partir de instruções em linguagem natural. Que tipo de IA faz isso?**

A. Regras de negócio fixas
B. Análise descritiva
C. IA generativa
D. Business intelligence

R4.2 · Inovando com a Inteligência Artificial do Google Cloud

**Uma empresa precisa de um modelo de visão em poucos dias e não tem especialistas. As APIs pré-treinadas não reconhecem seus produtos específicos. Qual é o melhor caminho?**

A. Usar o AutoML com imagens próprias
B. Desistir do uso de IA no projeto
C. Criar um modelo do zero sem equipe
D. Usar só a API genérica mesmo assim

R4.3 · Inovando com a Inteligência Artificial do Google Cloud

**Uma empresa quer usar IA para recomendar produtos aos clientes com base no histórico de compras. Qual é o valor principal para o negócio?**

A. Substituição da equipe de vendas humana
B. Proteção contra ataques DDoS
C. Redução do número de produtos
D. Personalização que aumenta vendas

R4.4 · Inovando com a Inteligência Artificial do Google Cloud

**Por que a qualidade dos dados é tão importante para um modelo de machine learning?**

A. A qualidade só importa em relatórios
B. A quantidade substitui a qualidade
C. O modelo corrige os dados sozinho
D. Dados ruins geram previsões ruins

## Respostas comentadas

### R4.1. C) IA generativa

Por quê: IA é o campo amplo; ML aprende padrões com dados; IA generativa cria conteúdo novo; análise de dados e BI descrevem e visualizam o que aconteceu.

✗ A) Incorreta. Regras fixas não criam conteúdo novo.
✗ B) Incorreta. Descreve o que já aconteceu nos dados.
✗ D) Incorreta. BI analisa e visualiza dados históricos.

> **Dica de prova:** Criar conteúdo novo a partir de prompts: IA generativa.

### R4.2. A) Usar o AutoML com imagens próprias

Por quê: O AutoML no Agent Platform equilibra personalização e baixo esforço quando as APIs pré-treinadas não bastam.

✗ B) Incorreta. Perde a oportunidade de usar IA.
✗ C) Incorreta. Exige especialistas que a empresa não tem.
✗ D) Incorreta. A API genérica não reconhece os produtos específicos.

> **Dica de prova:** Produtos específicos e sem especialistas: AutoML.

### R4.3. D) Personalização que aumenta vendas

Por quê: Recomendações personalizadas são um uso comum de ML no varejo.

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

> **Dica de prova:** Recomendar produtos: personalização.

### R4.4. D) Dados ruins geram previsões ruins

Por quê: Modelos aprendem padrões dos dados; dados incompletos, duplicados, desatualizados, inválidos, imprecisos ou inconsistentes geram resultados ruins.

✗ A) Incorreta. Afeta diretamente os modelos.
✗ B) Incorreta. Dados ruins em volume continuam ruins.
✗ C) Incorreta. Não corrige erros de origem.

> **Dica de prova:** Qualidade de dados: base de qualquer modelo de IA.

05

CAPÍTULO 5 · 18% DA PROVA

# Modernizar a Infraestrutura e as Aplicações com o Google Cloud

Neste domínio eu preciso escolher a opção de computação certa para cada cenário (Compute Engine, GKE, Cloud Run, Cloud Run functions), entender contêineres, Kubernetes e microsserviços, e dominar os seis termos de migração: retire, retain, rehost, replatform, refactor e reimagine. A prova também cobra APIs e o Apigee (cotas, monetização, governança) e as soluções híbridas e multinuvem, como GKE com gestão de frotas e AlloyDB Omni. As questões são de cenário: o enunciado diz quanto controle a empresa quer, se tolera interrupções, se quer pagar zero sem uso ou se vai reescrever o código. Onde mais erro: confundir rehost com replatform, refactor com reimagine e Cloud Run com Cloud Run functions.

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

## Tópicos-chave

### Compute Engine

Máquinas virtuais na infraestrutura do Google, com tipos de máquina sob medida, Spot VMs e migração em tempo real, que mantém as VMs ativas durante a manutenção do hardware. Indicado quando a aplicação exige SO específico e controle total, ao custo de gerenciar SO e capacidade.

### Spot VMs

Versão mais recente das VMs preemptivas: grande desconto, mas o Compute Engine pode interrompê-las. Servem para processamento em lote tolerante a interrupções com o menor custo.

### Bare Metal Solution

Servidores físicos dedicados para cargas especializadas, como Oracle. Aparece como distrator porque exige gerenciar hardware.

### Grupo gerenciado de instâncias + balanceamento + autoscaling

Combinação para rodar aplicações web em VMs com escala e resiliência: o escalonamento automático aumenta ou reduz o número de VMs conforme a carga e o balanceador distribui o tráfego entre instâncias. Uma VM única sem balanceador é ponto único de falha.

### Contêineres

Empacotam a aplicação e suas dependências; compartilham o kernel do host, são leves, iniciam mais rápido que VMs e são portáveis. Imagens padronizadas garantem ambientes iguais entre desenvolvimento e produção.

### Kubernetes

Padrão aberto de orquestração de contêineres, criado pelo Google: reinicia contêineres que falham, distribui a carga e faz atualizações graduais. Não confundir com Terraform (infraestrutura como código), PostgreSQL ou TensorFlow.

### GKE e modo Autopilot

Google Kubernetes Engine é o Kubernetes gerenciado, com escalonamento horizontal de pods e de nós. No Autopilot, o Google gerencia nós e capacidade do cluster; o GKE Standard exige gerenciar nós.

### GKE com gestão de frotas (GKE Enterprise)

Gerencia clusters no datacenter, no Google Cloud e em outras nuvens como uma frota, com ferramentas e políticas consistentes. O guia do exame chama esses recursos de GKE Enterprise (antes Anthos); hoje frotas, GKE Multi-Cloud e clusters anexados fazem parte do próprio GKE.

### Cloud Run

Serverless para contêineres: escala automática inclusive até zero e cobrança por uso. Ideal para quem não quer gerenciar servidores nem clusters e quer pagar zero quando não há acesso.

### Cloud Run functions

Antes chamado de Cloud Functions: executa código pequeno disparado por eventos, como um arquivo enviado ao Cloud Storage ou uma mensagem no Pub/Sub, também via Eventarc. Um cluster GKE dedicado seria mais complexo do que a tarefa exige.

### Computação sem servidor

Cobra só quando o código executa e escala automaticamente; reduz custo de cargas intermitentes e deixa a equipe focada no código, sem manter clusters nem pagar valor fixo sem uso.

