

Integración de pruebas visuales automatizadas con Applitoools en el proceso de pruebas funcionales

Integration of automated visual testing with Applitoools in the functional testing process

Liliana Rangel Cano

Dirección General de Cómputo y de
Tecnologías de Información y Comunicación
Universidad Nacional Autónoma de México

lilianarc@unam.mx

ORCID: 0009-0001-4100-4011

Información del reporte:

Licencia Creative Commons



El contenido de los textos es responsabilidad de los autores y no refleja forzosamente el punto de vista de los dictaminadores, o de los miembros del Comité Editorial, o la postura del editor y la editorial de la publicación.

Para citar este reporte técnico:

Rangel Cano, L. (2026). Integración de pruebas visuales automatizadas con Applitoools en el proceso de pruebas funcionales. *Cuadernos Técnicos Universitarios de la DGTIC*, 4 (1), páginas (8 - 18). <https://doi.org/10.22201/dgtic.30618096e.2026.4.1.148>

Resumen

En el marco de las pruebas funcionales de un sistema de información, se llevaron a cabo pruebas visuales con Applitoools, herramienta de pruebas visuales automatizadas que facilita la verificación de interfaces gráficas en múltiples navegadores, dispositivos y resoluciones. Ésta utiliza inteligencia artificial para detectar diferencias visuales. La metodología aplicada permitió confirmar su efectividad como apoyo en el proceso de pruebas, particularmente en la verificación visual y en pruebas de regresión sobre interfaces gráficas. La herramienta permite comparar versiones del software o sitios web con información estática, pero su uso en aplicaciones con contenido dinámico requiere mayor personalización para evitar falsos positivos. Applitoools genera líneas base de comparación específicas para cada ambiente, aunque se puede configurar una línea base unificada para facilitar la detección de inconsistencias visuales derivadas por diferencias en la renderización entre navegadores. Su implementación implica una curva de aprendizaje y demanda recursos tecnológicos adecuados, además de considerar limitaciones de la versión gratuita. La interpretación de los resultados depende del criterio del probador, quien determina si las diferencias corresponden a errores o cambios esperados. Una vez superada la curva de aprendizaje, su uso contribuye a reducir significativamente los tiempos de ejecución en comparación con las pruebas manuales. Asimismo, es necesario considerar aspectos

de seguridad de la información, la reutilización de la automatización y la conveniencia de adquirir una licencia institucional para su uso.

Palabras clave: Applitools, calidad del software, pruebas funcionales de caja negra, pruebas visuales.

Abstract

Within the framework of functional testing of an information system, visual tests were carried out using Applitools, an automated visual testing tool that facilitates the verification of graphical user interfaces across multiple browsers, devices, and resolutions. This tool uses artificial intelligence to detect visual differences. The applied methodology confirmed its effectiveness in supporting the testing process, particularly for visual verification and regression testing of graphical interfaces. The tool enables the comparison of software versions or websites with static information; however, its use in applications with dynamic content requires additional customization to avoid false positives. Applitools generates specific baselines for each environment, although a unified baseline can be configured to facilitate the detection of visual inconsistencies resulting from differences in browser rendering. Its implementation involves a learning curve and requires adequate technological resources, as well as consideration of the limitations of the free version. The interpretation of the results depends on the tester's judgment, who determines whether the differences correspond to errors or expected changes. Once the learning curve is overcome, its use significantly reduces execution times compared to manual testing. Furthermore, it is necessary to consider information security, the reusability of the automation, and the advisability of acquiring an institutional license for its use.

Keywords: Applitools, black box functional testing, software quality, visual testing.

1. INTRODUCCIÓN

Las pruebas funcionales son fundamentales para verificar que los sistemas de información cumplan con los requerimientos establecidos y respondan correctamente a las necesidades de los usuarios. Dentro de este proceso, se verifica tanto la funcionalidad como el diseño gráfico del software en distintos navegadores y tamaños de pantalla. Esto garantiza la correcta visualización y satisfacción del usuario al utilizar las diversas opciones disponibles en diferentes sistemas operativos, resoluciones y ambientes tecnológicos (Pressman & Maxim, 2020; Sommerville, 2016).

