

**Probador Certificado de ISTQB®**  
**Programa de Estudio**  
**Nivel Avanzado**  
**Ingeniería de Automatización de la Prueba**  
**Versión ES V00.18**

Traducción realizada por  
**Spanish Software Testing Qualifications Board**

Traducción del Programa de Estudio  
“ISTQB® Certified Tester - Advanced Level - Test Automation Engineering,  
Version V2.0”



## Nota sobre Derechos de Propiedad Intelectual

Nota sobre Derechos de Propiedad Intelectual (Copyright) © International Software Testing Qualifications Board (en adelante denominado ISTQB®) ISTQB® es una marca registrada de International Software Testing Qualifications Board.

Nota sobre Derechos de Propiedad Intelectual (Copyright) © 2024 los autores del programa de estudio "ISTQB® Certified Tester - Advanced Level - Test Automation Engineering, Version V2.0": Andrew Pollner (Presidente), Péter Földházi, Patrick Quilter, Gergely Ágnesz, László Szikszai

Nota sobre Derechos de Propiedad Intelectual (Copyright) © 2016 los autores Andrew Pollner (Presidente), Bryan Bakker, Armin Born, Mark Fewster, Jani Haukinen, Raluca Popescu, Ina Schieferdecker.

Todos los derechos reservados. Por la presente, los autores ceden los derechos de autor en favor de ISTQB®. Los autores (como actuales titulares de los derechos de autor) y el ISTQB® (como futuro titular de los derechos de autor) han acordado las siguientes condiciones de uso:

- Se podrán copiar extractos, para uso no comercial, de este documento siempre que se cite la fuente. Cualquier Proveedor de Formación Acreditado puede utilizar este programa de estudio como referencia 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 estudio y siempre que cualquier anuncio de dicho curso de formación pueda mencionar el programa de estudio sólo después de haber recibido la Acreditación oficial de los materiales de formación por parte de un Comité Miembro reconocido por el ISTQB®.
- Cualquier individuo o grupo de individuos puede utilizar este programa de estudio como referencia 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 estudio.
- Cualquier otro uso de este programa de estudio está prohibido sin obtener previamente la aprobación por escrito del ISTQB®.
- Cualquier Comité Miembro reconocido por el ISTQB® puede traducir este programa de estudio siempre y cuando reproduzca la Nota sobre Derechos de Autor antes mencionada en la versión traducida del programa de estudio.

## Historial de Revisiones

Versión	Fecha	Observaciones
Syllabus 2016	2016/10/21	CTAL-TAE GA Release
Syllabus v2.0	2024/05/03	CTAL-TAE v2.0 GA Release

BORRADOR

Historial de Revisiones – Traducción al idioma español		
Versión	Fecha	Observaciones
Syllabus 2016	2016/10/21	CTAL-TAE GA Release
Syllabus v2.0	2024/05/03	CTAL-TAE v2.0 GA Release

BORRADOR

# Tabla de Contenidos

Nota sobre Derechos de Propiedad Intelectual.....	2
Historial de Revisiones .....	3
Tabla de Contenidos.....	5
Agradecimientos .....	8
Notas de la Versión en Idioma Español.....	9
0 Introducción.....	10
0.1 Objetivo de este programa de estudio.....	10
0.2 Ingeniería de automatización de la prueba en la prueba de software.....	10
0.3 Trayectoria profesional para probadores e ingenieros de automatización de la prueba.....	11
0.4 Resultados de negocio .....	12
0.5 Objetivos de aprendizaje evaluables y nivel cognitivo de conocimiento.....	13
0.6 El examen del certificado de nivel avanzado de gestión de la prueba.....	13
0.7 Acreditación .....	14
0.8 Tratamiento de los estándares .....	14
0.9 Mantenerse actualizado.....	14
0.10 Nivel de detalle .....	14
0.11 Cómo está organizado este programa de estudio .....	15
1 Introducción y objetivos de la automatización de la prueba - (K-2).....	17
1.1 Propósito de la automatización de la prueba.....	19
1.1.1 Explicar las ventajas y desventajas de la automatización de la prueba .....	19
1.2 La automatización de la prueba en el ciclo de vida de desarrollo de software .....	20
1.2.1 Explicar cómo se aplica la automatización de la prueba en los diferentes modelos de ciclo de vida del desarrollo de software.....	20
1.2.2 Seleccionar las herramientas de automatización de prueba adecuadas para un sistema sujeto a prueba determinado.....	20
2 Preparación para la automatización de la prueba - (K4).....	22
2.1 Comprender la configuración de una infraestructura para hacer posible la automatización de la prueba 24	
2.1.1 Describir las necesidades de configuración de una infraestructura que posibilite la implementación de la automatización de la prueba.....	24
2.1.2 Explicar cómo se utiliza la automatización de la prueba en diferentes entornos.....	24
2.2 Proceso de evaluación para seleccionar las herramientas y estrategias adecuadas .....	26
2.2.1 Analizar un sistema sujeto a prueba para determinar la solución de automatización de la prueba adecuada.....	26
2.2.2 Aportar ejemplos de hallazgos técnicos de la evaluación de una herramienta .....	26
3 Arquitectura de automatización de la prueba - (K3).....	29
3.1 Conceptos de diseño utilizados en la automatización de la prueba.....	31
3.1.1 Explicar las principales capacidades en una arquitectura de automatización de la prueba	31
3.1.2 Explicar cómo diseñar una solución de automatización de la prueba .....	32
3.1.3 Aplicar la estructura por capas de los marcos de automatización de la prueba .....	33
3.1.4 Aplicar diferentes enfoques para automatizar casos de prueba.....	34
3.1.5 Aplicar principios de diseño y patrones de diseño en la automatización de pruebas .....	38
4 Implementación de la automatización de la prueba – (K4).....	39
4.1 Desarrollo de la automatización de la prueba.....	41

4.1.1	Aplicar directrices que soporten actividades efectivas de piloto y despliegue de automatización de la prueba .....	41
4.2	Riesgos asociados al desarrollo de la automatización de la prueba.....	42
4.2.1	Analizar los riesgos del despliegue y planificar estrategias de mitigación para la automatización de la prueba .....	42
4.3	Mantenibilidad de la solución de automatización de la prueba .....	43
4.3.1	Explicar los factores que soportan y afectan la mantenibilidad de la solución de automatización de la prueba .....	43
5	Estrategias de implementación y despliegue para la automatización de la prueba - (K3).....	45
5.1	Integración en canalizaciones de integración continua/entrega continua (IC/EC) .....	47
5.1.1	Aplicar la automatización de la prueba en diferentes niveles de prueba dentro de las canalizaciones.....	47
5.1.2	Explicar la gestión de la configuración para el producto de prueba.....	48
5.1.3	Explicar las dependencias de automatización de pruebas para la infraestructura de una IPA	49
6	Suministro de información de la automatización de la prueba y métricas .....	50
6.1	Recopilación, análisis y suministro de información de los datos de automatización de la prueba	52
6.1.1	Aplicar métodos de recopilación de datos de la solución de automatización de la prueba y del sistema sujeto a prueba .....	52
6.1.2	Analizar los datos de la solución de automatización de la prueba y del sistema sujeto a prueba para comprender mejor los resultados .....	55
6.1.3	Explicar cómo se elabora y publica un informe del avance de la prueba .....	56
7	Verificación de la solución de automatización de la prueba - (K3) .....	58
7.1	Verificación de la infraestructura de automatización de la prueba .....	60
7.1.1	Planificar la verificación del entorno de automatización de la prueba, incluyendo la configuración de la herramienta de prueba .....	60
7.1.2	Explicar el comportamiento correcto para un guion de prueba automatizado y/o un juego de prueba dado .....	61
7.1.3	Identificar dónde la automatización de la prueba produce resultados inesperados.....	62
7.1.4	Explicar cómo el análisis estático puede ayudar a la calidad del código de automatización de pruebas .....	62
8	Mejora continua - (K4).....	64
8.1	Oportunidades de mejora continua para la automatización de la prueba .....	66
8.1.1	Descubrir oportunidades para mejorar casos de prueba a través de la recopilación y análisis de datos.....	66
8.1.2	Analizar los aspectos técnicos de una solución de automatización de la prueba implementada y proporcionar recomendaciones para la mejora .....	67
8.1.3	Reestructurar el producto de prueba automatizado para alinearlos con las actualizaciones del sistema sujeto a prueba (SSP).....	70
8.1.4	Resumir las oportunidades para el uso de herramientas de automatización de la prueba..	71
9	Referencias .....	73
9.1	Estándares.....	73
9.2	Documentos de ISTQB® .....	74
9.3	Libros.....	74
9.4	Artículos.....	75

Anexo A - Objetivos de aprendizaje/nivel cognitivo de conocimiento .....	76
Anexo B - Matriz de trazabilidad de los resultados de negocio con respecto a los objetivos de aprendizaje 78	
Apéndice C - Notas de la entrega .....	83
Apéndice D - Términos específicos del dominio .....	84

BORRADOR

## Agradecimientos

Este documento fue entregado formalmente por la Asamblea General del ISTQB® el 3 de mayo de 2024. Fue elaborado por Test Automation Task Force of the Specialist Working Group from the International Software Testing Qualifications Board: Graham Bath (Specialist Working Group Chair) Andrew Pollner (Specialist Working Group Vice Chair and Test Automation Task Force Chair), Péter Földházi, Patrick Quilter, Gergely Ágneecz, László Szikszai. Test Automation Task Force reviewers included: Armin Beer, Armin Born, Geza Bujdoso, Renzo Cerquozzi, Jan Giesen, Arnika Hryszko, Kari Kakkonen, Gary Mogyorodi, Chris van Bael, Carsten Weise, Marc-Florian Wendland.

Revisor técnico: Gary Mogyorodi

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

Horváth Ágota, Laura Albert, Remigiusz Bednarczyk, Jürgen Beniermann, Armin Born, Alessandro Collino, Nicola De Rosa, Wim Decoutere, Ding Guofu, Istvan Forgacs, Elizabeta Fournet, Sudhish Garg, Jan Giesen, Matthew Gregg, Tobias Horn, Mattijs Kemmink, Hardik Kori, Jayakrishnan Krishnankutty, Ashish Kulkarni, Vincenzo Marrazzo, Marton Matyas, Patricia McQuaid, Rajeev Menon, Ingvar Nordström, Arnd Pehl, Michaël Pilaeten, Daniel Polan, Nishan Portoyan, Meile Posthuma, Adam Roman, Pavel Sharikov, Péter Sótér, Lucjan Stapp, Richard Taylor, Giancarlo Tomasig, Chris Van Bael, Koen Van Belle, Johan Van Berkel, Carsten Weise, Marc-Florian Wendland, Ester Zabar.

ISTQB Working Group Advanced Level Test Automation Engineer (Edition 2016): Andrew Pollner (Chair), Bryan Bakker, Armin Born, Mark Fewster, Jani Haukinen, Raluca Popescu, Ina Schieferdecker.



## Notas de la Versión en Idioma Español

Este “Programa de Estudio de Probador Certificado del ISTQB®” de Nivel Avanzado, Ingeniería de Automatización de la Prueba” ha sido traducido por Spanish Software Testing Qualifications Board (SSTQB).

El equipo de traducción y revisión para este programa de estudio es el siguiente (por orden alfabético):

**Responsable de la traducción:** Gustavo Márquez Sosa (España)

**Responsable de la revisión:** Luisa Morales Gómez Tejedor (España)

El Comité Ejecutivo del SSTQB agradece toda aportación que permita mejorar esta traducción del programa de estudio.

En una siguiente versión se podrán incorporar aportaciones adicionales. El SSTQB considera conveniente mantener abierta la posibilidad de realizar cambios en el “Programa de Estudio”.

Madrid, 07 de enero de 2025

## 0 Introducción

### 0.1 Objetivo de este programa de estudio

Este programa de estudio constituye la base para la formación como Probador Certificado del ISTQB® de Nivel Avanzado, Ingeniería de Automatización de la Prueba. El ISTQB proporciona este programa de estudio en los siguientes términos:

1. A los comités miembro, para que lo traduzcan a su idioma local y acrediten a los proveedores de formación. Los comités miembro pueden adaptar el programa de estudio a sus necesidades lingüísticas particulares y modificar las referencias para adaptarlas a sus publicaciones locales.
2. A los organismos de certificación, para que obtengan preguntas de examen en su idioma local adaptadas a los objetivos de aprendizaje de este programa de estudio.
3. A los proveedores de formación, para que elaboren material para formación y determinen los métodos de enseñanza adecuados.
4. A los candidatos a la certificación, para preparar 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 hacer avanzar la profesión de prueba de software y sistemas, y como fuente de libros y artículos.

### 0.2 Ingeniería de automatización de la prueba en la prueba de software

La cualificación en ingeniería de automatización de la prueba está dirigida a cualquier persona relacionada con la prueba de software y la automatización de la prueba. Esto incluye a personas con roles como probadores, analistas de prueba, ingenieros de automatización de la prueba, consultores de prueba, arquitectos de prueba, gestores de prueba y desarrolladores de software. Esta cualificación también es adecuada para cualquier persona que desee una comprensión básica de la automatización de la prueba, como directores de proyecto, directores de calidad, directores de desarrollo de software, analistas de negocio, directores de TI y consultores de gestión.

El programa de estudio de ingeniería de automatización de la prueba está dirigido al ingeniero de automatización de la prueba que busca implementar o mejorar la automatización de la prueba. Define métodos y prácticas que pueden respaldar una solución sostenible.

Otras directrices y modelos de referencia relacionados con las soluciones de automatización de la prueba son los estándares de ingeniería de software para los ciclos de vida de desarrollo de software seleccionados, las tecnologías de programación y los estándares de formato. Este programa de estudio no enseña ingeniería de software. Sin embargo, se espera que un ingeniero de automatización de la prueba tenga habilidades, experiencia y conocimientos en ingeniería de software.

Además, un ingeniero de automatización de la prueba necesita conocer los estándares de programación y documentación de la industria y las buenas prácticas para hacer uso de ellos mientras desarrolla una solución de automatización de la prueba. Estas prácticas pueden aumentar la mantenibilidad, fiabilidad y seguridad de la solución de automatización de la prueba. Estos estándares suelen basarse en características de calidad.

### 0.3 Trayectoria profesional para probadores e ingenieros de automatización de la prueba

El esquema ISTQB® proporciona apoyo a los profesionales de pruebas en todas las etapas de su carrera ofreciendo tanto amplitud como profundidad de conocimientos. Las personas que obtengan la certificación ISTQB® Ingeniero de automatización de la prueba también pueden estar interesadas en la cualificación Estrategia de automatización de la prueba (CT-TAS).

Las personas que obtengan la certificación ISTQB® Certified Tester de Ingeniería de Automatización de Pruebas también pueden estar interesadas en los niveles básicos avanzados (analista de pruebas, analista de pruebas técnicas y jefe de pruebas) y a partir de ahí en el nivel experto (gestión de la prueba o mejora del proceso de prueba). Cualquiera que busque desarrollar habilidades en prácticas de pruebas en un área de entorno Ágil podría tener en cuenta las certificaciones de Probador Técnico Ágil o Liderazgo de Pruebas Ágiles a Escala. La corriente de Especialista ofrece una inmersión profunda en áreas que tienen enfoques de prueba y actividades de prueba específicos, por ejemplo, en Estrategia de automatización de pruebas, Pruebas de rendimiento, Pruebas de seguridad, Pruebas de IA y Pruebas de aplicaciones móviles, o donde se requieren conocimientos específicos de dominio (por ejemplo, Pruebas de software de automoción o Pruebas de juegos). Visite [www.istqb.org](http://www.istqb.org) para obtener la información más reciente del Esquema de probador certificado de ISTQB®.

## 0.4 Resultados de negocio

Esta sección enumera los resultados de negocio esperados de un candidato que haya obtenido la certificación de ingeniero de automatización de la prueba.

Un candidato que haya logrado la certificación de Ingeniero de automatización de la prueba puede...