### Microsserviços

Aplicação composta por serviços independentes que podem ser atualizados e escalados separadamente, sem parar o sistema inteiro. Combinam bem com contêineres e Kubernetes; o oposto é o monólito.

### Serviços gerenciados

Serviços como Cloud SQL, Cloud Run e BigQuery transferem ao Google patches, backups e ajuste de capacidade, reduzindo a carga operacional e acelerando a entrega de valor.

### API e Apigee

API é a interface para sistemas conversarem. O Apigee é a plataforma de gestão de APIs do Google Cloud: segurança, análise, cotas, spike arrest, autenticação, monetização e governança central do ciclo de vida das APIs, internas e externas.

### Termos de migração

Retire desativa o que não se usa; retain mantém no local (por exemplo, por regulação); rehost (lift and shift) move sem alterar; replatform (move and improve) migra com otimizações como trocar o banco por um gerenciado; refactor reescreve, por exemplo para microsserviços; reimagine repensa o processo com recursos nativos de nuvem e agentes de IA.

### Carga de trabalho e descoberta

Carga de trabalho (workload) é a unidade que se avalia e migra: a aplicação, seus dados e a infraestrutura que a suporta. A migração começa por descoberta e avaliação (inventário, dependências, custo estimado); só depois se escolhe o termo de migração.

### Migrate to Virtual Machines

Migra VMs, por exemplo do VMware local, para o Compute Engine. Apoia o rehost: rápido e com poucas alterações, mas aproveitando menos os recursos nativos da nuvem.

### Produtos híbridos e multinuvem

AlloyDB Omni (o mesmo motor PostgreSQL de alto desempenho fora do Google Cloud, inclusive no datacenter), BigQuery Omni, recursos multinuvem do GKE, Cloud SQL e Looker (camada única de métricas sobre bancos em várias nuvens, como Amazon Redshift e Azure SQL).

## Macetes para memorizar

### 💡 Seis termos de migração

Os 6 R, do desligar ao reinventar: Retire (desliga), Retain (fica), Rehost (move igual), Replatform (move e melhora), Refactor (reescreve), Reimagine (repensa). Refactor e reimagine trazem mais benefício com mais esforço.

### 💡 Rehost x replatform

Lift and shift = levanta e solta; move and improve = move e dá um tapa (banco gerenciado).

**Opções de computação**

VM = controle (Compute Engine); K8s = orquestra (GKE); contêiner sem servidor = Cloud Run; evento = functions.

**Autopilot**

Autopilot = piloto automático: o Google dirige os nós.

**Apigee**

Apigee = porteiro e caixa das APIs: controla quem entra (segurança/cotas) e cobra (monetização).

**Omni**

Omni = em qualquer lugar: AlloyDB Omni e BigQuery Omni rodam fora do Google Cloud.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **Rehost (lift and shift)**<br>Mover para VMs sem alterar o código, para sair rápido do datacenter. | **Replatform (move and improve)**<br>Migrar com otimizações pontuais, como trocar o banco local por um gerenciado, sem reescrever a aplicação. |
| **Refactor**<br>Reescrever a aplicação existente, por exemplo de monólito para microsserviços. | **Reimagine**<br>Repensar o processo e criar uma experiência nova com serviços nativos de nuvem e agentes de IA. |
| **Retain**<br>A aplicação fica no datacenter, por exemplo por questão regulatória. | **Retire**<br>A aplicação ninguém usa e é desligada. |
| **Cloud Run**<br>Aplicação inteira em contêiner, serverless, escalando até zero. | **Cloud Run functions**<br>Código pequeno disparado por evento (arquivo no Cloud Storage, mensagem no Pub/Sub). |
| **GKE Autopilot**<br>Usar Kubernetes sem gerenciar nós nem capacidade do cluster. | **GKE Standard**<br>A equipe quer e aceita gerenciar os nós. |
| **Compute Engine**<br>SO específico e controle total da máquina. | **Cloud Run**<br>Contêiner sem servidor, pagando só quando há requisições. |
| **Máquina virtual**<br>Cada VM tem SO completo próprio e virtualiza o hardware. | **Contêiner**<br>Compartilha o kernel do host: mais leve, inicia mais rápido e é portável. |

## Números que caem

| Item | Valor |
|---|---|
| Termos de migração do guia | **6 (retire, retain, rehost, replatform, refactor, reimagine)** |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Marcar rehost quando a empresa troca o banco local por um gerenciado.
**Como não cair:** Rehost não muda nada. Qualquer otimização sem reescrever, como banco gerenciado, é replatform (move and improve).

⚠ **Alerta de prova:** Marcar refactor quando a clínica repensa o processo e cria uma experiência nova com agentes de IA.
**Como não cair:** Repensar a solução do zero com recursos nativos é reimagine; refactor reescreve a aplicação existente.

⚠ **Alerta de prova:** Confundir retain com retire.
**Como não cair:** Retain = manter onde está; retire = desligar. Desligar o que não se usa reduz custo e risco.

⚠ **Alerta de prova:** Achar que refatorar antes de mover é o caminho mais seguro na modernização gradual.
**Como não cair:** A sequência gradual é rehost primeiro e refatorar depois; refatorar antes é mais lento e arriscado.

⚠ **Alerta de prova:** Escolher rehost, refactor ou retire como a primeira etapa da migração.
**Como não cair:** Todos esses termos são decisões tomadas depois da descoberta e avaliação das cargas de trabalho.

⚠ **Alerta de prova:** Escolher Compute Engine para uma startup que quer contêineres sem gerenciar servidores.
**Como não cair:** Compute Engine exigiria gerenciar VMs; contêiner sem servidor e pagando por requisição é Cloud Run.

⚠ **Alerta de prova:** Usar um cluster GKE dedicado ou VM sempre ligada para reagir a um upload no Cloud Storage.
**Como não cair:** Código pequeno por evento, sem servidor, é Cloud Run functions; a VM paga mesmo ociosa.

⚠ **Alerta de prova:** Escolher GKE Standard para quem não quer gerenciar nós.
**Como não cair:** Standard exige gerenciar nós; o Autopilot entrega o cluster gerenciado pelo Google.

⚠ **Alerta de prova:** Achar que contêineres têm SO completo e iniciam mais devagar que VMs.
**Como não cair:** É o contrário: SO completo é da VM; contêineres compartilham o kernel, são leves e iniciam rápido.

⚠ **Alerta de prova:** Escolher VMs sob demanda reservadas ou GPUs para lote barato tolerante a interrupções.
**Como não cair:** O menor custo para carga interrompível é Spot VMs.

