

***Programa de Estudios
del Probador Certificado
Nivel Especialista en
Pruebas con IA Generativa (CT-GenAI)***

v1.0

International Software Testing Qualifications Board



Aviso de Derechos de Autor

Aviso de derechos de autor del © International Software Testing Qualifications Board (en adelante, ISTQB®)
ISTQB® es una marca registrada del International Software Testing Qualifications Board.

Copyright © 2025, los autores Abbas Ahmad, Gualtiero Bazzana, Alessandro Collino, Olivier Denoo y Bruno Legeard.

Todos los derechos reservados. Por la presente, los autores transfieren los derechos de autor al ISTQB®.
Los autores (como actuales titulares de los derechos de autor) e ISTQB® (como futuro titular de los derechos de autor) han aceptado las siguientes condiciones de uso:

Queda autorizada la reproducción parcial de este contenido para uso privado, sujeta al reconocimiento explícito de su procedencia.

Cualquier Proveedor de Formación Acreditado puede usar este programa de estudios como base para un curso de formación si los autores y el ISTQB® son reconocidos como la fuente y propietarios de los derechos de autor del programa de estudios y siempre que cualquier anuncio de dicho curso de formación pueda mencionar el programa de estudios solo después de que se haya recibido la Acreditación oficial de los materiales de formación de un Comité Miembro reconocido por ISTQB®.

Cualquier persona o grupo de personas puede utilizar este programa de estudios como base para artículos y libros, si los autores y el ISTQB® son reconocidos como la fuente y los propietarios de los derechos de autor del programa de estudios.

Cualquier otro uso de este programa de estudios está prohibido sin obtener primero la aprobación por escrito del ISTQB®.

Cualquier Comité de Miembros reconocido por ISTQB® puede traducir este programa de estudios siempre que reproduzca el Aviso de Derechos de Autor mencionado anteriormente en la versión traducida del programa de estudios.

Historial de revisión

Versión	Fecha	Observaciones
v1.0	25/07/2025	Entrega CT-GenAI v1.0
v1.0	22/03/2026	Entrega CT-GenAI v1.0 Español de España

Tabla de Contenidos

DERECHOS DE AUTOR

Historial de revisión

Índice

Agradecimientos

0	Introducción.....	8
0.1	Propósito de este programa de estudios.....	8
0.2	Pruebas de software con la IA generativa	8
0.3	Trayectoria profesional para los probadores.....	8
0.4	Resultados de negocio.....	8
0.5	Objetivos de aprendizaje evaluables, objetivos prácticos y nivel cognitivo de conocimiento.....	9
0.6	El examen de certificación del probador certificado en pruebas con la IA generativa.....	10
0.7	Acreditación.....	10
0.8	Tratamiento de los estándares.....	10
0.9	Nivel de detalle.....	10
0.10	Cómo se organiza este programa de estudios.....	11
1	Introducción a la IA generativa para las pruebas de software.....	13
1.1	Fundamentos de la IA generativa y los conceptos clave	14
1.1.1	Espectro de la IA: IA simbólica, aprendizaje automático clásico, aprendizaje profundo e IA generativa.....	14
1.1.2	Conceptos básicos de la IA generativa y los grandes modelos de lenguaje (MLG).....	14
1.1.3	MLG fundacionales, ajustados por instrucciones y de razonamiento.....	16
1.1.4	Los MLG multimodales y modelos de visión y lenguaje	16
1.2	Aprovechamiento de la IA generativa en las pruebas de software: Principios fundamentales.....	17
1.2.1	Capacidades clave de los MLG para las tareas de prueba.....	17
1.2.2	Chatbots con IA y aplicaciones de prueba impulsadas por los MLG para las pruebas de software.....	18
2	Ingeniería de instrucciones para las pruebas de software efectivas.....	19
2.1	Desarrollo eficaz de instrucciones.....	21
2.1.1	Estructura de las instrucciones para la IA generativa en las pruebas de software.....	21
2.1.2	Técnicas fundamentales de instrucción para las pruebas de software.....	22

2.1.3	Instrucción de sistema y instrucción de usuario.....	23
2.2	Aplicación de técnicas de ingeniería de instrucciones a las tareas de prueba de software	24
2.2.1	Análisis de prueba con la IA generativa.....	24
2.2.2	Diseño de pruebas e implementación de pruebas con la IA generativa.....	25
2.2.3	Pruebas de regresión automatizadas con la IA generativa	27
2.2.4	Monitorización de pruebas y control de pruebas con la IA generativa	28
2.2.5	Elección de técnicas de instrucciones para las pruebas de software.....	30
2.3	Evaluar los resultados de la IA generativa y afinar las instrucciones para las tareas de prueba de software.....	30
2.3.1	Métricas para evaluar los resultados de la IA generativa en las tareas de prueba.....	31
2.3.2	Técnicas para evaluar y refinar iterativamente las instrucciones	32
3	Gestión de riesgos de la IA generativa en las pruebas de software.....	33
3.1	Alucinaciones, errores de razonamiento y sesgos	34
3.1.1	Alucinaciones, errores de razonamiento y sesgos en la IA generativa.....	35
3.1.2	Identificación de alucinaciones, errores de razonamiento y sesgos en los resultados de los MLG.....	35
3.1.3	Técnicas de mitigación de las alucinaciones de la IA generativa, los errores de razonamiento y los sesgos en las tareas de prueba de software.....	37
3.1.4	Mitigación del comportamiento no determinista de los MLG.....	37
3.2	Privacidad de datos y riesgos de seguridad de la IA generativa en las pruebas de software.....	37
3.2.1	Riesgos de privacidad y seguridad de datos asociados con el uso de la IA generativa.....	38
3.2.2	Privacidad de datos y vulnerabilidades en la IA generativa para los procesos y las herramientas de prueba.....	39
3.2.3	Estrategias de mitigación para proteger la privacidad de los datos y mejorar la seguridad en las pruebas con la IA generativa.....	39
3.3	Consumo de energía e impacto ambiental de la IA generativa en las pruebas de software.....	39
3.3.1	El impacto del uso de la IA generativa en el consumo de energía y las emisiones de CO ₂	41
3.4	Regulaciones, estándares y marcos de trabajo de las mejores prácticas de la IA.....	41
3.4.1	Regulaciones, estándares y marcos de trabajo de la IA relevantes para la IA generativa en las pruebas de software.....	42
4	Infraestructura de pruebas impulsada por los MLG para las pruebas de software.....	43
4.1	Enfoques arquitectónicos para la infraestructura de pruebas impulsada por los MLG.....	43
4.1.1	Componentes arquitectónicos clave y conceptos de la infraestructura de pruebas impulsada por los MLG.....	44
4.1.2	Generación aumentada por recuperación (GAR).....	45
	El papel de los agentes impulsados por los MLG en la automatización de los procesos de prueba.....	46
4.2	Ajuste fino y las operaciones de grandes modelos de lenguaje (OpsMLG): puesta en marcha de la IA generativa para las pruebas de software.....	45

4.2.1	Ajuste fino de los MLG para las tareas de prueba.....	47
4.2.2	Las OpsMLG al desplegar y gestionar los MLG para las pruebas de software.....	48
5	Despliegue e integración de la IA generativa en las organizaciones de prueba.....	49
5.1	Hoja de ruta para la adopción de la IA generativa en las pruebas de software.....	49
5.1.1	Riesgos de la IA en la sombra (shadow AI)	50
5.1.2	Aspectos clave de una estrategia de la IA generativa en las pruebas de software.....	50
5.1.3	Selección de los MLG o modelos de lenguaje pequeños (MLP) para las tareas de las pruebas de software.....	51
5.1.4	Fases al adoptar la IA generativa en las pruebas de software	51
5.2	Gestionar el cambio al adoptar la IA generativa para las pruebas de software.....	51
5.2.1	Habilidades y conocimientos esenciales para realizar las pruebas con la IA generativa.....	52
5.2.2	Desarrollo de capacidades en la IA generativa en los equipos de prueba.....	52
5.2.3	Evolución de los procesos de prueba en las organizaciones con pruebas potenciadas por la IA.....	53
6	Referencias	
	Estándares.....	54
	Documentos ISTQB®	53
	Referencias del glosario	53
	Libros.....	53
	Artículos.....	53
	Páginas Web	54
7	Anexo A – Objetivos de aprendizaje o nivel cognitivo de conocimiento.....	56
	Nivel 1: Recordar (K1).....	55
	Nivel 2: Comprender (K2).....	55
	Nivel 3: Aplicar (K3)	56
8	Anexo B – Matriz de trazabilidad de los resultados de negocio con los objetivos de aprendizaje.....	58
9	Anexo C – Notas de entrega.....	64
10	Anexo D – Términos específicos de la IA generativa.....	65
11	Anexo E – Marcas comerciales.....	68
12	Índice.....	69

Agradecimientos

Este documento fue publicado formalmente por la Asamblea General del ISTQB® el 25/07/2025.

Fue producido por un equipo del International Software Testing Qualifications Board: Abbas Ahmad (propietario de producto), Gualtiero Bazzana, Alessandro Collino, Olivier Denoo y Bruno Legeard (director técnico).

El equipo agradece a Anne Kramer, Jędrzej Kwapinski, Samuel Ouko e Ina Schieferdecker por su revisión técnica y al equipo de revisión y a los Comités Miembros por sus sugerencias y aportaciones.

Las siguientes personas participaron en la revisión, comentarios y votación de este programa de estudios:

Albert Laura, Aneta Derkova, Anne Kramer, Arda Ender Torçuk, Baris Sarialioglu, Claire Van Der Meulen, Daniel van der Zwan, Derek Young, Dietmar Gehring, Francisca Cano Ortiz, Gary Mogyorodi, Gergely Ágnes, Horst Pohlmann, Ina Schieferdecker, Ingar Nordström, Jan Sabak, Jaroslaw Hryszko, Jędrzej Kwapinski, Joanna Kazun, Karol Frühaufer, Katalin Balla, Koray Yitmen, Laura Albert, Linda Vreeswijk, Lucjan Stapp, Lukáš Piška, Mario Winter, Marton Siska, Mattijs Kemmink, Matthias Hamburg, Meile Posthuma, Michael Stahl, Márton Siska, Nele Van Asch, Nils Röttger, Nishan Portoyan, Piet de Roo, Piotr Wicherski, Péter Földházi, Péter Sótér, Radoslaw Smilgin, Ralf Pichler, Renzo Cerquozzi, Rik Marselis, Samuel Ouko, Stephanie Ulrich, Stuart Reid, Tal Pe'er, Tamás Gergely, Thomas Letzkus, Wim Decoutere, Zsolt Hargitai, Mark Rutz, Patrick Quilter, Earl Burba, Taz Daughtrey, Judy McKay, Randall Rice, Thomas Adams, Tom Van Ongeval, Sander Mol, Miroslav Renda, Geng Chen, Chai Afeng, Xinghan Li, Klaudia Dussa-Zieger, Arnd Pehl, Florian Fieber, Ray Gillespie, József Kreis, Dénes Medzihradzky, Ferenc Hamori, Giorgio Pisani, Giancarlo Tomasi, Young jae Choi, Arnika Hryszko, Andrei Brovko, Iliia kulakov, Praveen, Kostas Pashalidis, Ferdinand Gramsamer, A. Berfin Öztaş, Abdullah Gök, Abdurrahman AKIN, Aleyna Zuhail İŞIK, Anil Şahin, Atakan Erdemgil, Aysel Bilici, Azmi YÜKSEL, Bilal Gelik, Bilge Yazıcı, Burak Gel, Burcu ÖZEL, Büşra İlayda Çevik Köken, Can Polat, Canan Ayten Dörtkol (Polat), Cansu Mercan Daldaban, Denizcan Orhun Karaca, Didem Çiçek Bay, Duygu Yalçınkaya, Efe Can Yemez, ELIF CERAV, Emine Tekiner, Emre Aman, Emre Can Akgül, Esra Küçük, Gençay GENÇ, Gül Çalışır Açı, Gül Nihal SİNGİL, Güler GÖK, Gulhanim Anulur, Hakan GÜVEZ, Haktan Bilgehan Dilber, Halil Ibrahim Tasdemir, Hasan Küçükayar, Hatice Erdoğan, Hatice Kübra Daşdoğan, Hüseyin Sevki ARI, Hyulya Gyuler, İLKNUR NEŞE TUNCAL, Kaan Eminçulu, Kamil Isik, Koray Danışman, Melisa Canbaz, Merve Guleroglu, Müjde CEYLAN, Mustafa Furkan CEYLAN, Nergiz Gençaslan, Nuh Soner Bozkurt, Omer Fatih Poyraz, Onurso Ery, Özlem Körpe, Özgür Öz Ödemzir, Sedat YOLTAY, Selahtinatıgazatıçlı, Seiko Seçli, Sebastian Sevyka Malysk, Ökire, Tayliana Ögian.

0 Introducción

0.1 Propósito de este programa de estudios

Este programa de estudios constituye la base para el Probador Certificado ISTQB®, Nivel Especialista en Pruebas con IA Generativa (CT-GenAI). El ISTQB® proporciona este programa de estudios de la siguiente manera:

1. A los comités miembro, para traducir a su idioma local y acreditar a los proveedores de formación. Los comités miembros pueden adaptar el programa de estudios a sus necesidades lingüísticas particulares y modificar las referencias para adaptarlas a sus publicaciones locales.
2. A los organismos de certificación, para derivar preguntas de examen en su idioma local adaptadas a los objetivos de aprendizaje de este programa de estudios.
3. A los proveedores de formación, para producir material didáctico y determinar los métodos de enseñanza apropiados.
4. A los candidatos de la certificación, para prepararse para el examen de certificación (ya sea como parte de un curso de formación o de forma independiente).
5. A la comunidad internacional de ingeniería de software y sistemas, para avanzar en la profesión de pruebas de software y sistemas, y como base para libros y artículos.

0.2 Pruebas de software con la IA generativa

La certificación pruebas con la IA generativa está dirigida a cualquier persona involucrada en el uso de la IA generativa (GenAI) para las pruebas de software. Esto incluye a personas en roles como probadores, analistas de pruebas, ingenieros de automatización de pruebas, gestores de pruebas, probadores de aceptación de usuarios y desarrolladores de software. Esta certificación de Pruebas con la IA generativa también es adecuada para cualquier persona que desee una comprensión básica del uso de la IA generativa para las pruebas de software, como directores de proyectos, directores de calidad, directores de desarrollo de software, analistas de negocios, directores de TI y consultores de gestión.

0.3 Trayectoria profesional para los probadores

El programa de certificación ISTQB® ofrece apoyo a los profesionales de pruebas en todas las etapas de sus carreras, ofreciendo amplitud y profundidad de conocimientos. Las personas que obtienen la certificación ISTQB® pruebas con la IA generativa también pueden estar interesadas en los niveles avanzados centrales (Analista de Pruebas, Analista de Pruebas Técnicas, Gestor de Pruebas e Ingeniería de Pruebas) y, posteriormente, en el Nivel Experto (Gestión de Pruebas o Mejora del Proceso de Prueba). Visite www.istqb.org para obtener la información más reciente del Programa de Certificación de Probadores ISTQB.

0.4 Resultados de negocio

Esta sección enumera los resultados de negocio que se esperan de un candidato que ha superado el examen con la certificación de IA generativa.

Un candidato que haya obtenido la certificación de Pruebas con la IA generativa puede:

GenAI-B1	Comprender los conceptos fundamentales, las capacidades y las limitaciones de la IA generativa
GenAI-B2	Desarrollar habilidades prácticas para impulsar grandes modelos de lenguaje para las pruebas de software
GenAI-B3	Adquirir información sobre los riesgos y las mitigaciones del uso de la IA generativa para las pruebas de software
GenAI-B4	Adquirir información sobre las aplicaciones de las soluciones de la IA generativa para las pruebas de software
GenAI-B5	Contribuir de manera efectiva a la definición e implementación de una estrategia de la IA generativa y una hoja de ruta para las pruebas de software dentro de una organización

0.5 Objetivos de aprendizaje evaluables, objetivos prácticos y nivel cognitivo de conocimiento

Los objetivos de aprendizaje y prácticos respaldan los resultados de negocio y se utilizan para crear exámenes de certificación para las pruebas con la IA generativa.

En general, todos los contenidos de este programa de estudios son evaluables en los niveles K1, K2 y K3, a excepción de la Introducción, los objetivos prácticos y los anexos. Las preguntas del examen confirmarán el conocimiento de las palabras clave en el nivel K1 (ver a continuación) o los objetivos de aprendizaje en todos los niveles de K.

Los niveles específicos de los objetivos de aprendizaje se muestran al comienzo de cada capítulo y se clasifican de la siguiente manera:

- K1: Recordar
- K2: Comprender
- K3: Aplicar

En el Anexo A se proporcionan más detalles y ejemplos de los objetivos de aprendizaje.

Se recordarán todos los términos enumerados como palabras clave justo debajo de los encabezados de los capítulos, incluso si no se mencionan explícitamente en los objetivos de aprendizaje.

Los objetivos prácticos específicos (HO) se muestran al comienzo de cada capítulo. Cada HO está vinculado a un LO en el nivel K2 o K3, con el objetivo de perfeccionar el aprendizaje a través de la práctica. El nivel de un HO se clasifica de la siguiente manera:

- H0: Esto puede incluir una demostración en vivo de un ejercicio o un video grabado. Dado que esto no lo realiza el alumno, no es estrictamente un ejercicio.
- H1: Ejercicio guiado. Los alumnos siguen una secuencia de pasos realizados por el formador
- H2: Ejercicio con pistas. Al alumno se le da un ejercicio con consejos relevantes para permitir que el ejercicio a resolver en el plazo establecido.

0.6 El examen de certificación del probador certificado en pruebas con la IA generativa

El examen de certificación del probador certificado en pruebas con la IA generativa se basará en este programa de estudios. Las respuestas a las preguntas del examen pueden requerir el uso del material basado en más de una sección de este programa de estudios. Todas las secciones del programa de estudios son evaluables, excepto la Introducción, los Objetivos prácticos y los Anexos. Se incluyen como referencias estándares, libros y artículos, pero su contenido no es evaluable, más allá de lo que se resume en el propio temario.

Consulte el documento Estructuras y reglas del examen v1.0 para obtener más detalles sobre los exámenes de las pruebas con la IA generativa.

Nota de requisitos del examen: El certificado del nivel fundamentos de ISTQB® se obtendrá antes de presentarse al examen de certificación ISTQB® Certified Tester Testing with Generative AI.

0.7 Acreditación

Un Comité Miembro de ISTQB® puede acreditar a los proveedores de formación cuyo material del curso sigue este programa de estudios. Los proveedores de formación deben obtener directrices de acreditación del Comité Miembro u organismo que realiza la acreditación. Se reconoce que un curso acreditado cumple con este programa de estudios y se le permite tener un examen ISTQB® como parte del curso.

Las directrices de acreditación para este programa de estudios se definen en el documento de Directrices de Acreditación ISTQB CT-GenAI.

0.8 Tratamiento de los estándares

Existen estándares asociados con las características de calidad y las pruebas de software, a saber, los referenciados en el programa de estudios del Nivel Fundamentos, como por IEEE e ISO. El propósito de estas referencias es proporcionar un marco o proporcionar una fuente de información adicional si el lector lo desea. Tenga en cuenta que los programas de estudio utilizan los documentos estándar como referencia. Los documentos de estándares no están destinados a ser evaluados. Consulte el Capítulo 6 para obtener más información sobre los Estándares.

0.9 Nivel de detalle

El nivel de detalle en este programa de estudios permite cursos y exámenes internacionalmente consistentes. Con el objetivo de lograr esta meta el programa de estudios consiste de:

- Objetivos de instrucción general que describen la intención de la certificación de pruebas con la IA generativa.
- Objetivos de aprendizaje para cada área de conocimiento, describiendo el resultado de aprendizaje cognitivo a alcanzar.
- Una descripción de los conceptos clave, incluidas las referencias a fuentes como la literatura o los estándares aceptados
- Una descripción para cada objetivo práctico de la práctica recomendada para apoyar el aprendizaje

El contenido del programa de estudios no es una descripción de toda el área de conocimiento de las pruebas con la IA generativa; refleja el nivel de detalle que se cubrirá en los cursos de formación de Probador Certificado ISTQB® para las Pruebas con IA Generativa. Se centra en los conceptos y técnicas de prueba que se pueden aplicar a todos los proyectos de software cuando se utiliza IA generativa para las pruebas.

El programa de estudios utiliza la terminología (es decir, el nombre y el significado) de los términos utilizados en las pruebas de software y el aseguramiento de la calidad de acuerdo con el Glosario ISTQB®.