- TAE-B01** Describir la finalidad de la automatización de la prueba

---

- TAE-B02** Comprender la automatización de la prueba a través del ciclo de vida de desarrollo del software

---

- TAE-B03** Comprender la configuración de una infraestructura para hacer posible la automatización de la prueba

---

- TAE-B04** Conocer el proceso de evaluación para seleccionar las herramientas y estrategias adecuadas

---

- TAE-B05** Comprender los conceptos de diseño para construir soluciones de automatización de la prueba modulares y escalables

---

- TAE-B06** Seleccionar un enfoque, incluido un piloto, para planificar la implantación de la automatización de la prueba dentro del ciclo de vida del desarrollo del software

---

- TAE-B07** Diseñar y desarrollar soluciones de automatización de la prueba (nuevas o modificadas) que satisfagan las necesidades técnicas

---

- TAE-B08** Tener en cuenta el alcance y el enfoque de la automatización de la prueba y el mantenimiento del producto de prueba

---

- TAE-B09** Comprender cómo se integran las pruebas automatizadas dentro de las canalizaciones de integración continua/entrega continua (IC/EC)

---

- TAE-B10** Comprender cómo recopilar, analizar e informar sobre los datos de automatización de la prueba para informar a los implicados

---

- TAE-B11** Verificar la infraestructura de automatización de la prueba

---

- TAE-B12** Definir las oportunidades de mejora continua para la automatización de la prueba

## 0.5 Objetivos de aprendizaje evaluables y nivel cognitivo de conocimiento

Los objetivos de aprendizaje respaldan los resultados de negocio y se utilizan para generar los exámenes de ingeniería de automatización de la prueba para probadores certificados.

En general, todos los contenidos de este programa de estudio son evaluables en los niveles K2, K3 y K4, excepto la Introducción y los Apéndices. Es decir, se puede pedir al candidato que reconozca, recuerde o rememore una palabra clave o un concepto mencionado en cualquiera de los ocho capítulos. Los niveles específicos de los objetivos de aprendizaje se muestran al principio de cada capítulo y se clasifican de la siguiente manera:

K2: Comprender

K3: Aplicar

K4: Analizar

En el Apéndice A se ofrecen más detalles y ejemplos de los objetivos de aprendizaje.

Deberán recordarse todos los términos que figuran como palabras clave inmediatamente debajo de los títulos de los capítulos, aunque no se mencionen explícitamente en los objetivos de aprendizaje.

## 0.6 El examen del certificado de nivel avanzado de gestión de la prueba

El examen del certificado de ingeniería de automatización de la prueba se basará en este programa de estudio. Las respuestas a las preguntas del examen pueden requerir el uso de material basado en más de una sección de este programa de estudio. Todas las secciones del programa de estudio son evaluables, excepto la Introducción y los Apéndices. Se incluyen normas y libros como referencias, pero su contenido no es evaluable, más allá de lo que se resume en el propio programa de estudio a partir de dichas normas y libros.

El examen del certificado de ingeniería de automatización de la prueba se basará en este programa de estudio. Las respuestas a las preguntas del examen pueden requerir el uso de material basado en más de una sección de este programa de estudio. Todas las secciones del programa de estudio son evaluables, excepto la Introducción y los Apéndices. Se incluyen normas y libros como referencias, pero su contenido no es evaluable, más allá de lo que se resume en el propio programa de estudio a partir de dichas normas y libros.

Para más información sobre el examen de certificación en ingeniería de automatización de pruebas, consultar el documento “Exam Structures and Rules v1.1 Compatible with Syllabus Foundation and Advanced Levels and Specialists Modules”.

El criterio de admisión para presentarse al examen de Ingeniero de automatización de la prueba es que los candidatos tengan interés en las pruebas de software y la automatización de la prueba. Sin embargo, se recomienda de forma especial que los candidatos también:

- Tengan al menos una formación mínima en desarrollo de software y pruebas de software, como por ejemplo seis meses de experiencia como ingeniero de pruebas de software o como desarrollador de software.
- Realicen un curso que haya sido acreditado según los estándares del ISTQB (por una de las juntas de miembros reconocidas por el ISTQB).

Nota sobre los requisitos de acceso: Deberá obtenerse el certificado de nivel básico ISTQB® antes de presentarse al examen de certificación de ingeniero de automatización de la prueba de ISTQB®.

## 0.7 Acreditación

Un Comité Miembro de ISTQB® puede acreditar a los proveedores de formación cuyo material de curso siga este programa de estudio. Los proveedores de formación deberán obtener las directrices de acreditación del Comité Miembro u organismo que realice la acreditación. Un curso acreditado es reconocido como conforme a este programa de estudio y se le permite tener un examen ISTQB® como parte del curso.

Las directrices de acreditación para este programa de estudio son las Directrices de Acreditación genéricas publicadas por el Grupo de Trabajo de Gestión de Procesos y Conformidad del ISTQB®.

## 0.8 Tratamiento de los estándares

Existen estándares a los que se hace referencia en el programa de estudio de Ingeniería de automatización de la prueba (por ejemplo, (IEEE e ISO). El propósito de estas referencias es proporcionar un marco de trabajo (como en las referencias a la norma ISO 25010 relativa a las características de calidad) o proporcionar una fuente de información adicional si así lo desea el lector. Tenga en cuenta que el programa de estudio utiliza los documentos estándar como referencia. Los documentos estándar no están destinados a ser evaluados. Para más información sobre los estándares, consultar el capítulo 9 Referencias.

## 0.9 Mantenerse actualizado

La industria del software cambia rápidamente. Para hacer frente a estos cambios y proporcionar acceso a los implicados a información relevante y actualizada, los grupos de trabajo del ISTQB han creado enlaces en la página web [www.istqb.org](http://www.istqb.org), que hacen referencia a documentos de apoyo y cambios en los estándares. Esta información no es evaluable en el marco del programa de estudio de ingeniería de automatización de la prueba.

## 0.10 Nivel de detalle

El nivel de detalle de este programa de estudio permite que los cursos y los exámenes sean consistentes a escala internacional. Para lograr este objetivo, el programa de estudio consta de:

- Objetivos generales de instrucción que describen la intención del programa de estudio de gestión de la prueba de nivel avanzado.
- Una lista de palabras clave que los alumnos deben ser capaces de recordar.
- Objetivos de aprendizaje para cada área de conocimiento, que describen el resultado cognitivo de aprendizaje que debe alcanzarse.
- Una descripción de los conceptos clave, incluyendo referencias a fuentes como la bibliografía aceptada o los estándares.

El contenido del programa de estudio no es una descripción de toda el área de conocimiento de las pruebas de software; refleja el nivel de detalle que se cubrirá en los cursos de formación de ingeniería de

automatización de la prueba. Se concentra en conceptos y técnicas de prueba que pueden aplicarse a todos los proyectos de software, incluidos los que siguen métodos ágiles. Este programa de estudio no contiene objetivos de aprendizaje específicos relacionados con la prueba Ágil, pero sí analiza cómo se aplican estos conceptos en los proyectos Ágiles y en otros tipos de proyectos.

## 0.11 Cómo está organizado este programa de estudio

Hay ocho capítulos con contenidos evaluables. El encabezamiento superior de cada capítulo especifica la duración de este; no se proporciona cronología por debajo del nivel de capítulo. Para los cursos de formación acreditados, el programa de estudio exige un mínimo de 21 horas de instrucción, distribuidas en los ocho capítulos de la siguiente manera:

- Capítulo 01: 45 minutos - Introducción y objetivos de la automatización de la prueba
  - El probador aprende las ventajas de la automatización de la prueba y sus limitaciones
  - Se cubre la automatización de la prueba dentro de diferentes modelos de ciclo de vida de desarrollo del software
  - El probador aprende cómo influye la arquitectura de un sistema sujeto a prueba (SSP) en la adecuación de las herramientas de prueba.
- Capítulo 2: 180 minutos - Preparación para la automatización de la prueba
  - Se trata el diseño de la capacidad de ser probado del sistema sujeto a prueba (SSP) a través de la observabilidad, la controlabilidad y una arquitectura definida de forma clara.
  - Un probador aprende sobre la automatización de la prueba en distintos entornos.
  - Se tratarán los factores necesarios para evaluar una solución de automatización de la prueba adecuada.
  - Un probador conocerá las consideraciones técnicas necesarias para elaborar recomendaciones sobre la automatización de la prueba.
- Capítulo 3: 210 minutos - Arquitectura de automatización de la prueba
  - Se aborda la arquitectura de automatización de la prueba, así como sus componentes que conducen a una solución de automatización de la prueba.
  - Un probador conocerá las capas y su aplicación en un marco de automatización de la prueba.
  - Se cubrirán múltiples enfoques para el uso de herramientas de automatización de la prueba.
  - El probador aprenderá cómo aplicar los principios y patrones de diseño a la automatización de pruebas.
- Capítulo 4: 150 minutos - Implementación de la automatización de la prueba
  - Se abordará cómo planificar y desplegar de forma efectiva un proyecto piloto de automatización de la prueba.
  - Un probador conocerá los riesgos de despliegue y las estrategias para mitigarlos.
  - Se tratarán los factores que mejoran la mantenibilidad del código de automatización de pruebas.
- Capítulo 5: 90 minutos - Estrategias de implementación y despliegue para la automatización de la prueba

- Un probador aprenderá acerca de las canalizaciones de integración continua/entrega continua (CI/CD) y la ejecución automatizada de la prueba en todos los niveles de prueba.
- Se tratará la gestión de la configuración de los componentes de la automatización de la prueba.
- Un probador aprenderá sobre las dependencias aplicadas a la prueba de interfaz de programación de aplicación (IPA) y de contrato.
- Capítulo 6: 150 minutos - Suministro de información de la automatización de la prueba y métricas
  - Un probador aprenderá dónde se pueden recopilar los datos de un sistema sujeto a prueba (SSP) y la automatización de la prueba para el análisis y la elaboración de informes.
  - Se tratará el análisis de datos de los informes de prueba y la automatización de la prueba para descubrir las causas de fallos.
  - Se tratará el uso de informes de prueba y paneles de control para informar a los implicados.
- Capítulo 7: 135 minutos - Verificación de la solución de automatización de la prueba
  - El probador aprenderá a examinar y verificar la correcta operación de los componentes y el entorno de automatización de la prueba.
  - Se tratará cómo asegurar que los guiones de prueba y los juegos de prueba se ejecutan correctamente.
  - El probador comprenderá cuándo hay que realizar un análisis de la causa raíz.
  - Se tratarán las técnicas para analizar la calidad del código de automatización de la prueba.
- Capítulo 8: 210 minutos - Mejora continua
  - Se tratarán áreas adicionales de análisis de datos para la mejora de casos de prueba.
  - Un probador aprenderá formas de realizar mejoras y actualizaciones en una solución de automatización de la prueba y sus componentes.
  - Se tratará la identificación de formas de consolidar y racionalizar la automatización de la prueba.
  - Un probador aprenderá cómo las herramientas de automatización de la prueba pueden ayudar con las necesidades de soporte y configuración de la prueba.



# 1 Introducción y objetivos de la automatización de la prueba - (K-2)

**Duración:** 45 minutos

## Palabras Clave<sup>1</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB:** identificarán palabras clave del proceso de prueba
- **ESPDOM:** identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	sistema sujeto a prueba (SSP)	system under test (SUT)
ISTQB	automatización de la prueba	test automation

<sup>1</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

## Objetivos de Aprendizaje para “Capítulo 1”

### 1.1 Propósito de la automatización de la prueba

**TAE - 1.1.1 (K2)** Explicar las ventajas y desventajas de la automatización de la prueba.

### 1.2 Automatización de la prueba en el ciclo de vida de desarrollo de software

**TAE - 1.2.1 (K2)** Explicar cómo se aplica la automatización de la prueba en diferentes modelos de ciclo de vida de desarrollo de software.

**TAE - 1.2.2 (K2)** Seleccionar las herramientas de automatización de la prueba adecuadas para un sistema sujeto a prueba determinado.

## 1.1 Propósito de la automatización de la prueba

### 1.1.1 Explicar las ventajas y desventajas de la automatización de la prueba

La automatización de la prueba, que incluye la ejecución automatizada de pruebas y la generación de informes de prueba, es una o más de las siguientes actividades:

- Uso de herramientas de software diseñadas específicamente para controlar y configurar conjuntos de prueba para la ejecución de pruebas.
- Ejecución de pruebas de forma automatizada.
- Comparación de los resultados reales con los esperados.

La automatización de la prueba proporciona importantes prestaciones y capacidades que pueden interactuar con un sistema sujeto a prueba (SSP). La automatización de la prueba puede abarcar un amplio campo del software. Las soluciones abarcan muchos tipos de software (por ejemplo, un sistema sujeto a prueba (SSP) con una interfaz de usuario (IU), un sistema sujeto a prueba (SSP) sin interfaz de usuario (IU), aplicaciones móviles, protocolos de red y conectividad).

La automatización de la prueba tiene diversas ventajas. Entre otras:

- Permite realizar más pruebas por construcción en comparación con las pruebas manuales.
- Ofrece la posibilidad de crear y ejecutar pruebas que no pueden ejecutarse manualmente (por ejemplo, respuesta en tiempo real, prueba remota y prueba en paralelo)
- Permite realizar pruebas más complejas que las pruebas manuales
- Se ejecuta más rápido que las pruebas manuales
- Está menos sujeta a errores humanos
- Es más eficaz y eficiente en el uso de los recursos de prueba
- Proporciona una retroalimentación más rápida con respecto a la calidad del sistema sujeto a prueba (SSP)
- Ayuda a mejorar la fiabilidad del sistema (por ejemplo, la disponibilidad y la capacidad de recuperación)
- Mejora la consistencia de la ejecución de pruebas a lo largo de los ciclos de prueba

Sin embargo, la automatización de la prueba tiene desventajas potenciales, entre las que se incluyen:

- Requisito de inversión inicial para establecer una solución de automatización de la prueba.
- Tiempo para desarrollar y mantener una solución de automatización de la prueba.
- Requisito de objetivos claros de automatización de la prueba para asegurar el éxito.
- Rigidez de las pruebas y menor adaptabilidad a los cambios en el sistema sujeto a prueba (SSP).
- Introducción de defectos adicionales mediante la automatización de la prueba.

Es necesario tener en cuenta la existencia de limitaciones en la automatización de la prueba:

- No todas las pruebas manuales pueden automatizarse
- Sólo se puede verificar lo que las pruebas automatizadas están programadas para hacer
- La automatización de la prueba sólo puede comprobar resultados de prueba interpretables por la máquina, lo que significa que algunas características de calidad pueden no ser objeto de prueba con la automatización

- La automatización de la prueba sólo puede comprobar resultados de prueba que puedan ser verificados por un oráculo de prueba automatizado

## 1.2 La automatización de la prueba en el ciclo de vida de desarrollo de software

### 1.2.1 Explicar cómo se aplica la automatización de la prueba en los diferentes modelos de ciclo de vida del desarrollo de software

- **Cascada**
  - El modelo en cascada es un modelo de ciclo de vida de desarrollo de software lineal y secuencial. Este modelo tiene distintas fases (es decir, requisitos, diseño, implementación, verificación y mantenimiento) y cada fase suele concluir con documentación que debe ser aprobada. La implementación de la automatización de la prueba suele tener lugar en paralelo o después de la fase de implementación. La realización de pruebas suele tener lugar durante la fase de verificación debido a que los componentes de software no están listos para ser probados hasta entonces.
- **Modelo V**
  - El modelo V es un modelo de ciclo de vida de desarrollo de software (CVDS) en el que un proceso se ejecuta de forma secuencial. A medida que se define un proyecto desde los requisitos de alto nivel hasta los de bajo nivel, se definen las correspondientes actividades de prueba e integración para validar dichos requisitos. De aquí se obtienen los niveles de prueba tradicionales: componente, integración de componentes, sistema, integración de sistemas y aceptación, como se describe en la sección 2.2 del CTFL. Proporcionar un marco de trabajo de automatización de prueba (MTAP) para cada nivel de prueba es posible y recomendable.
- **Desarrollo Ágil de Software**
  - En el desarrollo ágil de software existen innumerables posibilidades para la automatización de pruebas. A diferencia de la cascada o el modelo V, en el método de Desarrollo Ágil de Software, los ingenieros de automatización de pruebas (IAP) y los representantes de negocio pueden decidir la hoja de ruta, el calendario y la entrega prevista de las pruebas. En este método, existen buenas prácticas como las revisiones de código, la programación en pareja y la frecuente ejecución de pruebas automatizadas. La eliminación de silos (es decir, asegurarse de que desarrolladores, probadores y otros implicados trabajan juntos) permite a los equipos cubrir todos los niveles de prueba con la cantidad y profundidad adecuadas de automatización de la prueba, logrando un objetivo denominado automatización durante el esprint. Para más detalles, consultar la sección 3.2 del programa de estudio CT-TAS de ISTQB.