⚠ **Alerta de prova:** Marcar Terraform como a tecnologia que orquestra contêineres.
**Como não cair:** Terraform automatiza infraestrutura como código; a orquestração de contêineres é do Kubernetes, gerenciado no GKE.

⚠ **Alerta de prova:** Expor serviços internos abrindo o banco na Internet.
**Como não cair:** Exponha com APIs gerenciadas (Apigee), que dão segurança, controle de versões e governança.

⚠ **Alerta de prova:** Para PostgreSQL no próprio datacenter, escolher Bigtable ou Firestore.
**Como não cair:** Bigtable e Firestore são NoSQL gerenciados no Google Cloud; o PostgreSQL de alto desempenho fora do Google Cloud é o AlloyDB Omni.

⚠ **Alerta de prova:** Achar que multinuvem reduz automaticamente toda a complexidade ou é obrigação técnica.
**Como não cair:** Multinuvem traz flexibilidade e menos dependência, mas pode aumentar a complexidade e não é obrigatória.

## Como cai na prova

📝 **Como cai:** Cenário de migração descreve o que muda na aplicação e pede o termo.
**Como resolver:** Pergunte: desliga (retire)? fica (retain)? move igual (rehost)? troca peça como o banco (replatform)? reescreve (refactor)? recria o processo (reimagine)?

📝 **Como cai:** Cenário pede a opção de computação considerando controle, custo e gestão.
**Como resolver:** Controle do SO leva a Compute Engine; tolera interrupção e quer barato leva a Spot VMs; contêiner sem servidor leva a Cloud Run; evento leva a Cloud Run functions; Kubernetes sem nós leva a Autopilot.

📝 **Como cai:** Questão de múltipla resposta sobre benefícios de contêineres.
**Como resolver:** Marque portabilidade e implantação rápida e consistente; descarte dependência de hardware, inicialização lenta e um SO por aplicação (isso é VM).

📝 **Como cai:** Cenário de APIs para parceiros com abuso, cobrança ou controle.
**Como resolver:** Identifique Apigee e o recurso: abuso leva a cotas e limites de taxa; cobrar leva a monetização; várias APIs leva a governança central.

📝 **Como cai:** Cenário híbrido ou multinuvem pede o produto.
**Como resolver:** Contêineres em vários ambientes leva a GKE com frotas; PostgreSQL local leva a AlloyDB Omni; BI sobre bancos em várias nuvens leva a Looker; análise de dados em outra nuvem leva a BigQuery Omni.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| VMs com tamanho sob medida, SO específico, controle total da máquina | **Compute Engine** |
| lote tolerante a interrupções com menor custo | **Spot VMs** |
| contêineres sem gerenciar servidores, pagar zero sem acesso, escala até zero | **Cloud Run** |
| código pequeno quando um arquivo chega ao Cloud Storage ou mensagem no Pub/Sub | **Cloud Run functions** |
| Kubernetes gerenciado sem montar o cluster do zero | **GKE** |
| Kubernetes sem gerenciar nós nem capacidade | **GKE Autopilot** |
| contêineres no datacenter e em várias nuvens com políticas centralizadas | **GKE com gestão de frotas (GKE Enterprise)** |
| mover sem alterar o código, sair rápido do datacenter | **Rehost (lift and shift)** |
| trocar o banco local por gerenciado sem reescrever | **Replatform (move and improve)** |
| reescrever o monólito em microsserviços | **Refactor** |
| levantar servidores, dependências e custo antes de migrar | **Descoberta e avaliação** |
| cotas, limites de taxa, monetização de API, parceiros | **Apigee** |
| VMs com escala e tráfego distribuído | **Grupo gerenciado de instâncias + balanceador de carga + autoscaling** |

| | |
|---|---|
| PostgreSQL no próprio datacenter com o motor do Google Cloud | **AlloyDB Omni** |

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

- Replatform = move and improve; rehost = lift and shift.
- Cloud Run escala até zero; VM e cluster fixo cobram mesmo sem uso.
- A migração começa pela descoberta e avaliação das cargas de trabalho.
- Nomes atuais: Cloud Run functions (antes Cloud Functions); recursos de frota no GKE (no guia, GKE Enterprise, antes Anthos).

## Essenciais para a prova (checklist)

- ☐ Sei escolher entre Compute Engine, GKE, Cloud Run e Cloud Run functions por cenário.
- ☐ Sei quando usar Spot VMs e o que é migração em tempo real.
- ☐ Sei diferenciar VM de contêiner e explicar o papel do Kubernetes.
- ☐ Sei diferenciar GKE Autopilot de GKE Standard.
- ☐ Sei aplicar os seis termos de migração a cenários reais.
- ☐ Sei que a descoberta e avaliação vem antes de qualquer decisão de migração.
- ☐ Sei o que o Apigee oferece: segurança, cotas, análise, monetização e governança.
- ☐ Sei citar os produtos híbridos e multinuvem: GKE com frotas, AlloyDB Omni, BigQuery Omni, Looker.
- ☐ Sei explicar por que serverless reduz custo de cargas intermitentes.

## Questões de revisão do capítulo

R5.1 · Modernizar a Infraestrutura e as Aplicações com o Google Cloud

**Uma empresa quer controlar o acesso, medir o uso e aplicar limites de chamadas às suas APIs. Qual é o benefício de uma plataforma de gestão de APIs?**

A. Treinamento de modelos de visão
B. Armazenamento de vídeos grandes
C. Backup de bancos relacionais
D. Segurança, análise e controle de uso

R5.2 · Modernizar a Infraestrutura e as Aplicações com o Google Cloud

**Uma empresa quer rodar código em resposta a mensagens publicadas no Pub/Sub, sem gerenciar servidores. Qual opção é a mais adequada?**

A. VM dedicada sempre ligada
B. Planilha com atualização manual
C. Função sem servidor por evento
D. Servidor físico no escritório

R5.3 · Modernizar a Infraestrutura e as Aplicações com o Google Cloud

**O que é uma API?**

A. Uma interface para sistemas conversarem
B. Um tipo de disco de armazenamento
C. Uma classe de máquina virtual
D. Uma política de faturamento

R5.4 · Modernizar a Infraestrutura e as Aplicações com o Google Cloud

**Uma empresa decide reescrever uma aplicação monolítica em microsserviços para aproveitar melhor a nuvem. Qual estratégia de migração é essa?**

A. Estratégia refactor (refatorar)
B. Estratégia retire (desativar)
C. Estratégia rehost (mover como está)
D. Estratégia retain (manter no local)

## Respostas comentadas

### R5.1. D) Segurança, análise e controle de uso

Por quê: Apigee oferece segurança, análise, cotas e monetização de APIs.