0.10 Cómo se organiza este programa de estudios

Hay 5 capítulos con contenido evaluable. El título principal de cada capítulo especifica la duración del mismo; la duración no se proporciona por debajo del nivel del capítulo. Para los cursos de formación acreditados, el programa de estudios requiere un mínimo de 13,6 horas de instrucción, distribuidas en los 5 capítulos de la siguiente manera:

- Capítulo 1: 100 minutos Introducción a la IA generativa para las pruebas de software
 - El probador aprende los conceptos básicos de los grandes modelos de lenguaje (MLG), incluida la tokenización y las capacidades multimodales.
 - El probador explora las aplicaciones de la IA generativa (GenAI) en las pruebas de software, distinguiendo el chatbot con IA de las herramientas de prueba impulsadas por MLG y experimentando con la tokenización, las ventanas de contexto y las instrucciones multimodales.
- Capítulo 2: 365 minutos Ingeniería de instrucciones para las pruebas de software efectivas
 - El probador aprende a elaborar instrucciones efectivas y estructurados para la IA generativa en las pruebas de software.
 - El probador adquiere experiencia práctica con técnicas de ingeniería de instrucciones para las tareas de prueba de software y las aplica.
- Capítulo 3: 160 minutos Gestión de riesgos de la IA generativa en las pruebas de software
 - El probador aprende a identificar y mitigar las alucinaciones, los errores de razonamiento y los sesgos al realizar las pruebas con la IA generativa.
 - El probador aprende a abordar los problemas de privacidad y seguridad de datos de la IA generativa en las pruebas de software.
 - El probador aprende sobre el consumo de energía y el impacto ambiental de la IA generativa en las pruebas de software.
 - El probador aprende las regulaciones, los estándares y las mejores prácticas de IA para el uso ético, transparente y seguro de la IA generativa en las pruebas de software.
- Capítulo 4: 110 minutos Infraestructura de pruebas impulsada por los MLG para las pruebas de software
 - El probador explora la arquitectura de IA generativa como la generación aumentada por recuperación (GAR) y los agentes de IA generativa.
 - El probador aprende el proceso para ajustar los MLG para las tareas de prueba de software.

- El probador aprende los conceptos de operaciones de grandes modelos de lenguaje (OpsMLG) para desplegar y gestionar los MLG en las pruebas de software.
- Capítulo 5: 80 minutos Despliegue e integración de la IA generativa en las organizaciones de prueba
 - El probador aprende una hoja de ruta estructurada para integrar la IA generativa en los procesos de prueba.
 - El probador aprende la transformación organizacional para la integración de la IA generativa en los procesos de prueba.

1 Introducción a la IA generativa para las pruebas de software – 100 minutos

Palabras clave

Ninguna

Palabras clave específicas de IA generativa

Chatbot con IA, ventana de contexto, aprendizaje profundo, incrustación (incrustación), característica, MLG fundacional, IA generativa, transformador generativo preentrenado, MLG ajustado por instrucciones, gran modelo de lenguaje, aprendizaje automático, modelo multimodal, MLG de razonamiento, IA simbólica, tokenización, transformador

Objetivos de aprendizaje y objetivos prácticos para el Capítulo 1:

1.1 Fundamentos de la IA generativa y los conceptos clave

- | | | |
|-------------|------|---|
| GenAI-1.1.1 | (K1) | Recordar diferentes tipos de IA: IA simbólica, aprendizaje automático clásico, aprendizaje profundo e IA generativa |
| GenAI-1.1.2 | (K2) | Explicar los conceptos básicos de la IA generativa y los grandes modelos de lenguaje |
| HO-1.1.2 | (H1) | Practicar la tokenización y la evaluación del recuento de tokens cuando se utiliza un MLG para una tarea de prueba de software |
| GenAI-1.1.3 | (K2) | Distinguir entre los MLG fundacionales, ajustados por instrucciones y de razonamiento |
| GenAI-1.1.4 | (K2) | Resumir los principios fundamentales de los MLG multimodales y los modelos de visión y lenguaje |
| HO-1.1.4 | (H1) | Revisar y ejecutar una instrucción dada para un MLG multimodal utilizando entradas de texto e imágenes para una tarea de prueba de software |

1.2 Aprovechamiento de la IA generativa en las pruebas de software: Principios fundamentales

- | | | |
|-------------|------|---|
| GenAI-1.2.1 | (K2) | Dar ejemplos de capacidades clave de los MLG para las tareas de prueba |
| GenAI-1.2.2 | (K2) | Comparar los modelos de interacción al usar la IA generativa para las pruebas de software |

1.1 Fundamentos de la IA generativa y los conceptos clave

La inteligencia artificial generativa (GenAI) es una rama de la inteligencia artificial que utiliza modelos grandes y preentrenados para generar resultados similares a los humanos, como texto, imágenes o código. Los grandes modelos de lenguaje (MLG) son modelos de IA generativa que están preentrenados en grandes conjuntos de datos textuales, lo que les permite determinar el contexto y producir respuestas relevantes de acuerdo con las instrucciones de usuario.

Los conceptos clave incluyen la tokenización (es decir, dividir el texto en unidades para un procesamiento eficiente), las ventanas de contexto (limitar la cantidad de información considerada a la vez para mantener la relevancia) y los modelos multimodales (capaces de procesar múltiples tipos de datos, como texto, imágenes y audio para interacciones enriquecidas).

En las pruebas de software, estos MLG pueden respaldar tareas como revisar y mejorar los criterios de aceptación, generar casos de prueba o scripts de prueba, identificar defectos potenciales, analizar patrones de defectos, generar datos de prueba sintéticos o respaldar la generación de documentación a lo largo de todo el proceso de prueba.

1.1.1 Espectro de la IA: IA simbólica, aprendizaje automático clásico, aprendizaje profundo e IA generativa

La Inteligencia Artificial (IA) es un campo amplio que abarca diferentes tipos de tecnologías, cada una con su propia forma única de resolver problemas, como la IA simbólica, el aprendizaje automático clásico, el aprendizaje profundo y la IA generativa (entre otras tecnologías que están fuera del alcance de este programa):

- La IA simbólica utiliza un sistema basado en reglas para imitar la toma de decisiones humanas. Esencialmente, la IA simbólica representa el conocimiento utilizando símbolos y reglas lógicas.
- El aprendizaje automático clásico es un enfoque basado en datos que requiere la preparación de datos, selección de características y el entrenamiento de modelos, y se puede utilizar para tareas como la categorización de defectos y la predicción de problemas de software.
- El aprendizaje profundo utiliza estructuras de aprendizaje automático llamadas redes neuronales para aprender automáticamente las características de los datos. Los modelos de aprendizaje profundo pueden encontrar patrones en conjuntos de datos muy grandes y complejos, como imágenes, video, audio o texto, sin necesidad de que los usuarios definan manualmente las características, aunque en la práctica, aún puede requerir la participación humana en tareas como la anotación de datos, el ajuste fino de modelos o la validación de resultados.
- La IA generativa utiliza técnicas de aprendizaje profundo para crear nuevo contenido (texto, imágenes, código) aprendiendo e imitando patrones de sus datos de entrenamiento. Modelos como los MLG pueden generar texto, escribir código y simular razonamiento o resolución de problemas dentro del alcance de su entrenamiento.

En resumen, el campo de la IA ha evolucionado en varias direcciones, cada una con diferentes fortalezas y limitaciones. La ventaja clave de usar IA generativa para las pruebas de software es que utiliza modelos preentrenados que se pueden aplicar directamente a las tareas de prueba sin la necesidad de una fase de entrenamiento adicional, aunque esto conlleva algunos riesgos (consulte la Sección 3.1).

1.1.2 Conceptos básicos de la IA generativa y los grandes modelos de lenguaje (MLG)

Basado en el modelo generativo de aprendizaje profundo de transformadores preentrenados, los MLG son entrenados con conjuntos de datos muy grandes, incluidos libros, artículos y sitios web. Los modelos de lenguaje pequeños (MLP) son modelos compactos con menos parámetros en comparación con los grandes modelos de lenguaje, diseñados para proporcionar soluciones de IA generativa ligeras y enfocadas.

Los MLG pueden manejar los matices del lenguaje y generar contenido coherente. Dos conceptos clave que ayudan a los MLG a procesar y generar contenido son la tokenización y las incrustaciones (incrustaciones). La tokenización y las incrustaciones convierten el lenguaje en una forma numérica que el modelo puede procesar de manera efectiva.

- La tokenización en los modelos de lenguaje es el proceso de dividir el texto en unidades más pequeñas llamadas tokens. Los tokens pueden ser tan pequeños como un carácter o tan grandes como una subpalabra o palabra. Cuando un MLG procesa una oración, primero tokeniza la entrada para que cada token pueda entenderse individualmente, manteniendo el contexto general.
- Las incrustaciones son representaciones numéricas de tokens que codifican sus relaciones semánticas, sintácticas y contextuales en un formato adecuado para su procesamiento por modelos de la IA generativa. Cada token se transforma en un vector en un espacio de alta dimensión, capturando información matizada sobre su significado y uso. Los tokens con significados o roles contextuales similares tienen incrustaciones que se colocan muy juntas en este espacio. Esta proximidad permite a los MLG comprender las relaciones entre las palabras, retener el contexto y generar respuestas coherentes y contextualmente adecuadas.

Los MLG utilizan una arquitectura de red neuronal conocida como modelo de transformador. Los modelos de transformador sobresalen en las tareas de lenguaje al procesar el contexto de secuencias de texto extensas y aprender cómo los tokens se relacionan entre sí. Durante la inferencia, los MLG predicen el siguiente token en una secuencia, aprovechando estas relaciones aprendidas para generar texto coherente y contextualmente apropiado. El modelo de transformador se puede utilizar para generar un nuevo texto que sea estadísticamente plausible, basado en los datos de entrenamiento y la instrucción. Pero esto no es necesariamente correcto.

Los MLG exhiben un comportamiento no determinista principalmente debido a la naturaleza probabilística de sus mecanismos de inferencia y configuraciones de hiperparámetros. Esta aleatoriedad inherente puede conducir a variaciones en los resultados incluso cuando la misma entrada se proporciona varias veces.

En el ámbito de los MLG, la ventana de contexto se refiere a la cantidad de texto anterior, medido en tokens, que el modelo puede considerar al generar respuestas. Una ventana de contexto más grande permite que el modelo mantenga la coherencia en pasajes más largos, por ejemplo, al analizar registros de pruebas grandes. Sin embargo, aumentar el número de tokens en la ventana de contexto también aumenta la complejidad computacional y el tiempo de procesamiento necesarios para que el modelo funcione de manera efectiva.

Objetivo práctico HO-1.1.2 (H1): Practicar la tokenización y la evaluación del recuento de tokens

Esta actividad práctica está diseñada para ayudar a los alumnos a desarrollar una comprensión práctica de la tokenización y sus implicaciones cuando trabajen con los MLG. El ejercicio se dividió en dos partes:

- Tokenización: use un tokenizador para dividir un texto de muestra en tokens individuales. Examine el resultado para ver cómo se representan las palabras, la puntuación y las frases, e identifique patrones o matices en la tokenización.
- Evaluación del recuento de tokens: Mide la cantidad de tokens generados a partir de varios textos de entrada. Analizar cómo el recuento de tokens influye en el rendimiento del modelo, particularmente en relación con los límites de la ventana de contexto del modelo y las consideraciones de eficiencia.

Al final de este ejercicio, los alumnos podrán anticipar mejor cómo las diferentes estructuras de texto y longitudes de entrada pueden afectar a las interacciones con los MLG.

1.1.3 MLG fundacionales, ajustados por instrucciones y de razonamiento

Los grandes modelos de lenguaje se desarrollan a través de etapas de entrenamiento progresivamente especializadas para mejorar su efectividad en una amplia gama de tareas. Estas etapas dan lugar a tres categorías principales: MLG fundacionales, MLG ajustados por instrucciones y MLG de razonamiento.

- **Fundamentos de MLG:** Estos son modelos de propósito general entrenados en conjuntos de datos vastos y diversos que comprenden texto, código, imágenes y otras modalidades. Su amplio entrenamiento previo les permite apoyar a diversas tareas en dominios como el procesamiento del lenguaje natural, la visión artificial y el reconocimiento de voz. Si bien son potentes y flexibles, los modelos fundacionales generalmente requieren una mayor adaptación para cumplir con los requisitos específicos de la tarea.
- **MLG ajustados por instrucciones:** Derivados de los modelos fundacionales, los MLG ajustados por instrucciones se ajustan utilizando conjuntos de datos que combinan las instrucciones con las respuestas esperadas. Esta etapa mejora su alineación con las instrucciones humanas, mejorando la usabilidad en aplicaciones del mundo real. El proceso de ajuste fino implica la optimización de la adherencia a la tarea, el seguimiento de las instrucciones y la coherencia de la respuesta, mejorando así la capacidad del modelo para interpretar y actuar sobre la intención del usuario de manera efectiva.
- **MLG de razonamiento:** Los modelos de razonamiento amplían los modelos ajustados finamente por instrucciones al enfatizar las habilidades cognitivas estructuradas, como la inferencia lógica, la resolución de problemas de varios pasos y el razonamiento en cadena de pensamiento. Estos modelos están más entrenados o afinados en tareas cuidadosamente seleccionadas que exigen comprensión contextual, pasos de razonamiento intermedios y síntesis de información compleja. Como resultado, son más adecuados para tareas de alta carga cognitiva, incluidas aquellas en dominios técnicos.

En el contexto de las aplicaciones de IA generativa para las pruebas de software, se utilizan los MLG tanto por instrucción ajustada (a veces denominada “sin razonamiento”) como de razonamiento. La selección depende de la complejidad y las demandas de razonamiento de la tarea de prueba específica en cuestión.

1.1.4 Los MLG multimodales y modelos de visión y lenguaje

Los MLG multimodales amplían el modelo de transformador tradicional para procesar múltiples modalidades de datos, incluidos texto, imágenes, sonido y video. Estos modelos están entrenados en conjuntos de datos grandes y diversos que les permiten aprender las relaciones entre diferentes tipos de datos. Para manejar varias modalidades, la tokenización se adapta para cada tipo de datos, por ejemplo, las imágenes se convierten en incrustaciones utilizando modelos de lenguaje de visión antes de procesarse en el modelo de transformador.

Los modelos de visión y lenguaje, un subconjunto de los MLG multimodales, integran específicamente la información visual y textual para realizar tareas como subtítulos de imágenes, respuestas a preguntas visuales y el análisis de la coherencia entre la entrada textual y visual.

En las pruebas de software, los MLG multimodales, especialmente los MLG aumentados con modelos de visión y lenguaje, ofrecen oportunidades significativas. Pueden analizar elementos visuales de las aplicaciones, como las capturas de pantalla y los esquemas de interfaz gráfica del usuario, junto con descripciones textuales asociadas, como informes de defectos o historias de usuario. La capacidad permite a los probadores identificar discrepancias entre los resultados esperados y los elementos visuales reales en una captura de pantalla.

Además, los MLG aumentados con modelos de visión y lenguaje pueden generar casos de prueba ricos y realistas que incorporan datos textuales y señales visuales, lo que aumenta la cobertura general.

Objetivo práctico HO-1.1.4 (H1): Revisar y ejecutar una instrucción dada que aborde una tarea de prueba utilizando un modelo MLG multimodal

Este ejercicio implica revisar y ejecutar una instrucción dada para un MLG multimodal utilizando la entrada de texto e imagen para resolver una tarea de prueba en dos pasos:

- Revisar las entradas: Revise la instrucción y los datos de entrada (texto e imagen).
- Ejecutar la instrucción y verificar el resultado: Use un MLG multimodal para ingresar tanto la imagen como el texto y verifique la respuesta del MLG.

Este ejercicio demuestra cómo usar los MLG multimodales para una tarea que involucra la entrada de texto e imagen en casos de uso de pruebas de software, incluido el reconocimiento de los beneficios y los desafíos potenciales involucrados.

1.2 Aprovechamiento de la IA generativa en las pruebas de software: Principios fundamentales

La IA generativa proporciona capacidades transformadoras en diversas actividades de prueba. Los MLG son excelentes procesando el lenguaje natural y el código, generando texto y código coherentes, respondiendo preguntas, resumiendo información, traduciendo idiomas y analizando imágenes en un contexto multimodal.

Los profesionales de pruebas en todos los roles pueden aprovechar la IA generativa de dos maneras complementarias: a través de chatbots con la IA generativa que proporcionan respuestas instantáneas a las consultas, y a través de aplicaciones impulsadas por los MLG integradas en herramientas de prueba.

1.2.1 Capacidades clave de los MLG para las tareas de prueba

Los MLG pueden interpretar requisitos, especificaciones, capturas de pantalla, código, casos de prueba e informes de defectos, convirtiéndolos en herramientas para comprender y aclarar la información necesaria a lo largo del proceso de prueba y generar elementos del testware. A continuación se presentan algunas de las capacidades clave de los MLG relevantes para las pruebas de software:

- Análisis y mejora de requisitos: los MLG pueden ayudar a analizar los requisitos y otros elementos de la base de la prueba, identificando ambigüedades, inconsistencias o información faltante. Pueden generar preguntas significativas para ayudar a aclarar los requisitos durante las discusiones con las partes interesadas.
- Soporte de creación de casos de prueba: los MLG pueden ayudar a generar casos de prueba y sugerir objetivos de prueba basados en los requisitos del sistema, las historias de usuario o cualquier otro elemento de la base de prueba.
- Generación de oráculos de prueba: los MLG pueden ayudar a generar los resultados esperados.
- Generación de datos de prueba: los MLG pueden generar conjuntos de datos, establecer valores límite y crear diferentes combinaciones de datos de prueba.

- Soporte de la automatización de pruebas: los MLG pueden ayudar a generar scripts de prueba a partir de la descripción del caso de prueba y mejorar los scripts de prueba existentes al sugerir cambios e identificar las técnicas de diseño de prueba adecuadas.
- Análisis de los resultados de pruebas: los MLG pueden ayudar a analizar los resultados de las pruebas creando resúmenes y clasificando las anomalías en función de la severidad y la prioridad.
- Creación del testware: los MLG pueden ayudar a crear varios documentos, incluidos planes de prueba, informes de prueba e informes de defecto, y mantenerlos actualizados a medida que el proyecto evoluciona.

Estas capacidades demuestran cómo los MLG pueden afectar varios aspectos de las pruebas de software a través de todo el proceso de prueba.

1.2.2 Chatbots con IA y aplicaciones de prueba impulsadas por los MLG para las pruebas de software

Los chatbots con IA y las aplicaciones de prueba impulsadas por los MLG pueden ayudar a los probadores, aunque difieren en la funcionalidad, la flexibilidad y los enfoques de integración.

Los chatbots con IA proporcionan una interfaz conversacional fácil de usar que permite a los probadores comunicarse directamente con los MLG. Esta interacción en lenguaje natural permite a los probadores ingresar preguntas, comandos o instrucciones y recibir respuestas inmediatas y conscientes del contexto. A través de técnicas como el encadenamiento de instrucciones, los probadores pueden refinar iterativamente los resultados, lo que hace que los chatbots sean particularmente efectivos para tareas rutinarias, pruebas exploratorias e incluso la incorporación de nuevos probadores al proporcionar un acceso rápido a los conocimientos y las prácticas de prueba.

Estos chatbots con IA son especialmente beneficiosos en escenarios que requieren retroalimentación rápida, aclaración de conceptos de prueba o exploración dinámica de requisitos y posibles casos de prueba. Su interfaz intuitiva los hace accesibles incluso a las partes interesadas no técnicas, ampliando la base de usuarios potenciales y fomentando una adopción más amplia.

Las aplicaciones de prueba impulsadas por los MLG, por el contrario, implican la integración de las capacidades de los MLG a través de las API para realizar tareas de prueba bien definidas y, a menudo, automatizadas. Estas aplicaciones ofrecen una mayor personalización y escalabilidad, lo que permite a las organizaciones y los proveedores de herramientas integrar la IA generativa en los marcos de trabajo de prueba existentes. Esto permite la automatización de tareas repetitivas o complejas, como la generación de casos de prueba, el análisis de defectos o la síntesis de datos de prueba. En implementaciones más avanzadas, las organizaciones pueden crear agentes de IA diseñados específicamente para realizar ciertas funciones de prueba (consulte el Capítulo 4).

Independientemente de cómo interactúe el probador con los MLG, ya sea a través de chatbots o aplicaciones integradas alimentadas por los MLG, la implementación exitosa de la IA generativa en las pruebas requiere una sólida ingeniería de instrucciones (consulte el Capítulo 2). Las instrucciones cuidadosamente diseñadas y las instrucciones claras y específicas son esenciales para garantizar que los resultados generados por los MLG sean precisos, relevantes y estén alineados con los objetivos de las pruebas. Esta práctica ayuda a maximizar el valor derivado de la IA generativa y garantiza un soporte consistente y fiable para una amplia gama de actividades de prueba.