El *International Software Testing Qualifications Board* (ISTQB) define las pruebas funcionales como aquellas que “se centran en las funciones que realiza un sistema o componente, mediante la verificación de que la salida producida por el software coincide con la esperada, basada en las especificaciones funcionales” (ISTQB, 2024).

Como parte del proceso de pruebas funcionales, se realizan revisiones visuales de la interfaz de usuario, una tarea que demanda tiempo y recursos considerables. Para optimizar esta actividad, se evaluó la automatización de la validación visual, considerando a Applitools como una alternativa de herramienta tecnológica (Myers, Sandler & Badgett, 2011; Applitools, 2025).

Applitools es una plataforma especializada en pruebas visuales automatizadas para aplicaciones web, móviles y de escritorio. Utiliza inteligencia artificial para comparar las interfaces de una aplicación y detectar diferencias visuales no deseadas, como desplazamientos, cambios en color, alineación o tamaño

de elementos, sin necesidad de definir selectores o ubicaciones específicas en el código (Applitoools, 2025; Nguyen, 2023).

Las pruebas visuales se definen como un proceso mediante el cual se comparan imágenes o representaciones gráficas de una interfaz de usuario con una versión base o esperada, con el fin de identificar diferencias que puedan afectar la experiencia del usuario (Myers, Sandler & Badgett, 2011). Se consideran un complemento a las pruebas funcionales, ya que garantizan la coherencia y consistencia estética de la interfaz, además de su correcto comportamiento (Applitoools, 2025).

Con el objetivo de determinar el grado en que esta herramienta puede apoyar las actividades sustanciales del proceso de pruebas de software, se implementó una prueba piloto en un sistema de información institucional. La funcionalidad evaluada consistió en el acceso a tres formularios de captura de datos disponibles para usuarios de la comunidad universitaria y público externo. Dos formularios solicitan los mismos tres datos, mientras que el tercero requiere únicamente dos de ellos e incluye una funcionalidad que modifica la distribución de los campos. Una vez ingresada la información, el sistema valida los datos y, en caso de ser correctos, despliega un segundo formulario para capturar información complementaria. Por el alcance específico de la funcionalidad evaluada, no se considera necesario profundizar en los detalles del sistema utilizado.

Dado que Applitoools, al basarse en Inteligencia Artificial, tiene riesgos inherentes asociados al manejo de información, la prueba se enfocó exclusivamente en la comparación visual de formularios, sin capturar ni almacenar datos sensibles, con lo que se garantiza la confidencialidad y el cumplimiento de políticas de privacidad institucionales (UNAM, 2023; UNESCO, 2021).

En cumplimiento a las recomendaciones emitidas por la UNAM sobre el uso ético de inteligencia artificial generativa en la docencia, se atendieron los lineamientos que enfatizan la transparencia en el uso de datos, el control por parte de los usuarios y la recopilación mínima necesaria de información, con lo que se respetan los principios éticos y de protección de la dignidad humana (Grupo Académico de Inteligencia Artificial Generativa en Educación, 2023; UNESCO, 2021).

Asimismo, el documento “Recomendaciones para el uso educativo de la Inteligencia Artificial Generativa en la UNAM” (Grupo Académico de Inteligencia Artificial Generativa en Educación, 2025) destaca que la incorporación adecuada de herramientas basadas en IA requiere identificar procesos susceptibles de mejora y promover un uso crítico, responsable y experimental, derivado de experiencias individuales e institucionales. En ese sentido, la evaluación de Applitoools se enmarca en un ejercicio controlado y ético para mejorar procesos de aseguramiento de calidad.

2. DESARROLLO TÉCNICO

Las pruebas visuales dentro del contexto de las pruebas funcionales cumplen una función complementaria, ya que aseguran que la presentación gráfica y la interfaz de usuario sean coherentes, estables y libres de defectos visuales. Detectan inconsistencias como desalineaciones, cambios inadvertidos en el diseño, o diferencias en la renderización entre navegadores que podrían afectar la experiencia del usuario.