✗ A) Incorreta. Não é sua função.
✗ B) Incorreta. Não é sua função.
✗ C) Incorreta. Não é sua função.

> **Dica de prova:** Gestão de APIs: segurança e controle de uso.

### R5.2. C) Função sem servidor por evento

Por quê: Cloud Run functions e serviços do Cloud Run podem ser acionados por mensagens do Pub/Sub e por outros eventos via Eventarc, sem servidores para gerenciar.

✗ A) Incorreta. Exige gestão e custo contínuo.
✗ B) Incorreta. Não reage a eventos.
✗ D) Incorreta. Exige gestão e hardware.

> **Dica de prova:** Código por mensagem: função por evento.

### R5.3. A) Uma interface para sistemas conversarem

Por quê: APIs permitem integração entre sistemas e novos modelos de negócio.

✗ B) Incorreta. Não é o conceito.
✗ C) Incorreta. Não é o conceito.
✗ D) Incorreta. Não é o conceito.

> **Dica de prova:** API: sistemas conversando.

### R5.4. A) Estratégia refactor (refatorar)

Por quê: Refatorar exige mais esforço, mas aproveita melhor os recursos da nuvem.

✗ B) Incorreta. Desliga a aplicação.
✗ C) Incorreta. Não muda o código.
✗ D) Incorreta. Não migra.

> **Dica de prova:** Reescrever para microsserviços: refactor.

06

CAPÍTULO 6 · 18% DA PROVA

# Transformação Digital com o Google Cloud

Aqui eu preciso dominar o vocabulário de base da nuvem: modelos de serviço (IaaS, PaaS, SaaS), modelos de implantação (pública, privada, híbrida, multinuvem), responsabilidade compartilhada e os benefícios de negócio (agilidade, escalabilidade, elasticidade, confiabilidade, flexibilidade). A prova traz cenários curtos de empresa e pede o conceito que explica a situação, quase sempre com distratores absurdos e um ou dois muito parecidos com a certa. Também cai a infraestrutura global do Google (regiões, zonas, borda, cabos submarinos), noções de rede (latência, largura de banda, IP, DNS) e os diferenciais do Google Cloud, como abertura e IA de ponta. Onde mais erro: confundir híbrida com multinuvem, escalabilidade com elasticidade e o que fica com o cliente em cada modelo de serviço.

**Neste capítulo**

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

## Tópicos-chave

### Computação em nuvem

Acesso sob demanda a recursos de TI pela rede, pagando pelo uso. Servidores comprados e instalados no escritório são infraestrutura local, não nuvem.

### Transformação digital

Mudar como a empresa gera valor, repensando processos, produtos e cultura com tecnologia. Digitalizar papel ou comprar computadores novos é parte pequena ou nem faz parte do conceito.

### Motores e obstáculos da transformação

Motores: mudança nas expectativas dos clientes, concorrência e novas tecnologias. Obstáculos: sistemas legados e dívida técnica, falta de habilidades, cultura resistente, orçamento e preocupações com segurança. Não se transformar traz o risco de perder relevância, competitividade e clientes.

### Dívida técnica de sistemas legados

Sistemas antigos muito acoplados, mal documentados e caros de manter. É o obstáculo típico quando o enunciado descreve um legado que trava a modernização.

### IaaS (infraestrutura como serviço)

O provedor cuida do hardware; o cliente gerencia o sistema operacional, as aplicações e os dados. É o modelo com mais controle e mais responsabilidade de gestão, como as máquinas virtuais.

### PaaS (plataforma como serviço)

O provedor cuida de servidores e sistema operacional; o cliente foca em desenvolver e publicar o código. O tradeoff em relação ao IaaS é menos controle com menos gestão.

### SaaS (software como serviço)

Aplicação pronta, gerenciada pelo provedor; o cliente só usa e configura usuários. É o modelo que exige menos equipe de TI, como e-mail ou CRM na nuvem.

### Responsabilidade compartilhada

A infraestrutura física e a segurança dos datacenters são sempre do provedor. Dados, identidades e acessos são sempre do cliente, em qualquer modelo, inclusive SaaS. Quanto mais gerenciado o serviço, menos o cliente administra.

### Nuvem pública x privada

Pública compartilha a infraestrutura do provedor; privada é dedicada a uma organização. A privada é escolhida para atender exigências de controle, costuma custar mais e exige gestão.

### Nuvem híbrida

Integra a infraestrutura própria (datacenter local) com a nuvem pública. Aparece em cenários com sistemas sensíveis no local, exigência de latência em fábricas ou migração gradual.

### Multinuvem

Uso de mais de um provedor de nuvem pública para reduzir a dependência e escolher o melhor serviço de cada um. Um único provedor nunca é multinuvem.

### Escalabilidade e elasticidade

Escalabilidade é crescer a capacidade conforme a necessidade, sem planejar com anos de antecedência. Elasticidade é crescer e reduzir automaticamente acompanhando a demanda, como no pico de algumas horas ou na Black Friday, pagando só pelo uso.

### Agilidade

Criar ambientes rapidamente para reduzir o lançamento de novos serviços de meses para semanas e testar ideias sem grande investimento inicial. Instalação física de equipamentos, compra antecipada de servidores e aprovação de hardware em comitês aumentam o prazo.

### Confiabilidade

Manter o sistema disponível mesmo com falhas de servidor, graças a redundância e distribuição geográfica. No e-commerce, é a loja no ar mesmo quando um servidor cai.

### Abertura e interoperabilidade

Diferencial do Google Cloud: código aberto, padrões abertos e suporte a multinuvem para evitar aprisionamento ao fornecedor. Código aberto é código-fonte público e modificável; padrão aberto é especificação pública que garante interoperabilidade. Kubernetes, criado pelo Google, é exemplo de código aberto.

### Diferenciais do Google Cloud

IA de ponta (modelos Gemini), AI Hypercomputer, compromisso com abertura e interoperabilidade, plataforma de dados pronta para IA, segurança e rede global.

### Infraestrutura global do Google

Regiões compostas por várias zonas (áreas isoladas que contêm falhas dentro da região), cabos submarinos que dão conectividade global rápida e locais de borda (pontos de presença) que aproximam o conteúdo dos usuários. Implantar em regiões perto dos usuários reduz a latência sem construir datacenters.

### Latência, largura de banda, IP e DNS

Latência é o atraso, o tempo que os dados levam para chegar. Largura de banda é a capacidade, o volume transferido por segundo. Endereço IP é o identificador numérico na rede; DNS traduz nomes de domínio em endereços IP.

