

Databricks Certified Data Engineer Associate



[Enviar feedback sobre o guia do exame](#)

Objetivo deste guia

O objetivo deste guia é oferecer uma visão geral do exame e do conteúdo abordado, ajudando você a avaliar se está preparado para realizá-lo. Este documento é atualizado sempre que há mudanças em um exame (e quando essas mudanças entram em vigor), para que você possa se preparar. Este guia contempla a versão do exame vigente em **4 de maio de 2026**.

Descrição do público-alvo

O exame de certificação Databricks Certified Data Engineer Associate avalia a capacidade do candidato de utilizar a Databricks Data Intelligence Platform para executar tarefas fundamentais de engenharia de dados. O exame avalia o conhecimento do candidato sobre a Data Intelligence Platform, seu workspace, sua arquitetura e seus recursos, além das tarefas relacionadas a ingestão de dados, carga de dados, transformação e modelagem de dados — como a capacidade de executar tarefas de Extract, Transform, Load (ETL) com PySpark, trabalhar com Lakeflow Jobs e aplicar CI/CD. Por fim, o exame avalia o entendimento do candidato sobre técnicas de troubleshooting, monitoramento e otimização, bem como seu conhecimento sobre como garantir governança e segurança na plataforma Databricks.

Sobre o exame

- Número de itens pontuados: 45 questões de múltipla escolha pontuadas.
- Tempo limite: 90 minutos.
- Taxa de inscrição: USD 200, mais os impostos aplicáveis conforme exigido pela legislação local.
- Formato de aplicação: online ou centro de testes
- Materiais de apoio: nenhum é permitido.
- Pré-requisito: nenhum é obrigatório; recomendamos fortemente a participação no curso e seis meses de experiência prática em Databricks.
- Validade: 2 anos.
- Recertificação: obrigatória a cada 2 anos para manter sua certificação. Para se recertificar, você deve realizar a versão vigente do exame. Consulte a seção “Getting Ready for the Exam” na página do exame para se preparar para realizá-lo novamente.

- Conteúdo não pontuado: os exames podem incluir itens não pontuados para coletar informações estatísticas para uso futuro. Esses itens não são identificados no formulário e não afetam sua pontuação. Um tempo adicional é previsto para esse conteúdo.

Treinamento recomendado

- Com instrutor: Data Engineering with Databricks
- Self-paced (disponível na Databricks Academy):
 - Data Ingestion with Lakeflow Connect
 - Deploy Workloads with Lakeflow Jobs
 - DevOps Essentials for Data Engineering
 - Data Interoperability with Unity Catalog
 - Build Data Pipelines with Lakeflow Spark Declarative pipeline
 - Get Started with Data Governance on Databricks

Estrutura do exame

Seção 1: Databricks Intelligence Platform (6%)

- Compreender os componentes essenciais do Databricks Data Intelligence Platform, como sua arquitetura, o Delta Lake e o Unity Catalog.
- Compreender os serviços de compute do Databricks Data Intelligence Platform, incluindo suas características, limitações e modelos de custo, e selecionar a opção mais adequada para o use case de cada workload.

Seção 2: Ingestão e carga de dados (21%)

- Habilitar e detalhar padrões de ingestão de dados, incluindo carga em batch, por streaming e incremental, e importar dados de fontes como arquivos locais, conectores padrão do Lakeflow Connect e managed connectors do Lakeflow Connect.
- Usar o comando COPY INTO para carregar arquivos de forma incremental do cloud object storage (ADLS/S3/GCS) em tabelas governadas pelo Unity-Catalog.
- Usar o Auto Loader com schema enforcement e schema evolution em modos batch (por exemplo, directory listing ou file notification) para carregar dados em tabelas governadas pelo UnityCatalog.
- Configurar o Lakeflow Connect para ingerir dados de diversas fontes corporativas em tabelas governadas pelo UnityCatalog de forma confiável.
- Usar clientes JDBC/ODBC ou REST em notebooks para carregar dados no cloud storage ou diretamente em tabelas governadas pelo UnityCatalog, normalmente orquestrados e agendados com Lakeflow Jobs.
- Priorizar entre Auto Loader, Lakeflow Connect (conectores padrão e managed connectors), connectors de parceiros e outros métodos de ingestão com base em requisitos técnicos como volume de dados, frequência de ingestão, tipos de dados e necessidades de governança com o Unity Catalog.

- Ingerir dados semiestruturados e não estruturados (por exemplo, JSON e dados aninhados) por meio do Lakeflow Connect e de outros managed connectors em tabelas Delta governadas pelo UnityCatalog.

