

Microsoft Certified: Azure AI Fundamentals

Domínios, pesos oficiales y habilidades clave del temario de AI-901, para repasar antes del examen.

Nivel: Beginner Nota de corte: 700/1000 Fuente: guía de estudio oficial de Microsoft Learn

1. Identify AI concepts and capabilities

40–45%

Describe principles of responsible AI

- La equidad exige que un sistema de IA no trate de forma distinta a personas o grupos similares. Como los datos los eligen personas y pueden reflejar sesgos inconscientes, hay que minimizar el sesgo en los datos de entrenamiento y probar la equidad de los resultados por grupos sensibles, por ejemplo con la evaluación de equidad del panel de Responsible AI.
- La IA se basa en modelos probabilísticos y no es infalible, así que las aplicaciones deben mitigar sus riesgos: funcionar según lo diseñado, responder de forma segura ante condiciones inesperadas y evitar actuar si la confianza es baja. Los filtros de contenido ayudan, pero los principios se consideran desde el diseño hasta la operación.
- Los modelos se entrenan con datos que pueden incluir información personal, por lo que hay que mantenerlos seguros y evitar que el modelo entrenado revele detalles privados. Se aplica con control de acceso, cifrado en tránsito y en reposo, auditoría, borrado de datos que ya no se necesitan y herramientas como SmartNoise (privacidad diferencial) y Counterfit.
- La inclusión busca que el potencial de la IA esté abierto a todos y que las soluciones no excluyan a ningún usuario. En la práctica, un agente por voz ofrece también subtítulos de texto, y el diseño de agentes tiene en cuenta la accesibilidad, la variación lingüística y los distintos niveles de competencia técnica, con aportaciones de comunidades diversas desde el principio.
- La transparencia consiste en que los usuarios sepan cómo funciona un sistema de IA y qué limitaciones tiene: indicar que se usa IA, describir las características de los datos de entrenamiento sin revelar información confidencial y ofrecer Transparency Notes o avisos en el producto. La interpretabilidad (explicaciones globales, locales y por cohorte) ayuda a entender sus decisiones.

Identify AI model components and configurations

- Los modelos generativos de lenguaje dividen el texto en tokens, los convierten en embeddings con un transformador (codificador con atención) y un decodificador predice, token a token, la continuación más probable de un prompt. Los ejemplos del prompt condicionan solo esa inferencia y la aplicación reenvía el historial para dar contexto.
- Elegir modelo empieza por la tarea: embeddings para búsqueda semántica, multimodal para analizar imágenes con texto, modelos de imagen o vídeo para generar contenido, modelos pequeños para apps ligeras y modelos rápidos para chat de alto volumen. Si el caso es muy concreto, una herramienta de Foundry con modelos prediseñados da resultados más predecibles.
- Al desplegar un modelo eliges el tipo de implementación (Global Standard por defecto, Data Zone o Standard para residencia, Provisioned para capacidad reservada, Batch para trabajos asíncronos), la versión y los tokens por minuto. Temperatura, top_p y tokens máximos de salida se ajustan en el playground o en el código.

Identify AI workloads

- Las cargas de trabajo habituales son: IA generativa y agentes (modelo, instrucciones y herramientas que actúan), análisis de texto (sentimiento, frases clave, entidades, PII), voz (reconocimiento, síntesis y voz a voz), visión (análisis de imágenes y generación) y extracción de información de documentos, audio y vídeo con esquemas.
- La extracción de frases clave lista los conceptos principales de un texto, el reconocimiento de entidades detecta y clasifica elementos como personas, lugares, fechas o cantidades, y el análisis de sentimiento etiqueta frases y documentos como positivos, neutrales o negativos con puntuaciones de confianza. El resumen puede ser extractivo (frases originales) o abstractivo (redacción nueva).
- El reconocimiento de voz (voz a texto) convierte audio en texto mediante un modelo acústico y un modelo de lenguaje, y puede ser en tiempo real, rápido o por lotes, con modelos personalizados para vocabulario específico. La síntesis de voz (texto a voz) genera audio con voces neuronales y se ajusta con SSML. Voice Live une ambas para conversar en tiempo real con un agente.
- Los modelos multimodales con visión aceptan imágenes y texto en una misma solicitud (por URL o base64) para describir fotos y responder preguntas sobre ellas. Los modelos de generación de imágenes crean imágenes a partir de texto, y los de vídeo generan clips mediante trabajos asíncronos. Azure Vision aporta OCR y Face para leer texto y analizar rostros.
- Azure Content Understanding en Foundry Tools convierte documentos, imágenes, audio y vídeo en datos estructurados mediante analizadores con un esquema de campos, más allá del OCR básico. Los analizadores de audio transcriben con diarización, y los de vídeo añaden fotogramas clave, planos y segmentos. El texto plano se analiza con Azure Language, por ejemplo con el reconocimiento de entidades.