### IA generativa x IA agêntica

A generativa cria conteúdo novo. A agêntica é um subconjunto que orquestra agentes: eles usam modelos como cérebro, planejam e executam ações com ferramentas para atingir metas, com supervisão humana adequada.

## Macetes para memorizar

### Modelos de serviço

IaaS = Infra (eu cuido do SO); PaaS = Programo (só o código); SaaS = Só uso.

### Responsabilidade compartilhada

Físico é do provedor, sempre; dados e acessos são do cliente, sempre. O meio muda com o modelo.

### Híbrida x multinuvem

Híbrida = casa + nuvem; Multi = muitas nuvens públicas.

### Escala

Escalabilidade cresce; elasticidade estica e encolhe sozinha.

### Rede

Latência é demora; banda é largura do cano; DNS é a agenda que troca nome por IP.

### Diferenciais do Google Cloud

IA + Abertura + Dados prontos para IA + Segurança + Rede global.

## Tabela de memorização: não confunda

| Quando é... | ...e quando é |
|---|---|
| **Nuvem híbrida**<br>Datacenter próprio integrado com uma nuvem pública (sistemas sensíveis ou latência local ficam no local). | **Multinuvem**<br>Dois ou mais provedores de nuvem pública, para reduzir dependência ou usar o melhor de cada um. |
| **IaaS**<br>Precisa de controle do sistema operacional e das configurações, aceitando mais gestão. | **PaaS**<br>Quer só desenvolver e publicar o código, sem cuidar de servidores e SO: menos controle, menos gestão. |
| **PaaS**<br>A empresa vai desenvolver a própria aplicação. | **SaaS**<br>A empresa só usa uma aplicação pronta e quer o mínimo de gestão e de equipe de TI. |
| **Escalabilidade**<br>Crescer conforme a necessidade, sem planejar capacidade com anos de antecedência. | **Elasticidade**<br>Aumentar e reduzir automaticamente no pico diário ou sazonal, pagando só pelo uso. |
| **Latência**<br>Tempo de ida e volta, atraso da comunicação. | **Largura de banda**<br>Volume de dados transferido por segundo, capacidade do link. |
| **Nuvem pública**<br>Infraestrutura compartilhada do provedor, com elasticidade e pagamento por uso. | **Nuvem privada**<br>Infraestrutura dedicada a uma organização, para exigências de controle; costuma custar mais. |
| **IA generativa**<br>Criar textos, imagens e outros conteúdos a partir de prompts. | **IA agêntica**<br>Planejar e executar tarefas rumo a um objetivo, agindo por meio de ferramentas. |

## Alertas de prova (pegadinhas)

⚠ **Alerta de prova:** Achar que, em SaaS, a segurança é toda do provedor.
**Como não cair:** Em qualquer modelo, o cliente responde por quem acessa e quais dados entram. Patches do servidor do CRM são do provedor no SaaS, mas acessos e dados não.

⚠ **Alerta de prova:** Marcar nuvem híbrida quando a empresa usa dois provedores públicos.
**Como não cair:** Dois provedores públicos é multinuvem. Híbrida exige o ambiente local integrado à nuvem pública.

⚠ **Alerta de prova:** Marcar multinuvem quando há datacenter próprio + uma nuvem pública.
**Como não cair:** Próprio + nuvem pública integrados é híbrida; multinuvem sem ambiente local usaria só provedores de nuvem.

⚠ **Alerta de prova:** Escolher nuvem privada por ser mais barata ou por ter elasticidade ilimitada.
**Como não cair:** A privada costuma custar mais e exige gestão; elasticidade é benefício da pública. O motivo da privada é controle.

⚠ **Alerta de prova:** Confundir o tempo de ida e volta de um pacote com largura de banda.
**Como não cair:** Tempo é latência; volume por segundo é largura de banda.

⚠ **Alerta de prova:** Marcar nome de domínio como o identificador numérico do servidor.
**Como não cair:** O identificador numérico é o endereço IP; o nome de domínio é a forma legível que o DNS traduz para IP.

⚠ **Alerta de prova:** Atender a um pico sazonal com contrato fixo de capacidade máxima ou datacenter com folga.
**Como não cair:** Isso paga pelo pico o ano inteiro e gera ociosidade. Pico com pagamento por uso é elasticidade.

⚠ **Alerta de prova:** Achar que PaaS ou Função como serviço (FaaS) dão mais controle do SO que IaaS.
**Como não cair:** No PaaS e no FaaS o provedor gerencia o SO: menos controle e menos gestão, e o cliente ainda mantém responsabilidades de segurança. Maior controle com maior responsabilidade é IaaS.

⚠ **Alerta de prova:** Achar que migrar para a nuvem acaba com as responsabilidades de segurança da empresa.
**Como não cair:** A empresa herda a segurança em camadas da infraestrutura do Google, mas continua responsável por identidades, acessos e dados.

⚠ **Alerta de prova:** Tratar o valor da nuvem como apenas redução imediata de custos.
**Como não cair:** O valor estratégico é inovar mais rápido e criar serviços que antes não eram viáveis; custo é parte menor.

⚠ **Alerta de prova:** Na diferença generativa x agêntica, marcar que a agêntica cria conteúdo sem agir.
**Como não cair:** Criar conteúdo sem agir é a generativa. A agêntica planeja e executa ações. BI analisa dados históricos e não é IA generativa.

⚠ **Alerta de prova:** Achar que código aberto é software que roda em um único provedor ou vendido por fabricante exclusivo.
**Como não cair:** Código aberto tem código-fonte público e modificável e favorece a portabilidade; licença que proíbe alterar é software proprietário.

## Como cai na prova

📝 **Como cai:** Cenário de empresa descreve onde estão os sistemas e pergunta qual modelo de implantação é esse.
**Como resolver:** Conte os ambientes: tem datacenter próprio integrado? Híbrida. Só nuvens públicas de mais de um provedor? Multinuvem. Dedicada a uma organização? Privada.

📝 **Como cai:** Pergunta qual responsabilidade continua com o cliente em IaaS, PaaS ou SaaS.
**Como resolver:** Elimine tudo que é físico (datacenter, discos, rede do provedor). Sobram dados, identidades e acessos, e no IaaS também o SO e as aplicações.

📝 **Como cai:** Cenário de negócio pede o benefício da nuvem mais relevante.
**Como resolver:** Ache a palavra-chave: semanas ou testar rápido (agilidade), pico e pagar pelo uso (elasticidade), falha de servidor (confiabilidade), qualquer lugar (flexibilidade), IA nos dados (inteligência).

📝 **Como cai:** Definição de termo de rede ou de infraestrutura global.
**Como resolver:** Separe atraso (latência) de capacidade (banda), número (IP) de nome (DNS), região (local geográfico) de zona (isolamento de falha) e borda (perto do usuário).