2.1 METODOLOGÍA

La metodología empleada, representada en la Figura 1, está basada en enfoques recomendados por estándares de gestión de proyectos y pruebas de software. Ésta se estructuró en cuatro fases: planeación, preparación del ambiente, ejecución, así como análisis y evaluación (Project Management Institute, 2021; ISTQB, 2024). Cada fase estuvo orientada a verificar la utilidad, precisión y aplicabilidad de la herramienta en escenarios reales de pruebas funcionales y visuales.

Figura 1

Metodología aplicada en la prueba piloto del uso de Applitools



Fase 1. Planeación

La planeación de la prueba piloto para revisar la funcionalidad de Applitools fue tratada como un proyecto, donde se definieron objetivos, alcance, recursos y entregables (Project Management Institute, 2021). Sin embargo, dadas las limitaciones de recursos, esta planeación no incluyó un calendario exhaustivo ni una gestión detallada de riesgos, enfocándose principalmente en establecer objetivos específicos, criterios de éxito y métricas de evaluación.

Los objetivos específicos definidos fueron:

- I. Determinar el costo-beneficio en cuestión de tiempo de ejecución respecto a las pruebas manuales.
- II. Evaluar la precisión de la herramienta en la detección de diferencias visuales.
- III. Analizar la viabilidad técnica y ética para integrar Applitools en las actividades de pruebas funcionales.

Debido a la demanda, la mayoría de los proyectos evaluados por el equipo de trabajo de pruebas de software corresponden a sistemas de información, por lo que se delimitó el alcance técnico a la comparación visual en un sistema institucional con funcionalidad genérica y se excluyó la captura o almacenamiento de información sensible o confidencial. Se seleccionó un módulo representativo sin funcionalidad de autenticación, con formularios que presentan variaciones estructurales y de contenido, con el fin de evaluar la capacidad de Applitools para detectar diferencias visuales en diseño, espaciado y disposición de campos, aspectos cruciales para la experiencia del usuario (Myers et al., 2011).

La prueba piloto se integró al flujo habitual del proceso de pruebas funcionales, se le asignó a un probador la aplicación de la prueba en su equipo de trabajo regular, se le proporcionaron 24 horas adicionales para actividades de automatización, ejecución y análisis de resultados. La capacitación en Applitools no formó parte del piloto, ya que el *tester* contaba con conocimientos básicos previos.

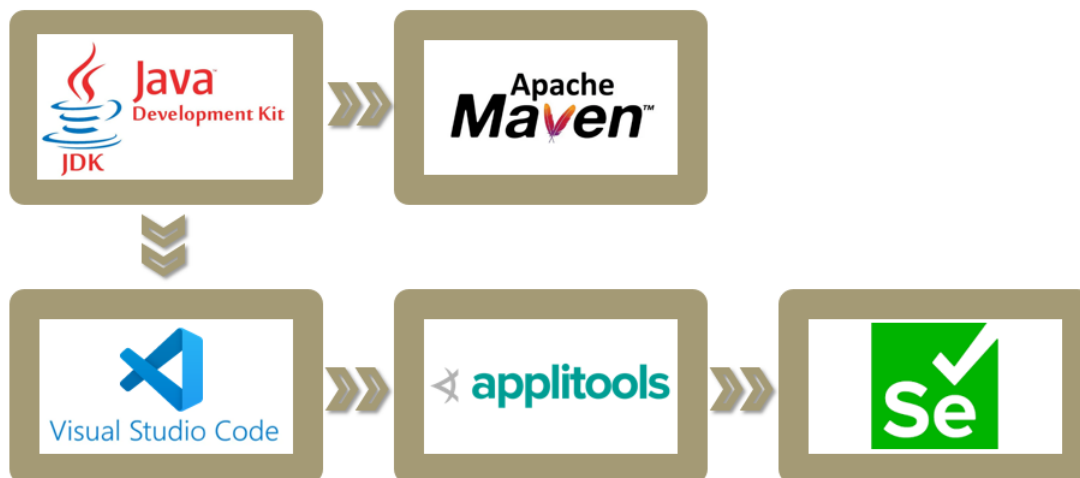
Conforme a buenas prácticas internacionales, se establecieron criterios y métricas de evaluación para decidir sobre la adopción de Applitools. Además, se diseñaron casos de prueba específicos para validar funcionalidades, que incluyen la comparación entre diferentes formularios del mismo sistema, del mismo formulario en distintos tamaños de pantalla y de la interfaz en diversos navegadores.