2 Ingeniería de instrucciones para las pruebas de software efectivas – 365 minutos

Palabras clave

criterios de aceptación, guión de prueba, caso de prueba, condición de prueba, datos de prueba, diseño de prueba, informe de prueba

Palabras clave específicas de IA generativa

formulación de instrucciones con pocos ejemplos, metainstrucción, procesamiento del lenguaje natural, formulación de instrucciones con un ejemplo, instrucción, encadenamiento de instrucciones, ingeniería de instrucciones, instrucción de sistema, instrucción de usuario, formulación de instrucciones sin ejemplos

Objetivos de aprendizaje y objetivos prácticos para el Capítulo 2:

2.1 Desarrollo eficaz de instrucciones

- | | | |
|-------------|------|--|
| GenAI-2.1.1 | (K2) | Dar ejemplos de la estructura de las instrucciones utilizadas en la IA generativa para las pruebas de software |
| HO-2.1.1 | (H0) | Observar varias instrucciones dadas para las tareas de prueba de software, identificando los componentes de rol, contexto, instrucciones, datos de entrada, restricciones y formato de salida en cada una de ellas |
| GenAI-2.1.2 | (K2) | Diferenciar las técnicas fundamentales de instrucciones para las pruebas de software |
| HO-2.1.2a | (H0) | Observar las demostraciones de encadenamiento de instrucciones, formulación de instrucciones con pocos ejemplos y metainstrucción aplicados a las tareas de prueba de software |
| HO-2.1.2b | (H1) | Identificar qué técnicas de ingeniería de instrucciones se están utilizando en los ejemplos dados |
| GenAI-2.1.3 | (K2) | Distinguir entre las instrucciones de sistema y las instrucciones de usuario |

2.2 Aplicación de técnicas de ingeniería de instrucciones a las tareas de prueba de software

- | | | |
|-------------|------|---|
| GenAI-2.2.1 | (K3) | Aplicar la IA generativa a las tareas de análisis de prueba |
| HO-2.2.1a | (H2) | Practicar las instrucciones multimodales para generar criterios de aceptación para una historia de usuario basada en un esquema de página de la interfaz gráfica de usuario |
| HO-2.2.1b | (H2) | Practicar el encadenamiento de instrucciones y la verificación humana para analizar progresivamente una historia de usuario determinada y refinar los criterios de aceptación |
| GenAI-2.2.2 | (K3) | Aplicar la IA generativa a las tareas del diseño de pruebas e implementación de pruebas |
| HO-2.2.2a | (H2) | Practicar la generación de casos de prueba funcionales a partir de historias de usuario con la IA generativa mediante el encadenamiento de instrucciones, instrucciones estructuradas y metainstrucción |
| HO-2.2.2b | (H2) | Utilizar la técnica de instrucciones con pocos ejemplos para generar casos de prueba al estilo Gherkin a partir de historias de usuario determinadas |
| HO-2.2.2c | (H2) | Utilizar el encadenamiento de instrucciones para priorizar los casos de prueba dentro de un conjunto de pruebas determinado, teniendo en cuenta sus prioridades y dependencias específicas |
| GenAI-2.2.3 | (K3) | Aplicar la IA generativa a las pruebas de regresión automatizadas |

HO-2.2.3a	(H2)	Practicar la formulación de instrucciones con pocos ejemplos para crear y gestionar los scripts de prueba basados en palabras clave
HO-2.2.3b	(H2)	Practicar la ingeniería estructurada de instrucciones para el análisis del informe de prueba
GenAI-2.2.4	(K3)	Aplicar la IA generativa a las tareas de monitorización de pruebas y control de pruebas
HO-2.2.4	(H0)	Observar las métricas de monitorización de las pruebas preparadas por la IA generativa a
GenAI-2.2.5		partir de los datos de prueba
HO-2.2.5	(K3)	Seleccionar y aplicar las técnicas de instrucciones adecuadas para un contexto dado y una tarea de prueba
	(H1)	Seleccionar y aplicar las técnicas de instrucciones adecuadas al contexto para una tarea de prueba determinada

2.3 Evaluar los resultados de la IA generativa y afinar las instrucciones para las tareas de prueba de software

GenAI-2.3.1	(K2)	Resumir las métricas para evaluar los resultados de la IA generativa en las tareas de prueba
HO-2.3.1	(H0)	Observar cómo se pueden usar las métricas para evaluar el resultado de la IA generativa en una tarea de prueba
GenAI-2.3.2	(K2)	Dar ejemplos de técnicas para evaluar y refinar iterativamente las instrucciones
HO-2.3.2	(H1)	Evaluar y optimizar una instrucción para una tarea de prueba determinada

2.1 Desarrollo eficaz de instrucciones

El diseño eficaz de instrucciones garantiza que las herramientas de IA generativa realicen las tareas de prueba de software de manera precisa y eficiente y que los probadores obtengan resultados útiles de los MLG. Una instrucción estructurada incluye diferentes componentes (consulte la sección 2.1.1). Cada uno de estos componentes contribuye a la claridad y precisión de una instrucción que comunica eficazmente los requisitos y las expectativas a los MLG.

Varias técnicas de ingeniería de instrucciones mejoran la efectividad de las instrucciones en las pruebas de software. Las técnicas como el encadenamiento de instrucciones, formulación de instrucciones con pocos ejemplos y metainstrucción ayudan a abordar los desafíos complejos de las pruebas (consulte la sección 2.1.2).

La combinación de instrucciones estructuradas (ver sección 2.1.1) con técnicas fundamentales de instrucciones tiene como objetivo lograr buenos resultados al consultar a un MLG para las tareas de prueba de software (ver sección 2.1.3).

2.1.1 Estructura de las instrucciones para la IA generativa en las pruebas de software

Una instrucción estructurada para las pruebas de software generalmente incluye seis componentes:

- **Rol:** El rol define la perspectiva o persona que debe tomar el modelo de la IA generativa al generar una respuesta. Especificar el rol ayuda al MLG a determinar sus responsabilidades y adoptar un tono o enfoque apropiado, como actuar como probador, gestor de pruebas o ingeniero de automatización de pruebas.
- **Contexto:** El contexto proporciona la información de fondo que el modelo de la IA generativa necesita para determinar las condiciones de la prueba. Esto incluye detalles sobre el objeto de prueba, la funcionalidad específica a probar y cualquier información contextual relevante.
- **Instrucciones:** Las instrucciones son directivas dadas a la IA generativa que describen la tarea específica a realizar. Las instrucciones claras, imperativas y concisas incluyen una descripción de la tarea y cualquier requisito relevante para la tarea.
- **Datos de entrada:** los datos de entrada incluyen cualquier información necesaria para realizar la tarea, como historias de usuario, criterios de aceptación, capturas de pantalla, código, casos de prueba existentes o ejemplos de salida. Proporcionar datos de entrada detallados y estructurados ayuda al MLG a generar resultados más precisos y sensibles al contexto.
- **Restricciones:** las restricciones describen cualquier restricción o consideración especial que el MLG debe cumplir. Las restricciones ayudan a especificar cómo se deben aplicar las instrucciones a los datos de entrada.
- **Formato de salida:** Las especificaciones de salida denotan el formato, la estructura o las características esperadas de la respuesta. Estos indicadores ayudan a dar forma a la salida del MLG.

Estos componentes forman la estructura básica de la instrucción. Esta estructura debe combinarse con la implementación de técnicas de instrucciones (ver Sección 2.1.2), dependiendo de la tarea a realizar y el MLG que se utilizará.

Objetivo práctico HO-2.1.1 (H0): Observar y analizar los componentes de una instrucción

En una demostración, se experimentan varias instrucciones estructuradas en un chatbot con IA, cada una adaptada a tareas específicas de pruebas de software. Estas instrucciones siguen un formato estructurado que consta de seis componentes clave: rol, contexto, instrucciones, datos de entrada, restricciones y formato de salida. La demostración tiene como objetivo facilitar la observación y el análisis de estas instrucciones estructuradas, destacando cómo cada componente contribuye a proporcionar la información precisa, relevante y procesable a un MLG utilizado para una tarea de pruebas de software.

2.1.2 Técnicas fundamentales de instrucción para las pruebas de software

En los últimos años, se han propuesto muchas técnicas de instrucciones de los MLG para diferentes casos de uso de la IA generativa (Schulhoff 2024). Entre estas, tres técnicas fundamentales de instrucciones se utilizan comúnmente para las tareas de prueba con IA generativa junto con la estructura de instrucciones de 6 componentes descrita anteriormente (consulte la sección 2.1.1): encadenamiento de instrucciones, instrucciones con pocos ejemplos y metainstrucción.

- El encadenamiento de instrucciones implica dividir una tarea en una serie de pasos intermedios (instrucciones múltiples). El resultado de cada paso se comprueba y refina manual o automáticamente antes de pasar al siguiente paso. Este enfoque conduce a una mayor precisión a medida que cada respuesta informa a la siguiente instrucción. El encadenamiento de instrucciones es particularmente útil en procesos de prueba donde las tareas son complicadas y requieren descomposición en subtareas y verificación sistemática de los resultados intermedios del MLG. También permite interacciones dinámicas en los procesos de prueba.
- La formulación de instrucciones con pocos ejemplos consiste en proporcionarle ejemplos al MLG dentro de la instrucción. Si bien la instrucción sin ejemplos se basa en el conocimiento preexistente del modelo para generar una respuesta, la instrucción con un solo ejemplo proporciona un ejemplo para demostrar el resultado deseado para una entrada dada. Las instrucciones con pocos ejemplos contienen más de un ejemplo (unos pocos) para consolidar aún más el comportamiento de respuesta deseado del modelo.

Esta técnica ayuda a guiar el modelo al proporcionar una referencia clara y garantizar que los resultados sean consistentes y estén en línea con las expectativas. Las instrucciones con pocos ejemplos son particularmente efectivos para tareas en las que los ejemplos pueden ilustrar el comportamiento requerido, lo que permite que el modelo se generalice de manera efectiva y produzca resultados fiables.

- La metainstrucción aprovecha la capacidad de la IA para generar o refinar sus propias instrucciones. En un ciclo iterativo, el MLG puede generar instrucciones que pueden ser evaluadas y refinadas por el probador. Este enfoque optimiza la calidad de las instrucciones aprovechando el conocimiento de los MLG sobre las instrucciones optimizadas. La metainstrucción es especialmente beneficiosa cuando la eficiencia y la optimización rápida son críticas, ya que reducen el esfuerzo manual requerido para diseñar instrucciones efectivas. Otra ventaja de la metainstrucción es que, si el probador no está seguro de cómo elaborar una instrucción efectiva, puede colaborar con el MLG para cocrearla. Esto refleja una forma de emparejamiento con la herramienta de IA generativa donde el probador y la IA trabajan juntos de forma interactiva para lograr un objetivo compartido. Este concepto de trabajo en pareja destaca una nueva forma de colaborar con las herramientas de IA, mejorando tanto la productividad como el aprendizaje no solo en la ingeniería de instrucciones, sino también en la programación en pareja y las pruebas en pareja.

Estas técnicas de instrucciones se pueden usar de manera efectiva en combinación para mejorar los resultados de los MLG (consulte la sección 2.2.5).

Objetivo práctico HO-2.1.2a (H0): Observar y discutir el encadenamiento de instrucciones, las instrucciones con pocos ejemplos y la metainstrucción en las tareas de prueba de software

Los participantes experimentarán con el encadenamiento de instrucciones, las instrucciones con pocos ejemplos y la metainstrucción en un chatbot con IA, cada una aplicada a tareas específicas de prueba de software.

La demostración tiene como objetivo explorar y discutir estas técnicas de instrucciones en el contexto de las pruebas de software, haciendo énfasis en cómo cada técnica contribuye a la precisión e integridad de los resultados del MLG.

Objetivo práctico HO-2.1.2b (H1): Identificar qué técnicas de ingeniería de instrucciones se están utilizando en ejemplos concretos

Los participantes leerán un conjunto de ejemplos de instrucciones relacionados con las pruebas de software para identificar las técnicas fundamentales de instrucciones aplicadas. La atención se centra en el reconocimiento de técnicas como el encadenamiento de instrucciones, formulación de instrucciones con pocos ejemplos y la metainstrucción, al tiempo que se destacan sus características distintivas y aplicaciones prácticas.

Esta actividad tiene como objetivo profundizar en la comprensión de los participantes de cómo las diferentes técnicas de instrucciones mejoran el uso eficaz de la IA generativa en las pruebas de software.

2.1.3 Instrucción de sistema y instrucción de usuario

Las instrucciones de sistema y las instrucciones de usuario tienen diferentes propósitos en las interacciones con los MLG, cada uno de los cuales desempeña un papel distinto en la configuración de la conversación. La instrucción de sistema suele ser definida por el desarrollador o probador, para guiar el comportamiento general del MLG, y no es visible o editable por el usuario del chatbot en la mayoría de las interfaces.

Una instrucción de sistema actúa como un conjunto de comandos predefinido que define el comportamiento, la personalidad y los parámetros operativos del MLG. Los parámetros operativos determinan cómo responde el MLG, por ejemplo, utilizando un tono formal, manteniendo las respuestas concisas, respetando las reglas específicas del dominio o evitando ciertos comportamientos. La instrucción de sistema establece las reglas para toda la conversación. Puede contener partes de una instrucción estructurada, como el rol, el contexto y las restricciones.

La instrucción de sistema se mantiene constante durante toda la sesión de interacción y establece el marco fundamental de cómo debe responder el MLG. Por ejemplo, una instrucción de sistema podría decir: "Eres un asistente profesional de pruebas de software. Responde siempre con claridad, usa un lenguaje formal y concéntrate en las prácticas alineadas con el ISTQB. Evita la especulación y cita los principios de prueba cuando sea relevante".

La instrucción de usuario, por otro lado, representa la entrada o pregunta real del usuario del chatbot. Cambia con cada interacción y puede incluir instrucciones, preguntas o tareas específicas que el usuario del chatbot desea que el MLG aborde. A diferencia de la instrucción de sistema, las instrucciones de usuario son directamente visibles y forman el contexto inmediato para cada respuesta. Por ejemplo, una instrucción de usuario podría ser: "Enumere las diferencias clave entre las pruebas de caja negra y caja blanca con ejemplos".

El uso típico implica configurar la instrucción de sistema una vez al comienzo de la conversación y luego enviar instrucciones sucesivas de usuario para cada interacción. El MLG genera respuestas considerando tanto la instrucción de sistema inmutable como la instrucción de usuario actual juntos. Para una implementación efectiva, las instrucciones de sistema deben ser claras y específicas sobre el rol del MLG y las posibles limitaciones. También puede contener contexto e instrucciones generales, por ejemplo, con respecto al resultado esperado.

Las instrucciones de usuario deben estar enfocadas y bien estructuradas, incluyendo instrucciones explícitas, así como instrucciones adicionales relevantes de contexto y formato de salida.

2.2 Aplicación de técnicas de ingeniería de instrucciones a las tareas de prueba de software

La aplicación de técnicas de ingeniería de instrucciones a las pruebas de software permite a la IA generativa apoyar tareas de prueba como el análisis de prueba, el diseño de prueba, la automatización de prueba, la priorización de casos de prueba, la detección de defectos, el análisis de cobertura y la monitorización de prueba y el control de prueba. Al usar y combinar las técnicas como el encadenamiento de instrucciones, formulación de instrucciones con pocos ejemplos y la metainstrucción, los equipos pueden adaptar las instrucciones de la IA a los objetivos específicos de la prueba, haciendo que los resultados sean más precisos, relevantes y efectivos. La información de alta calidad es crucial para obtener resultados significativos de la IA.

2.2.1 Análisis de prueba con la IA generativa

La IA generativa puede apoyar las tareas de análisis de pruebas generando y priorizando las condiciones de prueba, identificando defectos en la base de prueba y proporcionando análisis de cobertura. Los datos de entrada incluyen los requisitos, las historias de usuario, las especificaciones técnicas, los esquemas de página de la interfaz gráfica de usuario y otra información relevante. El resultado consiste en productos de trabajo del análisis de prueba típicos, como condiciones de prueba priorizadas (por ejemplo, criterios de aceptación).

Estas son algunas tareas típicas del análisis de pruebas que pueden ser asistidas por la IA generativa:

- **Identificar posibles defectos en la base de prueba:** la IA generativa puede ayudar a analizar la base de prueba en busca de inconsistencias, ambigüedades o información incompleta que pueda conducir a defectos. Al comparar patrones de requisitos similares o aplicar el conocimiento de informes de defectos anteriores, el MLG puede detectar posibles anomalías y sugerir mejoras.
- **Generación de condiciones de prueba basadas en la base de prueba,** por ejemplo, en los requisitos o historias de usuario: los MLG pueden analizar los requisitos y las historias de usuario para generar condiciones de prueba. Usando el procesamiento del lenguaje natural, pueden interpretar el significado de los requisitos y dividirlos en declaraciones medibles y comprobables. Esto puede ayudar a traducir los requisitos en condiciones de prueba específicas.
- **Priorizar las condiciones de prueba en función del nivel de riesgo:** con información sobre la probabilidad del riesgo y el impacto del riesgo del fallo para cada condición de prueba, un MLG puede ayudar a priorizar el esfuerzo de prueba. Al considerar aspectos como el cumplimiento regulatorio, las características orientadas al usuario (por ejemplo, la funcionalidad de inicio de sesión o el procesamiento de pagos) y los datos históricos de defectos, el MLG puede recomendar niveles de prioridad.
- **Soporte al análisis de cobertura:** al mapear los requisitos y las historias de usuario para probar las condiciones, un MLG puede realizar un análisis de cobertura para determinar si todos los aspectos de la base de prueba están cubiertos. Esto es particularmente útil para proyectos con requisitos complejos, donde las brechas en la cobertura pueden conducir a defectos escapados.
- **Sugerir técnicas de prueba:** la IA generativa puede sugerir técnicas de prueba relevantes (por ejemplo, el análisis de valores límite, la partición de equivalencia) en función del tipo de requisito o historia de usuario que se está probando. Esto puede ayudar a los probadores a aplicar las técnicas de prueba más efectivas para las condiciones de prueba específicas.

La calidad y relevancia de las entradas proporcionadas al MLG en relación con la tarea a completar afectan directamente la exactitud y precisión del resultado generado por el MLG.

Objetivo práctico HO-2.2.1a (H2): Practicar la creación de instrucciones multimodales estructurados para generar criterios de aceptación para una historia de usuario basada en un esquema de página de la interfaz gráfica de usuario

Este es un ejercicio para practicar la escritura de instrucciones estructuradas utilizando la entrada multimodal (texto e imagen). El objetivo es generar criterios de aceptación de alta calidad (es decir, bien formados, claros y completos) a partir de una historia de usuario y un esquema de la página de la interfaz gráfica de usuario. Se pueden agregar otros elementos de texto para proporcionar el contexto, como restricciones en los campos de entrada o reglas de negocio que se aplicarán al procesamiento de datos.

Los resultados obtenidos del MLG se comparan para evaluar el impacto de diferentes formulaciones de la instrucción estructurada (rol, contexto, instrucciones, datos de entrada de texto e imágenes, restricciones y formato de salida) para una tarea de análisis de prueba.

Este ejercicio proporciona experiencia práctica sobre la importancia de la estructuración de instrucciones, la contribución de instrucciones precisas y la importancia de los datos contextuales tanto textuales como de imagen para obtener resultados precisos y relevantes del MLG.

Objetivo práctico HO-2.2.1b (H2): Practicar el encadenamiento de instrucciones y la verificación humana para analizar progresivamente una historia de usuario determinada y refinar los criterios de aceptación

Este es un ejercicio para practicar el encadenamiento de instrucciones para analizar una historia de usuario determinada y refinar los criterios de aceptación, primero identificando ambigüedades, luego evaluando la capacidad de prueba y, finalmente, evaluando la integridad. Este ejercicio fomenta un enfoque paso a paso, refinando el análisis en cada paso para garantizar que los criterios de aceptación estén bien formados y sean procesables para lograr los objetivos de la prueba. En cada paso, los resultados proporcionados por el MLG se verifican y corrigen manualmente, si es necesario, ya sea ajustando la salida o mediante un proceso de encadenamiento de instrucciones con el MLG. De esta manera, la siguiente etapa utiliza un resultado limpio de la etapa anterior para abordar otro aspecto de la mejora de los criterios de aceptación.