📝 **Como cai:** Cenário pede um diferencial do Google Cloud ou o benefício de abertura, com distratores que descrevem o oposto.
**Como resolver:** Descarte aprisionamento, licenças fechadas, contratos sem portabilidade, hardware exclusivo e compra antecipada; fique com IA de ponta, abertura e interoperabilidade, dados prontos para IA, segurança ou rede global.

## Gatilhos do enunciado

| Se o enunciado trouxer... | Pense em |
|---|---|
| dois provedores de nuvem pública, evitar dependência de um só | **Multinuvem** |
| datacenter próprio + nuvem pública integrados, latência em fábricas locais | **Nuvem híbrida** |
| desenvolver e publicar sem se preocupar com servidores e SO | **PaaS** |
| aplicação pronta, só configura usuários, menos equipe de TI | **SaaS** |
| controle total do SO, atualizar o SO e proteger as aplicações | **IaaS** |
| pico de algumas horas ou Black Friday, pagar só pelo que usar | **Elasticidade** |
| não precisa mais planejar capacidade com anos de antecedência | **Escalabilidade sob demanda** |
| lançar em semanas, testar uma ideia rápido sem investimento inicial | **Agilidade** |
| sistema continua no ar mesmo se um servidor falhar | **Confiabilidade** |
| trabalhar de qualquer lugar com as mesmas ferramentas | **Flexibilidade** |
| entregar vídeo com baixa latência a vários continentes | **Pontos de presença na borda da rede** |
| objetivo das zonas dentro de uma região | **Isolar falhas** |
| tecnologias abertas, não ficar preso a um fornecedor | **Abertura e interoperabilidade, código aberto e padrões abertos** |
| sistemas antigos acoplados, mal documentados e caros | **Dívida técnica de sistemas legados** |

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

- Segurança física do datacenter é sempre do Google; dados, identidades e acessos são sempre do cliente.
- Mais controle = mais responsabilidade: IaaS > PaaS > SaaS.
- Híbrida tem ambiente local; multinuvem tem mais de um provedor público.
- O valor da nuvem é estratégico (inovar mais rápido), não só redução de custos.

## Essenciais para a prova (checklist)

- ☐ Sei classificar um cenário em IaaS, PaaS ou SaaS e dizer o que o cliente gerencia em cada um.
- ☐ Sei diferenciar nuvem pública, privada, híbrida e multinuvem.
- ☐ Sei separar escalabilidade, elasticidade, agilidade, confiabilidade e flexibilidade por cenário.
- ☐ Sei explicar regiões, zonas, borda da rede e cabos submarinos.
- ☐ Sei diferenciar latência, largura de banda, endereço IP e DNS.
- ☐ Sei citar motores, obstáculos e riscos de não fazer a transformação digital.
- ☐ Sei listar os diferenciais do Google Cloud, incluindo abertura e interoperabilidade.
- ☐ Sei distinguir código aberto de padrão aberto e IA generativa de IA agêntica.

## Questões de revisão do capítulo

R6.1 · Transformação Digital com o Google Cloud

**Qual cenário é mais adequado para uma estratégia multinuvem?**

A. Rodar tudo em um servidor local
B. Usar o melhor serviço de cada provedor
C. Evitar os serviços de nuvem pública
D. Manter um único provedor

R6.2 · Transformação Digital com o Google Cloud

**Uma empresa usa um CRM em SaaS. Qual responsabilidade de segurança continua sendo dela?**

A. Proteger fisicamente o datacenter
B. Manter o hardware de rede do provedor
C. Controlar quem acessa e quais dados entram
D. Aplicar patches no servidor do CRM

R6.3 · Transformação Digital com o Google Cloud

**Qual benefício do Google Cloud se refere a oferecer tecnologias abertas e evitar que o cliente fique preso a um fornecedor?**

A. Dependência de licenças fechadas
B. Uso exclusivo de hardware próprio
C. Abertura e interoperabilidade
D. Aprisionamento ao fornecedor

R6.4 · Transformação Digital com o Google Cloud

**Ao apresentar o Google Cloud à diretoria, um gerente quer citar um diferencial que o próprio Google destaca em relação a outros provedores. Qual é um deles?**

A. Contratos que impedem a portabilidade
B. Infraestrutura física gerida pelo cliente
C. Plataforma de dados pronta para IA
D. Compra antecipada de todo o hardware

## Respostas comentadas

### R6.1. B) Usar o melhor serviço de cada provedor

Por quê: Multinuvem permite escolher serviços e reduzir dependência.

✗ A) Incorreta. Não usa nuvem.
✗ C) Incorreta. Multinuvem usa mais de uma nuvem pública.
✗ D) Incorreta. Um único provedor não é multinuvem.

> **Dica de prova:** Melhor de cada provedor: multinuvem.

### R6.2. C) Controlar quem acessa e quais dados entram

Por quê: Em qualquer modelo, o cliente responde por dados, identidades e acessos.

✗ A) Incorreta. É do provedor.
✗ B) Incorreta. É do provedor.
✗ D) Incorreta. É do provedor no SaaS.

> **Dica de prova:** SaaS: acessos e dados com o cliente.

### R6.3. C) Abertura e interoperabilidade

Por quê: Entre os diferenciais do Google Cloud está o compromisso com abertura e interoperabilidade: código aberto, padrões abertos e suporte a multinuvem.

✗ A) Incorreta. É o oposto do benefício.
✗ B) Incorreta. Hardware exclusivo aumentaria a dependência.
✗ D) Incorreta. É o oposto do benefício.

> **Dica de prova:** Sem aprisionamento: abertura e interoperabilidade.

**R6.4. C) Plataforma de dados pronta para IA**

Por quê: Diferenciais destacados pelo Google Cloud: IA de ponta, compromisso com abertura e interoperabilidade, AI Hypercomputer, plataforma de dados pronta para IA, segurança e rede global.

✗ A) Incorreta. O Google enfatiza abertura e interoperabilidade, não aprisionamento.

✗ B) Incorreta. Na nuvem, o Google gerencia a infraestrutura física.

✗ D) Incorreta. A nuvem é cobrada pelo uso, sem compra antecipada de hardware.

> **Dica de prova:** Diferenciais do Google Cloud: IA, abertura, dados prontos para IA, segurança e rede global.