### 1.2.2 Seleccionar las herramientas de automatización de prueba adecuadas para un sistema sujeto a prueba determinado

Para identificar las herramientas de prueba más adecuadas para un proyecto determinado, primero se debe analizar el sistema sujeto a prueba (SSP). Los ingenieros de automatización de pruebas (IAP) necesitan identificar los requisitos del proyecto que pueden utilizarse como línea base para la selección de herramientas.

Dado que se utilizan diferentes prestaciones de la herramienta de automatización de la prueba para el software de interfaz de usuario y, por ejemplo, para los servicios web, es importante comprender lo que el proyecto quiere conseguir a lo largo del tiempo. No hay límite en el número de herramientas y prestaciones de automatización de la prueba que pueden utilizarse o seleccionarse, pero siempre hay que tener en cuenta los costes. Utilizar una herramienta comercial de mercado o implementar una solución a medida basada en tecnología de código abierto puede ser un proceso complejo.

El siguiente tema a evaluar es la composición y la experiencia del equipo en la automatización de la prueba. En el caso de que los probadores tengan poca o ninguna experiencia en programación, utilizar una solución baja en código o sin código puede ser una opción viable.

Para los probadores técnicos con conocimientos de programación, puede ser útil seleccionar herramientas cuyo lenguaje coincida con el del sistema sujeto a prueba (SSP). Esto ofrece ventajas como la posibilidad de trabajar con los desarrolladores en la depuración de los defectos de automatización de la prueba de forma más eficiente y la formación cruzada de los miembros entre equipos.

## 2 Preparación para la automatización de la prueba - (K4)

Duración: 180 minutos

### Palabras Clave<sup>2</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	prueba de IPA	API testing
ISTQB	prueba de IGU	GUI testing
ISTQB	capacidad de ser probado	testability

<sup>2</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

## Objetivos de Aprendizaje para “Capítulo 2”

### 2.1 Comprender la configuración de una infraestructura para hacer posible la automatización de la prueba

- TAE - 2.1.1 (K2)** Describir las necesidades de configuración de una infraestructura que posibilite la implementación de la automatización de la prueba.
- TAE - 2.1.2 (K2)** Explicar cómo se utiliza la automatización de la prueba en diferentes entornos

### 2.2 Proceso de evaluación para seleccionar las herramientas y estrategias adecuadas

- TAE - 2.2.1 (K4)** Analizar un sistema sujeto a prueba para determinar la solución de automatización de la prueba adecuada.
- TAE - 2.2.2 (K4)** Aportar ejemplos de hallazgos técnicos de la evaluación de una herramienta.

## 2.1 Comprender la configuración de una infraestructura para hacer posible la automatización de la prueba

### 2.1.1 Describir las necesidades de configuración de una infraestructura que posibilite la implementación de la automatización de la prueba

La capacidad de ser probado del sistema sujeto a prueba (es decir, la disponibilidad de interfaces de software que permitan la realización de pruebas, por ejemplo, para habilitar el control y la observabilidad del sistema sujeto a prueba) debe diseñarse e implementarse en paralelo con el diseño y la implementación de las demás prestaciones del sistema sujeto a prueba (SSP). En general, este trabajo lo realiza un arquitecto de software, ya que la capacidad de ser probado es un requisito no funcional del sistema, a menudo con la participación de un ingeniero de automatización de pruebas (IAP) para identificar las áreas específicas en las que se pueden realizar mejoras.

Para mejorar la capacidad de ser probado del sistema sujeto a prueba (SSP) pueden utilizarse diferentes soluciones que necesitan distintas configuraciones, por ejemplo:

- **Identificadores de accesibilidad**
  - Los diferentes marcos de trabajo de desarrollo pueden generar estos identificadores automáticamente o los desarrolladores pueden establecerlos manualmente.
- **Variables de entorno del sistema**
  - Algunos parámetros de la aplicación pueden modificarse para posibilitar una prueba más sencilla a través de la administración.
- **Variables de despliegue**
  - Similares a las variables del sistema, pero pueden establecerse antes de iniciar el despliegue.

Diseñar la capacidad de ser probado de un sistema sujeto a prueba (SSP) consta de los siguientes aspectos:

- **Observabilidad:** El sistema sujeto a prueba (SSP) necesita proporcionar interfaces que permitan conocerlo. Los casos de prueba pueden utilizar estas interfaces para determinar si los resultados reales son iguales a los resultados esperados.
- **Controlabilidad:** El sistema sujeto a prueba (SSP) necesita proporcionar interfaces que puedan utilizarse para realizar acciones en el sistema sujeto a prueba (SSP). Puede tratarse de elementos de la interfaz de usuario, llamadas a funciones, elementos de comunicación (por ejemplo, protocolos del protocolo de control de transmisiones/protocolo de Internet (TCP/IP) y del bus serie universal (USB)) o señales electrónicas para conmutadores físicos o lógicos de las distintas variables del entorno.
- **Transparencia de la arquitectura:** La documentación de una arquitectura necesita proporcionar componentes e interfaces claros y comprensibles que ofrezcan observabilidad y controlabilidad en todos los niveles de prueba y fomenten la calidad.

### 2.1.2 Explicar cómo se utiliza la automatización de la prueba en diferentes entornos

Diferentes tipos de pruebas automatizadas pueden ejecutarse en diferentes entornos. Estos entornos pueden diferir entre proyectos y metodologías, y la mayoría de los proyectos tienen uno o más entornos



que pueden utilizar para probar. Desde un punto de vista técnico, estos entornos pueden crearse a partir de contenedores, software de virtualización y utilizando otros enfoques.

Un conjunto de posibles entornos a tener en cuenta incluye los siguientes:

- **Entorno de desarrollo local**

- En un entorno de desarrollo local es donde se crea inicialmente el software y se prueban los componentes con automatización para verificar la adecuación funcional. En el entorno de desarrollo local pueden llevarse a cabo varios tipos de pruebas, como la prueba de componentes, la prueba de interfaz gráfica de usuario y la prueba de interfaz de programación de aplicación (IPA). También es importante señalar que si se utiliza un entorno de desarrollo integrado (EDI) en el ordenador dado, se puede realizar una prueba de caja blanca para identificar problemas de codificación y de calidad tan pronto como sea posible.

- **Entorno de construcción**

- Su propósito principal es construir el software y ejecutar pruebas que comprueben la corrección de la construcción resultante en un ecosistema DevOps. Este entorno puede ser un entorno de desarrollo local o un agente de integración continua/entrega continua (CI/CD) en el que se pueden realizar pruebas de bajo nivel (es decir, pruebas de componentes y pruebas de integración de componentes) y análisis estáticos sin necesidad de un despliegue real en otros entornos.

- **Entorno de integración**

- Tras realizar la prueba de bajo nivel y el análisis estático, la siguiente etapa es un entorno de integración de sistemas. Aquí se dispone de una versión candidata del sistema sujeto a prueba que está totalmente integrada con otros sistemas que pueden probarse. En este entorno se puede ejecutar un juego de pruebas totalmente automatizado, ya sean pruebas de interfaz de usuario o pruebas de Interfaz de Programación de Aplicación (IPA). En este entorno no hay pruebas de caja blanca, sólo pruebas de caja negra (es decir, pruebas de integración de sistemas y/o de aceptación). Es importante señalar que éste es el primer entorno en el que se debe monitorizar para ver qué ocurre en segundo plano durante el uso del sistema sujeto a prueba (SSP) y posibilitar así una investigación eficiente de defectos/fallos.

- **Entorno de preproducción**

- Un entorno de preproducción se utiliza sobre todo para evaluar las características de calidad no funcionales (por ejemplo, la eficiencia de desempeño). Aunque las pruebas no funcionales pueden realizarse en cualquier entorno, se concentra más en la preproducción porque guarda el mayor parecido posible con producción. A menudo, los implicados de negocio pueden realizar pruebas de aceptación de usuario para verificar el producto final y, si es necesario, también es posible ejecutar aquí el juego de pruebas automatizado existente. Este entorno también se monitoriza.

- **Entorno de producción/operación**

- Se puede utilizar un entorno de producción para evaluar las características de calidad funcionales y no funcionales en tiempo real mientras los usuarios interactúan con un sistema desplegado con supervisión y ciertas buenas prácticas que permiten realizar pruebas en producción (por ejemplo, lanzamiento canario, despliegue azul/verde y prueba A/B).

## 2.2 Proceso de evaluación para seleccionar las herramientas y estrategias adecuadas

### 2.2.1 Analizar un sistema sujeto a prueba para determinar la solución de automatización de la prueba adecuada

Cada sistema sujeto a prueba (SSP) puede ser diferente entre sí, pero hay varios factores y características que pueden analizarse para poder contar con una solución de automatización de la prueba (TAE) adecuada. Durante la investigación de un sistema sujeto a prueba (SSP), los Ingenieros de automatización de pruebas (IAP) necesitan reunir los requisitos teniendo en cuenta su alcance y sus capacidades dadas. Los distintos tipos de aplicaciones (por ejemplo, servicio web, móvil y web) necesitan diferentes tipos de automatización de la prueba desde un punto de vista técnico. La investigación se puede realizar -y se recomienda- en colaboración con otros implicados (por ejemplo, probadores manuales, implicados de negocio y analistas de negocio) para identificar tantos riesgos y sus mitigaciones como sea posible a fin de contar con una solución de automatización de la prueba beneficiosa para el futuro.

Los requisitos de un enfoque y una arquitectura de automatización de la prueba deberían tener en cuenta lo siguiente:

Qué actividades del proceso de prueba deben automatizarse (por ejemplo, gestión de la prueba, diseño de la prueba, generación de la prueba y ejecución de la prueba).

- Qué niveles de prueba deberían ser soportados.
- Qué tipos de prueba deberían ser soportados.
- Qué roles y conjuntos de habilidades de prueba deberían ser soportados.
- Qué productos software, líneas de productos y familias deberían ser soportados, (por ejemplo, para definir el alcance y la vida útil de la solución de automatización de la prueba (SAP) implementada).
- Qué tipo de sistemas sujetos a prueba (SSP) necesitan ser compatibles con la solución de automatización de la prueba (SAP).
- Disponibilidad de los datos de prueba y su calidad.
- Posibles métodos y formas de emular casos inalcanzables (por ejemplo, aplicaciones de terceros implicadas).

### 2.2.2 Aportar ejemplos de hallazgos técnicos de la evaluación de una herramienta

Una vez analizado el sistema sujeto a prueba (SSP) y recopilados los requisitos de todos los implicados, es probable que existan herramientas de automatización de la prueba que cumplan estos requisitos y que puedan tenerse en cuenta. Puede que no exista una única herramienta que se ajuste a todos los requisitos identificados, y los implicados deberían admitir esta posibilidad.

Resulta útil recopilar los hallazgos sobre las posibles herramientas y reflexionar sobre los distintos requisitos directos e indirectos en una tabla comparativa. El objetivo de la tabla comparativa es permitir a los implicados ver las diferencias entre las herramientas en función de los requisitos específicos. La tabla de comparación enumera las herramientas en las columnas y los requisitos en las filas. Las celdas contienen información sobre las propiedades de cada herramienta en relación con cada requisito y sobre las prioridades.

BORRADOR

En general, las herramientas de automatización de la prueba deberían evaluarse para determinar si se ajustan al requisito identificado en la sección anterior (2.2.1). Entre los requisitos que hay que tener en cuenta a la hora de evaluar y comparar las herramientas se incluyen:

- El idioma/tecnología de la herramienta y las herramientas del entorno de desarrollo integrado (EDI).
- La capacidad de configurar una herramienta, si admite distintos entornos de prueba, ejecuta configuraciones y utiliza valores de configuración dinámicos o estáticos.
- La capacidad de gestionar los datos de prueba dentro de la herramienta. La gestión de datos de prueba podría integrarse con un repositorio central para el control de versiones.
- Para distintos tipos de prueba, podría ser necesario seleccionar diferentes herramientas de automatización de la prueba.
- La capacidad de integrarse con otras herramientas utilizadas en un proyecto o en la organización, como integración continua/entrega continua (IC/EC), seguimiento de tareas, gestión de pruebas, suministro de información u otras herramientas.
- La capacidad de ampliar la arquitectura general de prueba y evaluar la escalabilidad, mantenibilidad, modificabilidad, compatibilidad y fiabilidad de las herramientas.

Esta tabla comparativa es una buena fuente para determinar la herramienta o conjunto de herramientas que se propone utilizar para la automatización de la prueba del sistema sujeto a prueba (SSP).

El proceso puede variar en cuanto a cómo se toma la decisión sobre qué herramienta(s) se utilizará(n), pero la propuesta debe demostrarse a los implicados adecuados para su aprobación.

### 3 Arquitectura de automatización de la prueba - (K3)

Duración: 210 minutos

#### Palabras Clave<sup>3</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB:** identificarán palabras clave del proceso de prueba
- **ESPDOM:** identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	desarrollo basado en el comportamiento	behavior-driven development
ISTQB	captura/reproducción	capture/playback
ISTQB	prueba guiada por datos	data-driven testing
ISTQB	arquitectura de automatización de pruebas genérica	generic test automation architecture
ISTQB	prueba guiada por palabra clave	keyword-driven testing
ISTQB	guionización lineal	linear scripting
ISTQB	prueba basada en modelos	model-based testing
ISTQB	guionizado estructurado	structured scripting
ISTQB	capa de adaptación de la prueba	test adaptation layer
ISTQB	marco de trabajo de automatización de la prueba	test automation framework
ISTQB	solución de automatización de la prueba	test automation solution
ISTQB	arnés de prueba	test harness
ISTQB	guion de prueba	test script
ISTQB	producto de prueba	testware
ISTQB	paso de prueba	test step
ISTQB	desarrollo guiado por prueba	test-driven development

<sup>3</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

## Objetivos de Aprendizaje para “Capítulo 3”

### 3.1 Conceptos de diseño utilizados en la automatización de la prueba

- TAE - 3.1.1 (K2)** Explicar las principales capacidades en una arquitectura de automatización de la prueba.
- TAE - 3.1.2 (K2)** Explicar cómo diseñar una solución de automatización de la prueba.
- TAE - 3.1.3 (K3)** Aplicar la estructura por capas de los marcos de automatización de pruebas.
- TAE - 3.1.4 (K3)** Aplicar diferentes enfoques para automatizar casos de prueba.
- TAE - 3.1.5 (K3)** Aplicar principios de diseño y patrones de diseño en la automatización de pruebas.

## 3.1 Conceptos de diseño utilizados en la automatización de la prueba

### 3.1.1 Explicar las principales capacidades en una arquitectura de automatización de la prueba

La arquitectura de automatización de pruebas genérica (AAPg) es un concepto de diseño de alto nivel que proporciona una visión abstracta de la comunicación entre la automatización de la prueba y los sistemas a los que está conectada la automatización de la prueba, es decir, el sistema sujeto a prueba (SSP), la gestión de proyectos, la gestión de la prueba y la gestión de la configuración (véase la figura 1). También proporciona las capacidades que es necesario cubrir a la hora de diseñar una arquitectura de automatización de la prueba (AAP).