Seção 3: Transformação e modelagem de dados (22%)

- Implementar a limpeza de dados lendo tabelas bronze com PySpark/SQL, tratando valores nulos, padronizando tipos de dados e gravando em novas tabelas silver.
- Combinar DataFrames com operações como inner join, left join, broadcast join, múltiplas chaves, cross join, union e union all.
- Manipular colunas, linhas e estruturas de tabelas adicionando, removendo, dividindo e renomeando nomes de colunas, aplicando filtros e explodindo arrays.
- Realizar operações de deduplicação de dados e operações de agregação em DataFrames, como count, approximate count distinct, mean e summary.
- Compreender os parâmetros básicos de ajuste (spark.sql.shuffle.partitions;
 - spark.default.parallelism, spark.executor/driver.memory,
 - spark.sql.autoBroadcastJoinThreshold) e medir novamente o desempenho.
- Compreender a diferença entre os objetos da camada Gold — como materialized views, views, tabelas e streaming tables — e saber como construí-los para as equipes de BI e analytics no Unity Catalog.
- Aplicar verificações de qualidade de dados e regras de validação para garantir datasets Silver e Gold confiáveis.

Seção 4: Trabalhando com Lakeflow Jobs (16%)

- Implementar fluxos de controle (retries e tarefas condicionais, como ramificações e loops) usando Lakeflow Jobs para a orquestração de pipelines
- Configurar tarefas comuns (tarefas de notebook, SQL query, dashboard e pipeline) e suas dependências usando Lakeflow Jobs e seu grafo de tarefas baseado em DAG
- Implementar agendamentos de jobs usando Lakeflow Jobs, compreendendo os tipos de trigger (agendado, chegada de arquivo e atualização de tabela)
- Escolher entre triggers baseados em tempo e orientados a dados conforme a disponibilidade dos dados e as dependências do pipeline.

Seção 5: Implementando CI/CD (10%)

- Gerenciar seu fluxo de desenvolvimento de código na UI do workspace do Databricks, incluindo criar e alternar entre branches em Databricks Git Folders (anteriormente conhecido como Databricks Repos), fazer commit e push de alterações e criar pull requests usando a integração com Git do Databricks.
- Compreender a configuração específica de cada ambiente usando variáveis e overrides do Automation Bundle (anteriormente conhecido como Databricks Asset Bundles) ao promover a mesma base de código nos targets dev, test e prod.
- Implantar Declarative Automation Bundles (anteriormente conhecido como Databricks Asset Bundles) para empacotar, configurar e promover Lakeflow Jobs, Lakeflow Spark Declarative Pipelines e outros ativos do workspace nos ambientes dev, test e prod.
- Usar o Databricks CLI para validar, implantar e gerenciar Declarative Automation Bundles

(anteriormente Databricks Asset Bundles) e outros ativos do workspace em fluxos automatizados de CI/CD.

Seção 6: Troubleshooting, monitoramento e otimização (10%)

- Identificar tendências no desempenho dos jobs usando a visualização de histórico de execuções do Lakeflow Jobs para comparar os tempos de execução atuais com as referências históricas.
- Usar a UI do Lakeflow Jobs para monitorar a saúde dos pipelines interpretando o estado dos jobs, visualizando grafos de tarefas baseados em DAG para identificar bloqueios upstream e acompanhando os tempos de execução e as taxas de falha dos pipelines.
- Identificar gargalos de desempenho comuns , como data skew, shuffling e disk spilling, interpretando métricas em nível de stage na Spark UI.
- Compreender os recursos de Liquid Clustering e predictive optimization.
- Diagnosticar falhas de startup de cluster, conflitos de bibliotecas e problemas de falta de memória.

Seção 7: Governança e segurança (15%)

- Diferenciar tabelas managed e external no Unity Catalog e realizar operações básicas com elas (criar, modificar, excluir e converter entre tabelas managed e external).
- Configurar controles de acesso pela UI e por SQL aplicando privilégios GRANT, REVOKE e DENY a principais (usuários, grupos e service principals) nos níveis apropriados da hierarquia de segurança.
- Compreender o mascaramento em nível de coluna e a segurança em nível de linha para restringir a visibilidade dos dados com base nos grupos de usuários.
- Compreender as políticas ABAC do Unity Catalog para controlar de forma centralizada a filtragem em nível de linha e o mascaramento de colunas para dados sensíveis.

Exemplos de questões

Estas questões foram retiradas de uma versão anterior do exame. O objetivo é mostrar os objetivos conforme descritos no guia do exame e apresentar um exemplo de questão alinhado a cada objetivo. O guia do exame lista os objetivos que podem ser abordados em um exame. A melhor forma de se preparar para um exame de certificação é revisar a estrutura do exame no guia.

Questão 1

Objetivo: identificar gargalos comuns de desempenho, como data skew, shuffling e disk spilling, interpretando métricas em nível de stage na Spark UI.

Um engenheiro de dados percebe que a duração de um job em batch dobrou depois que uma nova fonte de dados foi integrada. Na Spark UI, o stage mais longo mostra que a maioria das tarefas termina em menos de 30 segundos, mas uma tarefa leva mais de 10 minutos. O resumo de tarefas

do stage mostra leitura de shuffle mínima/mediana em torno de 400 MB, enquanto a leitura de shuffle máxima passa de 5 GB.