Fase 2. Preparación del ambiente

Para garantizar la validez técnica y la reproducibilidad de los resultados de la prueba piloto, se configuró un ambiente controlado y se instalaron los componentes mínimos requeridos en sus versiones más recientes. Se aseguró la compatibilidad con las librerías y dependencias más actuales (ISTQB, 2024; SeleniumHQ, 2022). La Figura 2 presenta el software que compone el ambiente para utilizar Applitools.

Figura 2

Ambiente con los recursos de software



Posteriormente, se configuraron las variables de entorno y las dependencias del proyecto, se incluyó la definición de la clave API de Applitools y la actualización de los archivos de configuración de Maven.

Como buena práctica, se verificó la conectividad con los servicios externos (Applitools Eyes Server, repositorios Maven y el navegador).

Durante esta fase, se identificaron errores derivados de actualizaciones en el SDK de Applitools, que modificaron la forma de definir las áreas de captura, lo cual requirió ajustes para estabilizar la automatización. Este hallazgo destacó un riesgo técnico de compatibilidad de versiones, que debe considerarse en futuras implementaciones.

Se estableció la *baseline* visual o imagen de referencia, la cual corresponde al estado esperado de un formulario del sistema.

Fase 3. Ejecución

En esta fase, se desarrollaron y ejecutaron los casos de prueba automatizados con la herramienta Applitools. El proceso inició al retomar la *baseline* visual, que sirvió como punto de comparación para detectar diferencias.

Las pruebas consistieron en las siguientes tareas estructuradas:

I. Ajustar el caso de prueba

De acuerdo con el propósito de la herramienta Applitools, una vez automatizada la prueba, ésta debería ejecutarse de forma repetible sin necesidad de modificaciones, incluso ante ajustes menores en el código fuente o el ambiente visual (Applitools, 2025). No obstante, en el contexto de la prueba piloto, fue necesario realizar ciertos ajustes manuales para verificar el comportamiento de la herramienta bajo condiciones diversas, tales como cambiar la URL del formulario a evaluar, modificar los navegadores o el tamaño de la pantalla para validar la consistencia visual, e incorporar datos dinámicos para evitar inconsistencias. Estas configuraciones permitieron simular escenarios, considerando la diversidad de ambientes en los que los usuarios finales interactúan con los sistemas institucionales.

II. Ejecutar el caso de prueba

Se enviaron las ejecuciones al panel de Applitools, donde el motor de validación visual asistido por inteligencia artificial procesó las capturas y comparó con la *baseline* visual. El enfoque de Applitools combina *machine learning* y análisis perceptual, lo que reduce falsos positivos y mejora la eficacia respecto a herramientas tradicionales (Myers, Sandler & Badgett, 2011; Applitools, 2025). Como resultado de las ejecuciones, Applitools presentó un informe visual con las diferencias detectadas. Esta funcionalidad permitió una evaluación objetiva y reproducible de la interfaz gráfica.

III. Analizar las diferencias visuales

Una vez obtenidos los resultados, el probador fue responsable de analizar las diferencias detectadas por Applitools. Se observó que la herramienta puede identificar una gran cantidad de diferencias, incluyendo los cambios esperados, lo cual requiere una revisión cuidadosa por parte del especialista en pruebas para distinguir entre defectos reales y variaciones por los cambios aplicados.