2. Implement AI solutions by using Microsoft Foundry

55–60%

Implement generative AI apps and agents by using Foundry

- El mensaje del sistema fija el rol, el tono, el alcance y las barreras del asistente, y el del usuario es la petición concreta. Un buen mensaje es claro, específico y conciso, indica el formato de salida, puede incluir ejemplos (few-shot), datos de referencia y una salida cuando no hay respuesta, y repite instrucciones clave al final si el contexto es largo.
- Desplegar un modelo en el portal de Foundry exige un proyecto y elegir un tipo de despliegue; el nombre del despliegue es el valor que se pasa en el parámetro model. Al terminar llegas al playground, donde pruebas prompts, ajustas parámetros como la temperatura, comparas hasta tres modelos y consultas la pestaña Code.
- Un cliente de chat ligero recoge la entrada del usuario, llama al modelo desplegado y muestra la respuesta. Con el SDK de Foundry en Python creas un AIProjectClient con el endpoint del proyecto y DefaultAzureCredential, obtienes el cliente OpenAI con get_openai_client y llamas a responses.create con model e input.
- Un agente de Foundry combina un modelo, instrucciones y herramientas; el conocimiento aporta contexto mediante RAG. En el playground de agentes configuras y pruebas conversaciones sin código, guardas cambios como versiones inmutables y usas trazado, evaluaciones y publicación para depurar, medir y compartir el agente.
- Para llamar a un agente desde un cliente ligero en Python, creas un AIProjectClient con el endpoint del proyecto y obtienes un cliente OpenAI con get_openai_client, apuntando al agente por nombre. Con una conversación y responses.create mantienes el historial entre turnos; el patrón recomendado es crear el agente una vez y reutilizarlo.

Implement AI solutions for text and speech by using Foundry

- Una app ligera puede analizar texto con un modelo de propósito general (flexible, salida probabilística) o con Azure Language en Foundry Tools (analizadores con salida estructurada y determinista). Con el SDK de Python usas TextAnalyticsClient y métodos como detect_language, recognize_pii_entities, analyze_sentiment o extract_key_phrases.
- Un modelo de chat con audio desplegado en Foundry recibe la pregunta hablada como un elemento input_audio con datos en base64 y su formato. Para obtener voz en la respuesta se indican las modalidades text y audio y un ajuste de voz y formato; el audio devuelto viene en base64 y hay que decodificarlo. La API realtime, separada, cubre la voz interactiva de baja latencia.
- Con el SDK de Azure Speech para Python se crea un SpeechConfig con el punto de conexión y la clave del recurso de Foundry, un SpeechRecognizer para voz a texto y un SpeechSynthesizer para texto a voz. La transcripción puede ser en tiempo real o por lotes (asíncrona), y Voice Live reúne reconocimiento, modelo generativo y síntesis para agentes de voz.

Implement AI solutions with computer vision and image-generation capabilities by using Foundry

- Un modelo multimodal con visión desplegado en Foundry acepta texto e imágenes en una misma solicitud y puede describir fotos, responder preguntas sobre ellas o interpretar gráficos y capturas. Las imágenes se envían como URL accesible públicamente o como datos en base64; el parámetro detail ajusta el nivel de detalle y max_tokens evita respuestas cortadas.
- Los modelos de generación de imágenes de Foundry crean imágenes a partir de texto y también permiten editar o transformar imágenes existentes; se despliegan en un recurso de Foundry, se prueban en el área de pruebas y se usan con la API Responses o la API de imágenes, que devuelve los datos en base64. La generación de vídeo es un trabajo asíncrono: crear, consultar el estado y descargar.
- Una aplicación ligera de visión usa el SDK de OpenAI para Python con la clave y el punto de conexión del recurso de Foundry (o Entra ID) y pasa como model el nombre del despliegue. Crea un mensaje de varias partes con texto e imagen y lee la respuesta en output_text; antes se puede probar todo en el área de pruebas, que ofrece código de ejemplo.

Implement AI solutions for information extraction by using Foundry

- Content Understanding procesa documentos con un analizador que combina extracción de contenido (OCR, tablas, secciones, marcas de selección, códigos de barras) y extracción de campos según un esquema, aplicado de forma semántica. Los campos usan los métodos extract, classify o generate, y la confianza y el anclaje se activan con estimateFieldSourceAndConfidence, contentCategories con enableSegment clasifica y divide archivos mixtos.
- Los analizadores de imágenes de Content Understanding convierten fotos en datos estructurados a partir de un esquema de campos, descripciones y tipos de salida, con usos como control de calidad o análisis de estanterías. Sus campos usan generate o classify, porque extract solo existe para documentos. Si el objetivo es sobre todo el texto, conviene un esquema de documentos: prebuilt-imageSearch genera una descripción para búsqueda.
- Los analizadores de audio transcriben en WebVTT con diarización, detección de rol, de idioma y transcripción multilingüe, y extraen campos como resumen o sentimiento; prebuilt-callCenter y prebuilt-audioSearch son los prebuilt. En vídeo, la extracción de contenido aporta transcripción, planos y fotogramas clave, y un modelo generativo añade campos y segmentación con enableSegment; prebuilt-videoSearch es el analizador RAG.
- Una aplicación ligera usa el SDK azure-ai-contentunderstanding: crea un ContentUnderstandingClient con el punto de conexión del recurso de Foundry y una credencial, y llama a begin_analyze con un analizador y sus entradas. El análisis es asíncrono: el poller espera el resultado, con markdown y campos con confianza; en REST se sondea Operation-Location. Los analizadores personalizados se crean con PUT.