Este ejercicio proporciona una experiencia práctica de los beneficios de dividir una tarea compleja en subtareas, con verificación humana de los resultados de cada etapa.

2.2.2 Diseño de pruebas e implementación de pruebas con la IA generativa

Como se describe en [ISTQB_CTFL_SYL], el diseño de pruebas implica la elaboración y el refinamiento de las condiciones de prueba, que luego se traducen en casos de prueba y otro testware. La implementación de las pruebas implica la creación o adquisición del testware necesario para realizar las pruebas.

Tanto las pruebas manuales como los scripts de prueba automatizados se pueden crear, priorizar y organizar dentro de un calendario de ejecución de pruebas con el apoyo de la IA generativa. La IA generativa puede apoyar significativamente este gran grupo de actividades de prueba al ayudar en la creación y evaluación de varios testware, incluidos los casos de prueba, los datos de prueba, los scripts de prueba y los entornos de prueba.

Estas son algunas tareas típicas de diseño e implementación de pruebas que pueden ser compatibles con la IA generativa:

- **Generación de casos de prueba:** el procesamiento del lenguaje natural permite a la IA generativa crear casos de prueba preliminares basados en requisitos funcionales y no funcionales. Cuando se le solicita la información adecuada, un MLG puede sugerir precondiciones e entradas de prueba, resultados esperados y criterios de cobertura, produciendo casos de prueba que cumplan con diferentes objetivos de prueba, desde la verificación funcional básica hasta las pruebas complejas de extremo a extremo.

- **Síntesis de datos de prueba:** la IA generativa puede crear datos de prueba sintéticos representativos que preserven la privacidad de los datos y que se asemejen a los datos de producción, cubriendo situaciones extremas y condiciones de prueba variadas. Estos datos de prueba sintéticos se pueden utilizar para las pruebas funcionales y no funcionales. Los datos de prueba generados por la IA se pueden adaptar a los requisitos de la aplicación, simulando escenarios realistas sin exponer información confidencial.
- **Generación automatizada de scripts de prueba:** la IA generativa puede generar procedimientos de prueba manuales y scripts de prueba automatizados a partir de casos de prueba estructurados, interpretando los pasos de prueba y traduciéndolos en código compatible con varios marcos de trabajo de automatización de pruebas. Estos scripts de prueba se pueden actualizar o ampliar en función de los nuevos requisitos.
- **Creación de calendario y priorización de la ejecución:** la IA generativa puede analizar casos de prueba y sus interdependencias, optimizando los calendarios de ejecución de pruebas en función de la prioridad, los riesgos asociados, la disponibilidad de recursos y los objetivos de la prueba.

Objetivo práctico HO-2.2.2a (H2): Practicar la generación de casos de prueba funcionales a partir de historias de usuario con la IA mediante el encadenamiento de instrucciones, las instrucciones estructuradas y la metainstrucción

Este ejercicio se centra en el desarrollo de casos de prueba funcionales a partir de historias de usuario con la IA generativa, utilizando el encadenamiento de instrucciones, las instrucciones estructuradas y las técnicas de metainstrucción para garantizar una cobertura completa. El primer paso es crear una instrucción que indique a la IA que genere casos de prueba funcionales basados en criterios de aceptación dados siguiendo un formato de salida específico. Un segundo paso es verificar la integridad de los casos de prueba generados. Aquí, la instrucción verifica que cada criterio de aceptación se cubra haciendo que la IA genere una tabla que resuma la cobertura. Finalmente, un tercer paso es crear una metainstrucción para ayudar en la creación de procedimientos de prueba de extremo a extremo. Este metainstrucción ayuda a refinar la instrucción para generar pruebas integrales de extremo a extremo, fomentando mejoras iterativas para maximizar la efectividad.

Este ejercicio mejora la comprensión del uso de los MLG para la generación de casos de prueba, la validación de la cobertura y las pruebas de extremo a extremo.

Objetivo práctico HO-2.2.2b (H2): Utilizar la técnica de instrucciones con pocos ejemplos para generar casos de prueba al estilo Gherkin a partir de historias de usuario determinadas

Este ejercicio se trata de usar instrucciones con pocos ejemplos para generar casos de prueba al estilo Gherkin a partir de historias de usuario determinadas. Comenzando con una revisión de ejemplos predefinidos y la sintaxis de Gherkin, el paso 1 es seleccionar n ejemplos para incluirlos en la instrucción, cada uno con una historia de usuario, condiciones de prueba y casos de prueba al estilo “dado-cuando-entonces” para modelar el resultado deseado. Esta instrucción se aplica a una nueva historia de usuario, generando escenarios de Gherkin que reflejan las condiciones de prueba originales. Si los resultados son inexactos, se debe refinar el mensaje o los ejemplos.

Este ejercicio ayuda a adquirir experiencia en la aplicación de técnicas de instrucciones con pocos ejemplos para el diseño realista de la prueba y las tareas de implementación de la prueba.

Objetivo práctico HO-2.2.2c (H2): Utilizar el encadenamiento de instrucciones para priorizar los casos de prueba dentro de un conjunto de pruebas determinado, teniendo en cuenta sus prioridades y dependencias específicas

Este ejercicio se centra en el uso de la IA generativa para mejorar la priorización de casos de prueba dentro de un conjunto de pruebas determinado con análisis de riesgos asociados y dependencias entre casos de prueba. La sesión comienza con una breve descripción general de los diferentes enfoques de prueba, como los basados en el riesgo, los basados en la cobertura y los basados en los requisitos, y una revisión del conjunto de pruebas dado. Luego, los participantes participarán en la creación de instrucciones para generar planes de priorización procesables para varias estrategias de priorización de pruebas. Los resultados del MLG basados en la instrucción y los datos de entrada dados deben verificarse manualmente para detectar cualquier error en el razonamiento del MLG.

El objetivo de este ejercicio es experimentar con la IA generativa en las tareas de prueba que requieren capacidades de razonamiento multicriterio (aquí, los diferentes riesgos y las dependencias a considerar para la priorización del caso de prueba).

2.2.3 Pruebas de regresión automatizadas con la IA generativa

A medida que se completa cada nueva iteración o entrega, el número de casos de prueba de regresión que se ejecutarán suele aumentar, lo que los convierte en candidatos ideales para la automatización, particularmente en las cadenas de integración y entrega continuas (pipelines CI/CD) debido a la alta frecuencia de ejecución de pruebas. La IA generativa puede agilizar este proceso ayudando en la creación, mantenimiento y optimización de conjuntos de pruebas de regresión automatizadas. Al adaptarse dinámicamente a los cambios en la base del código y realizar el análisis de impacto, la IA generativa puede identificar qué áreas del software tienen más probabilidades de verse afectadas por las modificaciones recientes, enfocando los esfuerzos de las pruebas de regresión donde más se necesitan.

Estas son algunas de las actividades típicas de pruebas de regresión automatizadas e informes de pruebas que pueden ser compatibles con las instrucciones de la IA generativa:

- **Implementación automatizada de los scripts de prueba con la automatización basada en palabras clave:** los MLG se pueden utilizar para implementar scripts de prueba basados en marcos de trabajo de automatización de pruebas basados en palabras clave, donde las palabras clave predefinidas representan pasos de prueba comunes. La IA generativa puede asignar estas palabras clave a casos de prueba específicos, generar scripts de prueba y ayudar a los probadores e ingenieros de automatización de pruebas en su trabajo.
- **Análisis de impacto y optimización de pruebas:** la IA generativa se puede utilizar para analizar los cambios de código con el fin de identificar áreas de alto riesgo, lo que permite realizar pruebas de regresión específicas donde más se necesita.
- **Pruebas autorreparables y pruebas adaptativas:** la IA generativa se puede utilizar para ajustar automáticamente los scripts de prueba para manejar cambios menores en la interfaz de usuario o la API, evitando fallos innecesarios de pequeñas modificaciones y asegurando que los conjuntos de pruebas permanezcan estables a lo largo del tiempo.
- **Informes e información de prueba automatizados:** la IA generativa permite la creación de informes de prueba detallados y disponibles en tiempo real, con métricas de éxito, fallos y análisis clave. Esto proporciona a las partes interesadas paneles de control que resaltan las tendencias de prueba y ofrecen información predictiva sobre posibles puntos de fallo.
- **Informes de defecto mejorados y análisis de causas raíz:** la IA generativa puede apoyar la recopilación automática de informes completos de defecto con registros de prueba, capturas de pantalla y datos del entorno de prueba.

Estas actividades se pueden aplicar a una variedad de pruebas de regresión, incluidas las pruebas de regresión funcional y no funcional. Sin embargo, los probadores deben ser conscientes de que la IA generativa puede cometer errores. Por lo tanto, la salida generada debe verificarse cuidadosamente, dependiendo del riesgo asociado (consulte el capítulo 3).

Además, la IA generativa puede ayudar a realizar pruebas de regresión automatizadas basadas en interfaz gráfica de usuario y API de extremo a extremo, cada una con sus desafíos y soluciones distintivos. Las pruebas de interfaz gráfica de usuario con frecuencia se vuelven inestables debido a cambios recurrentes en la interfaz de usuario. La IA generativa puede adaptar automáticamente los scripts de prueba para manejar cambios como localizadores dinámicos e interacciones modificadas, reduciendo la necesidad de intervención manual. Las pruebas de regresión de API enfrentan desafíos como cambiar los formatos de solicitudes y respuestas, los puntos finales y la autenticación. La IA generativa puede adaptar los scripts de prueba automáticamente a la evolución de las especificaciones de la API y generar diversos datos de prueba, manteniendo una cobertura completa y reduciendo la necesidad de actualizaciones manuales.

Objetivo práctico HO-2.2.3a (H2): Practicar la formulación de instrucciones con pocos ejemplos para crear y gestionar los scripts de prueba basados en palabras clave

Este ejercicio se centra en el desarrollo y la automatización de los scripts de prueba para una aplicación web determinada utilizando un marco de trabajo de automatización de las pruebas interfaz gráfica de usuario. El ejercicio está estructurado en dos secciones principales: automatización de pruebas y depuración de scripts de prueba. La primera parte del ejercicio proporciona la orientación sobre cómo crear documentación para una biblioteca de palabras clave, generar scripts de prueba iniciales, hacer que la IA valide estos scripts de prueba y ampliar la cobertura con scripts de prueba adicionales. La segunda parte pone énfasis en la asistencia en la depuración, utilizando las instrucciones de sistema para crear un asistente de IA que pueda verificar y corregir los scripts de prueba.

Este ejercicio combina la automatización de pruebas tradicional con la validación asistida por IA, lo que demuestra cómo se pueden utilizar eficazmente las instrucciones con pocos ejemplos para crear, mantener y depurar scripts de prueba basados en palabras clave.

Objetivo práctico HO-2.2.3b (H2): Practicar la ingeniería estructurada de instrucciones para el análisis del informe de prueba en el contexto de las pruebas de regresión

Este ejercicio ilustra un enfoque metódico para analizar los informes de las pruebas de regresión, utilizando instrucciones estructuradas. El proceso comienza con un análisis de los resultados de las pruebas proporcionados y una comparación con la especificación de las pruebas. Luego avanza a la agrupación de defectos similares, el mantenimiento de una lista de anomalías conocidas y una verificación cruzada de los hallazgos. Cada paso está vinculado al siguiente en una sola conversación del MLG.

El enfoque paso a paso demuestra cómo se pueden utilizar las instrucciones estructuradas para transformar los resultados de las pruebas de regresión y los registros de las pruebas en información procesable, lo que respalda un análisis eficaz de los informes de las pruebas en el contexto de las pruebas de regresión.

2.2.4 Monitorización de pruebas y control de pruebas con la IA generativa

Las tareas de monitorización de pruebas requieren la recuperación de grandes cantidades de datos (a veces no estructurados), que a menudo ya están disponibles en las herramientas de gestión de pruebas que la IA generativa puede ayudar a analizar y sintetizar.

La IA generativa facilita una serie de tareas de monitorización y control de pruebas, que incluyen:

- **Monitorización de pruebas y análisis de métricas:** la IA generativa puede facilitar la automatización de la monitorización de pruebas, así como el análisis de tendencias para predecir riesgos potenciales y alertar a los equipos de cualquier desviación del plan. Esto permite a los equipos mantenerse informados y tomar medidas para mantener los estándares de calidad.
- **Control de pruebas:** la IA generativa puede ayudar con el control de pruebas al proporcionar información para volver a priorizar las pruebas, ajustar los calendarios de las pruebas y reasignar los recursos según sea necesario. Esto garantiza que las pruebas sigan siendo flexibles y se centren en áreas de alta prioridad.
- **Información sobre la finalización de las pruebas y el aprendizaje continuo:** la IA generativa puede ayudar generando informes de finalización de las pruebas, destacando los éxitos y las lecciones aprendidas. Esto permite a los equipos perfeccionar las estrategias de prueba y mejorar los procesos de prueba futuros.
- **Mejora en la visualización de las métricas de prueba y los informes de prueba:** la IA generativa puede ayudar en la creación de paneles de control dinámicos y resúmenes en lenguaje natural, asegurando que todas las partes interesadas tengan acceso a las métricas relevantes. Esta asistencia proporciona la información necesaria para tomar decisiones rápidas y ofrece una visión clara del progreso de las pruebas.

Objetivo práctico HO-2.2.4 (H0): Observar las métricas de monitorización de pruebas preparadas por la IA a partir de los datos de prueba

Esta demostración ilustra cómo la IA generativa puede ayudar a los equipos de prueba al transformar los datos de prueba en métricas de monitorización de prueba procesables, lo que facilita la toma de decisiones informadas. A partir de los datos de prueba extraídos de las herramientas de prueba, un MLG los procesa para generar métricas clave como el progreso de las pruebas, las tendencias de los defectos o la cobertura, destacando los riesgos potenciales. Estas métricas generadas por la IA pueden mostrarse en un panel de control y resumirse en lenguaje natural para facilitar su comprensión por todas las partes interesadas.

Esta demostración ilustra cómo la IA generativa convierte los datos de las pruebas en información práctica, ayudando a los equipos de prueba a monitorizar el progreso de las pruebas, gestionar la calidad y adaptarse rápidamente a los cambios.

2.2.5 Elección de técnicas de instrucciones para las pruebas de software

La siguiente tabla muestra la idoneidad de las tres técnicas de instrucciones mencionadas en la sección 2.1.2 de acuerdo con las características de las tareas de prueba.

Técnicas de instrucciones	Recomendaciones de casos de uso	Características y aplicaciones clave
Encadenamiento de instrucciones	Tareas complejas que requieren precisión con verificación humana en cada paso	Divide las tareas en pasos más pequeños, útiles para el análisis de pruebas, el diseño de pruebas y la automatización de pruebas, donde se verifica la exactitud de cada paso de prueba.
Instrucciones con pocos ejemplos	Tareas de formato de salida repetitivas o específicas o restringidas	Proporciona ejemplos a la IA generativa para la generación repetitiva con un patrón específico, por ejemplo, en casos de prueba al estilo Gherkin (por ejemplo, basados en escenarios), pruebas basadas en palabras clave o informes de prueba con un formato de salida específico.
Metainstrucción	Tareas flexibles y dinámicas, útiles para elaborar las instrucciones para nuevas tareas	Descripción general del objetivo y la tarea a realizar, que guía al MLG en la creación de la instrucción. Útil para todo tipo de tareas complejas, como el análisis de informes de pruebas y la detección de anomalías.

Incluso es posible utilizar múltiples técnicas para un solo caso de uso. Por ejemplo, las metainstrucciones se pueden usar para crear una instrucción inicial. Esta indicación generada puede contener ejemplos que deben adaptarse y mejorarse (instrucciones con pocos ejemplos). Por último, puede ser útil dividir la tarea en subtareas más pequeñas para permitir la validación de los pasos intermedios (encadenamiento de instrucciones).

Objetivo práctico HO-2.2.5 (H1): Seleccionar las técnicas de instrucciones adecuadas al contexto para tareas de prueba determinadas

Este ejercicio se centra en seleccionar las técnicas de instrucciones adecuadas para las diferentes tareas de prueba. A los participantes se les asignan varias tareas de prueba con diferentes desafíos. Para cada tarea de prueba, los participantes deben evaluar la naturaleza de la tarea, ya sea que requiera precisión o estructura repetitiva, y sugerir la (s) técnica(s) de instrucciones que mejor se adapte (n) al contexto y satisfaga (n) las necesidades específicas de la tarea. Las opciones se discuten en el grupo.

Este ejercicio está diseñado para profundizar la comprensión de cómo las diferentes técnicas de instrucciones se pueden usar de manera efectiva en los esfuerzos de pruebas prácticas.

2.3 Evaluar los resultados de la IA generativa y afinar las instrucciones para las tareas de prueba de software

Evaluar el rendimiento de la IA generativa en las pruebas de software requiere un conjunto claro de métricas para evaluar la calidad, la relevancia y la efectividad de los resultados generados (Li 2024). Estas métricas, ya sean generales o específicas de la tarea, ayudan a optimizar las instrucciones del MLG.

2.3.1 Métricas para evaluar los resultados de la IA generativa en las tareas de prueba

Se pueden utilizar varias métricas para evaluar la calidad y la eficiencia de los resultados de la IA generativa en una tarea de prueba:

Métrica	Descripción	Ejemplo
Exactitud	Mide cuán correcto es todo el contenido generado con respecto a los casos de prueba, los requisitos redactados por expertos u otros estándares.	El grado en que los casos de prueba generados cubren todos los requisitos especificados.
Precisión	Evalúa cuán correcto es el resultado generado con respecto a un objetivo específico.	El grado en que los casos de prueba generados identifican correctamente las anomalías.
Recuperación	Mide la capacidad de un modelo para identificar todas las instancias relevantes dentro de un conjunto de datos.	El grado en que los casos de prueba generados cubren las particiones de equivalencia válidas e inválidas de una clase de datos.
Relevancia y adecuación contextual	Determina si el resultado generado es aplicable y adecuado para un contexto determinado.	El grado en que los casos de prueba generados son consistentes con la base de prueba e integran los requisitos específicos del dominio.
Diversidad	Asegura que se cubra una amplia gama de entradas y escenarios, evitando la repetición.	El grado en que los casos de prueba generados cubren diversos comportamientos de los usuarios y qué tan bien exploran los casos extremos
Tasa de éxito de ejecución	Mide la proporción de casos de prueba generados o scripts de prueba que se pueden ejecutar con éxito.	Determinar cuántos de los scripts de prueba generados se pueden ejecutar sin errores de sintaxis o problemas de formato de salida en un entorno de pruebas que funcione correctamente.
Eficiencia temporal	Evalúa el tiempo ahorrado en comparación con los esfuerzos de pruebas manuales.	Tiempo requerido por la IA para generar casos de prueba en comparación con el tiempo que un ser humano tardaría en crear manualmente las pruebas equivalentes.

Además de estas métricas generales, las métricas específicas de la tarea se pueden adaptar para evaluar qué tan bien la IA generativa apoya las actividades de prueba específicas.

Para evaluar estas métricas de manera efectiva, los probadores pueden realizar revisiones manuales o automatizarlas, por ejemplo, comparando la salida del MLG con una referencia predefinida. Dada la naturaleza no determinista de la IA generativa, las métricas deben basarse en datos estadísticamente relevantes.

Objetivo práctico HO-2.3.1 (H0): Observar cómo se pueden usar las métricas para evaluar el resultado de la IA generativa en una tarea de prueba

Durante una demostración en una tarea de prueba determinada, se muestran las métricas adaptadas a la tarea para evaluar los resultados de la IA generativa, así como su aplicación concreta a los resultados obtenidos con un MLG en esa tarea de prueba.

Esta demostración ilustra la importancia de las métricas de evaluación para proporcionar confianza en los resultados de la IA generativa para las pruebas de software.

2.3.2 Técnicas para evaluar y afinar iterativamente las instrucciones

Sobre la base de las métricas presentadas anteriormente, se utilizan técnicas específicas para la evaluación y el perfeccionamiento rápidos para mejorar los resultados de la IA:

- **Modificación iterativa de instrucciones:** comience con una instrucción base y modifíquela iterativamente en función de los resultados observados, agregando gradualmente más contexto o ajustando la redacción (por ejemplo, con respecto a la terminología) para mejorar la especificidad y la relevancia.
- **Pruebas A/B de instrucciones:** cree múltiples versiones de instrucciones y evalúe qué versión produce mejores resultados en función de las métricas predefinidas. Este enfoque ayuda a determinar qué fraseo rápido o estructura rápida produce los resultados más precisos y relevantes.
- **Análisis de resultados:** Examinar los resultados generados por la IA en busca de inexactitudes o inconsistencias, por ejemplo, con respecto a la base de prueba. Comprender los tipos de errores e inconsistencias puede ayudar a refinar las instrucciones para evitar defectos similares en futuras iteraciones.
- **Integrar la retroalimentación de los usuarios:** recopilar información de los probadores sobre la utilidad y la claridad de los resultados generados, por ejemplo, con respecto al nivel de detalle de las pruebas generadas. Analice sus conocimientos y utilícelos para refinar las instrucciones para satisfacer mejor las necesidades de las pruebas del mundo real.
- **Ajustar la longitud y la especificidad de la instrucción:** experimente con diferentes longitudes y niveles de detalle de la instrucción. A veces, agregar más contexto puede mejorar la calidad de la respuesta. En otros casos, las instrucciones más cortas pueden producir una mejor generalización.

