

Microsoft Certified: Fabric Data Engineer Associate

Nivel: Intermediate (Associate) Duración: 100 min Nota de corte: 700/1000 Fuente: guía de estudio oficial de Microsoft Learn

1. Implement and manage an analytics solution

30–35%

Configure Microsoft Fabric workspace settings

- Los administradores del área de trabajo eligen el grupo predeterminado (grupo de inicio o personalizado), un entorno predeterminado con runtime y bibliotecas, y la admisión de trabajos: optimista por defecto o reservando el máximo de núcleos. También fijan el tiempo de espera de sesión, una duración máxima que solo cancela trabajos de usuarios y la alta simultaneidad, que comparte sesiones de un mismo usuario.
- Un dominio agrupa áreas de trabajo por área de negocio; al asignar un área de trabajo, todos sus elementos heredan el dominio, pero el acceso sigue dependiendo de los roles y permisos. Los colaboradores de dominio deben ser administradores del área de trabajo y la asignan desde su configuración. Un dominio predeterminado asigna las áreas sin dominio de ciertos usuarios y las que creen después, sin cambiar las ya asignadas.
- La pestaña OneLake de la configuración del área de trabajo activa la caché de accesos directos, que guarda los archivos leídos desde S3, GCS o una puerta de enlace para reducir la salida, con retención configurable y opción de restablecerla. También activa los diagnósticos de OneLake, que escriben eventos de acceso en JSON en un almacén de lago de la misma capacidad y pueden hacerse inmutables.
- La configuración del área de trabajo define el runtime predeterminado de Apache Airflow. El grupo de inicio arranca al instante y se desaprovisiona cuando no se usa, ideal para desarrollo; un grupo personalizado está siempre activo hasta pausarlo, con tamaño de nodo Large o Small, escalado automático y nodos adicionales para más DAG simultáneos. Si se desactiva la personalización del proceso, se usa el grupo de inicio.

Implement lifecycle management in Fabric

- La integración con Git funciona por área de trabajo, conectada a una rama y carpeta de Azure DevOps o GitHub. Solo el administrador conecta o desconecta; los colaboradores confirman y actualizan, y el administrador puede permitirles cambiar de rama. Actualizar siempre trae toda la rama, y las confirmaciones se bloquean si hay cambios en ramas. Lo recomendado es que cada desarrollador trabaje en un área de trabajo ramificada.
- Los proyectos de base de datos de estilo SDK guardan los esquemas como archivos .sql, se compilan en un .dacpac y se publican con DacFx o SqlPackage; conviene generar el script antes de publicar y usar BlockOnPossibleDataLoss en producción. En una base de datos SQL de Fabric conectada a Git, Fabric mantiene el proyecto, cada actualización copia y publica, y los scripts posteriores gestionan datos estáticos.
- Una canalización de implementación asigna un área de trabajo a cada fase y copia metadatos, no datos, entre fases. Los elementos emparejados se sobrescriben conservando identificador y URL; los no emparejados se duplican. Las reglas (origen de datos, parámetro o almacén de lago predeterminado para cuadernos) se definen en la fase de destino. Implementar exige ser administrador de la canalización y colaborador en ambas áreas.

Configure security and governance

- Los roles del área de trabajo (Administrador, Miembro, Colaborador y Visor) se asignan a usuarios o grupos y se aplican a todos sus elementos; con varios grupos prevalece el permiso más alto. Solo el administrador elimina el área o añade administradores; el miembro añade roles inferiores; el colaborador escribe y lee datos con Spark; el visor solo lee por SQL salvo que uno de seguridad de OneLake le de más.
- Además de los roles del área de trabajo, puedes compartir un elemento concreto: compartir siempre concede Read, que permite descubrir y abrir el elemento y conectarse al punto de conexión SQL, y puedes añadir ReadData (consultas T-SQL), ReadAll (Spark y OneLake), Edit o Share. Los permisos de elemento y los roles del área se evalúan por separado, así que restar uno no resta el otro.
- En el almacén de datos y el punto de conexión SQL, la seguridad de objetos se aplica con GRANT, REVOKE y DENY; la de columnas, con un GRANT SELECT que enumera columnas; y la de filas, con CREATE SECURITY POLICY y una función insertada como predicado de filtro. En OneLake, los roles de seguridad conceden acceso a carpetas y tablas; los permisos se heredan hacia abajo y se pueden recorrer las carpetas primarias sin ver las hermanas.
- El enmascaramiento dinámico de datos oculta valores en los resultados de las consultas sin cambiar los datos almacenados; se define con MASKED WITH y las funciones default(), email(), random() y partial(). Ven datos reales quienes tienen UNMASK o CONTROL, incluidos los roles Administrador, Miembro y Colaborador. No impide deducir valores con filtros, así que se combina con seguridad de columna o de fila.
- Las etiquetas de confidencialidad de Microsoft Purview se aplican a cualquier elemento de Fabric desde el encabezado o la configuración, con permisos de edición sobre el elemento. Se propagan por herencia al crear (por ejemplo, de un almacén de lago a un cuaderno) y por herencia descendente, que nunca sobrescribe etiquetas manuales. Asociadas a directivas de protección, bloquean el acceso a quien no figure en la directiva.

