

# Microsoft Certified: DevOps Engineer Expert

Domínios, pesos oficiales y habilidades clave del temario de AZ-400 para repasar antes del examen.

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

## 1. Design and implement processes and communications

10–15%

### Design and implement traceability and flow of work

- GitHub flow es un flujo ligero basado en ramas: creas una rama descriptiva, confirmas cambios aislados, abres una PR (en borrador si quieres opinión temprana), atiendes la revisión, combinas y eliminas la rama. En Azure Repos, Microsoft recomienda ramas de característica, PR con validación de compilación y revisores automáticos en main, y en producción, ramas de producción. En Azure Pipelines, el archivo azure-pipelines.yml puede ocultar la incidencia en blanco, y las subincidencias crean jerarquías. Las notificaciones llegan a la bandeja web, móvil o correo según si participas o vigilas el repositorio. Azure DevOps entrega avisos por correo o Slack. Los requisitos de construcción se definen en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En Azure Boards, la configuración de requisitos de construcción se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En Azure Boards, la configuración de requisitos de construcción se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En Azure Boards, la configuración de requisitos de construcción se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.

### Design and implement appropriate metrics and queries for DevOps

- El tiempo de entrega va de la creación del elemento a su estado Completed y el de ciclo, desde que entra en In Progress hasta Completed; los widgets Lead Time y Cycle Time muestran la media móvil y la desviación estándar, y el CFD revela cuellos de botella y exceso de WIP. Para la entrega, Cloud Adoption Framework propone Lead Time.
- Para el tiempo de entrega, se puede usar el widget Lead Time y Cycle Time. El widget Lead Time muestra la media móvil y la desviación estándar, y el CFD revela cuellos de botella y exceso de WIP. Para la entrega, Cloud Adoption Framework propone Lead Time.
- Las métricas de entrega se definen en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- Las métricas de entrega se definen en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- Las métricas de entrega se definen en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- Las métricas de entrega se definen en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.

### Configure collaboration and communication

- La wiki de Azure DevOps admite Markdown y diagramas Mermaid con el contenedor :: mermaid o un bloque delimitado mermaid; los diagramas de flujo se escriben con graph, no con flowchart. Etiquetas como [[\_TOC\_]] y [[\_TOSP\_]] generan índices, y Publicar código como wiki asigna una rama y carpeta del repositorio, ordenadas con el nombre del código de la presentación. Las métricas de entrega se definen en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En GitHub, el documento de presentación se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En GitHub, el documento de presentación se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En GitHub, el documento de presentación se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En GitHub, el documento de presentación se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.
- En GitHub, el documento de presentación se define en los archivos de configuración de pipeline y se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción. Los requisitos de construcción se vinculan a los requisitos de construcción.

## 2. Design and implement a source control strategy

10-15%

### Design and implement branching strategies for the source code

- Microsoft recomienda una estrategia sencilla: ramas de características cortas desde main, combinadas mediante pull requests y merge con main.
- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

### Configure and manage repositories

- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

## 3. Design and implement build and release pipelines

50-55%

### Design and implement a package management strategy

- GitHub Packages aloja paquetes npm, RubyGems, Maven, Gradle, NuGet e imágenes de contenedor junto al código.
- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

### Design and implement a testing strategy for pipelines

- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

### Design and implement pipelines to deploy artifacts

- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

### Design and implement deployments

- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

### Design and implement infrastructure as code (IaC)

- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

### Maintain pipelines

- En Azure DevOps, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.
- En GitHub, las ramas de características se crean desde la rama de producción y se devuelven a producción mediante pull requests.