Mediante el uso de estas técnicas, los equipos de prueba pueden organizar sesiones de evaluación y optimización rápidas para garantizar la mejora continua de las instrucciones de IA generativa. Compartir las prácticas entre el equipo de pruebas o la organización de pruebas no solo ayuda a estandarizar las técnicas rápidas y mantener una calidad constante, sino que también promueve una cultura de aprendizaje y mejora iterativa. Este enfoque colaborativo contribuye a la evolución de las metodologías de prueba la IA generativa permitiendo que los equipos de prueba se basen en conocimientos colectivos, eviten errores repetidos y perfeccionen su uso de las herramientas de IA generativa de manera más efectiva a lo largo del tiempo, por ejemplo, compartiendo bibliotecas de instrucciones.

Objetivo práctico HO-2.3.2 (H1): Evaluar y optimizar una instrucción para una tarea de prueba determinada

Este ejercicio se centra en aplicar técnicas de optimización de instrucciones a una tarea de prueba determinada. Los participantes comenzarán con una instrucción inicial y la refinarán iterativamente para mejorar los resultados generados por la IA. Utilizarán técnicas como las pruebas A/B y la verificación humana para evaluar y mejorar la calidad de las instrucciones. El objetivo es que los participantes experimenten cómo el refinamiento iterativo conduce a una generación de casos de prueba más efectiva y contextualmente relevante.

Al final del ejercicio, los participantes habrán realizado varias iteraciones de refinamiento rápido y evaluado cada iteración utilizando las métricas discutidas para mejorar la calidad de los resultados de la IA.

3 Gestión de riesgos de la IA generativa en las pruebas de software – 160 minutos

Palabras clave

seguridad, vulnerabilidad, privacidad de datos

Palabras clave específicas de IA generativa

alucinación, temperatura, error de razonamiento, sesgo

Objetivos de aprendizaje y objetivos prácticos para el Capítulo 3:

3.1 Alucinaciones, errores de razonamiento y sesgos

- GenAI-3.1.1 (K1) Recordar las definiciones de alucinaciones, errores de razonamiento y sesgos en los sistemas de IA generativa
- GenAI-3.1.2 (K3) Identificar las alucinaciones, los errores de razonamiento y los sesgos en los resultados de los MLG
- HO-3.1.2a (H1) Experimentar con alucinaciones en las pruebas con la IA generativa
- HO-3.1.2b (H1) Experimentar con errores de razonamiento en las pruebas con la IA generativa
- GenAI-3.1.3 (K2) Resumir las técnicas de mitigación de las alucinaciones de la IA generativa, los errores de razonamiento y los sesgos en las tareas de prueba de software
- GenAI-3.1.4 (K1) Recordar las técnicas de mitigación para el comportamiento no determinista de los MLG

3.2 Privacidad de datos y riesgos de seguridad de la IA generativa en las pruebas de software

- GenAI-3.2.1 (K2) Explicar los riesgos clave de privacidad y seguridad de datos asociados con el uso de la IA generativa en las pruebas de software
- GenAI-3.2.2 (K2) Dar ejemplos de privacidad de datos y vulnerabilidades en el uso de la IA generativa en las pruebas de software
- GenAI-3.2.3 (K2) Resumir las estrategias de mitigación para proteger la privacidad de los datos y mejorar la seguridad en la IA generativa para las pruebas de software
- HO-3.2.3 (H0) Reconocer los riesgos de privacidad y seguridad de datos en un determinado caso de estudio de pruebas con la IA generativa

3.3 Consumo de energía e impacto ambiental de la IA generativa en las pruebas de software

- GenAI-3.3.1 (K2) Explicar el impacto de las características de las tareas y el uso del modelo en el consumo de energía de la IA generativa en las pruebas de software
- HO-3.3.1 (H1) Usar un simulador para calcular la energía y las emisiones de CO₂ para determinadas tareas de prueba con la IA generativa

3.4 Regulaciones, estándares y marcos de trabajo de las mejores prácticas de la IA

- GenAI-3.4.1 (K1) Recordar ejemplos de las regulaciones, los estándares y los marcos de trabajo de las mejores prácticas que son relevantes para la IA generativa en las pruebas de software

3.1 Alucinaciones, errores de razonamiento y sesgos

Los sistemas IA generativa, especialmente los MLG, son propensos a ciertos defectos, como alucinaciones, errores de razonamiento y sesgos. Estos defectos reducen la calidad de la salida de la IA generativa en las tareas de prueba, lo que resulta en un testware generado que no cumple con las expectativas de los probadores. Estas alucinaciones, estos errores de razonamiento y estos sesgos en el resultado del MLG deben ser identificados por los probadores, y se deben tomar medidas para mitigar estos riesgos.

El comportamiento no determinista de los MLG (ver sección 1.1.2) dificulta la corrección de este tipo de defectos; pueden parecer corregidos en una salida del MLG, pero reaparecer en otra conversación con el mismo MLG.

3.1.1 Alucinaciones, errores de razonamiento y sesgos en la IA generativa

Las alucinaciones ocurren cuando un MLG genera resultados que parecen objetivamente incorrectos o irrelevantes para una tarea determinada. En las pruebas de software, las alucinaciones pueden manifestarse cuando los MLG crean casos de prueba ficticios o irrelevantes, generan scripts de prueba incorrectos o que no funcionan, o sugieren casos de prueba que verifican criterios de aceptación inexistentes. Esto puede confundir a los probadores y comprometer la validez de los resultados de las pruebas.

Los errores de razonamiento ocurren cuando los MLG malinterpretan las estructuras lógicas, como las relaciones de causa y efecto, la lógica condicional o los procesos de resolución de problemas paso a paso, lo que lleva a conclusiones incorrectas. A diferencia de los humanos, los MLG carecen de un verdadero razonamiento lógico y dependen de la coincidencia de patrones, lo que puede conducir a una lógica defectuosa al realizar tareas como el razonamiento matemático (Mirzadeh 2024). La planificación de las pruebas y la priorización de los casos de prueba son ejemplos de tareas de prueba que requieren razonamiento lógico y donde los MLG pueden cometer errores de razonamiento.

Los sesgos de los MLG (Gallegos 2024) provienen de los datos sobre los que se entrenó el modelo. Estos sesgos pueden conducir a resultados que favorecen ciertos tipos de información, enfoques o suposiciones. Por ejemplo, los MLG entrenados principalmente con datos en inglés pueden subestimar las perspectivas que no son en inglés. En las pruebas de software, los sesgos pueden influir en las respuestas de los MLG cuando, por ejemplo, se generan datos de prueba o se refinan los criterios de aceptación para los casos de prueba.

Las alucinaciones, los errores de razonamiento y los sesgos en la salida de la IA generativa son el resultado de la naturaleza de sus datos de entrenamiento y las limitaciones inherentes del modelo transformador (ver Capítulo 1). Reconocer y abordar estos desafíos aumenta la calidad de los resultados de la IA generativa en los procesos de prueba.

3.1.2 Identificación de alucinaciones, errores de razonamiento y sesgos en los resultados de los MLG

La integración efectiva de los sistemas con la IA generativa en las pruebas de software requiere la capacidad de detectar alucinaciones, errores de razonamiento y sesgos en la salida de los MLG. Dependiendo del tipo de problema, se pueden aplicar diferentes enfoques de detección. Los siguientes son enfoques comunes que se aplican a través de la revisión o una combinación de revisión y verificación automatizada:

Detección de alucinaciones:

- **Verificación cruzada:** compare los resultados generados por la IA con la documentación existente, los requisitos y el comportamiento conocido del sistema. Las herramientas automatizadas pueden ayudar a cruzar el resultado con las fuentes de datos establecidas para marcar las discrepancias.

- Consulta a expertos en el dominio: involucre a expertos en la materia para validar la exactitud del contenido generado. Su experiencia es esencial para capturar información matizada que los sistemas automatizados podrían pasar por alto.
- Comprobaciones de consistencia: Verifique que las salidas generadas sean consistentes entre sí y con la información conocida. Los sistemas automatizados pueden ayudar a identificar patrones y marcar inconsistencias.

Detección de errores de razonamiento:

- Validación lógica: Evaluar el flujo lógico (por ejemplo, la consistencia, la coherencia y el razonamiento estructurado dentro del texto generado) del contenido generado por la IA en cuanto a la coherencia y la capacidad de ser correcto a través de ciclos de revisión. Las herramientas automatizadas pueden ayudar, pero los casos complejos pueden requerir juicio humano.
- Pruebas de las salidas: por ejemplo, ejecutar los casos de prueba generados o los scripts de prueba contra los objetos de prueba para verificar los resultados de las pruebas. Esto puede ser parcial o totalmente automatizado, dependiendo del tipo de testware que se genere.

Detección del sesgo:

- Revisar cómo el testware generado, como los datos de prueba sintéticos, se representa de manera justa y precisa en relación con la estrategia de pruebas
- Evaluar los sesgos relacionados con los tipos de prueba, como las pruebas no funcionales subrepresentadas en el resultado generado del MLG.

La implementación real de estos métodos de detección dependerá del nivel de riesgo estimado de alucinaciones, errores de razonamiento o sesgos en las tareas de prueba que se realiza la IA generativa.

Objetivo práctico HO-3.1.2a (H1): Experimentar con las alucinaciones de la IA generativa relacionadas con una tarea de las pruebas de software

Este ejercicio se centra en experimentar con ejemplos de alucinaciones de la IA generativa en relación con el conjunto de conocimientos de las pruebas de software. Los participantes intentarán confrontar al menos dos MLG con una situación en la que los MLG inventan elementos irrelevantes, por ejemplo, agregar criterios no relacionados que no existen en los datos contextuales dados. Las variaciones en las instrucciones se prueban para examinar la influencia de las instrucciones en las alucinaciones.

Este ejercicio aumenta la comprensión de la identificación de alucinaciones de la IA generativa en las pruebas de software.

Objetivo práctico HO-3.1.2b (H1): Experimentar con los errores de razonamiento de la IA generativa en una tarea de la planificación de pruebas

Este ejercicio se centra en presentar un ejemplo de un error de razonamiento de la IA generativa. Un ejemplo de un problema a resolver en el área de la planificación de pruebas, como la estimación del esfuerzo de las pruebas y la priorización de los casos de prueba (ver [ISTQB_CTFL] - Capítulo 5). El ejercicio está diseñado con una cierta complejidad de datos de entrada, lo que requiere habilidades de resolución de problemas y destaca las limitaciones de los MLG para este propósito. El resultado del MLG se comparará con el resultado exacto que se debe lograr. Se probarán tres tipos diferentes de MLG (MLG, MLP y el modelo de razonamiento) y se utilizarán variaciones de la instrucción para tratar de mejorar los resultados.

Este ejercicio aumenta la comprensión de cómo identificar los errores de razonamiento de la IA generativa en las tareas de prueba de software que requieren habilidades de resolución de problemas lógicos.

3.1.3 Técnicas de mitigación de las alucinaciones de la IA generativa, los errores de razonamiento y los sesgos en las tareas de prueba de software

Para minimizar los resultados indeseables de la IA generativa en las pruebas de software, se pueden emplear varias estrategias para reducir las alucinaciones, los errores de razonamiento y los sesgos. Es más probable que estos problemas ocurran cuando las instrucciones no están diseñadas correctamente (consulte el Capítulo 2) o cuando faltan datos de entrada contextuales relevantes para una tarea de prueba determinada. Las técnicas clave para mitigar los riesgos asociados con las alucinaciones, los errores de razonamiento y los sesgos de la IA incluyen:

- Proporcionar un contexto completo: asegurarse de que la instrucción contenga toda la información relevante (consulte la sección 2.1.1), ofreciendo un contexto integral para guiar a la IA en la producción de resultados exactos.
- Dividir las instrucciones en segmentos manejables: dividir las instrucciones complejas en pasos más pequeños mediante el uso de técnicas de encadenamiento de instrucciones (consulte la sección 2.1.2), verificando sistemáticamente cada resultado antes de pasar al siguiente. Este enfoque paso a paso puede ayudar a detectar errores de razonamiento al principio del proceso de generación.
- Utilizar formatos de datos claros e interpretables: evitar formatos que puedan ser ambiguos o difíciles de interpretar para la IA generativa. Los formatos estructurados y sencillos ayudan al modelo a centrarse en los aspectos esenciales de la tarea.
- Seleccionar el modelo de la IA generativa adecuado para la tarea: utilizar un MLG específicamente entrenado para la tarea en cuestión (consulte la sección 5.1.3).
- Comparar los resultados entre los modelos: cuando sea adecuado, evaluar la instrucción con varios MLG y comparar las salidas lo que ayuda a detectar errores de salida y seleccionar los resultados más fiables.

El capítulo 4 presenta dos técnicas complementarias para mejorar los resultados de los MLG: generación aumentada por recuperación (GAR) y el ajuste fino.

3.1.4 Mitigación del comportamiento no determinista de los MLG

El comportamiento no determinista inherente de los MLG (Shuyin 2023) puede conducir a variaciones en la producción, incluso cuando se proporciona la misma entrada. Esto surge de los procesos de muestreo probabilístico utilizados durante la inferencia. En consecuencia, lograr resultados consistentes y reproducibles al usar MLG puede ser un desafío, particularmente para resultados largos, lo que aumenta el riesgo de variabilidad.

Si bien no se puede garantizar la reproducibilidad completa, ciertas estrategias pueden ayudar a reducir la variabilidad:

- Ajustar la configuración de los parámetros de temperatura del MLG: bajar la temperatura durante la generación de las respuestas (inferencia) reduce la distribución de probabilidad, reduciendo la aleatoriedad y dando como resultado resultados más consistentes. Sin embargo, esto también limitará la creatividad y la diversidad en las respuestas, haciendo que los resultados sean más repetitivos o demasiado deterministas.
- Configuración de semillas aleatorias: algunas implementaciones de los MLG permiten establecer un valor de semilla para el generador de números aleatorios, asegurando que se use la misma secuencia pseudoaleatoria (es decir, valores aleatorios deterministas), lo que mejora la reproducibilidad.

Reducir el riesgo de las alucinaciones y los errores de razonamiento en la producción de los MLG implica abordar este comportamiento no determinista, por ejemplo, automatizando algunos aspectos de la verificación de la producción para garantizar un proceso de evaluación estructurado y consistente.

3.2 Privacidad de datos y riesgos de seguridad de la IA generativa en las pruebas de software

La IA generativa en las pruebas introduce riesgos relacionados con la privacidad y seguridad de datos debido al manejo de información confidencial y posibles vulnerabilidades en la infraestructura de pruebas impulsada por los MLG. Una protección de datos sólida es esencial para evitar infracciones, el acceso no autorizado y la exposición de datos confidenciales.

3.2.1 Riesgos de privacidad y seguridad de datos asociados con el uso de la IA generativa

La IA generativa puede procesar grandes cantidades de datos que pueden contener información confidencial o de identificación personal. Esto plantea las siguientes preocupaciones:

- Exposición no intencional de datos: los modelos la IA generativa pueden generar resultados que accidentalmente revelan información confidencial.
- Falta de control sobre el uso de datos: las herramientas la IA generativa pueden almacenar y procesar datos confidenciales sin el consentimiento o control explícito del usuario. Esto puede dar lugar a un posible uso indebido o acceso no autorizado.
- Riesgos de cumplimiento: el uso de herramientas con la IA generativa sin cumplir con las regulaciones de protección de datos, como el RGPD (Reglamento General de Protección de Datos, Reglamento (UE) 2016/679), podría dar lugar a disputas legales.

Además, surgen riesgos de seguridad específicos al realizar pruebas con la IA generativa, como:

- La Infraestructura de pruebas impulsada por los MLG puede ser vulnerable a ataques de seguridad, como violaciones de datos o acceso no autorizado.
- Los actores maliciosos pueden explotar las vulnerabilidades en los MLG, como ataques de manipulación (ver sección 3.2.2), para alterar su comportamiento o extraer información confidencial.
- Los atacantes pueden introducir intencionalmente datos de entrada maliciosos para engañar a los MLG y comprometer su exactitud o seguridad.

3.2.2 Privacidad de datos y vulnerabilidades en la IA generativa para los procesos y las herramientas de prueba

La siguiente tabla ofrece algunos ejemplos de vectores de ataque en los procesos y herramientas de prueba con la IA generativa.

Vector de ataque	Descripción	Ejemplo
Exfiltración de datos (también fuga de datos)	Envío de instrucciones diseñados para extraer datos de entrenamiento confidenciales.	Exceder la ventana de contexto del MLG con instrucciones largas para sobrecargar la memoria de la IA podría llevarla a revelar fragmentos aleatorios de sus datos de entrenamiento y exponer potencialmente información confidencial.
Manipulación de las solicitudes	Envío de instrucciones diseñados para extraer datos de entrenamiento confidenciales.	Exceder la ventana contextual del MLG con instrucciones largas para sobrecargar la memoria de la IA podría llevarla a revelar fragmentos aleatorios de sus datos de entrenamiento y exponer potencialmente información confidencial.
Envenenamiento de datos (también contaminación de datos)	Manipular los datos de entrenamiento.	Proporcionar evaluaciones falsas al calificar los resultados de un informe de prueba generado por la IA.
Generación de código malicioso	Manipular un MLG para generar puertas traseras (por ejemplo, llamadas a comandos externos) durante el uso.	Generación de código para abrir un canal de comunicación con una IP específica y maliciosa.

3.2.3 Estrategias de mitigación para proteger la privacidad de los datos y mejorar la seguridad en las pruebas con la IA generativa

A medida que la IA generativa se convierte en la tecnología predominante, y con los riesgos inherentes involucrados, surgen regulaciones y estándares para mitigarlos (ver sección 3.4.1).

Las regulaciones de protección de datos como el RGPD no restringen las aplicaciones de la IA generativa explícitamente, pero proporcionan salvaguardas que pueden limitar lo que se puede hacer, particularmente con respecto a la legalidad y las limitaciones en los fines de recopilación, procesamiento y almacenamiento de datos.

Para mitigar estos riesgos, las organizaciones deben implementar medidas sólidas de privacidad de datos, que incluyen:

- **Minimización de datos:** evitar el procesamiento de datos confidenciales a menos que esté legalmente permitido y utilizar solo la cantidad necesaria de datos no confidenciales en las pruebas con IA para reducir los riesgos de privacidad de los datos.
- **Anonimización y seudonimización de datos:** Enmascarar o reemplazar información sensible con datos no identificables.
- **Almacenamiento y transmisión de datos seguros:** implementación de un fuerte cifrado y controles de acceso.

- Formación de recursos humanos: las organizaciones deben establecer programas y políticas de formación claros para garantizar el uso responsable de las herramientas de IA generativa, promover prácticas éticas y mitigar los riesgos potenciales.

Se pueden considerar estrategias de mitigación adicionales al implementar la IA generativa para las pruebas:

- Revisión sistemática del resultado generado: la evaluación humana es esencial para garantizar la calidad y exactitud de las tareas de prueba impulsadas por la IA generativa.
- Evaluación en comparación con otro MLG: Esto implica el uso de varios MLG en una tarea determinada para evaluar los resultados mediante la comparación de sus respuestas.
- Elección de un entorno seguro y operativo: dependiendo del nivel de confidencialidad requerido, las organizaciones pueden optar por diferentes soluciones seguras: utilizar una oferta comercial segura de un proveedor de MLG, operar el MLG en una nube segura o instalar el MLG en la infraestructura de la organización.
- Auditorías de seguridad y evaluaciones de vulnerabilidad periódicas: identificar y abordar las debilidades en los sistemas de IA generativa.
- Mantenerse actualizado con las mejores prácticas de seguridad: mantenerse al día con las últimas directrices y tecnologías de seguridad.

Las estrategias con frecuencia se complementan entre sí y se requiere una combinación de estas para garantizar la seguridad de datos mientras se usa la IA generativa. Se recomienda encarecidamente involucrar a ingenieros de seguridad sénior, asesores legales, el director de tecnología (CTO) o el responsable de seguridad de la información (DSI/CISO), si está presente en la organización.