Qual solução reduz o tempo de execução do job?

- A. Aumentar o tamanho do cluster para adicionar mais executors, de modo que a tarefa lenta termine mais rápido
- B. Confirmar que o adaptive query execution com tratamento de skew join está ativo para dividir automaticamente a partição superdimensionada em tempo de execução
- C. Reduzir spark.sql.shuffle.partitions para concentrar mais trabalho em menos tarefas
- D. Reparticionar manualmente o dataset usando uma salt key antes do join para distribuir de forma uniforme as keys com skew

Questão 2

Objetivo: compreender os serviços de compute da Databricks Data Intelligence Platform, incluindo suas características, limitações e modelos de custo, e selecionar a opção mais adequada para o use case de cada workload.

Um engenheiro de dados precisa iterar rapidamente em pipelines, mantendo rollbacks confiáveis após ingestões com problemas, garantindo trilhas de auditoria para conformidade regulatória e oferecendo acesso consistente a uma única fonte da verdade tanto para workloads de IA quanto de BI.

Qual estratégia o engenheiro de dados deve usar para atender a esses requisitos?

- A. Armazenamento de CSV no DBFS com versionamento manual de arquivos e cópias noturnas para rollback.
- B. Transações ACID e time travel do Delta Lake, governadas pelo Unity Catalog para acesso consistente e linhagem.
- C. Apenas cloud object storage, com SQL queries ad hoc para recuperação e governança.
- D. DataFrames efêmeros em memória para trilhas de auditoria e distribuição de BI.

Questão 3

Objetivo: habilitar e detalhar padrões de ingestão de dados, incluindo carga em batch, por streaming e incremental e importar dados de fontes como arquivos locais, conectores padrão do Lakeflow Connect e managed connectors do Lakeflow Connect.

Um engenheiro de dados está construindo pipelines downstream para consumir os logs de auditoria do Databricks a partir de um bucket S3 pertencente ao cliente. Antes de implementar a inferência de schema e o checkpointing, ele quer entender o formato de entrega, a latência típica

de ingestão e se os arquivos podem ser sobrescritos.

Qual é o comportamento de armazenamento dos logs de auditoria do Databricks?

- A. Os arquivos são entregues em JSON, com o registro típico de eventos em menos de 15 minutos após o início da entrega, e novas entregas podem sobrescrever arquivos existentes
- B. Os arquivos são entregues em CSV, com garantias de latência inferior a um minuto, e sobrescritas nunca ocorrem depois que um arquivo é gravado, para preservar a imutabilidade
- C. Os arquivos são entregues em Parquet, com consistência eventual acima de 24 horas, e as sobrescritas são desabilitadas para simplificar a ingestão por streaming
- D. Os arquivos são entregues em JSON, com entrega em cadência semanal de batch, e as sobrescritas substituem completamente o conteúdo anterior, sem anexá-lo

Questão 4

Objetivo: diagnosticar falhas de startup de cluster, conflitos de bibliotecas e problemas de falta de memória.

Uma equipe de engenharia de dados atende vários analistas de negócios que executam SQL queries ad hoc ao longo do dia em tabelas Delta curadas. A equipe precisa garantir bom desempenho das queries, startup rápido do cluster e suporte a vários usuários simultâneos, controlando os custos ao evitar o escalonamento desnecessário para clusters muito grandes.

Qual configuração de cluster atende a esses requisitos?

- A. Um job cluster com autoscaling projetado para fluxos de ETL agendados
- B. Um all-purpose cluster configurado com um número fixo de worker nodes
- C. Um cluster de alta concorrência com autoscaling habilitado
- D. Um cluster single-node configurado para tarefas leves de desenvolvimento

Questão 5

Objetivo: gerenciar seu fluxo de desenvolvimento de código na UI do workspace do Databricks, incluindo criar e alternar entre branches em Databricks Git Folders (anteriormente chamado de Databricks Repos), fazer commit e push de alterações e criar pull requests usando a integração com Git do Databricks.

Uma equipe quer uma forma modular de implantar, versionar e orquestrar pipelines de ETL no Databricks — viabilizando CI/CD e repetibilidade.

Qual recurso atende a esse requisito?

- A. Usar models no Unity Catalog para representar jobs de ETL, em que cada model armazena o

artefato de código do pipeline e o CI/CD promove versões atualizando os aliases do model vinculados às tarefas do Job.

B. Empacotar a lógica de transformação como bibliotecas wheel armazenadas em Volumes do Unity Catalog e vinculá-las às tarefas de Jobs para garantir uma implantação determinística entre os ambientes.

C. Empacotar a lógica de API dentro de um notebook montado em Volume e usar a Jobs API v2 para acionar o notebook, contando com o histórico de revisões do notebook como sistema de versionamento.

D. Usar DABs para definir recursos e ativos de código, versioná-los no Git e promover implantações entre os ambientes por meio de ações automatizadas de CI/CD.

Respostas:

1. *B*
2. *B*
3. *A*
4. *C*
5. *D*