IV. Determinar la causa de las diferencias

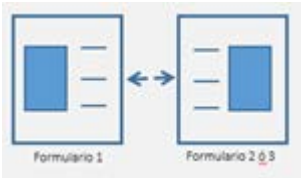
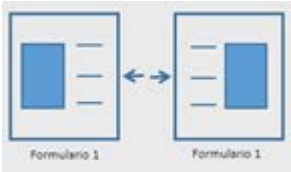
Finalmente, el probador evaluó si las diferencias eran esperadas (por ejemplo, asociados a los cambios en el contenido) o no esperadas (errores visuales o inconsistencias). Esta tarea, aunque automatizada parcialmente, puede demandar más tiempo de análisis que una revisión manual cuando se presentan

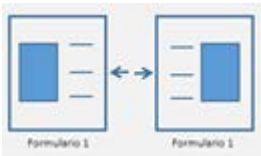
numerosas diferencias, especialmente si la interfaz es dinámica o dependiente de datos variables. Por lo anterior, al analizar los resultados, el probador generó ajustes en la automatización del caso de prueba para que, al aplicar nuevamente la prueba, AppliTools eliminará inconsistencias inexistentes.

En la Tabla 1 se presentan los resultados obtenidos en las pruebas aplicadas:

Tabla 1

Resultados obtenidos en las pruebas realizadas

Prueba	Resultados	
	Esperados	Obtenidos
Comparar diferentes formularios para conocer la funcionalidad de la herramienta. 	Se espera que AppliTools reconozca consistentemente los mismos elementos visuales en los diferentes formularios evaluados. AppliTools genera una línea base (baseline) a partir de las imágenes de referencia de los formularios. Los recuadros rojos destacan las diferencias visuales identificadas por el usuario entre el baseline y la versión actual del formulario. 	AppliTools identificó las diferencias en los textos y las resaltó en ambas pantallas con un sombreado rosa. Como parte de la prueba, se configuró la automatización para ignorar los desfases de posición, por lo que no se mostraron diferencias en ese aspecto, como en el caso del recuadro de solicitud de datos. 
Comparar un mismo formulario en diferentes tamaños de pantalla para determinar las diferencias que presenta la herramienta. 	Se espera que AppliTools reconozca consistentemente los elementos en diferentes tamaños de pantalla, a partir de la línea base, y marcar únicamente las diferencias entre elementos. 	A partir de la línea base generada por AppliTools para un tamaño de pantalla específico, se identificaron diferencias al comparar con otros tamaños, las cuales fueron resaltadas mediante un sombreado en color rosa. Al realizar la comparación con un tamaño distinto, prácticamente la totalidad de la pantalla fue marcada como diferente. 

Prueba	Resultados	
	Esperados	Obtenidos
Comparar un mismo formulario en diferentes navegadores respecto a uno establecido para determinar las diferencias que presenta la herramienta.	Se espera que Applitoools reconozca consistentemente los elementos en diferentes navegadores, a partir de la línea base, y marcar únicamente las diferencias entre elementos.	A partir de la línea base generada por Applitoool, no se detectaron diferencias entre navegadores cuando se mantuvo el mismo tamaño de pantalla; en caso contrario, se presentaron variaciones en la distribución de los elementos.
		

Durante la ejecución, se identificaron aspectos positivos de la herramienta, como la posibilidad de realizar múltiples comparaciones en una sola ejecución, la capacidad de agrupar pruebas en lotes, lo que facilita la gestión de resultados, ignorar diferencias causadas por desplazamientos y la flexibilidad en la generación de la *baseline* visual del área visible en pantalla o de la página completa al desplazarse de manera vertical. Éstos elementos confirman que Applitoools es una herramienta robusta para la automatización de pruebas visuales en escenarios donde la interfaz gráfica es crítica para la calidad.

Fase 4. Análisis y evaluación

Finalmente, se realizó un análisis entre los resultados obtenidos mediante Applitoools y los derivados de las pruebas manuales tradicionales. Esta fase tuvo como objetivo determinar la viabilidad técnica y operativa para incorporar la herramienta dentro del proceso institucional de pruebas de software. Se evaluó su precisión, eficiencia y aplicabilidad en distintos escenarios.