Orchestrate processes

- Dataflow Gen2 es la opción de poco código basada en Power Query para preparar y transformar datos; la actividad Copy de una canalización mueve grandes volúmenes sin código; y un cuaderno ejecuta código de Spark con bibliotecas de código abierto para transformaciones complejas. Las canalizaciones también orquestan flujos de datos y cuadernos con programaciones, bucles, condiciones y rutas de error.
- Una canalización se inicia a petición, con programaciones (varias por canalización, cada una con frecuencia, inicio y fin obligatorios) o con desencadenadores de eventos de archivos, trabajos o área de trabajo, que usan Eventstream y Activator. Los eventos de trabajo como ItemJobSucceeded encadenan elementos sin orquestador central. Un cuaderno programado se ejecuta con la identidad de quien creó o actualizó la programación.
- La actividad Notebook pasa parámetros base que sustituyen los valores de la celda de parámetros del cuaderno. notebookutils.notebook.run ejecuta un cuaderno hijo y devuelve su valor de exit, siempre como cadena; runMultiple ejecuta varios en paralelo con un DAG de dependencias. En las expresiones, @{ } interpola dentro de texto y siempre devuelve una cadena, y @item() da el elemento actual de un ForEach.

2. Ingest and transform data

30–35%

Design and implement loading patterns

- Una carga completa copia todo en cada ejecución; una incremental copia todo la primera vez y después solo lo nuevo o cambiado. Copy job admite marca de agua (solo inserciones y actualizaciones) o CDC (también eliminaciones) y métodos Append, Merge, Overwrite o SCD tipo 2. En canalizaciones, el patrón de marca de agua usa Lookup, Copy y un procedimiento almacenado; Dataflow Gen2 reemplaza buckets por fecha.
- El proceso ETL carga el almacenamiento provisional, después las dimensiones en orden de dependencia y al final los hechos, buscando la clave suplente de cada dimensión. No se truncan dimensiones con claves suplentes generadas; los cambios se aplican como SCD tipo 1, 2 o 3 y las bajas como eliminación lógica. Si falta un miembro, se usa un miembro Unknown o un miembro inferido.
- Un Eventstream puede cargar en un Eventhouse con ingesta directa o con procesamiento antes de la ingesta si se aplican operadores, o en un almacén de lago, que aplica el esquema del primer registro y no conviene para datos CDC. Con Spark Structured Streaming se escribe en Delta con append, checkpoint y un trigger para lotes mayores; en producción se usa una definición de trabajo de Spark con reintentos.