Objetivo práctico HO-3.2.3 (H0): Reconocer los riesgos de privacidad y seguridad de datos en un determinado caso de estudio de la IA generativa para las pruebas

Esta demostración ilustra cómo pueden surgir riesgos para la privacidad y la seguridad de datos cuando se utiliza la IA generativa en las pruebas de software. Los participantes explorarán casos de estudio para identificar posibles amenazas, como vulnerabilidades de modelos, acceso no autorizado a datos o uso malicioso de los resultados generados. Explorarán estrategias de mitigación, incluido el manejo seguro de datos, controles de acceso sólidos y prácticas de monitorización de la IA, mientras reflexionan sobre las implicaciones éticas y prácticas.

Al final, los participantes comprenderán los principios de privacidad de datos y aprenderán a reconocer y abordar los riesgos de seguridad en las condiciones de prueba de la IA generativa.

3.3 Consumo de energía e impacto ambiental de la IA generativa en las pruebas de software

Estudios como (Luccioni 2024a) muestran que el entrenamiento y el procesamiento de los MLG requieren el uso intensivo de una gran cantidad de recursos informáticos especializados. Los MLG están disponibles como servicios basados en la web, y su uso aumenta la carga en dispositivos, redes y centros de datos, lo que lleva a un mayor consumo de energía.

3.3.1 El impacto del uso de la IA generativa en el consumo de energía y las emisiones de CO₂

No se debe subestimar el impacto ambiental de la IA generativa, ya que el consumo de energía aumenta bruscamente a medida que aumenta el uso. La complejidad de la tarea y los recursos computacionales requeridos influyen en el consumo de energía. Por ejemplo, generar una sola imagen utilizando un modelo de IA potente puede consumir tanta energía como cargar completamente un teléfono inteligente, mientras que generar texto consume solo un pequeño porcentaje de la carga de un teléfono inteligente (Heikkilä 2023).

Incluso si es difícil obtener datos precisos sobre el impacto ambiental de la IA generativa (Luccioni 2024b), está claro que estas operaciones de uso intensivo de energía contribuyen colectivamente a emisiones significativas de CO₂ (Berthelot 2024). Si bien una sola búsqueda o tarea de generación de texto puede parecer insignificante, su efecto acumulativo en millones de usuarios en todo el mundo resulta en una tensión ambiental sustancial.

Adoptar las mejores prácticas, como limitar las interacciones innecesarias de los modelos, es fundamental para mitigar los riesgos ambientales que plantea la IA generativa.

Objetivo práctico HO-3.3.1 (H1): Utilizar un simulador para calcular la energía y las emisiones de CO₂ para determinadas tareas de prueba con la IA generativa

Este ejercicio se centra en evaluar el consumo de energía y las emisiones de CO₂ asociadas de varias tareas de la IA generativa dentro de las pruebas de software. Los participantes utilizarán simulaciones para calcular estas métricas y examinar cómo las diferentes características de la tarea y el uso del modelo afectan el impacto ambiental.

Al observar cómo los diferentes factores afectan el consumo de energía y las emisiones, los participantes comprenden los factores que impulsan el consumo de energía con los MLG.

3.4 Regulaciones, estándares y marcos de trabajo de las mejores prácticas de la IA

La IA generativa está transformando las pruebas de software al ayudar a los probadores en una variedad de tareas de prueba (consulte el Capítulo 2). Sin embargo, estas oportunidades también conllevan riesgos significativos, como errores de razonamiento, privacidad de datos, vulnerabilidades e impactos ambientales (consulte las secciones 3.1, 3.2 y 3.3). Para abordar estos riesgos se debe considerar las regulaciones generales, los estándares y los marcos de trabajo de las mejores prácticas para la IA.

3.4.1 Regulaciones, estándares y marcos de trabajo de la IA relevantes para la IA generativa en las pruebas de software

A continuación, se muestra una descripción general de las directrices clave relevantes para el uso de la IA generativa en las pruebas de software:

Nombre / Tipo	Descripción	Aplicación en las pruebas de software
ISO/IEC 42001:2023 Tecnología de la información – Sistema de gestión de inteligencia artificial Tipo: Estándar	Especifica los requisitos para gestionar los sistemas de IA dentro de una organización	Asegura que el uso de la IA generativa en las pruebas se adhiera a las prácticas recomendadas, promoviendo la consistencia y la fiabilidad
ISO/IEC 23053:2022 Marco de trabajo los para sistemas de IA que utilizan el aprendizaje automático. Tipo: Estándar	Proporciona un marco de trabajo para los procesos del ciclo de vida de la IA, haciendo hincapié en la tolerancia a fallos y la transparencia.	Proporciona un marco de trabajo para la calidad de los datos, la transparencia y la tolerancia a fallos al utilizar la IA generativa en las pruebas.
Ley de IA de la Unión Europea Tipo: Reglamento	Establece un marco de trabajo jurídico que aborda los riesgos de la IA, clasificando las aplicaciones por su nivel de riesgo. Fuente: (Ley de la IA 2024)	Exige el cumplimiento en materia de transparencia, rendición de cuentas y mitigación de sesgos para la IA generativa utilizada en las pruebas.
Marco de trabajo de la Gestión de Riesgos de la IA del NIST (EE. UU.) Tipo: Marco de trabajo	Ofrece las directrices para la gestión de riesgos de la IA, centrándose en la equidad, la transparencia y la seguridad. Fuente: (NIST AI RMF 1.0)	Garantiza la equidad y mitiga los riesgos en la IA generativa, evitando resultados de las pruebas sesgados.

A medida que las tecnologías de la IA y sus entornos regulatorios continúan evolucionando, es imperativo que las organizaciones de prueba se mantengan actualizadas sobre el desarrollo de las regulaciones, los estándares, las leyes nacionales y marcos de trabajo de las mejores prácticas, como los de esta tabla.

4 Infraestructura de pruebas impulsada por los MLG para las pruebas de software – 110 minutos

Palabras clave

infraestructura de pruebas

Palabras clave específicas de IA generativa

ajuste fino, agente impulsado por los MLG, operaciones de grandes modelos de lenguaje, generación aumentada por recuperación, base de datos vectorial

Objetivos de aprendizaje y objetivos prácticos para el Capítulo 4:

4.1 Enfoques arquitectónicos para la infraestructura de pruebas impulsada por los MLG

- | | | |
|-------------|------|---|
| GenAI-4.1.1 | (K2) | Explicar los componentes arquitectónicos clave y los conceptos de la infraestructura de pruebas impulsada por los MLG |
| GenAI-4.1.2 | (K2) | Resumir la generación aumentada por recuperación (GAR) |
| HO-4.1.2 | (H1) | Experimentar con la generación aumentada por recuperación para una determinada tarea de prueba |
| GenAI-4.1.3 | (K2) | Explicar el rol y la aplicación de los agentes impulsados por los MLG en la automatización de los procesos de prueba |
| HO-4.1.3 | (H0) | Observar cómo un agente impulsado por los MLG ayuda a automatizar una tarea de prueba repetitiva |

4.2 Ajuste fino y las OpsMLG: puesta en marcha de la IA generativa para las pruebas de software

- | | | |
|-------------|------|--|
| GenAI-4.2.1 | (K2) | Explicar el ajuste fino de los modelos de lenguaje para tareas de prueba específicas |
| HO-4.2.1 | (H0) | Observar un ejemplo de un proceso de ajuste fino para una tarea de prueba y un modelo de lenguaje determinados |
| GenAI-4.2.2 | (K2) | Explicar las OpsMLG y su papel en la implementación y gestión de los MLG para las tareas de prueba |

4.1 Enfoques arquitectónicos para la infraestructura de pruebas impulsada por los MLG

Los chatbots con IA y las herramientas de prueba alimentadas por los MLG son dos tipos de infraestructuras de prueba que utilizan los MLG (consulte la sección 1.2.2).

Más allá de la arquitectura básica de una infraestructura de prueba alimentada por los MLG (consulte la sección 4.1.1), las arquitecturas de generación aumentada por recuperación (consulte la sección 4.1.2) y de agente alimentado por los MLG (consulte la sección 4.1.3) amplían la funcionalidad y la utilidad del uso de los MLG en las pruebas de software.

4.1.1 Componentes arquitectónicos clave y conceptos de la infraestructura de pruebas impulsada por los MLG

Una infraestructura de prueba impulsada por un MLG se refiere a un sistema que integra un MLG en el proceso de prueba de software para mejorar la automatización, el razonamiento y la toma de decisiones. A diferencia de un chatbot con IA tradicional, que se centra principalmente en las interacciones conversacionales, una herramienta de pruebas impulsada por un MLG está diseñada para apoyar las pruebas de software mediante el procesamiento de consultas relacionadas con las pruebas, el análisis de requisitos, la generación de casos de prueba y la evaluación de resultados.

La arquitectura típica de una infraestructura de prueba alimentada por un MLG sigue un diseño multicomponente que facilita una interacción segura y eficiente con el MLG. La arquitectura consta de componentes de la capa de presentación (front-end) y la capa de servicios (back-end), junto con fuentes de datos externas y un MLG integrado:

- El front-end sirve como la interfaz de usuario donde los probadores interactúan con el sistema ingresando consultas o comandos.
- El back-end procesa la entrada del usuario y gestiona funciones críticas como la autenticación, la recuperación de datos, la preparación de instrucciones y la interacción con el MLG.
- El MLG, que puede alojarse como un servicio de terceros (al que se accede a través de la API) o un modelo interno personalizado, genera respuestas basadas en instrucciones estructuradas.

Esta arquitectura va más allá de un modelo cliente-servidor tradicional al incorporar componentes de procesamiento inteligentes, como los MLG y back-ends de múltiples fuentes:

1. El MLG no es solo un servidor, sino un componente de procesamiento inteligente que interpreta y razona en función del testware.
2. A diferencia de los chatbots basados en reglas que siguen respuestas guionizadas, una infraestructura de pruebas impulsada por MLG genera conocimientos de prueba de forma dinámica a partir del contexto, como requisitos, código o resultados de pruebas.
3. El back-end integra múltiples fuentes de datos, tales como:
 - Bases de datos relacionales (para datos estructurados utilizados en pruebas, como los casos de prueba).
 - Bases de datos vectoriales (para la recuperación semántica de contenido relacionado utilizando incrustaciones; consulte la sección 4.1.2).
4. El back-end mejora la salida bruta del MLG mediante un posprocesamiento, asegurando que sus respuestas se ajusten a las condiciones del proceso de pruebas antes de presentarlas al front-end.

4.1.2 Generación aumentada por recuperación (GAR)

La Generación Aumentada por Recuperación (GAR) mejora los MLG al incorporar fuentes de datos adicionales en su proceso de generación de respuestas (Zhao 2024), lo que aumenta la relevancia y precisión de sus resultados.

GAR combina sistemas de recuperación con modelos de lenguaje para generar respuestas conscientes del contexto. Durante el preprocesamiento, los documentos grandes se dividen en fragmentos más pequeños (por ejemplo, 256-512 tóquenes) para garantizar la recuperación enfocada y la compatibilidad con la ventana de contexto del modelo. Cada fragmento se limpia, procesa y codifica en un vector de alta dimensión (incrustación) utilizando modelos preentrenados. Estas incrustaciones, que se pueden almacenar en bases de datos vectoriales, permiten una recuperación eficiente basada en la similitud en tiempo de ejecución (inferencia). Se codifica una consulta de usuario, se recuperan fragmentos relevantes en función de la similitud semántica y estos fragmentos se utilizan como contexto para que el modelo de lenguaje genere una respuesta fundamentada.

Una respuesta relevante es esencialmente un resultado generado por el modelo de lenguaje que está profundamente arraigado en la información relevante, precisa y contextualmente adecuada recopilada durante el proceso de recuperación. Asegura que la respuesta no solo se base en el entrenamiento preexistente del modelo, sino que también se enriquezca con datos precisos pertinentes a la solicitud. Esta sinergia entre la recuperación y la generación mejora la exactitud y la relevancia de las respuestas, haciéndolas más fiables e informativas para el usuario.

En la fase de procesamiento de instrucciones del usuario, un sistema GAR funciona a través de un proceso de dos pasos:

1. Recuperación: Dada una consulta de usuario, el sistema recupera información relevante de las bases de datos vectoriales creadas previamente. Esta recuperación generalmente se basa en la similitud semántica entre las incrustaciones de la solicitud y las de los fragmentos.
2. Generación: La información recuperada se envía al MLG, que genera una respuesta que combina su conocimiento existente con los datos recién adquiridos, lo que resulta en un resultado más exacto y adecuado al contexto.

El GAR en las pruebas de software permite que la infraestructura de pruebas impulsada por MLG acceda a las fuentes de datos corporativos de la empresa, como bases de datos, documentación y repositorios, para recuperar información contextual en tiempo real, asegurando que las tareas de prueba, como el análisis de pruebas o el diseño de pruebas, estén alineadas con las últimas especificaciones, los requisitos y los datos de prueba existentes.

Objetivo práctico HO-4.1.2 (H1): Experimentar con la generación aumentada por recuperación para una tarea de prueba determinada

Este ejercicio práctico se centra en la aplicación de las técnicas con GAR para una tarea de prueba determinada. Los participantes experimentarán con un sistema GAR incorporando documentos y observarán cómo genera respuestas más o menos exactas basadas en información compleja. Los participantes compararán el resultado del MLG con y sin GAR en la tarea de prueba dada. Este ejercicio tiene como objetivo identificar las fortalezas y limitaciones del sistema GAR en el manejo de diferentes tipos de tareas de prueba.

Al examinar los datos recuperados y los resultados generados, los participantes obtendrán información sobre el rol del GAR en la mejora de los procesos de prueba impulsados por los MLG.

4.1.3 El papel de los agentes impulsados por los MLG en la automatización de los procesos de prueba

Los agentes impulsados por los MLG (Wang 2024) son aplicaciones de IA generativa especializadas impulsadas por los MLG y diseñadas para el procesamiento semiautónomo o autónomo de tareas definidas. En esencia, estos agentes confían en los MLG para la comprensión y generación del lenguaje natural, complementado con la posibilidad de procesar instrucciones, recuperar el contexto y tomar acciones inteligentes.

A diferencia de los chatbots con IA tradicionales que se centran únicamente en las interacciones pregunta-respuesta, los agentes impulsados por los MLG pueden realizar tareas o "actuar" invocando un conjunto predefinido de funciones, comúnmente denominadas "herramientas". Esta capacidad les permite interactuar y manipular sistemas externos, lo que los hace muy versátiles en la ejecución de tareas. El grado de autonomía de los agentes impulsados por los MLG puede variar:

- Los agentes autónomos operan de forma independiente, realizando tareas con una intervención humana mínima utilizando reglas predefinidas, aprendizaje por refuerzo y ciclos de retroalimentación adaptativa.
- Los agentes semiautónomos realizan tareas con supervisión humana periódica para asegurar que el resultado cumpla con los objetivos definidos por el usuario.

Las arquitecturas multiagente implican un sistema colaborativo donde varios agentes, cada uno con roles especializados, se comunican y coordinan para resolver problemas complejos de manera más eficiente que un solo agente. Este esfuerzo coordinado entre múltiples agentes de IA se conoce como "orquestación".

En los procesos de prueba, los agentes impulsados por los MLG pueden automatizar las tareas de prueba emulando el razonamiento humano y la toma de decisiones. Sin embargo, estos agentes sufren los mismos problemas de posibles alucinaciones, errores de razonamiento y sesgos observados al usar los MLG (ver Sección 3.1). Estos agentes pueden producir resultados incorrectos o engañosos, lo que puede debilitar la fiabilidad de los procesos de prueba automatizados. Estos riesgos se pueden mitigar mediante la implementación de procedimientos de verificación automatizados para los resultados de los agentes o el uso de agentes semiautónomos para tareas críticas.

Objetivo práctico HO-4.1.3 (H0): Observar cómo un agente impulsado por los MLG ayuda a automatizar una tarea de prueba repetitiva

La demostración se centra en una tarea de prueba realizada por un agente impulsado por los MLG. Los datos de entrada pasados al agente, su comportamiento y los resultados de sus acciones se demostrarán para ilustrar los diversos aspectos de la integración de soluciones basadas en agentes en un proceso de prueba.

Esta demostración muestra un ejemplo concreto de un agente impulsado por los MLG en el contexto de una tarea de prueba.

4.2 Ajuste fino y las OpsMLG: puesta en marcha de la IA generativa para las pruebas de software

Dos prácticas clave para poner en funcionamiento la infraestructura de prueba impulsada por los MLG para las pruebas incluyen el ajuste fino de los MLG y la gestión de la canalización operativa a través de las OpsMLG (Mailach 2024).

4.2.1 Ajuste fino de los MLG para las tareas de prueba

El ajuste adapta un modelo de lenguaje (LM) previamente entrenado, como un MLG o un modelo de lenguaje pequeño (MLP, consulte la sección 1.1.2), para realizar tareas específicas o adaptarlo a dominios particulares (Parthasarathy 2024). Esto implica un mayor entrenamiento del modelo en un conjunto de datos específico, lo que le permite aprender conocimientos y matices específicos del dominio. Mediante el ajuste, el rendimiento del modelo se mejora para las aplicaciones especializadas, lo que lo hace más preciso y relevante para el caso de uso previsto.

En la práctica, el ajuste fino es adecuado para equipar a los MLG genéricos con habilidades de razonamiento especializado relevantes para un dominio específico o para adoptar un vocabulario único para ese campo. El ajuste también se puede aplicar a modelos más pequeños, conocidos como los MLP, que requieren menos recursos. Al ajustar un MLP, se pueden lograr niveles de rendimiento más altos para tareas específicas sin la misma sobrecarga computacional requerida para los MLG. Esta comparación destaca la flexibilidad y la eficiencia en el uso de los MLG y los MLP en función de los requisitos específicos de la tarea.

Por ejemplo, en las pruebas de software, el ajuste fino puede permitir que un MLG o MLP genere casos de prueba a partir de historias de usuario en un formato de salida específico para el contexto de la organización. Al entrenar el modelo sobre las historias de usuario de la organización y los casos de prueba correspondientes, el modelo se alinea con el proceso de prueba y la terminología específicos de la organización.

El ajuste de un modelo de la IA generativa para las pruebas de software presenta varios desafíos:

- Evitar resultados sesgados o inexactos asegurando el uso de conjuntos de datos de entrenamiento de alta calidad y específicos de la tarea.
- Mitigar el sobreajuste (el modelo se vuelve demasiado especializado para los datos de entrenamiento, lo que afecta negativamente su rendimiento en datos nuevos e invisibles) para mantener la generalización en diferentes escenarios.
- Abordar la opacidad (falta de transparencia en la forma en que un MLG toma sus decisiones o produce sus resultados) en el razonamiento del modelo, lo que complica la depuración y la validación
- Gestionar los recursos computacionales significativos necesarios para el proceso de ajuste fino (para los MLG).

Objetivo práctico HO-4.2.1 (H0): Observar un ejemplo del proceso de ajuste fino para una tarea de prueba determinada y un MLG/MLP

Esta demostración muestra los diversos pasos involucrados en el ajuste fino de un MLG para una tarea de prueba determinada. Comienza con la selección de un MLG o MLP adecuado. A continuación, se presenta un conjunto de datos que se adapta a la tarea de prueba determinada. A continuación, se muestra una solución ejemplar para el proceso de ajuste fino (por ejemplo, un marco de trabajo de aprendizaje automático). Finalmente, se envía una instrucción al modelo afinado y se analiza la calidad de la salida generada.

Esta demostración del proceso de ajuste fino MLG/MLP para una tarea de prueba muestra varios aspectos clave de este proceso y aborda en particular la calidad de los datos de entrenamiento.

4.2.2 Las OpsMLG al desplegar y gestionar los MLG para las pruebas de software

Las OpsMLG se refieren al conjunto de prácticas, herramientas y procesos diseñados para agilizar el desarrollo, el despliegue y el mantenimiento de los MLG en entornos de producción (Sinha 2024).