Las interfaces de la arquitectura de automatización de pruebas genérica (AAPg) describen lo siguiente:

- La interfaz del sistema sujeto a prueba (SSP) describe la conexión entre el sistema sujeto a prueba (SSP) y el marco de trabajo de automatización de pruebas (MTAP) (consultar el apartado 3.1.3 sobre el marco de automatización de pruebas).
- La interfaz de gestión del proyecto describe el avance del desarrollo de la automatización de la prueba.
- La interfaz de gestión de la prueba describe la correspondencia entre las definiciones de los casos de prueba y los casos de prueba automatizados.
- La interfaz de gestión de la configuración describe las canalizaciones de integración continua/entrega continua (IC/EC), los entornos y el producto de prueba.

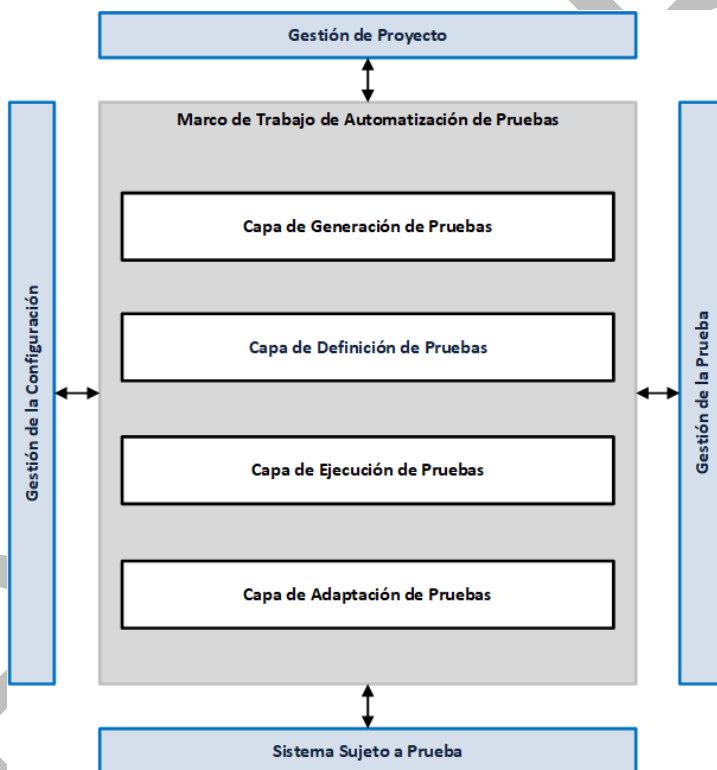


Ilustración 1 - Diagrama de AAPg

## Capacidades proporcionadas por las herramientas de automatización de la prueba y librerías

Las capacidades básicas de automatización de la prueba deben identificarse y seleccionarse de entre las herramientas disponibles según los requisitos de un proyecto determinado.

- **Generación de pruebas:**
  - Soporta el diseño automatizado de casos de prueba basados en un modelo de prueba. Las herramientas de pruebas basadas en modelos pueden aprovecharse en el proceso de generación (véase el programa de estudios ISTQB CT-PBM). La generación de pruebas es una capacidad opcional.
- **Definición de pruebas**
  - Soporta la definición e implementación de casos de prueba y/o conjuntos de pruebas, que opcionalmente pueden derivarse de un modelo de prueba. Separa la definición de la prueba del sistema sujeto a prueba (SSP) y/o de las herramientas de prueba. Contiene los medios para definir las pruebas de alto y bajo nivel, que se gestionan en los datos de prueba, los casos de prueba y los componentes de la biblioteca de pruebas o sus combinaciones.
- **Ejecución de pruebas**
  - Soporta la ejecución de pruebas y el registro en bitácora de prueba. Proporciona una herramienta de ejecución de la prueba para ejecutar automáticamente las pruebas seleccionadas, y un componente de registro de pruebas y de suministro de información de prueba.
- **Adaptación de pruebas**
  - Proporciona la funcionalidad necesaria para adaptar las pruebas automatizadas a los distintos componentes o interfaces del sistema sujeto a prueba (SSP). Proporciona diferentes adaptadores para conectarse al sistema sujeto a prueba a través de la Interfaz de programación de aplicación (IPA), los protocolos y los servicios.

### 3.1.2 Explicar cómo diseñar una solución de automatización de la prueba

Una solución de automatización de la prueba (SAP) se define mediante la comprensión de los requisitos funcionales, no funcionales y técnicos del sistema sujeto a prueba (SSP), las herramientas existentes o requeridas que son necesarias para implementar una solución. Una SAP se implementa con herramientas comerciales o de código abierto y puede necesitar adaptadores adicionales específicos del sistema sujeto a prueba (SSP).

La arquitectura de automatización de la prueba (AAP) define el diseño técnico de la SAP en su conjunto. Debería abordar:

- La selección de herramientas de automatización de la prueba y librerías específicas de la herramienta.
- El desarrollo de complementos y/o componentes.
- La identificación de los requisitos de conectividad e interfaz (por ejemplo, cortafuegos, base de datos, localizadores uniformes de recursos (URL)/conexiones, mocks/stubs, colas de mensajes y protocolos).
- Conexión a las herramientas de gestión de la prueba y de gestión de defectos.
- Uso de un sistema de control de versiones y repositorios.



### 3.1.3 Aplicar la estructura por capas de los marcos de automatización de la prueba

- **Marco de trabajo de automatización de la prueba**

- El marco de trabajo de automatización de la prueba (MTAP) es la base de una solución de automatización de la prueba (SAP). Suele incluir un arnés de prueba, también conocido como ejecutor de pruebas, y librerías de prueba, guiones de prueba y juegos de prueba.

- **Capas del marco de trabajo de automatización de la prueba (MTAP)**

- Las capas del marco de trabajo de automatización de la prueba (MTAP) definen una frontera distinta de clases que tienen propósitos similares, como los casos de prueba, los informes de prueba, el registro en bitácora de prueba, el cifrado y los arneses de prueba. Al introducir una capa para cada propósito individual, el diseño puede complicarse. Por lo tanto, se recomienda mantener bajo el número de capas del marco de trabajo de automatización de la prueba (MTAP).

- **Guiones de prueba**

- Su propósito es proporcionar un repositorio de casos de prueba del sistema sujeto a prueba (SSP) y de las anotaciones del juego de prueba. Invoca los servicios de la capa lógica de negocio, lo que puede implicar pasos de prueba, flujos de usuarios o llamadas a la Interfaz de Programación de Aplicación (IPA). Sin embargo, no deben realizarse llamadas directas a las librerías base desde los guiones de prueba.

- **Lógica de negocio**

- En esta capa se almacenan todas las librerías dependientes del sistema sujeto a prueba (SSP). Estas librerías heredarán los archivos de clase de las librerías base o utilizarán las fachadas proporcionadas por ellas (véase la sección 3.1.5 relativa a la herencia y las fachadas). La capa de lógica de negocio se utiliza para configurar el marco de trabajo de automatización de la prueba (MTAP) para que se ejecute contra el sistema sujeto a prueba (SSP) y las configuraciones adicionales.

- **Librerías base**

- En esta capa se almacenan todas las bibliotecas que son independientes de cualquier sistema sujeto a prueba (SSP). Estas librerías base pueden reutilizarse en cualquier tipo de proyecto que comparta la misma pila tecnológica de desarrollo.

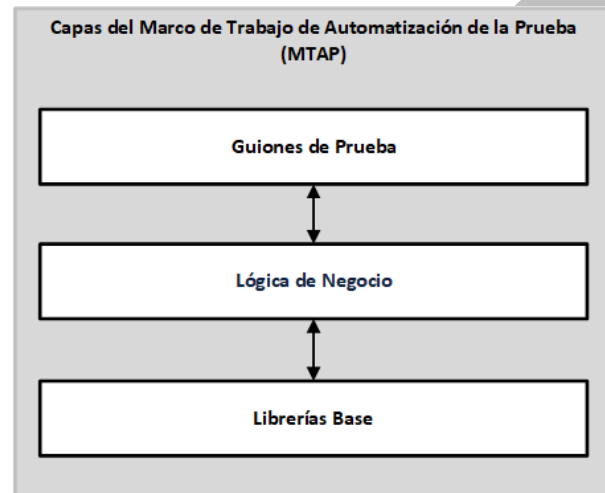


Ilustración 2 - Capas del Marco de Trabajo de Automatización de la Prueba (MTAP)

- **Escalado de la automatización de la prueba**

- El siguiente ejemplo (figura 3) muestra cómo las librerías base proporcionan una base reutilizable para múltiples marcos de trabajo de automatización de la prueba (MTAP). En el proyecto n.º 1 hay dos marcos de trabajo de automatización de la prueba (MTAP) contruidos sobre las librerías base, y otro proyecto aprovecha las librerías base ya existentes para construir su marco de trabajo de automatización de la prueba (MTAP) para probar la aplicación n.º 3. Un ingeniero de automatización de pruebas (IAP) construye los marcos de trabajo de automatización de la prueba (MTAP) para el proyecto n.º 1, mientras que un segundo ingeniero de automatización de pruebas (IAP) construye el marco de trabajo de automatización de la prueba (MTAP) para el proyecto n.º 2.

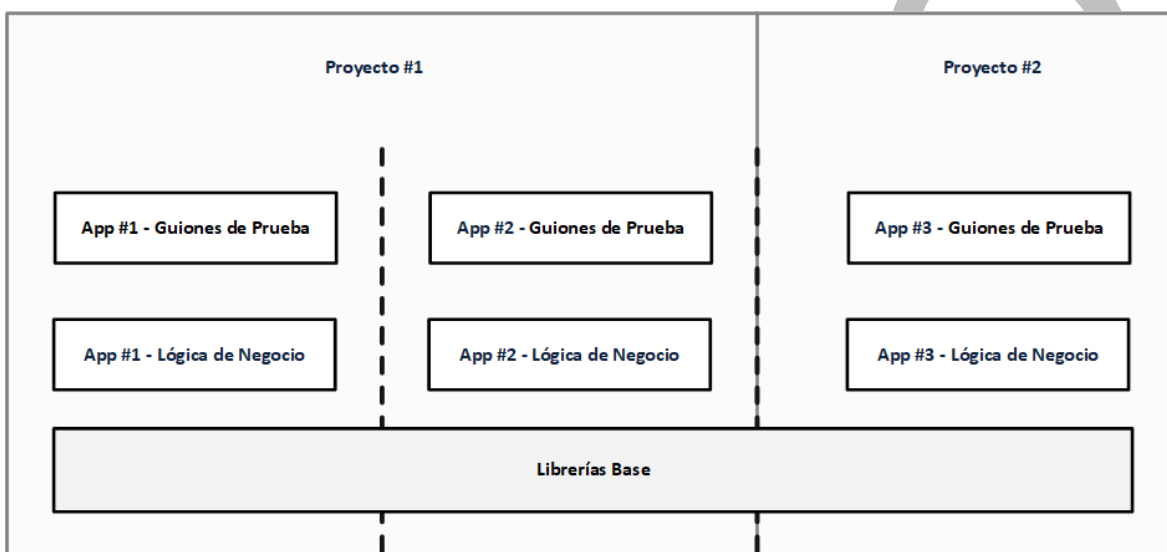


Ilustración 3 - Aplicación de librerías base a múltiples marcos de trabajo de automatización de la prueba (MTAP)

### 3.1.4 Aplicar diferentes enfoques para automatizar casos de prueba

Hay varios enfoques de desarrollo que los equipos pueden elegir para elaborar casos de prueba automatizados. Estos pueden incluir lenguajes de guion interactivos o lenguajes de programación compilados. Los distintos enfoques proporcionan diferentes beneficios de automatización y se pueden aprovechar en diferentes circunstancias. Aunque el desarrollo guiado por pruebas (DGP) y el desarrollo guiado por el comportamiento (DGC) son metodologías de desarrollo, si se siguen correctamente, dan como resultado el desarrollo automatizado de casos de prueba.

- **Captura/reproducción**

- La captura/reproducción es un enfoque en el que se capturan las interacciones con el sistema sujeto a prueba (SSP) mientras se llevan a cabo acciones de forma manual. Estas herramientas generan guiones de prueba durante la captura y, en función de la herramienta utilizada, el código de automatización de la prueba puede ser modificable. Las herramientas que no exponen código se denominan en ocasiones de automatización de prueba sin código, mientras que las herramientas que exponen código se denominan de automatización de prueba baja en código.

- **Ventajas**
  - Inicialmente fáciles de configurar y utilizar.
- **Desventajas**
  - Difícil de mantener, escalar y evolucionar.
  - El sistema sujeto a prueba (SSP) necesita estar disponible mientras se captura un caso de prueba.
  - Sólo es factible para un alcance pequeño y un sistema sujeto a prueba (SSP) que cambia con una frecuencia muy reducida.
  - La ejecución del sistema sujeto a prueba (SSP) capturado depende en gran medida de la versión del sistema sujeto a prueba (SSP) de la que se haya tomado la captura.
  - Grabar cada caso de prueba individual en lugar de reutilizar los bloques de construcción existentes requiere mucho tiempo.
- **Guionizado lineal**
  - El guionizado lineal es una actividad de programación que no requiere librerías de prueba personalizadas realizadas por un Ingeniero de automatización de pruebas (IAP) y se utiliza para redactar y ejecutar los guiones de prueba. Un Ingeniero de automatización de pruebas (IAP) puede aprovechar cualquier guion de prueba que haya sido grabado por una herramienta de captura/reproducción, que luego puede modificarse.
  - **Ventajas**
    - Fácil de configurar y de empezar a redactar guiones de prueba.
    - En comparación con la captura/reproducción, los guiones de prueba pueden modificarse más fácilmente.
  - **Desventajas**
    - Difícil de mantener, escalar y evolucionar.
    - El sistema sujeto a prueba (SSP) necesita estar disponible mientras se captura un caso de prueba.
    - Sólo es factible para un alcance pequeño y un sistema sujeto a prueba (SSP) que cambia con una frecuencia muy reducida.
    - En comparación con la captura/reproducción, son necesarios algunos conocimientos de programación.
- **Guionizado estructurado**
  - Se introducen librerías de prueba con elementos reutilizables, pasos de prueba y/o recorridos del usuario. Para la creación y el mantenimiento de guiones de prueba con este enfoque son necesarios conocimientos de programación.
  - **Ventajas**
    - Fácil de mantener, escalar, portar, adaptar y evolucionar.
    - Se puede separar la lógica de negocio de los guiones de prueba.
  - **Desventajas**
    - Son necesarios conocimientos de programación.
    - La inversión inicial en el desarrollo del marco de trabajo de automatización de la prueba (MTAP) y la definición del producto de prueba requieren mucho tiempo.

- **Desarrollo guiado por prueba**

- Los casos de prueba se definen como parte del proceso de desarrollo antes de que se implemente una nueva prestación del sistema sujeto a prueba (SSP). El enfoque DGP es probar, codificar y refactorizar o también conocido como rojo, verde y refactorizar. Un desarrollador identifica y crea un caso de prueba que fallará (rojo). A continuación, desarrolla la funcionalidad que satisfará el caso de prueba (verde). A continuación, se refactoriza el código para optimizarlo y atenerse a los principios del código limpio. El proceso continúa con la siguiente prueba y el siguiente incremento de funcionalidad.
- **Ventajas**
  - Simplifica el desarrollo de casos de prueba a nivel de componente.
  - Mejora la calidad del código y su estructura.
  - Mejora la capacidad de ser probado.
  - Facilita la consecución de la cobertura de código deseada.
  - Reduce la propagación de defectos a niveles de prueba superiores.
  - Mejora la comunicación entre desarrolladores, representantes de negocio y probadores.
  - Las historias de usuario que no se verifican mediante pruebas de interfaz gráfica de usuario (IGU) y pruebas de interfaz de programación de aplicación (IPA) pueden alcanzar rápidamente los criterios de salida siguiendo el DGP (“test-driven development – TDD”).
- **Desventajas**
  - Inicialmente lleva más tiempo acostumbrarse al DGP (“testing-driven development – TDD”).
  - No seguir correctamente el DGP (“test-driven development – TDD”) puede dar lugar a una falsa confianza en la calidad del código.