Ingest and transform batch data

- El almacén de lago encaja con Spark y datos de cualquier tipo; su punto de conexión SQL es de solo lectura. El almacén de datos se elige para T-SQL, datos estructurados y transacciones de varias tablas. El Eventhouse sirve para eventos en streaming con KQL y su disponibilidad en OneLake es opcional. SQL database in Fabric cubre OLTP y Cosmos DB in Fabric, NoSQL.
- Dataflow Gen2 ofrece transformaciones de Power Query sin código o con poco código, para analistas y volúmenes de bajos a medios. Los cuadernos de Spark manejan cualquier tipo de dato y lógica compleja con código. En una base de datos KQL, una directiva de actualización transforma los datos al ingerirlos. En un almacén de datos, T-SQL (INSERT...SELECT, CTAS) transforma por conjuntos.
- Un acceso directo de OneLake apunta a datos internos o externos sin copiarlos. En Tables solo se crea en el nivel superior y, si el destino es Delta y el nombre no tiene espacios, se reconoce como tabla; en Files no hay detección de tablas. Se aplica el permiso más restrictivo entre origen y destino, y borrar el acceso directo no borra los datos de destino.
- El reflejo replica de forma casi continua bases de datos (Azure SQL, SQL Server, Cosmos DB, Snowflake...) en tablas Delta de OneLake con un punto de conexión de análisis SQL de solo lectura. El reflejo de metadatos (Azure Databricks) usa accesos directos sin copiar datos, y el reflejo abierto permite que una aplicación deposite archivos con `__rowMarker__` y un `__metadata.json` con `keyColumns` en la zona de aterrizaje.
- Las canalizaciones combinan la actividad Copy con actividades de control: Lookup lee configuración o marcas de agua (`output.firstRow` o `output.value`), ForEach recorre colecciones con `item()`, Stored procedure guarda la nueva marca e Invoke pipeline reutiliza otras canalizaciones. Se ejecutan a petición, con programaciones o con desencadenadores de eventos, y una actividad Copy solo admite una puerta de enlace local.

Ingest and transform streaming data

- Eventstream ingiere, transforma con operaciones ligeras y enruta eventos sin código a destinos como Eventhouse, almacén de lago o Activator. Spark Structured Streaming usa código PySpark para transformaciones complejas, puntos de control y receptores Delta, de forma continua o con el desencadenador `available-now`. En un Eventhouse, las directivas de actualización transforman los datos en KQL al ingerirlos.
- Las tablas nativas de un Eventhouse almacenan los datos ingeridos con índices, ofrecen la menor latencia y admiten directivas de actualización y vistas materializadas. Los accesos directos de OneLake consultan datos en su ubicación con `external_table()`, sin copiarlos, y la aceleración de consultas los almacena en caché para acercarse al rendimiento nativo. La disponibilidad en OneLake expone las tablas KQL en formato Delta a otros motores.
- Un acceso directo de OneLake estándar en una base de datos KQL se consulta con `external_table()` sin mover datos, pero es más lento que una tabla nativa. La aceleración de consultas almacena en caché e indexa los datos Delta durante un período activo en días (o según `HotDateTimeColumn`), con rendimiento similar al ingerido y coste de SSD. Se comporta como tabla externa: sin vistas materializadas.
- Eventstream transforma eventos con operadores sin código (Filter, Manage fields, Aggregate, Group by, Expand, Union, Join) o con un operador SQL basado en el lenguaje de Stream Analytics, que admite varias salidas con `SELECT ... INTO`. Solo almacén de lago, Eventhouse (procesamiento previo a la ingesta), flujo derivado y Activator admiten operadores previos; para otros destinos se usa un flujo derivado.
- Structured Streaming trata un flujo como una tabla a la que se añaden filas; con `writeStream.format("delta")`, `checkpointLocation` y `toTable` escribe en tablas del almacén de lago. Los desencadenadores por intervalo y Optimized Write reducen archivos pequeños, `foreachBatch` permite MERGE y las marcas de agua acotan el estado. En producción se usan definiciones de trabajo de Spark con directiva de reintento.

3. Monitor and optimize an analytics solution

30–35%

Monitor Fabric items

- La ingesta se supervisa en varios sitios: el Monitoring hub y la vista de detalles de ejecución para canalizaciones y trabajos de copia (filas leídas y escritas, límites incrementales), Data insights y Runtime logs en Eventstream, Monitor replication en bases de datos reflejadas y la supervisión del área de trabajo, que guarda registros (por ejemplo, EventhouseIngestionResultsLogs) consultables con KQL.
- Las transformaciones se supervisan según el motor: Dataflow Gen2 con el historial de actualizaciones (estadísticas por actividad y registros detallados), Spark con el Monitor hub, Recent runs y la página de detalles de la aplicación (Jobs, Logs, Resources, Item snapshots y una pestaña de Structured Streaming), y los almacenes de datos con las vistas de queryinsights, como long_running_queries.
- El historial de actualizaciones de un modelo semántico muestra cada intento automático, las fases de datos y de caché de consultas y los estados completados con advertencias. El Monitoring hub y la página Semantic Model Refresh Detail centralizan estado, errores y métricas, y la supervisión del área de trabajo guarda registros consultables con KQL. Solo la actualización programada envía correos de error.
- Para alertar de una actividad concreta se añade una actividad Outlook o Teams tras ella; para toda un área de trabajo, la supervisión escribe ItemJobEventLogs y una regla de Activator sobre un conjunto de consultas KQL avisa de cualquier fallo. Activator envía una alerta por registro devuelto, las reglas nacen detenidas y solo evalúan datos nuevos.