En la Tabla 2 se presenta la comparativa del tiempo invertido en la prueba automatizada respecto a la manual, así como los recursos de software necesarios.

Tabla 2

Comparativa del tiempo invertido y recursos necesarios en la prueba automatizada respecto a la manual

Elemento de evaluación	Prueba	
	Manual	Automatizada
Tiempo invertido		
Automatización	No aplica	La automatización del caso de prueba: 2 h

Elemento de evaluación	Prueba	
	Manual	Automatizada
Tiempo invertido		
Ajustes aplicados en cada prueba	No aplica	Ajustes por cada prueba: de 5 a 30 min (depende del ajuste a realizar)
Aplicación de la prueba	Por cada comparación entre 2 navegadores o tamaños: 10 min	Inmediato
Identificación de defectos asociados al diseño	Consideraciones: La identificación de defectos depende tanto de la experiencia del probador como de su nivel de observación.	Consideraciones: La identificación de diferencias es exacta, sin embargo, determinar si es algo esperado o no depende del probador.
Recursos de software		
Consumo de recursos	Navegadores y emuladores de dispositivos o herramientas para desarrolladores de Chrome se simularon diferentes tamaños de pantalla	Java Development Kit (JDK), para ejecutar scripts de prueba. Maven o Gradle, administrador de dependencias del proyecto. IDE de desarrollo (IntelliJ IDEA, Visual Studio Code o Eclipse), ambiente para codificación e integración. Applitools SDK, biblioteca para integrar Applitools con frameworks de prueba. Framework de automatización (Selenium, Cypress, Playwright o WebdriverIO), herramienta base para ejecutar pruebas funcionales. Cuenta en Applitools Cloud (API Key activa), plataforma principal para almacenar, analizar y visualizar resultados. Dashboard de Applitools Eyes (Accesible desde navegador -Chrome, Edge, Firefox-), interfaz web para revisión y gestión de resultados visuales.

3. RESULTADOS

Applitools se enfoca en comparar una imagen o un conjunto de imágenes base en diferentes navegadores y resoluciones respecto a sus diversas versiones, conforme evoluciona el *software*. Permite comparar pantallas con funcionalidad similar mediante la automatización de casos de prueba, centrándose en la revisión de regiones específicas (elementos fijos de pantalla) como encabezados, menús y títulos, e ignorando diferencias en datos dinámicos o en desplazamientos. Al comparar una imagen base con distintos tamaños de pantalla o diferentes formularios produce inconsistencias causadas por diferencias estructurales y de contenido.

Por defecto, Applitools crea una línea base distinta para cada navegador y resolución (además de diferentes tamaños que consideren o no un diseño responsivo), aunque es posible unificar estas referencias visuales para todos los ambientes. Resultó muy efectiva para comparar una misma interfaz entre diferentes navegadores, siempre que se mantengan la resolución y el tamaño de pantalla. Se recomienda generar un conjunto de imágenes base en distintos navegadores y resoluciones para compararlas durante las pruebas de regresión, las cuales verifican qué cambios en el software no introducen errores en funcionalidades previamente validadas (ISTQB, 2024).

Por lo anterior, se concluye que la herramienta resulta especialmente útil en pruebas de regresión cuando se mantiene el mismo contexto, usuario, elementos de pantalla y volumen de información. También es adecuada para revisar sitios y portales web. No obstante, puede requerir personalización al comparar funcionalidades con contenido dinámico, como carruseles, noticias o elementos de interacción con los usuarios. Asimismo, puede integrarse en pruebas funcionales de sistemas de información, ya que permite comparar diferentes resultados esperados según las entradas ingresadas. Sin embargo, su uso exige un conocimiento previo del sistema y una inversión considerable de tiempo en la automatización.