DIA DA PROVA

# Estratégia de prova

O Cloud Digital Leader avalia visão de negócio e de produto, não configuração: cada questão é um cenário curto de empresa que pede o conceito ou o produto do Google Cloud que resolve o problema. Muitas alternativas erradas são absurdas, mas a decisão costuma ficar entre dois itens da mesma família, como Nearline x Coldline, SLO x SLA ou rehost x replatform. A prova tem 50 a 60 questões de múltipla escolha e múltipla resposta em 90 minutos, e o Google não divulga a nota de aprovação. Os cinco primeiros domínios pesam cerca de 18% cada e Operações cerca de 10%, então nenhum domínio pode ser deixado de lado. O guia vigente usa os nomes atuais dos produtos, como Gemini Enterprise Agent Platform no lugar de Vertex AI.

## Como responder

✓ Leia a última frase do enunciado primeiro para saber se pedem conceito, produto, benefício ou termo de migração; depois sublinhe mentalmente as pistas (frequência, escala, controle, sem servidor, tempo real).

✓ Em múltipla resposta (Escolha DUAS), marque exatamente o número pedido e avalie cada alternativa como verdadeira ou falsa de forma independente.

✓ Elimine primeiro as alternativas absurdas ou de sentido oposto ao benefício; depois decida entre os dois itens da mesma família pelo detalhe do enunciado.

✓ Com 90 minutos para 50 a 60 questões, responda as diretas logo e deixe para o fim as que ficarem entre duas opções.

✓ Distribua o estudo pelos pesos: cinco domínios de cerca de 18% e Operações com cerca de 10%; não pule nenhum.

✓ Quando dois produtos parecem servir, escolha o mais gerenciado e mais simples que atende ao cenário, a menos que o enunciado peça controle total.

✓ Reconheça os nomes antigos e novos: Vertex AI = Gemini Enterprise Agent Platform; Cloud Functions = Cloud Run functions; Dataproc = Managed Service for Apache Spark; Chronicle = Google Security Operations; Anthos = GKE Enterprise (hoje recursos de frota do próprio GKE); Cloud DLP = Sensitive Data Protection.

✓ Antes de encerrar, volte às questões que deixou para o fim e confira que nenhuma ficou sem resposta.

## Erros que derrubam candidatos

✗ Trocar nuvem híbrida por multinuvem e escalabilidade por elasticidade.

✗ Errar a classe do Cloud Storage por confundir acesso mensal (Nearline) com trimestral (Coldline).

✗ Confundir rehost com replatform e refactor com reimagine.

✗ Inverter SLO e SLA, ou autenticação e autorização.

✗ Achar que, em SaaS ou na nuvem em geral, o cliente deixa de ter responsabilidade sobre dados e acessos.

✗ Escolher a solução de IA mais complexa quando o cenário pede rapidez em caso comum (API pré-treinada).

✗ Travar em nomes de produtos atualizados por ter estudado apenas os nomes antigos.

## Na semana da prova

- ☐ Refazer as provas simuladas e revisar o porquê de cada alternativa errada, não só a certa.
- ☐ Montar uma folha de pares confusos: classes do Cloud Storage, bancos, termos de migração, SLI/SLO/SLA, estados dos dados, APIs de IA.
- ☐ Revisar a tabela de nomes antigos e novos dos produtos.
- ☐ Reler os gatilhos de cada domínio na véspera, em vez de estudar conteúdo novo.
- ☐ Treinar o ritmo fazendo ao menos uma prova cronometrada em 90 minutos.
- ☐ No dia, chegar descansado e deixar para o fim as questões que ficarem entre duas opções.

REVISÃO FINAL

# Folha de cola

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

## 1. Confiança e Segurança com o Google Cloud

‣ quem pode fazer o quê, permissões mínimas → **Cloud IAM com papéis e menor privilégio**

‣ quem alterou a configuração, quem acessou e quando → **Cloud Audit Logs**

‣ controlar e revogar as próprias chaves → **CMEK com Cloud KMS**

‣ cifrado inclusive na memória durante o processamento → **Confidential Computing**

‣ bloquear países, ataques web, milhões de requisições → **Cloud Armor e proteção DDoS da rede global**

⚠ Ver é confidencialidade; não alterar é integridade; acessar quando preciso é disponibilidade.

⚠ Autenticação só comprova identidade; definir o que pode fazer é autorização pelo IAM.

⚠ Registrar ações é auditoria (Cloud Audit Logs).

## 2. Escalando com as Operações do Google Cloud

‣ avisar quando o gasto chegar a 80% do planejado → **Orçamento com alerta de limite**

‣ evitar criar centenas de VMs por erro → **Cotas de recursos**

‣ quanto cada equipe gasta, etiquetas → **Rótulos (labels)**

‣ impedir recursos em regiões não autorizadas → **Políticas da organização**

‣ mesma política para todos os projetos do departamento → **Aplicar na pasta (herança)**

⚠ É o contrário: SLA = contrato; SLO = meta interna, geralmente mais rigorosa.

⚠ Meta é SLO, contrato é SLA; SLI é só a medida.

⚠ Ferramentas sem processos e responsáveis não mudam o comportamento: pessoas, processos e tecnologia.

## 3. Explorando a Transformação de Dados com o Google Cloud

‣ SQL em petabytes sem gerenciar servidores, destino do warehouse local → **BigQuery**

‣ fotos e vídeos de clientes, dados não estruturados → **Cloud Storage**

‣ dados brutos sem esquema para uso futuro, clickstream → **Data lake no Cloud Storage**

‣ acesso imprevisível, não quer escolher a classe → **Autoclass**

‣ retenção longa, auditoria, quase nunca acessado → **Archive**

⚠ Nearline = mensal; Coldline = trimestral; Archive = anual ou menos; Standard = frequente.

⚠ Archive encarece as leituras dos que voltam a ser acessados e Standard fixa mantém custo alto nos pouco usados; acesso imprevisível é Autoclass.

⚠ Réplica aumenta a disponibilidade, mas não ajusta a classe ao uso.

## 4. Inovando com a Inteligência Artificial do Google Cloud

‣ analistas que sabem SQL, dados já no BigQuery, sem Python → **BigQuery ML**

‣ imagens próprias rotuladas, pouca experiência em ML → **AutoML no Agent Platform**

‣ resultado rápido, caso comum, sem treinar → **API pré-treinada**

‣ extrair nome e CPF de formulários digitalizados → **Document AI**

‣ placas, logotipos, texto em fotos → **Cloud Vision API**

⚠ É o contrário: ML está dentro da IA.

⚠ Produtos específicos com imagens próprias e sem especialistas pedem AutoML.

⚠ AutoML exige coletar e rotular dados; caso comum e rápido é API pré-treinada.