El uso de la IA generativa en los procesos de prueba de una organización se puede lograr de varias maneras diferentes, lo que influirá en las decisiones de las OpsMLG que se tomarán. Tres posibles enfoques:

- **Uso de un chatbot con IA:** Las consideraciones principales para este enfoque incluyen la gestión de la privacidad de los datos y los riesgos de seguridad al tiempo que se optimizan los costos. Las organizaciones podrían utilizar plataformas de MLG como servicio si se ofrecen las garantías necesarias, o bien desplegar infraestructura propia mediante el uso de MLG de código abierto para obtener un mayor control. Una evaluación rigurosa de las garantías de los proveedores o las capacidades internas es fundamental para mitigar los riesgos de privacidad y seguridad de datos (consulte la sección 3.2) y garantizar la eficiencia operativa.
- **Uso de una herramienta de prueba con capacidades generativas de IA:** Este enfoque tiene consideraciones similares a los chatbots con IA, como la privacidad de los datos, la seguridad y los costos operativos. Además, las organizaciones deben evaluar la seguridad de datos y las garantías de rendimiento ofrecidas por el proveedor de la herramienta de prueba. Estas herramientas de prueba generalmente complementan los procesos de prueba existentes, que requieren un análisis riguroso de costo-beneficio y una evaluación de riesgos.
- **Desarrollo interno de una herramienta de pruebas basada en IA generativa:** Este enfoque enfatiza el control integral de la privacidad de los datos y los riesgos de seguridad, así como una planificación cuidadosa de los costos operativos de la IA, como los recursos computacionales, el almacenamiento de datos y la formación del personal. Las organizaciones también deben establecer procesos estructurados para validar y mantener los desarrollos específicos de la IA generativa. El desarrollo de soluciones internas requiere experiencia en la implementación y despliegue de una infraestructura de pruebas impulsada por los MLG.

Estos enfoques no son mutuamente excluyentes, ya que una organización podría utilizar un chatbot con IA para algunas tareas mientras desarrolla herramientas personalizadas para otras. Por lo tanto, pueden implementarse simultáneamente dependiendo de las actividades de prueba específicas involucradas. Además, pueden incorporar tecnologías adicionales, como GAR y ajuste fino de los MLG/MLP, para mejorar la efectividad y adaptabilidad de los procesos de prueba con la IA generativa.

5 Despliegue e integración de la IA generativa en las organizaciones de prueba – 80 minutos

Palabras clave

Ninguno

Palabras clave específicas de IA generativa

IA en la sombra (shadow AI)

Objetivos de aprendizaje y objetivos prácticos para el Capítulo 5:

5.1 Hoja de ruta para la adopción de la IA generativa en las pruebas de software

- | | | |
|-------------|------|---|
| GenAI-5.1.1 | (K1) | Recordar los riesgos de la IA en la sombra (shadow AI) |
| GenAI-5.1.2 | (K2) | Explicar los aspectos clave a considerar al definir una estrategia de IA generativa para las pruebas de software |
| GenAI-5.1.3 | (K2) | Resumir los criterios clave para seleccionar los MLG o los MLP para las tareas de prueba de software en un contexto determinado |
| HO-5.1.3 | (H1) | Estimar los costos recurrentes del uso de la IA generativa para una tarea de prueba determinada |
| GenAI-5.1.4 | (K1) | Recordar las fases clave en la adopción de la IA generativa en una organización de prueba |

5.2 Gestionar el cambio al adoptar la IA generativa para las pruebas de software

- | | | |
|-------------|------|---|
| GenAI-5.2.1 | (K2) | Explicar las habilidades esenciales y las áreas de conocimiento requeridas para que los probadores trabajen de manera efectiva con la IA generativa en los procesos de prueba |
| GenAI-5.2.2 | (K1) | Recordar las estrategias para cultivar las habilidades de la IA en los equipos de prueba, con el fin de apoyar la adopción de la IA generativa en las actividades de prueba |
| GenAI-5.2.3 | (K1) | Reconocer cómo cambian los procesos y las responsabilidades de las pruebas dentro de una organización de pruebas al adoptar la IA generativa para las pruebas |

5.1 Hoja de ruta para la adopción de la IA generativa en las pruebas de software

Una estrategia de prueba con la IA generativa debe considerar cuidadosamente los aspectos clave como los objetivos de la prueba que se deben lograr, la selección adecuada de los MLG, los problemas con los datos de entrada utilizados para las instrucciones y el cumplimiento de los estándares y las regulaciones de la IA. Con base en esta estrategia, la organización puede establecer una hoja de ruta y monitorizar el progreso en la integración de la IA generativa en los procesos de prueba.

5.1.1 Riesgos de la IA en la sombra (shadow AI)

La IA en la sombra puede generar riesgos con respecto a la seguridad, el cumplimiento y la privacidad de los datos:

- Debilidades en la seguridad de la información y la privacidad de los datos: Las herramientas personales de IA pueden carecer de una seguridad sólida, lo que puede dar lugar a posibles violaciones de datos.
- Cumplimiento y problemas regulatorios: el uso de herramientas de IA no aprobadas puede llevar al incumplimiento de los estándares y regulaciones de la industria (consulte la Sección 3.4.1), lo que puede tener consecuencias legales.
- Propiedad intelectual difusa: el uso de herramientas de IA con acuerdos de licencia poco claros puede exponer a los usuarios de los MLG a disputas de propiedad intelectual, especialmente si los datos con derechos de autor se procesan sin la autorización adecuada.

Una estrategia y pasos para integrar e implementar la IA generativa pueden ayudar a las organizaciones de prueba a evitar el riesgo de la IA en la sombra.

5.1.2 Aspectos clave de una estrategia de la IA generativa en las pruebas de software

Para implementar con éxito una estrategia la IA generativa en las pruebas, las organizaciones deben considerar cuidadosamente varios factores clave para garantizar una integración fluida y resultados óptimos. Esto comienza con la definición de objetivos de prueba medibles para la IA generativa, como aumentar la productividad de las pruebas, acortar los ciclos de las pruebas y mejorar la calidad de las pruebas. Seleccionar los MLG correctos es fundamental (consulte la sección 5.1.3) y debe alinearse con estos objetivos de prueba, al tiempo que asegura la compatibilidad con la infraestructura de prueba existente y cumple con los requisitos de escalabilidad del sistema.

La calidad de los datos juega un papel fundamental, ya que la efectividad de las pruebas impulsadas por los MLG depende de los datos de entrada precisos y relevantes, protegidos por procedimientos de seguridad sólidos. Por lo tanto, mantener datos de entrada de alta calidad es clave para lograr resultados en los que se pueda confiar que sean correctos.

Se deben proporcionar programas integrales de formación para garantizar que los equipos de prueba tengan las habilidades técnicas y éticas necesarias para utilizar las herramientas de la IA generativa de manera efectiva. Además de la formación, se deben recopilar métricas específicas para medir la efectividad de los resultados de la IA generativa (Ver sección 2.3.1).

Para garantizar el cumplimiento de los estándares reglamentarios y el cumplimiento de las directrices éticas, las organizaciones deben establecer directrices de proceso para el uso de la IA generativa, incluidas las reglas para el uso de datos confidenciales, las obligaciones de transparencia (por ejemplo, lo que se generó utilizando la IA generativa) y los controles de calidad con revisión del testware generado.

5.1.3 Selección de los MLG o modelos de lenguaje pequeños (MLP) para las tareas de prueba de software

Existe una amplia gama de MLG/MLP, cada uno con diferentes capacidades funcionales (por ejemplo, entrada multimodal, capacidades de razonamiento), características técnicas (por ejemplo, tamaño de ventana de contexto) y tipos de licencia (por ejemplo, comercial frente a código abierto). Si bien hay muchos puntos de referencia disponibles para evaluar los MLG/MLP para las tareas como el procesamiento del lenguaje natural, la generación de código o el análisis de imágenes, solo unos pocos se centran específicamente en las tareas de prueba de software (Wenhan 2024). Por lo tanto, la selección de los MLG/MLP para las tareas de prueba requiere una cuidadosa consideración de varios criterios clave:

- Rendimiento del modelo: Evaluar el rendimiento del modelo para las tareas de prueba específicas contra los puntos de referencia de la organización utilizando métricas como las presentadas en la sección 2.3.1.
- Potencial de ajuste fino: Evaluar si es posible y útil ajustar el modelo de lenguaje (MLG o MLP) con datos específicos del dominio para mejorar el rendimiento para un caso de uso determinado, aumentando la exactitud y la relevancia en contextos especializados.
- Costo recurrente: Considere los costos recurrentes de usar el MLG/MLP, incluidas las tarifas de licencia y los gastos operativos, para garantizar que se ajuste al presupuesto de la organización para las tareas de prueba específicas.
- Comunidad y asistencia: Elegir modelos con apoyo activo de la comunidad y documentación detallada para ayudar en la implementación y solución de problemas.

Al evaluar cuidadosamente estos criterios, las organizaciones de prueba pueden seleccionar uno o más MLG/MLP que satisfagan sus necesidades específicas y limitaciones organizativas.

Objetivo práctico HO-5.1.3 (H1): Estimar los costos recurrentes del uso de la IA generativa para una tarea de prueba determinada

Este ejercicio se centra en estimar los costos recurrentes del uso de la IA generativa para una tarea de prueba específica basada en varios supuestos. Estos incluyen factores como el número de tóquenes en los datos de entrada y salida, las instrucciones utilizadas y la frecuencia de la tarea. Se explorarán y compararán los modelos de precios de varios proveedores de MLG/MLP, incluyendo al menos una solución comercial y un modelo con licencia de código abierto.

Este ejercicio ofrece la oportunidad de calcular y experimentar con los costos recurrentes de la IA generativa utilizando condiciones de prueba prácticas, lo que ayuda a comprender las implicaciones financieras de los diferentes enfoques y proveedores.

5.1.4 Fases al adoptar la IA generativa en las pruebas de software

La adopción de la IA generativa dentro de una organización de prueba implica tres fases clave:

1. Descubrimiento: La primera fase se centra en la sensibilización y el desarrollo de capacidades. Las actividades incluyen formar a los equipos de prueba sobre los conceptos de la IA generativa, proporcionar acceso a los MLG/MLP y experimentar con casos de uso iniciales para familiarizar a los probadores con la IA generativa y generar confianza.
2. Inicio y definición del uso: Una vez que se establece la comprensión básica, la segunda fase se centra en identificar y priorizar casos de uso prácticos para la IA generativa en las pruebas de software. Esta fase incluye evaluar la infraestructura de pruebas impulsada por los MLG, desarrollar experiencia y garantizar la alineación con las necesidades de la organización (consulte la sección 6 de [ISTQB_CTFL_SYL]).

3. Uso e iteración: en esta fase avanzada, las organizaciones integran completamente la IA generativa en sus procesos de prueba. Se implementa una monitorización continua del progreso de la IA generativa para las pruebas de software y las herramientas relacionadas, así como la medición y gestión de la transformación para garantizar beneficios sostenibles y la escalabilidad.

Estas fases pueden ejecutarse en paralelo para diferentes casos de uso. Por ejemplo, el análisis del informe de prueba puede estar más avanzado en la hoja de ruta mientras que la automatización de las pruebas está en las primeras fases. También es importante reconocer y abordar las preocupaciones tempranas, como el temor a la pérdida de puestos de trabajo, que puede afectar a la adopción y a la moral del equipo.

5.2 Gestionar el cambio al adoptar la IA generativa para las pruebas de software

La implementación exitosa de la IA generativa en una organización de prueba requiere un enfoque estructurado para los procesos de gestión de cambios. Los aspectos clave incluyen el desarrollo de habilidades esenciales de la IA generativa y la evolución de los roles de prueba tradicionales para adoptar procesos de prueba habilitados para la IA. La transformación involucra tanto habilidades técnicas como aspectos organizativos.

5.2.1 Habilidades y conocimientos esenciales para realizar las pruebas con la IA generativa

La integración exitosa de la IA generativa en las pruebas requiere dominar las técnicas de ingeniería de instrucciones, comprender las ventanas de contexto del modelo y desarrollar métodos de revisión de pruebas. Los probadores deben combinar la experiencia en el dominio y las pruebas con las habilidades de la IA para evaluar las pruebas impulsadas por los MLG en tareas como la generación de casos de prueba, el análisis de informes de defectos y la generación de datos de prueba.

Las competencias clave incluyen la evaluación de las capacidades de los MLG, la comprensión de las técnicas de refinamiento rápido y la evaluación del testware generado por la IA. El conocimiento esencial incluye la comprensión de los riesgos inherentes de la IA generativa, junto con la conciencia de las estrategias comunes de mitigación. Los probadores deben comprender las implicaciones de seguridad de datos al compartir el testware con los MLG, implementar una limpieza de datos adecuada (eliminar o enmascarar información confidencial, personal o confidencial) y seguir prácticas de ingeniería de instrucciones que preserven la privacidad de los datos. Las consideraciones medioambientales incluyen la optimización de la selección de modelos y los patrones de uso para reducir los gastos generales computacionales, la selección de modelos de tamaño adecuado para las tareas de prueba y el equilibrio de los beneficios de la automatización de la IA generativa con el impacto en el costo y el consumo de energía.

5.2.2 Desarrollo de capacidades en la IA generativa en los equipos de prueba

Un enfoque práctico es esencial para formar estratégicamente a los equipos de prueba en la IA generativa para las pruebas. Esto incluye practicar con varios MLG/MLP, seguir rutas de aprendizaje estructuradas y desarrollar gradualmente conocimientos mediante el intercambio dentro de la organización. La formación se centra en el desarrollo de habilidades prácticas a través de ejercicios guiados, aprendizaje entre pares y la integración gradual de la IA en las tareas de prueba diarias.

Los miembros del equipo de pruebas progresan desde el dominio de la creación básica de instrucciones hasta el uso de técnicas más enfocadas, como instrucciones específicas de las pruebas. Un patrón de instrucciones es una plantilla reutilizable para elaborar instrucciones efectivas para guiar a la IA generativa hacia resultados consistentes y fiables. Las comunidades de práctica internas apoyan el intercambio continuo de conocimientos, con reuniones periódicas para destacar las aplicaciones de la IA generativa exitosas, debatir los retos y perfeccionar las mejores prácticas. Estas comunidades fomentan la mejora continua compartiendo bibliotecas de patrones de instrucciones y documentando las lecciones aprendidas de la IA generativa para implementaciones de pruebas en diversos proyectos y dominios.

5.2.3 Evolución de los procesos de prueba en las organizaciones con pruebas potenciadas por la IA

La integración de la IA generativa transforma los procesos de prueba tradicionales de los probadores y gestores de prueba dentro de las organizaciones de prueba.

Los probadores evolucionan de especialistas en diseño y ejecución de pruebas a especialistas en pruebas asistidas por la IA, combinando su experiencia en técnicas de prueba con habilidades para guiar y verificar el testware generado por la IA. Sus tareas de prueba se amplían para incluir la revisión de la producción general basada en la IA, el refinamiento rápido y el mantenimiento de bibliotecas de instrucciones específicas de las pruebas.

Las responsabilidades de los gestores de pruebas se actualizan para incluir el desarrollo de una estrategia de pruebas basada en la IA, la gestión de riesgos basada en la IA y la monitorización y el control de los procesos de prueba basados en la IA. Los gestores de pruebas se centran en equilibrar las capacidades humanas y de la IA, establecer marcos de gobernanza de la IA para casos de uso y asegurar que sus equipos de prueba mantengan tanto las competencias de prueba tradicionales como la alfabetización en la IA. Los gestores de pruebas no solo liderarán a los probadores humanos, sino que también se coordinarán con los agentes de prueba impulsados por IA generativa, lo que requerirá nuevas habilidades de gestión para supervisar los equipos híbridos de personas y las herramientas de IA generativa.

6 Referencias

Estándares

- **ISO/IEC 42001:2023** (2023), Information technology — Artificial intelligence — Management system
- **ISO/IEC 23053:2022** (2022), Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)

Documentos de ISTQB®

- **[ISTQB_CTFL_SYL]** ISTQB® Foundation Level Syllabus v4.0, 2023

Referencias del Glosario

- **Glosario ISTQB®** <https://glossary.istqb.org/>

Libros

- Winteringham M. (2024) *Software Testing with Generative AI*, Manning Publications (5 Mar. 2025), ISBN-13: 978-1633437364, 10 Dec. 2024 - 304 pages

Artículos

- (Berthelot 2024) Berthelot, Adrien, et al. "Estimating the environmental impact of Generative-AI services using an LCA-based methodology." *Procedia CIRP* 122 (2024): 707-712.
- (Gallegos 2024) Gallegos, Isabel O., et al. "Bias and fairness in large language models: A survey." *Computational Linguistics* (2024): 1-79.
- (Li 2024) Yihao Li, Pan Liu, Haiyang Wang, Jie Chu, W. Eric Wong, Evaluating Large Language Models for Software Testing, *Computer Standards & Interfaces* (2024), doi: <https://doi.org/10.1016/j.csi.2024.103942>
- (Luccioni 2024a) Luccioni, Sasha, Yacine Jernite, and Emma Strubell. "Power hungry processing: Watts driving the cost of AI deployment?." *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. 2024.
- (Mailach 2024) Mailach, Alina, et al. "Practitioners' Discussions on Building LLM-based Applications for Production." *arXiv preprint arXiv:2411.08574* (2024).
- (Mirzadeh 2024) Mirzadeh, Iman et al. "GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models." *ArXiv abs/2410.05229* (2024)
- (NIST AI RMF 1.0) National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1, U.S. Department of Commerce, 2023, <https://doi.org/10.6028/NIST.AI.100-1>.

- (Parthasarathy 2024) Parthasarathy, Venkatesh Balavadhani, et al. "The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities." arXiv preprint arXiv:2408.13296 (2024).
- (Schulhoff 2024) Schulhoff, S., "The Prompt Report: A Systematic Survey of Prompting Techniques", Art. no. arXiv:2406.06608, 2024. doi:10.48550/arXiv.2406.06608.
- (Shuyin 2023) Ouyang, Shuyin, et al. "LLM is Like a Box of Chocolates: the Non-determinism of ChatGPT in Code Generation." arXiv preprint arXiv:2308.02828 (2023).
- (Sinha 2024) Sinha, Megha, Sreekanth Menon, and Ram Sagar. "LLMOps: Definitions, Framework and Best Practices." 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET. IEEE, 2024.
- (Wang 2024) Wang, Yanlin, et al. "Agents in Software Engineering: Survey, Landscape, and Vision." arXiv preprint arXiv:2409.09030 (2024).
- (Wenhan 2024) Wang, Wenhan, et al. "TESTEVAL: Benchmarking Large Language Models for Test Case Generation." arXiv preprint arXiv:2406.04531 (2024).
- (Zhao 2024) Zhao, Penghao, et al. "Retrieval-augmented generation for ai-generated content: A survey." arXiv preprint arXiv:2402.19473 (2024).

Páginas Web

(AI Act 2024) European Commission. "European Approach to Artificial Intelligence." *Shaping Europe's Digital Future*, European Commission, <https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence>. Accessed 24 Nov. 2024.

(Heikkilä 2023) Heikkilä, M. (2023, December 1). Making an image with generative AI uses as much energy as charging your phone. MIT Technology Review. Retrieved from <https://www.technologyreview.com/2023/12/01/1084189/making-an-image-with-generative-ai-uses-as-much-energy-as-charging-your-phone/>

(Luccioni 2024b) Luccioni, S. (2024, February 22). Generative AI's environmental costs are soaring. Nature. Retrieved from <https://www.nature.com/articles/d41586-024-00478-x>

(Google Dev Glossary 2024) Google Developers. (n.d.). Machine learning glossary: Generative AI. Retrieved November 24, 2024, from <https://developers.google.com/machine-learning/glossary/generative>

(MIT 2024) "Glossary of Terms: Generative AI Basics." *MIT Sloan Teaching & Learning Technologies*, MIT Sloan School of Management, <https://mitsloanedtech.mit.edu/ai/basics/glossary>. Accessed 24 Nov. 2024.

Las referencias anteriores apuntan a información disponible en Internet y en otros lugares. A pesar de que esas referencias se verificaron en el momento de la publicación de este programa de estudios, el ISTQB® no se hace responsable si las referencias ya no están disponibles.

7 Anexo A – Objetivos de aprendizaje o nivel cognitivo de conocimiento

Los objetivos de aprendizaje específicos que se aplican a este programa de estudios se muestran al comienzo de cada capítulo. Cada tema del programa de estudios se examinará de acuerdo con su objetivo de aprendizaje.

Los objetivos de aprendizaje comienzan con un verbo de acción correspondiente a su nivel cognitivo de conocimiento como se indica a continuación.

Nivel 1: Recordar (K1)

El candidato recordará, reconocerá y rememorará un término o concepto.

Verbos de acción: Recordar, reconocer.

Ejemplos:
Recordar los conceptos de la pirámide de prueba.
Reconocer los objetivos típicos de las pruebas.

Nivel 2: Comprender (K2)

El candidato puede seleccionar las razones o explicaciones para las declaraciones relacionadas con el tema, y puede resumir, comparar, clasificar y dar ejemplos para el concepto de prueba.