## 4. Develop a security and compliance plan

10–15%

### Design and implement authentication and authorization methods

- Las identidades administradas evitan gestionar credenciales. La asignada por el sistema vive y muere con su recurso y solo ese recurso la usa; la asignada por el usuario es un recurso independiente, se comparte y admite asignaciones de roles previas al despliegue. Fuera de Azure, como en GitHub Actions, se usa una aplicación de Entra o una identidad asignada por el usuario con credencial federada.
- El GITHUB\_TOKEN es un token de instalación creado para cada trabajo, limitado al repositorio y ajustable con permissions; lo que hace no desencadena nuevas ejecuciones salvo excepciones. Las GitHub Apps son la opción para integraciones duraderas: permisos específicos, tokens de corta duración y sin depender de un usuario. Los tokens personales sirven para scripts breves; se prefieren los específicos.
- Las conexiones de servicio tienen los roles Reader, User, Creator y Administrator, y cada canalización debe autorizarse por separado en lugar de abrirlas a todas. Para Azure se recomienda la federación de identidades de carga de trabajo. Los PAT de Azure DevOps son un último recurso: ámbito de organización, ámbitos mínimos y vida corta; nunca superan los permisos de su propietario y regenerarlos invalida el valor anterior.
- Los roles de repositorio de GitHub van de menos a más acceso: Read, Triage, Write, Maintain y Admin. Triage gestiona incidencias sin escribir, Maintain administra el repositorio sin acciones sensibles y Admin controla protección de ramas, webhooks y borrado. En la organización hay roles predefinidos como security manager, billing manager, moderador o GitHub App manager, y los propietarios tienen acceso total.
- En Azure DevOps los permisos se asignan a grupos de seguridad en ámbito de organización, proyecto u objeto, con estados Allow, Deny y Not set. En el mismo ámbito un Deny suele prevalecer sobre un Allow, y una asignación explícita en un nodo hijo sustituye a la heredada. Project Collection Administrators gestiona procesos, directivas y extensiones; Project Administrators, equipos, rutas y repositorios.

### Design and implement a strategy for managing sensitive information in automation

- En Azure Pipelines, la tarea AzureKeyVault@2 descarga secretos como variables; con RunAsPreJob quedan disponibles para todo el trabajo, y la identidad necesita Get y List sobre secretos. Un grupo de variables vinculado a Key Vault solo asigna nombres de secretos (no claves ni certificados) y obtiene el valor más reciente en cada ejecución. En GitHub Actions se inicia sesión con OIDC y se enmascara el valor con ::add-mask::.
- La federación de identidades de carga de trabajo evita guardar secretos: en Azure Pipelines se recomienda para las conexiones de servicio de Azure Resource Manager, y las creadas automáticamente y usadas por un solo proyecto se pueden convertir. En GitHub Actions, el trabajo necesita id-token: write y azure/login intercambia el token OIDC por un token de Azure de corta duración, válido solo durante el trabajo.
- Los archivos seguros de Azure Pipelines (certificados, almacenes de claves, perfiles de aprovisionamiento, claves SSH) se guardan cifrados, son un recurso protegido con permisos de canalización, aprobaciones y comprobaciones, y no se pueden editar tras subirlos. DownloadSecureFile@1 expone la ruta en \$(nombre.secureFilePath) y borra el archivo al terminar el trabajo. En GitHub Actions se usan secretos en Base64 o archivos cifrados con gpg.
- Para evitar fugas, nunca imprimas secretos ni los pases por la línea de comandos: asignalos a variables de entorno y no uses datos estructurados como secreto, porque la ocultación depende de coincidencias exactas. Los secretos no llegan a compilaciones de bifurcaciones ni, en GitHub, a flujos reutilizables de forma automática. Ancla las acciones a un SHA completo y usa variables de entorno intermedias frente a la inyección.

### Automate security and compliance scanning

- Una estrategia de análisis combina análisis de secretos (todo el historial más protección de inserción), análisis de código con CodeQL, análisis de dependencias y control de licencias. En GitHub, las alertas de Dependabot detectan dependencias vulnerables en la rama predeterminada y la acción de revisión de dependencias bloquea solicitudes con paquetes vulnerables o licencias no permitidas (allow-licenses o deny-licenses).
- La seguridad de DevOps de Defender for Cloud conecta Azure DevOps, GitHub y GitLab mediante conectores y centraliza resultados de código, secretos, dependencias e IaC. La extensión y la acción Microsoft Security DevOps ejecutan analizadores como Bandit, Checkov, Template Analyzer o Trivy; en Azure DevOps los SARIF se publican en el artefacto CodeAnalysisLogs. Las anotaciones en solicitudes requieren validación de compilación.
- En GitHub, Advanced Security ofrece análisis de secretos con protección de inserción, análisis de código con CodeQL y alertas de Dependabot. En Azure Repos, GitHub Advanced Security for Azure DevOps usa tareas como AdvancedSecurity-Codeql-Init@1, AdvancedSecurity-Codeql-Analyze@1 y AdvancedSecurity-Dependency-Scanning@1, y las comprobaciones de estado NewHighAndCritical o AllHighAndCritical bloquean solicitudes.
- La integración de GitHub Advanced Security con Defender for Cloud usa el conector de GitHub para asignar repositorios a cargas de trabajo en ejecución y añadir factores de riesgo (exposición a Internet, datos sensibles, recursos críticos, movimiento lateral) a las alertas de GHAS. Las campañas se crean en la organización de GitHub con el filtro Runtime Risks, y las incidencias generadas desde Defender se sincronizan en ambos sentidos.
- Para analizar imágenes en la canalización se usan Microsoft Security DevOps con Trivy (break: true hace fallar la compilación) o la CLI de Defender for Cloud (defender scan image); Defender for Containers evalúa además las imágenes del registro y las que están en ejecución. CodeQL debe ejecutarse en el mismo contenedor en que se compila el código (container: image: en el trabajo) y no admite imágenes Alpine con musl.