- **Prueba guiada por datos**

- La prueba guiada por datos (PGD) se basa en el enfoque de guionizado estructurado. Los guiones de prueba se suministran con datos de prueba (por ejemplo, archivos .csv, archivos .xlsx y volcados de bases de datos). Esto permite ejecutar los mismos guiones de prueba varias veces con diferentes datos de prueba.
- **Ventajas**
  - Permite ampliar los casos de prueba de forma rápida y sencilla mediante la alimentación de datos.
  - El coste de añadir nuevas pruebas automatizadas puede reducirse significativamente.
  - Los analistas de prueba pueden especificar pruebas automatizadas rellenando uno o varios archivos de datos de prueba que describan las pruebas. Esto da a los analistas de prueba más libertad para especificar pruebas automatizadas con menos dependencia de los analistas de pruebas técnicas.
- **Desventajas**
  - Puede ser necesaria una gestión de datos de prueba adecuada.

- **Prueba guiada por palabras clave**

- Los casos de prueba guiados por palabras clave (PGPC) son una lista o tabla de pasos de prueba derivados de palabras clave y de los datos de prueba sobre los que operan las palabras clave. Las palabras clave se definen desde la perspectiva del usuario. Esta técnica suele construirse a partir de la PGD ("data-driven testing - DDT").
- **Ventajas**
  - Los analistas de pruebas y los analistas de negocio pueden participar en la generación de casos de prueba automatizados siguiendo el enfoque de la PGPC ("keyword-driven testing - KDT").
  - La PGPC también puede utilizarse para pruebas manuales independientes de la automatización de la prueba (véase el estándar ISO/IEC/IEEE 29119-5 relativo a la prueba guiada por palabras clave).
- **Desventajas**
  - La implementación y el mantenimiento de las palabras clave es una tarea compleja que los ingenieros de automatización de pruebas (IAP) necesitan cubrir, lo que puede convertirse en un reto cuando el alcance crece.
  - Supone un gran esfuerzo para los sistemas más pequeños.

- **Desarrollo guiado por el comportamiento**

- El DGC ("Behavior-driven development - BDD") aprovecha un formato de lenguaje natural (es decir, dado, cuándo y entonces) para formular criterios de aceptación que pueden utilizarse como casos de prueba automatizados y almacenarse en archivos de prestaciones. A continuación, una herramienta de BDD puede entender el lenguaje y ejecutar los casos de prueba.
- **Ventajas**
  - Mejora la comunicación entre desarrolladores, representantes de negocio y probadores
  - Los escenarios de DGC automatizados actúan como casos de prueba y aseguran la cobertura de las especificaciones.
  - El DGC puede aprovecharse para producir múltiples tipos de prueba en distintos niveles de la pirámide de prueba ("behavior-driven development - BDD")
- **Desventajas**
  - Los casos de prueba adicionales, normalmente las condiciones de prueba negativas y los casos extremos, siguen necesitando ser definidos por el equipo, normalmente por un analista de prueba o un IAP.
  - Muchos equipos confunden la DGC con una forma de escribir casos de prueba en lenguaje natural y no implican a los representantes de negocio y a los desarrolladores en todo el enfoque.
  - La implementación y el mantenimiento de los pasos de prueba en lenguaje natural es una tarea compleja para los IAP.
  - Unos pasos de prueba demasiado complejos convertirán la depuración en una actividad difícil y costosa.

### 3.1.5 Aplicar principios de diseño y patrones de diseño en la automatización de pruebas

La automatización de la prueba es una actividad de desarrollo de software. Por lo tanto, los principios de diseño y los patrones de diseño son tan importantes para un IAT como para un desarrollador de software.

- **Principios de programación orientada a objetos**
  - Existen cuatro grandes principios de programación orientada a objetos: encapsulamiento, abstracción, herencia y polimorfismo.
- **Principios SOLID**
  - Es un acrónimo de responsabilidad única, abierto-cerrado, sustitución de Liskov, sustitución de interfaces e inversión de dependencias. Estos principios mejoran la legibilidad, mantenibilidad y escalabilidad del código.
- **Patrones de diseño**
  - El patrón de fachada oculta los detalles de implementación para exponer sólo lo que los probadores necesitan crear en los casos de prueba, y el patrón de singleton se utiliza a menudo para asegurarse de que sólo hay un controlador que se comunica con el sistema sujeto a prueba (SSP).

En el modelo de objetos de página, se crea un archivo de clase al que se denomina modelo de página. Siempre que cambie la estructura del sistema sujeto a prueba (SSP), el Ingeniero de automatización de pruebas (IAP) tendrá que hacer actualizaciones en un solo lugar, el localizador dentro de un modelo de página, en lugar de actualizar los localizadores en cada caso de prueba.

El patrón del modelo de flujo es una ampliación del modelo de objetos de página. Introduce una fachada adicional sobre los modelos de objetos de página, que almacena todas las acciones del usuario que interactúan con los objetos de página. Al introducir un diseño de doble fachada, el patrón del modelo de flujo proporciona una abstracción y una mantenibilidad mejoradas, ya que los pasos de prueba pueden reutilizarse en múltiples guiones de prueba.

## 4 Implementación de la automatización de la prueba – (K4)

Duración: 150 minutos

### Palabras Clave<sup>4</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	riesgo	risk
ISTQB	dispositivo de prueba	test fixture

<sup>4</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

## Objetivos de Aprendizaje para “Capítulo 4”

### 4.1 Desarrollo de la automatización de la prueba

**TAE - 4.1.1 (K3)** Aplicar directrices que soporten actividades efectivas de piloto y despliegue de automatización de la prueba.

### 4.2 Riesgos asociados al desarrollo de la automatización de la prueba

**TAE - 4.2.1 (K4)** Analizar los riesgos del despliegue y planificar estrategias de mitigación para la automatización de la prueba.

### 4.3 Mantenibilidad de la solución de automatización de la prueba

**TAE - 4.3.1 (K2)** Explicar los factores que soportan y afectan la mantenibilidad de la solución de automatización de la prueba.



## 4.1 Desarrollo de la automatización de la prueba

### 4.1.1 Aplicar directrices que soporten actividades efectivas de piloto y despliegue de automatización de la prueba

Es importante definir el alcance de la validación de un proyecto piloto de automatización de la prueba. La realización de un proyecto piloto no lleva mucho tiempo, pero el resultado puede tener un impacto significativo en la dirección que tome el proyecto.

A partir de la información recopilada sobre el sistema sujeto a prueba (SSP) y los requisitos del proyecto, para establecer unas directrices que optimicen los esfuerzos de automatización de la prueba, se debería evaluar lo siguiente:

- Lenguaje(s) de programación que se utilizará(n).
- Herramientas de código abierto/comerciales adecuadas.
- Niveles de prueba que se deben cubrir.
- Casos de prueba seleccionados.
- Enfoque de desarrollo de casos de prueba.

Basándose en los puntos enumerados anteriormente, los ingenieros de automatización de pruebas (IAP) pueden definir un enfoque inicial a seguir. Basándose en los requisitos, se pueden crear varios prototipos iniciales diferentes para mostrar los pros y los contras de los distintos enfoques. A partir de ahí, el ingeniero de automatización de pruebas (IAP) puede decidir qué camino seguir.

Definir los plazos es una parte importante para cumplir los calendarios y asegurar el éxito del proyecto piloto. Una recomendación habitual es comprobar periódicamente el avance del proyecto piloto para identificar cualquier riesgo y mitigarlo.

Durante el proyecto piloto, también se recomienda intentar integrar la solución y el código ya implementado en la integración continua/entrega continua (IC/EC). Esto puede sacar a la luz dificultades de forma temprana, ya sea en el sistema sujeto a prueba (SSP), en el solución de automatización de la prueba (SAP) o en la integración general de las distintas herramientas dentro de la organización.

A medida que crece el número de casos de prueba, los Ingeniero de Automatización de Pruebas (IAP) pueden pensar en cambiar la configuración inicial de la integración continua/entrega continua (IC/EC) para ejecutar las pruebas de diferentes maneras y en diferentes momentos.

Además, durante el proyecto piloto, es necesario evaluar otros aspectos no técnicos, como:

- Los conocimientos y la experiencia de los miembros del equipo.
- La estructura del equipo.
- Las reglas de licencias y de la organización.
- El tipo de pruebas previstas y los niveles de prueba previstos que se deben cubrir durante la automatización de los casos de prueba.

Una vez completado el proyecto piloto, el esfuerzo debe ser evaluado por los Ingenieros de automatización de pruebas (IAP) y los jefes de prueba para valorar el éxito o el fallo y tomar una decisión adecuada.

## 4.2 Riesgos asociados al desarrollo de la automatización de la prueba

### 4.2.1 Analizar los riesgos del despliegue y planificar estrategias de mitigación para la automatización de la prueba

Es necesario tener en cuenta la interconexión del marco de trabajo de automatización de la prueba (MTAP) con el sistema sujeto a prueba (SSP) como parte del diseño arquitectónico. A continuación, se pueden seleccionar el empaquetado, el registro en bitácora de prueba y las herramientas de arnés de prueba.

Durante la implementación del piloto, hay que tener en cuenta la ampliación y el mantenimiento del código de automatización de las pruebas. Estos son factores cruciales de la fase de evaluación del piloto y pueden afectar seriamente a la decisión final.

A partir del piloto se pueden identificar diferentes riesgos de implantación:

- Aperturas de cortafuegos.
- Utilización de recursos (por ejemplo, CPU y RAM).

Es necesario prepararse para los riesgos de despliegue, como las dificultades con los cortafuegos, la utilización de recursos, la conexión a la red y la fiabilidad. Estos aspectos no están estrictamente relacionados con la automatización de la prueba, pero los Ingenieros de automatización de pruebas (IAP) necesitan asegurar que se cumplen todas las condiciones para proporcionar puertas de calidad fiables y beneficiosas en su proceso de desarrollo.

El uso de dispositivos reales para la automatización de la prueba móvil ofrece un ejemplo. Los dispositivos móviles deben estar encendidos, tener batería suficiente para funcionar durante la prueba, estar conectados a una red y tener acceso al sistema sujeto a prueba (SSP).

Los riesgos técnicos del despliegue pueden incluir:

- Empaquetado
- Registro
- Estructuración de la prueba
- Actualización
- **Empaquetado**
  - Es necesario tener en cuenta el empaquetado, ya que el control de versión de la automatización de la prueba es tan importante como para el sistema sujeto a prueba (SSP). Es posible que el producto de prueba necesite cargarse en un repositorio para compartirlo en toda la organización, ya sea en las instalaciones o en la nube.
- **Registro**
  - El registro en bitácora de prueba proporciona la mayor parte de la información sobre los resultados de prueba. Existen varios niveles de registro de la prueba y todos ellos son útiles en la automatización de la prueba por diversas razones:
    - **Fatal:** este nivel se utiliza para registrar eventos de error que pueden llevar a abortar la ejecución de prueba.
    - **Error:** este nivel se utiliza cuando falla una condición o interacción y, por tanto, falla también el caso de prueba
    - **Advertencia:** Este nivel se utiliza cuando se produce una condición/acción inesperada pero que no rompe el flujo del caso de prueba.

- **Info:** Este nivel se utiliza para mostrar información básica sobre un caso de prueba y lo que ocurre durante la ejecución de prueba.
  - **Depuración:** Este nivel se utiliza para almacenar detalles específicos de la ejecución que, por lo general, no son necesarios para los registros básicos, pero que resultan útiles durante la investigación de un fallo en la prueba.
  - **Traza:** Este nivel es similar al de **Depuración**, pero tiene aún más información.
- **Estructuración de la prueba**
    - La parte más importante de la solución de automatización de la prueba (SAP) es el arnés de prueba y los dispositivos de prueba incluidos en él, elementos que deben estar disponibles para ejecutar las pruebas. Las instalaciones de prueba proporcionan libertad para controlar el entorno de prueba y los datos de prueba. Pueden definirse precondiciones y postcondiciones para la ejecución de las pruebas y los casos de prueba pueden agruparse en conjuntos de pruebas de varias maneras. Estos aspectos también son importantes para la evaluación durante un proyecto piloto. Además, las instalaciones de prueba posibilitan la creación de pruebas automatizadas que son repetibles y atómicas.
  - **Actualización**
    - Uno de los riesgos técnicos más comunes son las actualizaciones automáticas de los arneses de prueba (por ejemplo, los agentes) y los cambios de versión en los dispositivos. Estos riesgos pueden mitigarse disponiendo de fuentes de alimentación adecuadas, conexiones de red apropiadas y planes de configuración de dispositivos adecuados.

## 4.3 Mantenibilidad de la solución de automatización de la prueba

### 4.3.1 Explicar los factores que soportan y afectan la mantenibilidad de la solución de automatización de la prueba

Una regla básica es intentar seguir los principios de código limpio de Robert C. Martin (Robert C Martin, “Clean Code: A Handbook of Agile Software Craftsmanship”, 2008).

En pocas palabras, los principios del código limpio hacen hincapié en los siguientes puntos:

- Utilizar una convención de nomenclatura común para las clases, los métodos y las variables para que los nombres tengan sentido.
- Utilizar una estructura de proyecto lógica y común.
- Evitar la codificación rígida.
- Evitar demasiados parámetros de entrada para los métodos.
- Evitar métodos largos y complejos.
- Utilizar registros.
- Utilizar patrones de diseño cuando sean beneficiosos y necesarios.
- Concentrarse en la capacidad de ser probado.

Las convenciones de nomenclatura son muy útiles para identificar el objetivo de una variable determinada. Disponer de nombres de variables comprensibles como “botónInicioSesión”, “botónRestablecimientoContraseña” ayuda a los Ingenieros de automatización de pruebas (IAP) a entender qué componente utilizar.

La codificación rígida es el proceso de incrustar valores en el software sin la posibilidad de cambiarlos directamente. Puede evitarse utilizando pruebas guiadas por datos, de modo que los datos de prueba procedan de una fuente común que pueda mantenerse más fácilmente. La codificación rígida reduce el tiempo de desarrollo, pero no se recomienda utilizarla porque los datos pueden cambiar con frecuencia, lo que puede llevar aparejado un tiempo considerable de mantenimiento. También se aconseja utilizar constantes para aquellas variables que no se espera que cambien con frecuencia. Al hacerlo, se pueden reducir las fuentes y los lugares que necesitan mantenimiento.

También es muy recomendable aprovechar los patrones de diseño. Los patrones de diseño -como se describe en 3.1.5- posibilitan la implementación de un código de automatización de la prueba estructurado y adecuadamente mantenible, siempre y cuando los patrones de diseño se utilicen correctamente.

Para asegurar código de automatización de pruebas de alta calidad, se recomienda hacer uso de analizadores estáticos. Los formateadores de código como los utilizados habitualmente en los entornos de desarrollo integrado (EDI) mejorarán la legibilidad del código de automatización de la prueba.

Aparte de los principios de código limpio, se recomienda utilizar una estructura y una estrategia de ramificación acordadas en el control de versión. Utilizar diferentes ramas para las prestaciones, las entregas y las correcciones de defectos ayuda a comprender el contenido de la rama.

## 5 Estrategias de implementación y despliegue para la automatización de la prueba - (K3)

Duración: 90 minutos

### Palabras Clave<sup>5</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	prueba de contrato	contract testing

<sup>5</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

## Objetivos de Aprendizaje para “Capítulo 5”

### 5.1 Integración en canalizaciones de integración continua/entrega continua (IC/EC)

- TAE - 5.1.1 (K3)** Aplicar la automatización de la prueba en diferentes niveles de prueba dentro de las canalizaciones.
- TAE - 5.1.2 (K2)** Explicar la gestión de la configuración para el producto de prueba.
- TAE - 5.1.3 (K2)** Explicar las dependencias de automatización de pruebas para la infraestructura de una interfaz de programación de aplicación (IPA).

## 5.1 Integración en canalizaciones de integración continua/entrega continua (IC/EC)

### 5.1.1 Aplicar la automatización de la prueba en diferentes niveles de prueba dentro de las canalizaciones

Uno de los principales beneficios de la automatización de la prueba es que las pruebas implementadas pueden ejecutarse sin supervisión, lo que las convierte en candidatas ideales para ejecutarse dentro de canalizaciones. Esto puede lograrse mediante canalizaciones de integración continua/entrega continua (CI/CD) o mediante la canalización utilizada para ejecutar pruebas con regularidad.