Verbos de acción: Clasificar, comparar, diferenciar, distinguir, explicar, dar ejemplos, interpretar, resumir

Ejemplos	Notas
Clasificar las herramientas de prueba de acuerdo con su propósito y las actividades de prueba que apoyan.	
Comparar los diferentes niveles de prueba.	Se puede utilizar para buscar similitudes, diferencias o ambas.
Diferenciar las pruebas de la depuración.	Busca diferencias entre conceptos.
Distinguir entre los riesgos de proyecto y de producto.	Permite clasificar por separado dos (o más) conceptos.
Explicar el impacto del contexto en el proceso de pruebas	
Dar ejemplos de por qué es necesario realizar pruebas.	

Ejemplos	Notas
Inferir la causa raíz de los defectos a partir de un perfil determinado de fallos.	
Resumir las actividades del proceso de revisión de los productos de trabajo.	

Nivel 3: Aplicar (K3)

El candidato puede llevar a cabo un procedimiento cuando se enfrenta a una tarea familiar, o seleccionar el procedimiento correcto y aplicarlo a un contexto determinado.

Verbos de acción: Aplicar, implementar, preparar, usar

Ejemplos	Notas
Aplicar el análisis de valores límite para derivar casos de prueba de requisitos determinados.	Debe referirse a un procedimiento, técnica, proceso, etc.
Implementar métodos de recopilación de métricas para respaldar los requisitos técnicos y de gestión.	
Preparar pruebas de instalabilidad para aplicaciones móviles.	
Utilizar la trazabilidad para monitorizar el progreso de las pruebas para verificar su integridad y coherencia con los objetivos de pruebas, la estrategia de pruebas y el plan de pruebas.	Podría usarse en un objetivo de aprendizaje que quiere que el candidato pueda usar una técnica o procedimiento. Similar a "aplicar".

Referencia

(Para los niveles cognitivos de los objetivos de aprendizaje)

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

8 Anexo B – Matriz de trazabilidad de los resultados de negocio con los objetivos de aprendizaje

Esta sección enumera la trazabilidad entre los resultados de negocio y los objetivos de aprendizaje del Probador Certificado en Pruebas con la IA generativa. Los objetivos prácticos no se mencionan en esta tabla, ya que cada objetivo práctico (HO) está asociado a un único objetivo de aprendizaje (LO). La trazabilidad entre un HO y un resultado de negocio (BO) es a través del objetivo de aprendizaje (LO) al que está asociado el HO.

Resultados de negocio: Probador certificado en pruebas con la IA generativa		BO1	BO2	BO3	BO4	BO5
GenAI-B1	Comprender los conceptos fundamentales, las capacidades y las limitaciones de la IA generativa	8				
GenAI-B2	Desarrollar habilidades prácticas para impulsar grandes modelos de lenguaje para las pruebas de software		10			
GenAI-B3	Adquirir información sobre los riesgos y las mitigaciones del uso de la IA generativa para las pruebas de software			11		
GenAI-B4	Adquirir información sobre las aplicaciones de las soluciones de la IA generativa para las pruebas de software				19	
GenAI-B5	Contribuir de manera efectiva a la definición e implementación de una estrategia de IA generativa y una hoja de ruta para las pruebas de software dentro de una organización					13

Resultados de negocio: Probador certificado en pruebas con la IA generativa			BO1	BO2	BO3	BO4	BO5
LO único	Objetivo de Aprendizaje	Nivel K					
1	Introducción a la IA generativa para las pruebas de software:						
1.1	Fundamentos de la IA generativa y los conceptos clave						
GenAI-1.1.1	Recordar diferentes tipos de IA: IA simbólica, aprendizaje automático clásico, aprendizaje profundo e IA generativa	K1	X				
GenAI-1.1.2	Explicar los conceptos básicos de la IA generativa y los grandes modelos de lenguaje	K2	X				
GenAI-1.1.3	Distinguir entre los MLG fundacionales, ajustados por instrucciones y de razonamiento	K2	X				
GenAI-1.1.4	Resumir los principios fundamentales de los MLG multimodales y los modelos de visión y lenguaje	K2	X				
1.2	Aprovechamiento de la IA generativa en las pruebas de software: Principios fundamentales						
GenAI-1.2.1	Dar ejemplos de capacidades clave de los MLG para las tareas de prueba	K2	X			X	
GenAI-1.2.2	Comparar los modelos de interacción al usar la IA generativa para las pruebas de software	K2	X			X	
2	Ingeniería de instrucciones para las pruebas de software efectivas						
2.1	Desarrollo eficaz de instrucciones						
GenAI-2.1.1	Dar ejemplos de la estructura de las instrucciones utilizadas en la IA generativa para las pruebas de software	K2		X			
GenAI-2.1.2	Diferenciar las técnicas fundamentales de instrucciones para las pruebas de software	K2		X			
GenAI-2.1.3	Distinguir entre las instrucciones de sistema y las instrucciones de usuario	K2		X			
2.2	Aplicación de técnicas de ingeniería de instrucciones a las tareas de prueba de software						
GenAI-2.2.1	Aplicar la IA generativa a las tareas de análisis de prueba	K3		X			
GenAI-2.2.2	Aplicar la IA generativa a las tareas del diseño de pruebas e implementación de pruebas	K3		X			
GenAI-2.2.3	Aplicar la IA generativa a las pruebas de regresión automatizadas	K3		X			
GenAI-2.2.4	Aplicar la IA generativa a las tareas de monitorización de pruebas y control de pruebas	K3		X			
GenAI-2.2.5	Seleccionar y aplicar las técnicas de instrucciones adecuadas para un contexto dado y una tarea de prueba	K3		X		X	
2.3	Evaluar los resultados de la IA generativa y afinar las instrucciones para las tareas de prueba de software						
GenAI-2.3.1	Resumir las métricas para evaluar los resultados de la IA generativa en las tareas de prueba	K2		X	X	X	
GenAI-2.3.2	Dar ejemplos de técnicas para evaluar y refinar iterativamente las instrucciones	K2		X	X	X	

Resultados de negocio: Probador certificado en pruebas con la IA generativa			BO1	BO2	BO3	BO4	BO5
3	Gestión de riesgos de la IA generativa en las pruebas de software						
3.1	Alucinaciones, errores de razonamiento y sesgos						
GenAI-3.1.1	Recordar las definiciones de alucinaciones, errores de razonamiento y sesgos en los sistemas de IA generativa	K1	X		X	X	
GenAI-3.1.2	Identificar las alucinaciones, los errores de razonamiento y los sesgos en los resultados de los MLG	K3			X	X	
GenAI-3.1.3	Resumir las técnicas de mitigación de las alucinaciones de la IA generativa, los errores de razonamiento y los sesgos en las tareas de prueba de software	K2			X	X	
GenAI-3.1.4	Recordar las técnicas de mitigación para el comportamiento no determinista de los MLG	K1	X		X	X	
3.2	Privacidad de datos y riesgos de seguridad de la IA generativa en las pruebas de software						
GenAI-3.2.1	Explicar los riesgos clave de privacidad y seguridad de datos asociados con el uso de la IA generativa	K2			X	X	
GenAI-3.2.2	Dar ejemplos de privacidad de datos y vulnerabilidades en el uso de la IA generativa en las pruebas de software	K2			X	X	
GenAI-3.2.3	Resumir las estrategias de mitigación para proteger la privacidad de los datos y mejorar la seguridad en la IA generativa para las pruebas de software	K2			X	X	
3.3	Consumo de energía e impacto ambiental de la IA generativa en las pruebas de software						
GenAI-3.3.1	Explicar el impacto de las características de las tareas y el uso del modelo en el consumo de energía de la IA generativa en las pruebas de software	K2			X	X	
3.4	Regulaciones, estándares y marcos de trabajo de las mejores prácticas de la IA						
GenAI-3.4.1	Recordar ejemplos de las regulaciones, los estándares y los marcos de trabajo de las mejores prácticas que son relevantes para la IA generativa en las pruebas de software	K1			X	X	X

Resultados de negocio: Probador certificado en pruebas con la IA generativa		BO1	BO2	BO3	BO4	BO5
4	Infraestructura de pruebas impulsada por los MLG para las pruebas de software					
4.1	Enfoques arquitectónicos para la infraestructura de pruebas impulsada por los MLG					
GenAI-4.1.1	Explicar los componentes arquitectónicos clave y los conceptos de la infraestructura de pruebas impulsada por los MLG	K2			X	X
GenAI-4.1.2	Resumir la generación aumentada por recuperación (GAR)	K2			X	X
GenAI-4.1.3	Explicar el rol y la aplicación de los agentes impulsados por los MLG en la automatización de los procesos de prueba	K2			X	X
4.2	Ajuste fino y las OpsMLG: puesta en marcha de la IA generativa para las pruebas de software					
GenAI-4.2.1	Explicar el ajuste fino de los modelos de lenguaje para tareas de prueba específicas	K2			X	X
GenAI-4.2.2	Explicar las OpsMLG y su papel en la implementación y gestión de los MLG para las tareas de prueba	K2			X	X
5	Despliegue e integración de la IA generativa en las organizaciones de prueba					
5.1	Hoja de ruta para la adopción de la IA generativa en las pruebas de software					
GenAI-5.1.1	Recordar los riesgos de la IA en la sombra (shadow AI)	K1				X
GenAI-5.1.2	Explicar los aspectos clave a considerar al definir una estrategia de IA generativa para las pruebas de software	K2				X
GenAI-5.1.3	Resumir los criterios clave para seleccionar los MLG/MLP para las tareas de prueba de software en un contexto determinado	K2				X
GenAI-5.1.4	Recordar las fases clave en la adopción de la IA generativa en una organización de prueba	K1				X
5.2	Gestionar el cambio al adoptar la IA generativa para las pruebas de software					
GenAI-5.2.1	Explicar las habilidades esenciales y las áreas de conocimiento requeridas para que los probadores trabajen de manera efectiva con la IA generativa en los procesos de prueba	K2				X
GenAI-5.2.2	Recordar las estrategias para cultivar las habilidades de la IA en los equipos de prueba, con el fin de apoyar la adopción de la IA generativa en las actividades de prueba	K1				X
GenAI-5.2.3	Reconocer cómo cambian los procesos y las responsabilidades de las pruebas dentro de una organización de pruebas al adoptar la IA generativa para las pruebas	K1				X

9 Anexo C – Notas de entrega

Esta entrega es la v1.0. No hay notas de entrega para esta primera versión.

10 Anexo D – Términos específicos de la IA generativa

Nombre del Término	Definición
Chatbot con IA	Un agente conversacional que utiliza los MLG para procesar consultas y generar respuestas de texto similares a las humanas, lo que permite la comunicación interactiva con los usuarios.
Ventana de contexto	El alcance del texto, medido en tóquenes, que un modelo de lenguaje considera al generar respuestas, influyendo en la relevancia y coherencia de sus resultados.
Aprendizaje profundo	Aprendizaje automático (ML) utilizando redes neuronales con múltiples capas.
Incrustación	Técnica utilizada para representar tóquenes como vectores densos en un espacio continuo, aprendida durante el entrenamiento para capturar relaciones semánticas, sintácticas y contextuales.
Característica	Un atributo medible individual de los datos de entrada utilizados para el entrenamiento por un algoritmo de aprendizaje automático (ML) y para la predicción por un modelo de aprendizaje automático (ML)
Instrucción con pocos ejemplos	Una técnica en la que a un modelo se le dan algunos ejemplos dentro de la instrucción para guiarlo en la generación de respuestas adecuadas.
Ajuste fino	Un proceso de aprendizaje supervisado que utiliza un conjunto de datos de ejemplos etiquetados para actualizar los pesos del MLG y adaptarlos a tareas o dominios específicos.
MLG fundacional	Modelos de propósito general preentrenados en una amplia gama de datos de texto, capaces de predecir la siguiente palabra en función de los patrones lingüísticos aprendidos. Sinónimo: MLG base
IA generativa (GenAI)	Un tipo de sistema de inteligencia artificial que utiliza modelos de aprendizaje automático para generar contenido intelectual (nuevo) que se asemeja al contenido creado por humanos.
Transformador generativo preentrenado (GPT)	Un tipo de modelo de aprendizaje profundo basado en transformadores entrenado previamente en grandes cantidades de datos de texto para comprender y generar texto similar al humano.
Alucinación	Información incorrecta creada por un MLG.
MLG ajustado por instrucciones	Un MLG fundacional entrenado para seguir instrucciones, a menudo reforzado por comentarios para fomentar respuestas correctas.
Gran modelo de lenguaje (MLG)	Un programa informático que utiliza colecciones muy grandes de datos del lenguaje para comprender y producir texto de una manera similar a la forma en que lo hacen los humanos.
Agente impulsado por MLG	Una aplicación que integra el razonamiento, la toma de decisiones y la memoria de los MLG, utilizando herramientas para realizar tareas.

Operaciones de grandes modelos de lenguaje (OpsMLG)	Prácticas y herramientas enfocadas en el despliegue, la monitorización y el mantenimiento de los MLG en entornos de producción.
Aprendizaje automático (ML)	El proceso que utiliza técnicas computacionales para permitir que los sistemas aprendan de los datos o la experiencia (ISO/IEC TR 29119-11).
Metainstrucción	La elaboración de instrucciones de nivel superior que generan instrucciones específicas para explorar o automatizar las capacidades.
Modelo multimodal	Modelos con IA generativa que son capaces de procesar y generar contenido en múltiples tipos de datos, como texto, imágenes y audio.
Procesamiento del lenguaje natural (NLP)	El procesamiento de datos codificados en lenguaje natural por ordenadores para recuperar información y para la representación del conocimiento.
Instrucción con un ejemplo	Una técnica de escritura de instrucciones donde la instrucción contiene un ejemplo para guiar la respuesta del MLG.
Instrucción	Una entrada de lenguaje natural proporcionada para obtener una respuesta específica en la IA generativa y los grandes modelos de lenguaje.
Encadenamiento de instrucciones	Una técnica de instrucciones que implica usar el resultado de una instrucción como entrada para otro, creando una secuencia de instrucciones.
Ingeniería de instrucciones	El proceso de diseño y perfeccionamiento de las instrucciones de entrada para guiar a los MLG hacia la producción de los resultados deseados.
MLG de razonamiento	Un MLG que se basa en modelos ajustados por instrucciones, que refina su capacidad para emular procesos de razonamiento similares a los humanos.
Generación aumentada por recuperación (GAR)	Una técnica que combina las capacidades de los MLG con un recuperador para obtener datos relevantes para generar respuestas precisas y contextualmente relevantes.
IA en la sombra	El uso de herramientas o sistemas con IA generativa dentro de una organización sin aprobación o supervisión formal.
Modelo de lenguaje pequeño (MLP)	Modelos de lenguaje diseñados y entrenados intencionalmente para ser pequeños, ofreciendo un equilibrio entre la eficiencia y la comprensión del lenguaje específico de la tarea.
IA simbólica	Un enfoque de IA que utiliza símbolos, reglas y conocimiento estructurado para modelar el razonamiento.
Instrucción de sistema	Un conjunto de instrucciones predefinido, generalmente oculto a los usuarios del chatbot, que establece consistentemente el contexto, el tono y los límites de las respuestas de un MLG y guía su comportamiento a lo largo de las interacciones.
Temperatura	Un parámetro que controla la aleatoriedad o creatividad de los resultados de un MLG.

Toquenización	El proceso de dividir el texto en unidades más pequeñas para su procesamiento por modelos de lenguaje.
Transformador	Una arquitectura de modelo de aprendizaje profundo que utiliza mecanismos de autoatención para capturar dependencias de largo alcance en secuencias de entrada.
Instrucción de usuario	Una instrucción o consulta introducida por un usuario en un MLG que dirige la respuesta del modelo para cumplir tareas específicas o proporcionar la información deseada.
Base de datos vectorial	Una base de datos optimizada para almacenar y consultar representaciones vectoriales de alta dimensión de datos.
Modelo de visión y lenguaje	Un sistema con IA generativa que procesa conjuntamente datos visuales y textuales para realizar tareas vinculando y generando contenido en ambas modalidades.
Instrucción sin ejemplos	Una técnica de escritura de instrucción donde la instrucción no contiene ejemplos, confiando en el conocimiento preexistente del modelo para generar una respuesta.

11 Anexo E – Marcas comerciales

ISTQB® es una marca registrada del International Software Testing Qualifications Board

Índice

Todos los términos de prueba se definen en el Glosario del ISTQB® (<http://glossary.istqb.org/>).

- 1 criterios de aceptación, 14, 19, 21, 24, 25, 26, 34
- 2 chatbot con IA, 11, 13, 21, 23, 43, 48
- 3 sesgos, 11, 33, 34, 35, 36, 46, 61
- 4 chatbots, 17, 46, 48
- 5 ventana de contexto, 13, 15, 44, 51
- 6 privacidad de datos, 11, 26, 33, 37, 38, 39, 40, 48, 7 49, 52, 61
- 8 aprendizaje profundo, 13, 14, 59, 65
- 9 incrustación, 13, 44
- 10 características, 13, 14
- 11 instrucciones con pocos ejemplos, 19, 20, 21, 22, 23, 24, 12 26, 28
- 13 ajuste fino, 42, 46, 47, 48, 55, 62
- 14 MLG fundacional, 13, 65
- 15 IA generativa, 1, 3, 8, 9, 10, 11, 12, 13, 14, 16 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 17 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 42, 18 45, 47, 48, 49, 50, 51, 52, 53, 58, 59, 60, 19 60, 61, 62, 63, 65, 66
- 20 IA generativa, 8, 9, 11, 13, 19, 20, 31, 33, 34, 21 40, 48, 49, 55, 60, 63
- 22 transformador generativo preentrenado, 13, 14
- 23 alucinación, 33
- 24 modelo de lenguaje grande, 13
- 25 MLG, 11, 13, 15, 17, 18, 21, 22, 23, 24, 25, 26 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 27 42, 43, 44, 46, 47, 48, 50, 51, 52, 54, 55, 28 58, 61, 62, 65, 66, 67
- 29 OpsMLG, 12, 42, 46, 48, 55, 62, 66
- 30 MLG, 11, 12, 13, 14, 15, 16, 17, 18, 21, 23, 31 24, 26, 27, 29, 33, 34, 35, 36, 37, 38, 39, 32 40, 42, 43, 44, 46, 47, 48, 49, 50, 51, 52, 33 58, 61, 65, 66
- 34 aprendizaje automático, 13, 14, 59
- 35 metainstrucción, 19, 21, 22, 23, 24, 26, 29
- 36 modelo multimodal, 13
- 37 procesamiento del lenguaje natural, 19, 24, 51
- 38 instrucciones con un solo ejemplo, 19
- 39 instrucción, 11, 13, 15, 17, 19, 20, 21, 22, 23, 24, 40 25, 26, 27, 31, 32, 35, 36, 43, 44, 52, 53, 41 65, 66, 67
- 42 encadenamiento de instrucciones, 19, 21, 22, 23, 24, 25, 26, 43 36
- 44 ingeniería de instrucciones, 11, 19, 20, 21, 23, 24, 52
- 45 GAR, 44, 48, 66
- 46 razonamiento, 11, 13, 14, 27, 33, 34, 35, 36, 40, 48 43, 46, 47, 51, 59, 61, 65, 66
- 48 recuperación por generación aumentada, 11, 36, 42, 50 43, 44, 62, 66
- 50 seguridad, 11, 33, 37, 38, 39, 40, 48, 50, 52, 61
- 51 IA en la sombra, 49
- 52 MLP, 14, 47, 48, 49, 51, 52
- 53 IA simbólica, 13, 14, 59
- 54 instrucción de sistema, 19, 23
- 55 Instrucción de sistema, 23
- 56 temperatura, 33, 36
- 57 automatización de pruebas, 8, 19, 21, 24, 27, 28, 29, 52
- 58 caso de prueba, 18, 19, 24, 26, 27, 29, 32, 34, 52
- 59 condición de prueba, 19, 24, 30
- 60 datos de prueba, 14, 17, 19, 20, 25, 26, 28, 29, 34, 61 43, 44, 52
- 62 diseño de prueba, 18, 19, 24, 25, 26, 29, 44, 53, 60
- 63 infraestructura de prueba, 37, 42, 43, 44, 46, 48, 50, 64 51, 62

65 informe de prueba, 19, 20, 28,
52
11, 13, 14, 15, 16
67 Toquenización, 15, 67
68 tóquenes, 15, 44, 51, 65
69 transformador, 13, 15, 16, 34,
65
70 instrucción de usuario, 19, 23
76

71 instrucción de usuario, 14, 19, 23, 60
72 base de datos vectorial, 42
73 Modelos de visión y lenguaje, 16
74 vulnerabilidad, 33, 38
75 instrucciones sin ejemplos, 19