## 5. Implement an instrumentation strategy

5–10%

### Configure monitoring for a DevOps environment

- Azure Pipelines se integra con Azure Monitor mediante la comprobación o puerta Query Azure Monitor alerts, que solo deja avanzar si no hay reglas de alertas activadas. Application Insights crea elementos de trabajo en Azure DevOps o incidencias en GitHub con plantillas que son libros de Azure Monitor. La auditoría de Azure DevOps puede enviarse a Azure Monitor Logs para consultarla con KQL en la tabla AzureDevOpsAuditing.
- Para las máquinas virtuales se instala el agente de Azure Monitor y se asocian reglas de recopilación de datos; las métricas del host se recogen sin agente. En AKS se combinan Prometheus administrado, la recopilación de registros de contenedores y una configuración de diagnóstico para el plano de control. Application Insights usa una cadena de conexión y OpenTelemetry, y Storage insights y Network insights funcionan sin configuración.
- Las estadísticas de GitHub Projects permiten crear gráficos actuales y gráficos históricos (eje X en Time), filtrarlos con la misma sintaxis que las vistas y sumar campos numéricos en el eje Y; no siguen elementos archivados. Los repositorios ofrecen Pulse y el gráfico de tráfico, y la organización ofrece métricas de uso (minutos) y de rendimiento (colas, errores) de Actions.
- En GitHub, la configuración de notificaciones de Actions permite recibir correos o avisos web y limitarlos a los flujos con error; para automatizar alertas, workflow\_run con types: [completed] y una condición sobre github.event.workflow\_run.conclusion reacciona a otro flujo. En Azure DevOps, las suscripciones de equipo usan plantillas como Build fails o Run stage waiting for approval y se entregan por correo o enlace de servicio.

### Analyze metrics from instrumentation

- Las métricas del host de una máquina virtual, como Percentage CPU, Available Memory Bytes, OS Disk Queue Depth o Network In Total, se recopilan sin agente y se analizan en el explorador de métricas con agregaciones, filtros y divisiones. La pestaña Performance de VM insights usa datos del invitado y muestra gráficos Top N de CPU, memoria, disco y red, incluido el tráfico entre redes virtuales.
- Application Insights ofrece herramientas de uso: Users, Sessions y Events, Funnels para la conversión en una secuencia fija, User Flows para explorar rutas, Cohorts, retención e Impact Analysis. Los usuarios se cuentan por identificadores anónimos salvo que se establezca un identificador autenticado. Las vistas Performance y Failures enumeran operaciones con recuento, duración media y usuarios afectados.
- Application Insights correlaciona la telemetría distribuida con operation\_Id, compartido por todos los elementos de una operación, y operation\_ParentId, que apunta al padre inmediato; con W3C Trace Context, traceparent lleva esos identificadores. El mapa de aplicación identifica componentes por el nombre del rol en la nube, y los detalles de la transacción muestran un diagrama de Gantt de solicitudes y dependencias.
- Una consulta KQL empieza por una tabla y encadena operadores con |: where filtra (== distingue mayúsculas y ~= no), project elige columnas, extend añade columnas, summarize agrega con funciones como count() o avg() y bin() agrupa por intervalos de tiempo. take devuelve registros arbitrarios, top ordena en el servidor y render debe ir al final.