En cuanto al tiempo total de ejecución y revisión, se observó un incremento inicial debido a la curva de aprendizaje y la necesidad de ajustar la automatización para los casos de prueba específicos. Es importante mencionar que, para probadores con menor experiencia, la curva de aprendizaje puede ser significativa. Es fundamental contar con equipos de cómputo adecuados (Procesador Intel i5 o superior, 8 GB RAM, 10 GB de espacio libre) que aseguren pruebas fluidas.

Una vez comprendido su alcance y uso, las pruebas automatizadas con Applitools reducen significativamente el tiempo invertido respecto a las pruebas manuales, especialmente al ejecutar múltiples validaciones visuales simultáneamente y automatizar la detección de inconsistencias. Es posible realizar pruebas en paralelo en distintos navegadores y dispositivos, lo que amplía la cobertura.

Aunque Applitools genera comparaciones entre pantallas en diferentes navegadores y resoluciones, la interpretación final de las diferencias recae en el probador, quien debe determinar si se trata de errores o de cambios esperados.

4. CONCLUSIONES

La metodología aplicada permitió validar de manera objetiva la efectividad de Applitools como herramienta de apoyo en las actividades del proceso de pruebas, destacando su valor en la verificación visual.

Applitools resulta particularmente viable y eficiente para pruebas de regresión en ambientes con interfaces gráficas estables, facilita la detección precisa de diferencias entre versiones o en la revisión de sitios y portales web con información estática.

La aplicación de pruebas visuales en sistemas de información con contenido dinámico presenta retos importantes, que requieren una planificación detallada, un manejo controlado de las líneas base y una capacitación técnica adecuada para maximizar su eficacia.

Debido a que el uso de Applitools se hace en la nube, se recomienda extremar precauciones respecto a las imágenes a generar, con el fin de evitar la captura de información sensible y confidencial. Por lo anterior, se sugiere el uso de datos de prueba para garantizar la privacidad durante las pruebas de regresión.

La creación de casos de prueba bien diseñados permitirá la reutilización de la automatización en distintas secciones y sistemas, lo que reducirá los tiempos de generación y ejecución en general.

Para incorporar la herramienta en las actividades de pruebas de software, es importante considerar que la versión gratuita de Applitools está limitada tanto en funcionalidad como en número de ejecuciones; por ello, para su uso en todos los proyectos, se observa la necesidad de evaluar la adquisición de una licencia de pago.

REFERENCIAS

- Applitools. (2025). *AI-powered Visual Testing Overview*. Recuperado de <https://applitools.com>
- Grupo Académico de Inteligencia Artificial Generativa en Educación UNAM. (2025). *Recomendaciones para el uso educativo de la inteligencia artificial generativa en la UNAM*. Coordinación de Evaluación, Innovación y Desarrollo Educativos (CEIDE). <https://www.ceide.unam.mx/index.php/recomendaciones-para-el-uso-educativo-de-la-inteligencia-artificial-generativa-en-la-unam/>
- International Software Testing Qualifications Board (ISTQB). (2024). *ISTQB glossary* (Versión 4.6.2). Recuperado de https://glossary.istqb.org/en_US/search?term=&exact_matches_first=true
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). John Wiley & Sons.
- Pressman, R. S., & Maxim, B. R. (2020). *Ingeniería de Software: Un enfoque práctico* (9a ed.). McGraw-Hill.
- Project Management Institute. (2021). *A guide to the project management body of knowledge (PMBOK® Guide)* (7ª ed.). Project Management Institute.
- Selenium HQ. (2022). *WebDriver* [Documentación]. Recuperado de <https://www.selenium.dev/documentation/webdriver/>
- UNESCO. (2021). *Recommendation on the Ethics of Artificial Intelligence* (SHS/BIO/PI/2021/1). Paris: UNESCO. <https://unesdoc.unesco.org/ark:/48223/pf0000381137>
- Universidad Nacional Autónoma de México. (2023, Octubre). *Recomendaciones para el uso de la inteligencia artificial generativa en la docencia* [PDF]. Recuperado de <https://cuaed.unam.mx/descargas/recomendaciones-uso-iagen-docencia-unam-2023.pdf>