Los niveles de prueba suelen integrarse de la siguiente forma:

- Pruebas de configuración para marco de trabajo de automatización de la prueba (MTAP)/solución de automatización de la prueba (SAP), durante la construcción pueden tenerse en cuenta como una subespecie de las pruebas de componente. Estas pruebas se ejecutan durante la construcción de un proyecto de automatización de la prueba (MTAP/SAP) y comprueban que todos los caminos a los archivos utilizados en los guiones de prueba son correctos, que los archivos existen realmente y se encuentran en los caminos especificados.
- Las pruebas de componentes forman parte del paso de construcción de la canalización, ya que se ejecutan en los componentes individuales, (por ejemplo, las clases de la librería y los componentes web). Actúan como puertas de calidad para la canalización, por lo que son una parte crucial de una canalización de integración continua.
- Las pruebas de integración de componentes pueden formar parte de la canalización de integración continua si son pruebas de componentes de bajo nivel o del sistema sujeto a prueba (SSP). En tales casos, estas pruebas y las pruebas de componentes se ejecutan conjuntamente.
- Las pruebas de sistema pueden integrarse a menudo en una canalización de despliegue continuo, donde actúan como última puerta de calidad del sistema sujeto a prueba (SSP) entregado.
- Las pruebas de integración de sistemas entre los distintos componentes del sistema suelen formar parte de una canalización de entrega continua como puertas de calidad. Estas pruebas de integración de sistemas aseguran que los componentes del sistema desarrollados por separado están trabajando juntos.

Muchos sistemas modernos de integración continua diferencian entre las fases de construcción y despliegue de las canalizaciones de entrega continua. En estos casos, las pruebas de componentes y las pruebas de integración de componentes forman parte de la primera fase de construcción. Cuando esta primera fase tiene éxito (es decir, construcción y prueba de forma conjunta), los componentes/sistema(s) sujeto(s) a prueba (SSP)(s) se despliegan.

En el caso de pruebas de integración de sistemas, pruebas de sistema y pruebas de aceptación, existen dos enfoques principales para integrarlas en tales canalizaciones:

1. Los casos de prueba se ejecutan como parte de la fase de despliegue tras el despliegue del componente. Esto puede ser beneficioso, ya que, en función de los resultados de las pruebas, el despliegue puede fallar y también ser revertido. Sin embargo, en este caso, si se necesitan volver a ejecutar las pruebas, hay que volver a realizar un despliegue.
2. Los casos de prueba se ejecutan como una canalización independiente, desencadenada por el despliegue satisfactorio. Esto puede ser beneficioso si se espera que en cada despliegue se ejecuten diferentes conjuntos de pruebas y diversos códigos de automatización de la prueba. En este caso, las pruebas no actúan como puerta de calidad. Por lo tanto, se requieren otras acciones,

normalmente manuales, para retrotraer un despliegue fallido. En este caso, pueden utilizarse algunos guiones de prueba automatizados sencillos, como comprobaciones de despliegue, para asegurar que el sistema sujeto a prueba (SSP) se despliega, pero estos guiones de prueba automatizados no comprueban la adecuación funcional de forma amplia.

Las canalizaciones también pueden utilizarse para otros fines de automatización de la prueba, tales como:

- Ejecución periódica de diferentes juegos de prueba: Se puede ejecutar un juego de regresión cada noche (es decir, la regresión nocturna), especialmente para los juegos de prueba de larga duración, de modo que el equipo tenga una imagen clara de la calidad del sistema sujeto a prueba (SSP) por la mañana.
- Ejecución de pruebas no funcionales: Ya sea como parte de una canalización de despliegue continuo, o por separado, para monitorizar periódicamente determinadas características de calidad no funcionales del sistema, como la eficiencia de desempeño.

### 5.1.2 Explicar la gestión de la configuración para el producto de prueba

La gestión de la configuración es una parte integral de la automatización de la prueba, ya que la automatización se ejecutará a menudo en múltiples entornos de prueba y versiones del sistema sujeto a prueba (SSP).

La gestión de la configuración en la automatización de la prueba incluye:

- Configuración del entorno de prueba
- Datos de prueba
- Juegos de prueba/casos de prueba
- **Configuración del entorno de prueba**
  - Cada entorno de prueba utilizado en la canalización del desarrollo puede tener diferentes configuraciones, como varias URL o credenciales. La configuración del entorno de prueba suele almacenarse con el producto de prueba. Sin embargo, en el caso de la automatización de la prueba utilizada en múltiples proyectos o múltiples marcos de trabajo de automatización de la prueba (MTAP) para el mismo proyecto, la configuración del entorno de prueba puede formar parte de la librería base común o estar en un repositorio compartido.
- **Datos de prueba**
  - Los datos de prueba también pueden ser específicos para el entorno de prueba o para la entrega y el conjunto de prestaciones del sistema sujeto a prueba (SSP). Al igual que ocurre con la configuración del entorno de prueba, los datos de prueba suelen almacenarse en marcos de trabajo de automatización de la prueba (MTAP) más pequeños, pero también pueden utilizarse sistemas de gestión de datos de prueba.
- **Juegos de prueba/casos de prueba**
  - Una práctica habitual consiste en configurar distintos conjuntos de casos de prueba en función de su finalidad, como la prueba de humo o la prueba de regresión. Estos conjuntos de prueba suelen ejecutarse en niveles de prueba separados, aprovechando diferentes canalizaciones y entornos de prueba.

Cada entrega del sistema sujeto a prueba (SSP) determina un conjunto de prestaciones que incluye casos de prueba y conjuntos de prueba que evalúan la calidad de la entrega en cuestión. Existen diferentes opciones en el producto de prueba para su tratamiento:



- Se puede definir un conmutador de configuración de la prestación por cada entrega o entorno de prueba. Hay casos de prueba y conjuntos de prueba para probar cada prestación. El conmutador de prestaciones puede utilizarse en el software de prueba para identificar los conjuntos de prueba que deben ejecutarse en una determinada versión/entorno de prueba.
- El producto de prueba también puede entregarse con el sistema sujeto a prueba (SSP) utilizando la misma versión de entrega. De este modo, existe una coincidencia exacta entre la versión del sistema sujeto a prueba (SSP) y el producto de prueba que puede probarlo. Una entrega de este tipo suele implementarse mediante un sistema de gestión de la configuración que utiliza etiquetas o ramas.

### 5.1.3 Explicar las dependencias de automatización de pruebas para la infraestructura de una IPA

Al realizar la automatización de la prueba de la interfaz de programación de aplicación (IPA), es crucial disponer de la siguiente información sobre las dependencias para construir una estrategia adecuada:

- Conexiones de la interfaz de programación de aplicación (IPA): Comprender la lógica de negocio que puede probarse automáticamente y la relación entre las interfaces de programación de aplicación (IPA).
- Documentación de la interfaz de programación de aplicación (IPA): Sirve como línea base para la automatización de la prueba con toda la información pertinente (por ejemplo, parámetros, encabezados y distintos tipos de solicitud-respuesta de objetos).

La prueba de interfaz de programación de aplicación (IPA) integrada y automatizada puede ser realizada tanto por los desarrolladores como por los ingenieros de automatización de pruebas (IAP). Sin embargo, con el desplazamiento a la izquierda se recomienda apoyar y dividir las pruebas entre distintos niveles. En el programa de estudio CTFL de ISTQB se mencionan las pruebas de integración de componentes y las pruebas de integración de sistemas, que pueden ampliarse con una buena práctica denominada prueba de contrato.

- **Prueba de contrato**
  - La prueba de contrato es un tipo de prueba de integración que verifica que los servicios pueden comunicarse entre sí y que los datos compartidos entre los servicios son consistentes con un conjunto especificado de reglas. El uso de la prueba de contrato proporciona compatibilidad entre dos sistemas separados (por ejemplo, dos microservicios) para comunicarse entre sí. Va más allá de la validación de esquemas, ya que requiere que ambas partes lleguen a un consenso sobre el conjunto de interacciones permitidas, al tiempo que prevé la evolución a lo largo del tiempo. Captura las interacciones que se intercambian entre cada servicio, almacenándolas en un contrato, que luego puede utilizarse para verificar que ambas partes se adhieren a él. Una de las principales ventajas de este tipo de prueba es que los defectos que se producen en los servicios subyacentes pueden encontrarse antes en el ciclo de vida de desarrollo de software (CVDS) y el origen de estos defectos puede identificarse más fácilmente.

En el enfoque de las pruebas de contrato impulsadas por el consumidor, éste establece sus expectativas determinando cómo responderá el proveedor a las peticiones que provengan de él. En el enfoque de las pruebas de contrato impulsado por el proveedor, éste crea el contrato, que muestra cómo funcionan sus servicios.

## 6 Suministro de información de la automatización de la prueba y métricas

Duración: 150 minutos

### Palabras Clave<sup>6</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	medición	measurement
	métrica	metric
	registro de prueba	test log
	informe del avance de la prueba	test progress report
	paso de prueba	test step

<sup>6</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

## Objetivos de Aprendizaje para “Capítulo 6”

### 6.1 Recopilación, análisis y suministro de información de los datos de automatización de la prueba

- TAE - 6.1.1 (K3)** Aplicar métodos de recopilación de datos de la solución de automatización de la prueba y del sistema sujeto a prueba.
- TAE - 6.1.2 (K4)** Analizar los datos de la solución de automatización de la prueba y del sistema sujeto a prueba para comprender mejor los resultados.
- TAE - 6.1.3 (K2)** Explicar cómo se elabora y publica un informe del avance de la prueba.

## 6.1 Recopilación, análisis y suministro de información de los datos de automatización de la prueba

### 6.1.1 Aplicar métodos de recopilación de datos de la solución de automatización de la prueba y del sistema sujeto a prueba

Se pueden recopilar datos de las siguientes fuentes:

- Registros del sistema sujeto a prueba (SSP).
  - Interfaz de Usuario Web/móvil.
  - interfaz de programación de aplicación (IPA).
  - Aplicaciones.
  - Servidores web.
  - Servidores de bases de datos.
- Registros del marco de trabajo de automatización de la prueba (MTAP) para proporcionar un rastro de auditoría.
- Registros de construcción.
- Registros de despliegue.
- Registros de producción para monitorizar los datos en producción (véase el programa de estudio ISTQB CT-PT, sección 2.3).
  - Monitorización del rendimiento en producción para realizar análisis de tendencias.
  - Registros de pruebas de eficiencia de rendimiento en un entorno de prueba de rendimiento (por ejemplo, pruebas de carga, estrés y picos).
- Capturas y grabaciones de pantalla (nativas de la herramienta de automatización o de terceros).

Dado que una solución de automatización de la prueba (SAP) tiene un producto de prueba automatizado en su núcleo, éste puede mejorarse para registrar información sobre su uso. Las mejoras del software de prueba realizadas en el software de prueba subyacente pueden ser utilizadas por todos los guiones de prueba automatizados de nivel superior. Por ejemplo, la mejora del software de pruebas subyacente para registrar la hora de inicio y fin de la ejecución de la prueba puede aplicarse a todas las pruebas.

- **Prestaciones de la automatización de la prueba que soportan la medición y la generación de informes de prueba**
  - Los lenguajes de guion de muchas herramientas de pruebas admiten la medición y la generación de informes a través de funciones que pueden utilizarse para registrar y anotar información antes, durante y después de la ejecución de pruebas individuales y de juegos de prueba completos.

El suministro de información de prueba sobre cada una de una serie de ejecuciones de prueba necesita tener en cuenta una prestación de análisis de los resultados de prueba de las ejecuciones de prueba anteriores para poder resaltar tendencias, como los cambios en la tasa de éxito de la prueba.

La automatización de la prueba suele requerir la automatización tanto de la ejecución de la prueba como de la verificación de la prueba; esta última se consigue comparando elementos específicos de los resultados reales con los resultados esperados. La mejor

forma de realizar esta comparación es mediante una herramienta de prueba que utilice aseveraciones. Hay que tener en cuenta el nivel de información que se informa como resultado de esta comparación. Es importante que el estado de la prueba se determine correctamente (es decir, si la prueba ha pasado o ha fallado). En el caso del estado fallado, se requerirá más información sobre la causa del fallo (por ejemplo, capturas de pantalla).

Las diferencias entre los resultados reales y los resultados esperados de una prueba no siempre están claras, y el soporte de la herramienta puede ayudar de forma significativa a definir comparaciones que ignoren las diferencias esperadas, como las fechas y las horas, al tiempo que resaltan cualquier diferencia inesperada.

- **Registro en bitácora de prueba**
  - Los registros de pruebas son una fuente que se utiliza con frecuencia para analizar posibles defectos en la solución de automatización de la prueba (SAP) y en el sistema sujeto a prueba (SSP). En la siguiente sección se presentan ejemplos de registro en bitácora de prueba, categorizados por solución de automatización de la prueba (SAP) y sistema sujeto a prueba (SSP).
- **Registro de la solución de automatización de la prueba (SAP)**
  - El contexto determina si el marco de trabajo de automatización de la prueba (MTAP) o la ejecución de prueba es responsable del registro de la información, que debe incluir lo siguiente:
    - Qué caso de prueba se encuentra actualmente en la ejecución de prueba, incluida la hora de inicio y de finalización.
    - El estado de la ejecución de la prueba porque, aunque los fallos pueden identificarse fácilmente en los registros de prueba, el marco de trabajo de automatización de la prueba (MTAP) también debería disponer de esta información e informar a través de un panel de control. El estado de ejecución de prueba puede ser paso, fallo o fallo de solución de automatización de la prueba (SAP). A veces, los estados pueden no ser concluyentes y es importante que una organización establezca definiciones claras y consistentes para ellos. Un fallo de solución de automatización de la prueba (SAP) se aplica a situaciones en las que el defecto no se encuentra en el sistema sujeto a prueba (SSP).
    - Detalles de bajo nivel en el registro de la prueba (por ejemplo, registro en bitácora de prueba de los pasos significativos), incluida la información sobre la cronología.
    - Información dinámica sobre el sistema sujeto a prueba (SSP) (por ejemplo, fugas de memoria) que el caso de prueba pudo identificar con la ayuda de herramientas de terceros. Los resultados reales y los fallos deben registrarse con el juego de prueba que se ejecutó cuando se detectó el fallo.
    - En el caso de pruebas de fiabilidad o pruebas de esfuerzo en las que se realizan numerosos ciclos de prueba, debe registrarse un contador para determinar fácilmente cuántas veces se han ejecutado los casos de prueba.
    - Cuando los casos de prueba tengan elementos aleatorios (por ejemplo, parámetros aleatorios o pasos de prueba aleatorios en las pruebas de transición de estado), deberá registrarse el número/las opciones aleatorias.
    - Todas las acciones que realice un caso de prueba deben registrarse de tal forma que los registros de prueba, o partes de ellos, puedan reproducirse para volver a ejecutar la prueba con los mismos pasos de prueba y la misma cronología. Esto es útil para reproducir un fallo identificado y para capturar información adicional. La información de la acción del caso de prueba también puede ser registrada por el sistema sujeto a prueba (SSP) para utilizarla al reproducir fallos identificados por el cliente. Si un cliente ejecuta un juego de prueba, la información del registro

de prueba se captura y puede ser reproducida por el equipo de desarrollo al solucionar un defecto.