## 5. Modernizar a Infraestrutura e as Aplicações com o Google Cloud

‣ VMs com tamanho sob medida, SO específico, controle total da máquina → **Compute Engine**

‣ lote tolerante a interrupções com menor custo → **Spot VMs**

‣ contêineres sem gerenciar servidores, pagar zero sem acesso, escala até zero → **Cloud Run**

‣ código pequeno quando um arquivo chega ao Cloud Storage ou mensagem no Pub/Sub → **Cloud Run functions**

‣ Kubernetes gerenciado sem montar o cluster do zero → **GKE**

⚠ Rehost não muda nada. Qualquer otimização sem reescrever, como banco gerenciado, é replatform (move and improve).

⚠ Repensar a solução do zero com recursos nativos é reimagine; refactor reescreve a aplicação existente.

⚠ Retain = manter onde está; retire = desligar. Desligar o que não se usa reduz custo e risco.

## 6. Transformação Digital com o Google Cloud

‣ dois provedores de nuvem pública, evitar dependência de um só → **Multinuvem**

‣ datacenter próprio + nuvem pública integrados, latência em fábricas locais → **Nuvem híbrida**

‣ desenvolver e publicar sem se preocupar com servidores e SO → **PaaS**

‣ aplicação pronta, só configura usuários, menos equipe de TI → **SaaS**

‣ controle total do SO, atualizar o SO e proteger as aplicações → **IaaS**

⚠ Em qualquer modelo, o cliente responde por quem acessa e quais dados entram. Patches do servidor do CRM são do provedor no SaaS, mas acessos e dados não.

⚠ Dois provedores públicos é multinuvem. Híbrida exige o ambiente local integrado à nuvem pública.

⚠ Próprio + nuvem pública integrados é híbrida; multinuvem sem ambiente local usaria só provedores de nuvem.

GABARITO

# Respostas do diagnóstico

**D1. B) Certificações e relatórios de auditoria** → revise o capítulo 1

Por quê: Certificações como ISO 27001 e relatórios SOC sustentam a confiança.

✗ A) Incorreta. Não são evidência.
✗ C) Incorreta. Não são evidência.
✗ D) Incorreta. Não são evidência.

> **Dica de prova:** Evidência de auditoria: certificações.

**D2. C) Logs de auditoria do Cloud** → revise o capítulo 1

Por quê: Cloud Audit Logs registram atividades administrativas e de acesso a dados.

✗ A) Incorreta. É armazenamento.
✗ B) Incorreta. Mostra custos.
✗ D) Incorreta. Traduz textos.

> **Dica de prova:** Quem alterou a configuração: logs de auditoria.

**D3. D) Garante dados sem alteração indevida** → revise o capítulo 1

Por quê: Hashes, assinaturas e controles de acesso preservam a integridade.

✗ A) Incorreta. Isso é desempenho.
✗ B) Incorreta. Não é segurança.
✗ C) Incorreta. Isso é confidencialidade.

> **Dica de prova:** Sem alteração indevida: integridade.

**D4. D) Políticas da organização** → revise o capítulo 2

Por quê: Políticas da organização aplicam restrições herdadas pela hierarquia.

✗ A) Incorreta. Mostram custos, sem restringir.
✗ B) Incorreta. Não restringe recursos.
✗ C) Incorreta. Não restringem recursos.

> **Dica de prova:** Restringir regiões: políticas da organização.

**D5. D) Objetivo de nível de serviço (SLO)** → revise o capítulo 2

Por quê: SLI mede, SLO define a meta e SLA é o compromisso contratual.

✗ A) Incorreta. Trata de credenciais.
✗ B) Incorreta. Trata de gastos.
✗ C) Incorreta. Trata de uso de software.

> **Dica de prova:** Meta de 99,9%: SLO.

**D6. D) Nó de organização** → revise o capítulo 2

Por quê: Hierarquia: organização, pastas, projetos e recursos.

✗ A) Incorreta. É o nível mais baixo.

✗ B) Incorreta. Fica abaixo das pastas.

✗ C) Incorreta. Fica abaixo da organização; não é o topo.

> **Dica de prova:** Topo da hierarquia: organização.

**D7. B) Autoclass no bucket** → revise o capítulo 3

Por quê: O Autoclass transiciona objetos entre Standard, Nearline, Coldline e Archive conforme o acesso, sem cobrança de recuperação nem de exclusão antecipada.

✗ A) Incorreta. Aumenta a disponibilidade, mas não ajusta a classe ao uso.

✗ C) Incorreta. Encarece as leituras dos arquivos que voltam a ser acessados.

✗ D) Incorreta. Mantém o custo de acesso frequente até para arquivos pouco usados.

> **Dica de prova:** Acesso imprevisível e classe automática: Autoclass.

**D8. B) Cloud Storage** → revise o capítulo 3

Por quê: Armazenamento de objetos é ideal para arquivos não estruturados.

✗ A) Incorreta. Cria modelos com SQL.

✗ C) Incorreta. É para dados relacionais.

✗ D) Incorreta. É banco relacional global.

> **Dica de prova:** Fotos e vídeos: Cloud Storage.

**D9. C) Estruturados seguem um esquema definido** → revise o capítulo 3

Por quê: A nuvem permite extrair valor de dados estruturados e não estruturados.

✗ A) Incorreta. É o contrário.

✗ B) Incorreta. É o contrário.

✗ D) Incorreta. Exigem tratamentos diferentes.

> **Dica de prova:** Estruturado: esquema definido.

**D10. D) ML é um subconjunto da IA** → revise o capítulo 4

Por quê: Machine learning é uma das abordagens para construir sistemas de IA.

✗ A) Incorreta. Estão relacionados.

✗ B) Incorreta. É o contrário.

✗ C) Incorreta. Não é o caso.

> **Dica de prova:** ML dentro da IA.

**D11. B) Treinar um modelo no Agent Platform** → revise o capítulo 4

Por quê: Modelos personalizados exploram dados exclusivos para diferenciação.

✗ A) Incorreta. Não aproveita os dados exclusivos.
✗ C) Incorreta. Não é ML.
✗ D) Incorreta. Não é ML.

> **Dica de prova:** Dados exclusivos e equipe: modelo próprio.

**D12. A) BigQuery ML** → revise o capítulo 4

Por quê: BigQuery ML permite criar e usar modelos de ML com SQL padrão.

✗ B) Incorreta. Traduz textos.
✗ C) Incorreta. Protege aplicações.
✗ D) Incorreta. Transcreve áudio.

> **Dica de prova:** Modelo com SQL no BigQuery: BigQuery ML.


## Agora, treine no formato real

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

**Ver as 6 provas completas e o cupom do mês →**

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

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

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

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