Identify and resolve errors

- La vista de detalle de una ejecución muestra el error de cada actividad y permite volver a ejecutar desde la actividad con error. Las actividades desactivadas no se ejecutan ni tienen campo error. La guía de solución de problemas explica códigos como 2002, 2003 o LSROBOTokenFailure, y la supervisión del área de trabajo guarda ejecuciones de canalización y de actividad sin detalle del error.
- El historial de actualizaciones muestra tablas y actividades con su error, y los registros detallados del motor Mashup se descargan desde cada ejecución. Las reglas de validación de destinos (DestinationQueryNotFound, DestinationQueryChallenge, DestinationQueryHasUnsupportedScript) tienen soluciones concretas, y con puerta de enlace la lectura de datos provisionales usa TDS por el puerto 1433.
- Los errores de cuadernos se diagnostican con la interfaz de Spark (pestañas Executors y Stages) y los registros del controlador desde el centro de supervisión. El código 137 indica falta de memoria del contenedor, collect() o toPandas() saturan el controlador y las opciones spark.executor.* deben ir en una celda %%configure inicial, no en spark.conf.set().
- En un Eventhouse, .show ingestion failures muestra los errores de los comandos de ingesta con Details y FailureKind, y filtrando por OriginatesFromUpdatePolicy revela los fallos de directivas de actualización. Con IsTransactional en false los datos llegan solo al origen. E_QUERY_RESULT_SET_TOO_LARGE se resuelve reduciendo el resultado y E_RUNAWAY_QUERY con más memoria por iterador o shuffle.
- Cada nodo de un Eventstream ofrece Data insights, con métricas como errores de deserialización o de conversión, retraso de marca de agua y eventos acumulados, y Runtime logs, con mensajes filtrables por gravedad. La supervisión del área de trabajo añade tablas consultables con KQL, pero exige activar Log Eventstream activity en cada Eventstream.

Optimize performance

- OPTIMIZE compacta archivos pequeños; con ZORDER BY agrupa valores para omitir archivos y con VORDER aplica V-Order. La compactación automática lanza OPTIMIZE tras las escrituras solo si hay fragmentación, y el tamaño objetivo adaptable ajusta el tamaño al crecer la tabla. VACUUM borra archivos no referenciados y OPTIMIZE FULL reagrupa tras cambiar claves de liquid clustering.
- Para acelerar canalizaciones, ForEach sin Sequential procesa elementos en paralelo y Batch count limita cuántos a la vez. En la actividad Copy, Intelligent throughput optimization asigna recursos, Degree of copy parallelism fija los subprocesos y las opciones de partición de SQL reparten la lectura. Copy job con copia incremental (marca de agua o CDC) evita recargar tablas completas.
- En un almacén de datos de Fabric conviene cargar por lotes con COPY INTO o un único INSERT en vez de inserciones fila a fila, que fragmentan los archivos; la compactación es automática. V-Order está activo por defecto y deshabilitarlo es irreversible. Usa tipos de datos ajustados (varchar(n), enteros, fechas) y OPTION (FORCE DISTRIBUTED PLAN) ante operaciones no escalables.
- En un Eventhouse, la directiva de caché mantiene datos activos en SSD y la de retención decide cuánto tiempo se pueden consultar. Las vistas materializadas guardan agregaciones y materialized_view() consulta solo la parte materializada; la aceleración de consultas acelera accesos directos de OneLake. En Eventstream, la configuración de rendimiento de eventos escala orígenes y destinos.
- El motor de ejecución nativo acelera consultas Spark sobre Parquet y Delta sin cambiar código y vuelve a JVM con operaciones no admitidas, como structured streaming o JSON. El modo de alta simultaneidad comparte una sesión entre cuadernos del mismo usuario y configuración. Los perfiles de recursos (writeHeavy, readHeavyForSpark, readHeavyForPBI) aplican configuraciones de lectura o escritura.