- Se pueden guardar capturas de pantalla durante la ejecución de prueba para su uso posterior durante el análisis de la causa raíz.
  - Siempre que un juego de prueba inicie un fallo, la solución de automatización de la prueba (SAP) debe asegurarse de que toda la información necesaria para analizar el defecto está disponible/almacenada, así como cualquier información relativa a la continuación de las pruebas, si es aplicable. La solución de automatización de la prueba (SAP) debe guardar cualquier volcado de fallos y trazas de pila asociados. Asimismo, cualquier registro de prueba que pueda sobrescribirse (por ejemplo, a menudo se utilizan memorias intermedias cíclicas para los registros de prueba en el sistema sujeto a prueba (SSP)) debe almacenarse donde esté disponible para su análisis posterior.
  - El uso del color puede ayudar a distinguir los distintos tipos de información del registro de pruebas (por ejemplo, los defectos en rojo y la información sobre el avance en verde).
- **Registro del sistema sujeto a prueba (SSP)**
    - La correlación de los resultados de la automatización de la prueba con los registros del sistema sujeto a prueba (SSP) ayuda a identificar la causa raíz de los defectos en el sistema sujeto a prueba (SSP) y en la solución de automatización de la prueba (SAP).
      - Cuando se identifica un defecto en el sistema sujeto a prueba (SSP), se debería registrar toda la información necesaria para analizar el defecto, incluidas las marcas de fecha y hora, la ubicación de origen del defecto y los mensajes de error.
      - Al iniciar un sistema se debería registrar en un archivo la información de configuración, consistente, por ejemplo, en las diferentes versiones de software/firmware, la configuración del sistema sujeto a prueba (SSP) y la configuración del sistema operativo.
      - Mediante la automatización de la prueba, los registros del sistema sujeto a prueba (SSP) pueden buscarse fácilmente. Se debería poder identificar fácilmente un fallo identificado en el registro de pruebas por la solución de automatización de la prueba (SAP) en el registro de prueba del sistema sujeto a prueba (SSP), y viceversa, con o sin herramientas adicionales. La sincronización de varios registros de prueba con una marca de tiempo facilita la correlación de lo ocurrido cuando se informa de un fallo.
  - **Integración con otras herramientas de terceros (por ejemplo, hojas de cálculo, XML, documentos, bases de datos y herramientas de informes)**
    - Cuando la información procedente de la ejecución de casos de prueba automatizados se utiliza en otras herramientas para el seguimiento y la elaboración de informes (por ejemplo, la actualización de la información de trazabilidad), es posible proporcionar la información en un formato adecuado para herramientas de terceros. Esto se consigue a menudo a través de la funcionalidad existente de la herramienta de prueba (por ejemplo, formatos de exportación para los informes de prueba) o mediante la creación de informes personalizados que se emiten en un formato consistente con otro software.
  - **Visualización de los resultados de prueba**
    - Los resultados de las pruebas pueden hacerse visibles mediante gráficos. Tener en cuenta el uso de iconos de colores, como semáforos, para indicar el estado general de la ejecución de la prueba/la automatización de la prueba, de modo que se puedan tomar decisiones basadas en el suministro de información. El personal directivo está especialmente interesado en los resúmenes visuales para ver los resultados de prueba, lo

que ayuda en la toma de decisiones. Si necesitan más información, pueden profundizar en los detalles.

## 6.1.2 Analizar los datos de la solución de automatización de la prueba y del sistema sujeto a prueba para comprender mejor los resultados

Tras la ejecución de pruebas, es importante analizar los resultados de prueba para identificar posibles fallos tanto en el sistema sujeto a prueba (SSP) como en la solución de automatización de la prueba (SAP). Para dicho análisis, los datos recogidos del TAS son primarios y los recogidos del sistema sujeto a prueba (SSP) son secundarios.

- Analizar los datos del entorno de prueba para prestar apoyo al dimensionamiento adecuado de la automatización de la prueba (por ejemplo, en la nube).
  - Agrupamientos y recursos (por ejemplo, CPU y RAM)
  - Ejecución de pruebas con un único navegador o con varios (es decir, con varios navegadores)
- Comparar los resultados de pruebas de ejecuciones de prueba anteriores.
- Determinar cómo utilizar los registros web para monitorizar el uso del software.

Los fallos en la ejecución de pruebas necesitan ser analizados debido a posibles dificultades:

1. Comprobar si se ha producido el mismo fallo en ejecuciones de prueba previas. Puede tratarse de un defecto conocido en el sistema sujeto a prueba (SSP) o en la solución de automatización de la prueba (SAP). La solución de automatización de la prueba (SAP) puede construirse para registrar los resultados históricos de los casos de prueba, lo que facilita aún más el análisis.
2. Si el defecto no es conocido es necesario identificar el caso de prueba y qué está probando. La prueba puede ser auto-explicativa, o el caso de prueba puede ser identificado en el sistema de gestión de la prueba, basado en su ID registrado con la ejecución de prueba.
3. Encontrar en qué paso de la prueba del caso de prueba se produjo el fallo. solución de automatización de la prueba (SAP) lo registra.
4. Analizar la información del registro de prueba sobre el estado del sistema sujeto a prueba (SSP) y si coincide con los resultados esperados mediante capturas de pantalla, registros de interfaz de programación de aplicación (IPA) y de red, o cualquier registro que muestre el estado del sistema sujeto a prueba (SSP).
5. Si el estado del sistema sujeto a prueba (SSP) no es el esperado, registrar un defecto en el sistema de gestión de defectos. Asegurarse de incluir toda la información necesaria sobre el defecto y los registros que justifiquen que se trata de un defecto.

Cuando se produce un fallo, es posible que el resultado real y el resultado esperado del sistema sujeto a prueba (SSP) coincidan. En este caso, lo más probable es que la solución de automatización de la prueba (SAP) contenga un defecto que necesita ser corregido o que exista un desajuste invisible.

Otra situación que puede darse es que el entorno de prueba no esté disponible durante la realización de prueba, o sólo lo esté parcialmente. En este caso, todos los casos de prueba pueden fallar, ya sea con el mismo defecto o si partes del sistema no funcionan, con fallos aparentemente reales. Para identificar la causa raíz de estos defectos, pueden analizarse los registros del sistema sujeto a prueba (SSP), que mostrarán si se produjo alguna interrupción del entorno de prueba en el momento de la realización de la prueba.

Si el sistema sujeto a prueba (SSP) implementa registros de auditoría para las interacciones de los usuarios (es decir, sesiones de interfaz de usuario o llamadas a la interfaz de programación de aplicación (IPA)), ayuda a analizar los resultados de prueba. Suele haber un ID único añadido a la interacción con el

mismo ID para cada llamada e integración posteriores en el sistema. De este modo, conociendo el ID único de una solicitud/interacción, se puede observar y trazar el comportamiento del sistema.

Este ID único suele denominarse ID de correlación o ID de traza. La solución de automatización de la prueba (SAP) puede registrarlo para ayudar a analizar los resultados de prueba.

### 6.1.3 Explicar cómo se elabora y publica un informe del avance de la prueba

Los registros de prueba proporcionan información detallada sobre los pasos de prueba, las acciones a realizar y las respuestas esperadas de un caso de prueba y/o juego de prueba. Sin embargo, los registros en bitácora de prueba por sí solos no pueden proporcionar una buena visión de conjunto de los resultados de prueba. Para ello, es necesario disponer de una funcionalidad de suministro de información de prueba. Tras la ejecución de un juego de prueba, debe crearse y publicarse un informe conciso del avance de la prueba. Para ello puede utilizarse un generador de informes.

- **Contenido de un informe del avance de la prueba**

- El informe del avance de la prueba debe contener los resultados de la prueba, información sobre el sistema sujeto a prueba (SSP) y documentación sobre el entorno de prueba en el que se han realizado las pruebas en un formato adecuado para cada uno de los implicados.

Es necesario saber qué pruebas han fallado y los motivos del fallo. Para facilitar la resolución de problemas, es importante conocer el historial de ejecución de pruebas y quién lo ha informado (es decir, generalmente la persona que lo creó o lo actualizó por última vez). La persona responsable tiene que investigar la causa del fallo, informar del defecto relacionado con él, hacer un seguimiento de la solución del defecto y probar que la solución se ha implementado correctamente.

El suministro de información de prueba también se utiliza para diagnosticar cualquier fallo de los componentes del marco de trabajo de automatización de la prueba (MTAP).

- **Publicación de los informes de prueba**

- Una opción es identificar las partes problemáticas del sistema sujeto a prueba (SSP) y mantener un historial de los informes de prueba, de modo que puedan recopilarse estadísticas sobre casos de prueba o juegos de prueba con regresiones frecuentes para el análisis de tendencias.

- **Entre los implicados a los que se informa se encuentran los siguientes:**

- Implicados en la gestión
  - Roles típicos: arquitecto de la solución o de la empresa, jefe de proyecto/entrega, director de programa, jefe de prueba o director de prueba.
- Implicados operativos
  - Roles típicos: propietario de producto/gestor, representante de negocio o analista de negocio.
- Implicados técnicos
  - Roles típicos: jefe de equipo, scrum master, administrador web, desarrollador/administrador de bases de datos, líder de la prueba, ingeniero de automatización de pruebas (IAP), probador o desarrollador.

Los informes de prueba pueden variar en contenido o detalle en función de los destinatarios. Mientras que las partes interesadas técnicas pueden estar más interesadas en detalles de bajo nivel, la gestión se centrará en las tendencias, como cuántos casos de prueba se han añadido desde la última realización de prueba, los cambios en la proporción de aprobados y suspensos y la fiabilidad de la solución de



automatización de la prueba (SAP). Los implicados en la operación suelen hacer más hincapié en las métricas relacionadas con el uso del producto.

- **Creación de paneles de control**

- Las herramientas modernas de suministro de información ofrecen varias opciones para informar a través de paneles de control, gráficos de colores, colecciones detalladas de registros y análisis automatizado de la bitácora de prueba. Hay muchas herramientas disponibles en el mercado entre las que elegir.

Estas herramientas admiten la agregación de datos de fuentes como registros en bitácora de prueba de ejecución de la prueba, herramientas de gestión de proyecto y repositorios de código. La visualización de datos que proporcionan estas herramientas ayuda a los implicados a ver tendencias y tomar decisiones en consecuencia. Estas tendencias pueden incluir agrupamientos de defectos, aumento/disminución de la propagación de defectos a determinados entornos de prueba, degradación del rendimiento del sistema sujeto a prueba (SSP) y fiabilidad de las construcciones.

- **Inteligencia artificial/aprendizaje automático: análisis de registros de prueba**

- En los últimos años, algunas herramientas de automatización de la prueba incluyen algoritmos de aprendizaje automático (AA) o se basan en ellos. El análisis automatizado de grandes cantidades de datos en los registros de prueba ayuda al IA a reducir el tiempo dedicado a encontrar localizadores dañados, analizar el motivo de los fallos de prueba (es decir, ¿se trata de un defecto en el sistema sujeto a prueba (SSP) o en la solución de automatización de la prueba (SAP)?) y agrupar los defectos comunes para los informes de prueba (véase el programa de estudio ISTQB CT-IA).

## 7 Verificación de la solución de automatización de la prueba - (K3)

Duración: 135 minutos

### Palabras Clave<sup>7</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	análisis estático	static analysis

<sup>7</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.

## Objetivos de Aprendizaje para “Capítulo 7”

### 7.1 Verificación de la infraestructura de automatización de la prueba

- TAE - 7.1.1 (K3)** Planificar la verificación del entorno de automatización de la prueba, incluyendo la configuración de la herramienta de prueba.
- TAE - 7.1.2 (K2)** Explicar el comportamiento correcto para un guion de prueba automatizado y/o un juego de prueba dado.
- TAE - 7.1.3 (K2)** Identificar dónde la automatización de la prueba produce resultados inesperados
- TAE - 7.1.4 (K2)** Explicar cómo el análisis estático puede ayudar a la calidad del código de automatización de pruebas.

## 7.1 Verificación de la infraestructura de automatización de la prueba

### 7.1.1 Planificar la verificación del entorno de automatización de la prueba, incluyendo la configuración de la herramienta de prueba

Es necesario comprobar si el entorno de automatización de la prueba y todos los demás componentes de la solución de automatización de la prueba (SAP) funcionan como se espera. Estas comprobaciones se realizan, por ejemplo, antes de iniciar la automatización de la prueba. Se puede tomar una serie de medidas para verificar los componentes del entorno de automatización de la prueba. A continuación, se explica cada uno de ellos con más detalle.

- **Instalación, configuración y personalización de la herramienta de prueba**
  - La solución de automatización de la prueba (SAP) consta de muchos componentes. Es necesario tener en cuenta a cada uno de ellos para asegurar un rendimiento fiable y repetible. En el núcleo de una solución de automatización de la prueba (SAP) se encuentran los componentes ejecutables, las librerías de funciones correspondientes y los archivos de datos y configuración de apoyo. El proceso de configuración de una solución de automatización de la prueba (SAP) puede ir desde el uso de guiones de instalación automatizados hasta la introducción manual de los archivos en las carpetas correspondientes. Las herramientas de prueba, al igual que los sistemas operativos y otros programas informáticos, suelen tener paquetes de servicios o pueden tener complementos opcionales o necesarios para garantizar la compatibilidad con cualquier entorno de sistema sujeto a prueba (SSP).  
  
La instalación automatizada o la copia desde un repositorio tiene ventajas. Puede garantizar que las pruebas sobre diferentes sistemas sujetos a prueba (SSP) se han realizado con la misma versión de la solución de automatización de la prueba (SAP) y con la misma configuración de SAP, cuando proceda. Las actualizaciones de la solución de automatización de la prueba (SAP) pueden realizarse a través del repositorio. El uso de un repositorio y el proceso de actualización a una nueva versión de la solución de automatización de la prueba (SAP) deben ser los mismos que los de las herramientas de desarrollo estándar.
- **Repetibilidad en la configuración/desmontaje del entorno de prueba**
  - Se implementará una solución de automatización de la prueba (SAP) en diversos sistemas, servidores y para dar soporte a canalizaciones de integración continua/entrega continua (IC/EC). Para asegurar que la SAP funciona correctamente en cada entorno de prueba, es necesario tener un enfoque sistemático para cargar y descargar la SAP de cualquier entorno de prueba dado. Esto se logra de forma satisfactoria cuando la construcción y reconstrucción de la solución de automatización de la prueba (SAP) no proporciona ninguna diferencia perceptible en la forma en que funciona dentro y a través de múltiples entornos de prueba. La gestión de la configuración de los componentes de la solución de automatización de la prueba (SAP) asegura que se pueda crear una configuración determinada de forma fiable. Una vez hecho esto, la documentación de los distintos componentes de la SAP proporcionará los conocimientos necesarios sobre los aspectos de la SAP que puedan verse afectados o que deban modificarse cuando cambie el entorno del sistema sujeto a prueba (SSP).
- **Conectividad con sistemas/interfaces internos y externos**
  - Una vez instalada una solución de automatización de la prueba (SAP) en un determinado entorno de sistema sujeto a prueba (SSP), y antes de utilizarla, debe llevarse a cabo una

serie de comprobaciones o precondiciones para asegurar la disponibilidad de la conectividad con los sistemas internos, los sistemas externos y las interfaces. Por ejemplo, es una buena práctica iniciar sesión en los servidores, poner en marcha las herramientas de automatización de la prueba, verificar que las herramientas de automatización de la prueba pueden acceder al sistema sujeto a prueba (SSP), inspeccionar manualmente los ajustes de configuración y asegurarse de que los permisos están configurados correctamente para el registro en bitácora de prueba y el suministro de información de prueba entre sistemas. Establecer las condiciones previas para la automatización de la prueba es esencial para asegurar que la solución de automatización de la prueba (SAP) se ha instalado y configurado correctamente.

- **Prueba de componentes del marco de trabajo de automatización de la prueba (MTAP)**
  - Al igual que cualquier proyecto de desarrollo de software, los componentes del marco de trabajo de automatización de la prueba (MTAP) necesitan ser probados y verificados individualmente. Esto puede incluir pruebas funcionales y no funcionales (por ejemplo, eficiencia de rendimiento y utilización de recursos). Por ejemplo, los componentes que proporcionan verificación de objetos en sistemas IGU (“Graphic User Interface - GUI”) necesitan ser probados para una amplia gama de clases de objetos para establecer que la verificación de objetos funciona correctamente. Del mismo modo, los registros en bitácora de prueba y los informes de prueba deben producir información precisa sobre el estado de la automatización de la prueba y el comportamiento del sistema sujeto a prueba (SSP). Algunos ejemplos de pruebas no funcionales pueden ser la comprensión de la degradación del rendimiento del marco de trabajo de automatización de la prueba (MTAP), la utilización de los recursos del sistema que pueden indicar defectos como fugas de memoria y la falta de interoperabilidad de los componentes dentro y/o fuera del marco de trabajo de automatización de la prueba (MTAP).

### 7.1.2 Explicar el comportamiento correcto para un guion de prueba automatizado y/o un juego de prueba dado

Los juegos de prueba automatizados necesitan ser probados para comprobar su compleción, consistencia y comportamiento correcto. Se pueden aplicar distintos tipos de comprobaciones para asegurarse de que el juego de prueba automatizado está disponible en un momento dado, o para determinar que es apto para su uso.

Se pueden tomar medidas para verificar el juego de prueba automatizado. Entre ellas se incluyen:

- Comprobar la composición del juego de prueba.
- Verificar las nuevas pruebas que se concentran en las nuevas prestaciones del marco de trabajo de automatización de la prueba (MTAP).
- Tener en cuenta la repetibilidad de las pruebas.
- Tener en cuenta la intrusividad de las herramientas de prueba automatizadas.

A continuación, se explica cada una de ellas con más detalle.

- **Comprobar la composición del juego de prueba**
  - Comprobar la completitud (por ejemplo, que todos los casos de prueba tengan los resultados esperados y que los datos de prueba estén presentes) y la versión correcta del marco de trabajo de automatización de la prueba (MTAP) y del sistema sujeto a prueba (SSP).
- Verificar las nuevas pruebas que se concentran en las nuevas prestaciones del marco de trabajo de automatización de la prueba (MTAP)

- La primera vez que se utiliza una nueva prestación del marco de trabajo de automatización de la prueba (MTAP) en casos de prueba, se debería verificar y monitorizar de cerca para asegurar que la prestación funciona correctamente.
- Tener en cuenta la repetibilidad de las pruebas
  - Al repetir las pruebas, los resultados de las mismas deben ser siempre los mismos. Los casos de prueba del juego de casos de prueba que no den un resultado fiable (por ejemplo, debido a condiciones de secuencia) deben retirarse del juego de casos de prueba automatizado activo y analizarse por separado para encontrar la causa raíz. De lo contrario, se perderá tiempo repetidamente en estas realizaciones de prueba para analizar el fallo.
- Tener en cuenta la intrusividad de las herramientas de prueba automatizadas
  - La solución de automatización de la prueba (SAP) suele estar estrechamente acoplada al sistema sujeto a prueba (SSP). Esto es por diseño para que haya una mejor compatibilidad en lo que respecta al nivel de interacciones. Sin embargo, esta estrecha integración también puede dar lugar a resultados adversos. Por ejemplo, cuando la solución de automatización de la prueba (SAP) se encuentra en el entorno del sistema sujeto a prueba (SSP) la funcionalidad de éste puede diferir de la de las pruebas realizadas manualmente, lo que puede afectar también al rendimiento.  
  
Un alto nivel de intrusión puede manifestar fallos durante las pruebas que no son evidentes en producción. Si esto provoca fallos en las pruebas automatizadas, la confianza en la solución de automatización de la prueba (SAP) puede disminuir drásticamente. Los desarrolladores pueden requerir que los fallos identificados por la automatización de la prueba se reproduzcan manualmente, si es posible, para asistir en el análisis.

### 7.1.3 Identificar dónde la automatización de la prueba produce resultados inesperados

Cuando un guion de prueba falla o pasa de forma inesperada, se debe realizar un análisis de la causa raíz. Esto incluirá la inspección de registros de prueba, datos de rendimiento, configuración y desmontaje del guion de prueba.

También es útil ejecutar algunas pruebas aisladas. Los fallos intermitentes son más difíciles de analizar. El defecto puede estar en el caso de prueba, el sistema sujeto a prueba (SSP), el marco de trabajo de automatización de la prueba (MTAP), el hardware o la red. Monitorizar los recursos del sistema puede dar pistas sobre la causa raíz. El análisis del archivo de registro en prueba del caso de prueba, el sistema sujeto a prueba (SSP) y el marco de trabajo de automatización de la prueba (MTAP) puede ayudar a identificar la causa raíz del defecto. También puede ser necesaria la depuración. Para ayudar a identificar la causa raíz puede ser necesario el apoyo de un analista de pruebas, un analista de negocio, un desarrollador o un ingeniero de sistemas.

Comprobar si se cumplen todas las aseveraciones. Si faltan aseveraciones, los resultados de prueba pueden no ser concluyentes.

### 7.1.4 Explicar cómo el análisis estático puede ayudar a la calidad del código de automatización de pruebas

El análisis estático de código puede ayudar a encontrar vulnerabilidades y defectos en el código del programa. Puede incluir el sistema sujeto a prueba (SSP) o el marco de trabajo de automatización de la prueba (MTAP).

Los análisis automatizados pueden inspeccionar el código para mitigar los riesgos. Esto proporciona una revisión del sistema sujeto a prueba (SSP) en busca de defectos y asegura que se cumplen y aplican los estándares de calidad del código. También se puede tener en cuenta como una técnica proactiva de detección de defectos, y esto tiene un rol importante en las implementaciones de DevSecOps (es decir, DevOps con énfasis en la seguridad). Estas exploraciones se producen al principio del ciclo de vida de desarrollo de software (CVDS) a través de canalizaciones para proporcionar a los equipos de desarrollo una retroalimentación inmediata.

La salida correspondiente a los defectos se suele clasificar como de severidad crítica, alta, media o baja para que los equipos de desarrollo puedan priorizar los defectos que deciden corregir. Algunas herramientas de análisis estático también tienen la capacidad de sugerir correcciones de código para solucionar los defectos que encuentran. Presentarán a los equipos de desarrollo una copia de las líneas de código infractoras y ofrecerán una posible solución para que los desarrolladores la implementen. Además, estas herramientas ayudan a los Ingeniero de Automatización de Pruebas (IAP) midiendo la calidad, sugiriendo áreas en las que comentar el código, mejorando el diseño del código para una gestión optimizada de los recursos (por ejemplo, utilizando bloques try/catch y mejores estructuras de bucle) y eliminando las llamadas a bibliotecas deficientes.

## 8 Mejora continua - (K4)

Duración: 210 minutos

### Palabras Clave<sup>8</sup>

En la siguiente tabla se presentan las palabras clave del capítulo. En este documento, se identifican dos tipos de palabras clave:

- **ISTQB**: identificarán palabras clave del proceso de prueba
- **ESPDOM**: identificarán palabras clave específicas de dominio

Tipo Palabra Clave	Español	Inglés
ISTQB	validación de esquemas	schema validation
ISTQB	histograma de prueba	test histogram

<sup>8</sup> Las palabras clave se encuentran ordenadas por orden alfabético de los términos en inglés.



## Objetivos de Aprendizaje para “Capítulo 8”

### 8.1 Oportunidades de mejora continua para la automatización de la prueba

- TAE - 8.1.1 (K3)** Descubrir oportunidades para mejorar casos de prueba a través de la recopilación y análisis de datos.
- TAE - 8.1.2 (K4)** Analizar los aspectos técnicos de una solución de automatización de la prueba implementada y proporcionar recomendaciones para la mejora.
- TAE - 8.1.3 (K3)** Reestructurar el producto de prueba automatizado para alinearlos con las actualizaciones del sistema sujeto a prueba (SSP)
- TAE - 8.1.4 (K2)** Resumir las oportunidades para el uso de herramientas de automatización de la prueba.

## 8.1 Oportunidades de mejora continua para la automatización de la prueba

### 8.1.1 Descubrir oportunidades para mejorar casos de prueba a través de la recopilación y análisis de datos

La recolección y el análisis de datos pueden mejorarse al tener en cuenta diversos tipos de datos con los enfoques que se describen a continuación.

- **Histograma de pruebas**
  - Un informe visual de los datos de prueba representado como un histograma de prueba proporciona áreas potenciales de mejora con respecto a las tendencias de los datos de casos de prueba. Los Ingeniero de Automatización de Pruebas (IAP) pueden decidir sobre posibles áreas de mejora ya que muchas herramientas de integración continua/entrega continua (CI/CD) y de suministro de información de prueba tienen la capacidad de mostrar diferentes resultados de prueba y sus respectivos datos de prueba (por ejemplo, registros de excepciones, mensajes de error y capturas de pantalla). El histograma de pruebas también permite al Ingeniero de Automatización de Pruebas (IAP) identificar y seleccionar aquellos casos de prueba que son frágiles y refactorizarlos con mejoras adicionales o replanteando la implementación actual.
- **Inteligencia artificial**
  - Otra oportunidad reciente es utilizar inteligencia artificial (IA) para apoyar la prueba y la automatización de la prueba. Por ejemplo, en los casos de prueba de interfaz de usuario, los datos también incluyen valores de localizadores de interfaz de usuario que pueden tratarse como entradas. Recientes herramientas de vanguardia muestran la capacidad de detectar si un determinado localizador se ha modificado con respecto al utilizado. Basándose en AA (“machine learning - ML”) y en el reconocimiento de imágenes, pueden identificar nuevos selectores y utilizar un algoritmo de autorreparación para corregir el caso de prueba e incluir los localizadores modificados en el informe de prueba. Esto puede acelerar los pasos siguientes, como el control de versión y el mantenimiento del código.
- **Validación de esquemas**
  - La validación de esquemas puede aplicarse en el análisis de datos de la interfaz de programación de aplicación (IPA) (por ejemplo, propiedades obtenidas de los puntos finales de destino) y en el análisis de bases de datos (por ejemplo, reglas de validación para campos de software, como tipos de datos permitidos y rangos de valores).

Con la validación de esquemas, la solución de automatización de la prueba (SAP) es capaz de comprobar si una respuesta coincide con la especificación real del negocio. Este tipo de comprobación puede utilizarse para determinar si los elementos de respuesta obligatorios están presentes en la respuesta del servicio y si su tipo de objeto coincide con el definido en el esquema. En caso de que se rompa el esquema, la solución devuelve la validación real que ayuda al Ingeniero de Automatización de Pruebas (IAP) a identificar la causa raíz del problema.

Ejemplo: una API tiene seis elementos de respuesta obligatorios que deben ser cadenas en la respuesta. Con las herramientas de validación de esquemas no es necesario escribir aserciones individuales para comprobar si estos tipos son cadenas y sus valores no son nulos. La validación de esquemas se encargará de estas comprobaciones, haciendo que

el código de automatización de la prueba implementada sea mucho más corto y aumentando la eficiencia de la detección de defectos en el servicio de extremo posterior.

## 8.1.2 Analizar los aspectos técnicos de una solución de automatización de la prueba implementada y proporcionar recomendaciones para la mejora

Además de las tareas de mantenimiento continuas que son necesarias para mantener la SAP sincronizada con el sistema sujeto a prueba (SSP), existen muchas oportunidades para mejorar la solución de automatización de la prueba (SAP). Estas mejoras pueden realizarse para lograr una serie de beneficios, entre los que se incluyen una mayor eficiencia (por ejemplo, reducir aún más la intervención manual), una mayor facilidad de uso, capacidades adicionales y un mejor soporte para las pruebas. La decisión sobre cómo mejorar la solución de automatización de la prueba (SAP) depende de qué prestaciones aportan más valor a un proyecto.

Las áreas específicas de una SAP que pueden tenerse en cuenta para su mejora incluyen la creación de guiones, la ejecución de prueba, la verificación, la arquitectura de automatización de la prueba (AAP), el MTAP, la configuración y el desmontaje, la documentación, las prestaciones de la SAP y las actualizaciones y mejoras de la SAP. A continuación, se describen con más detalle.

- **Guionizado**

- Las técnicas de guionizado varían desde el guionizado lineal hasta el enfoque de prueba guiada por datos, pasando por el enfoque de prueba guiada por palabra clave más sofisticado, como se describe en el apartado 3.1.4. Puede ser conveniente actualizar la actual técnica de guionizado de solución de automatización de la prueba (SAP) para todas las nuevas pruebas automatizadas. La técnica puede reacondicionarse para todas las pruebas automatizadas existentes o, al menos, para las que impliquen un mayor esfuerzo de mantenimiento.

Otra área de mejora de la SAP para guiones de prueba puede concentrarse en su implementación. Por ejemplo:

- Evaluar la superposición de guion de prueba/caso de prueba/pasos de prueba para consolidar las pruebas automatizadas. Los casos de prueba que contienen secuencias de acciones similares no deberían implementar estos pasos de prueba varias veces. Estos pasos de prueba deben convertirse en una función y añadirse a una librería, para que puedan reutilizarse. Estas funciones de librería pueden ser utilizadas por distintos casos de prueba. Esto aumenta la mantenibilidad del producto de prueba. Cuando los pasos de la prueba no son idénticos, pero sí similares, puede ser necesaria la parametrización. Nota: se trata de un enfoque típico de las pruebas guiadas por palabra clave.
- Establecer un proceso de recuperación de fallos para la SAP (“solución de automatización de pruebas - SAS”) y el sistema sujeto a prueba (SSP). Cuando se produce un fallo durante la ejecución de un juego de prueba, la solución de automatización de la prueba (SAP) debería ser capaz de recuperarse de esta condición para continuar con la siguiente prueba posible. Cuando se produce un fallo en el sistema sujeto a prueba (SSP), la SAP necesita llevar a cabo las acciones de recuperación necesarias en el sistema sujeto a prueba (SSP) (por ejemplo, un reinicio del sistema sujeto a prueba (SSP)) cuando sea factible y práctico.
- Evaluar mecanismos de espera para asegurar que se utiliza el mejor tipo. Hay tres mecanismos de espera habituales:
  - **Esperas codificadas de forma rígida** (es decir, esperar un número determinado de milisegundos), que pueden ser la causa raíz de muchos

defectos de automatización de la prueba dada la imprevisibilidad de los tiempos de respuesta del software.

- **Espera dinámica por sondeo** (por ejemplo, comprobar que se ha producido un determinado cambio de estado o acción) es mucho más flexible y eficiente:
  - solución de automatización de la prueba (SAP) sólo espera el tiempo necesario y no se pierde tiempo de prueba.
  - Si el proceso tarda más de lo esperado, el sondeo esperará hasta que la condición se cumpla. Es necesario incluir un mecanismo de tiempo de espera, de lo contrario la prueba puede esperar para siempre si hay un defecto.
- Una forma aún mejor es suscribirse al mecanismo de eventos del sistema sujeto a prueba (SSP). Esto es mucho más fiable que las otras dos opciones, pero el lenguaje de guionizado de prueba necesita soportar la suscripción a eventos y el sistema sujeto a prueba (SSP) necesita ofrecer estos eventos a la SAP. También se requiere un mecanismo de tiempo de espera, o la prueba puede esperar eternamente si hay un defecto.
- **Ejecución de la prueba**
  - Cuando un juego de prueba de regresión automatizado no finaliza debido a que la ejecución de la prueba lleva demasiado tiempo, puede ser necesario probar de forma concurrente en diferentes entornos de prueba si esto es posible. Cuando se utilizan sistemas costosos para las pruebas, puede ser una restricción que todas las pruebas deban realizarse en un único sistema de destino. Puede ser necesario dividir el juego de pruebas de regresión en varias partes, cada una de ellas ejecutada en un periodo de tiempo definido (por ejemplo, en una sola noche). Un análisis más detallado de la cobertura de la automatización de la prueba puede revelar duplicación. La eliminación de la duplicación puede reducir el tiempo de ejecución de prueba y puede producir una mayor eficiencia. En el caso de integración continua/entrega continua (IC/EC), una buena práctica es ejecutar trabajos por lotes en paralelo para optimizar el tiempo de ejecución de prueba. También es una buena práctica establecer un calendario de trabajos por lotes automatizados para ejecutar las diferentes canalizaciones a una hora determinada, por ejemplo, cada mañana, para reducir las interacciones manuales y acelerar el proceso de desarrollo.
- **Verificación**
  - Antes de crear nuevas funciones de verificación, adoptar un conjunto de métodos de verificación estándar para uso de todas las pruebas automatizadas. Esto evitará la reimplementación de acciones de verificación en múltiples pruebas. Cuando los métodos de verificación no sean idénticos, pero sí similares, el uso de la parametrización ayudará a que una función pueda utilizarse en varios tipos de objetos.
- **Arquitectura de automatización de la prueba (AAP)**
  - Puede ser necesario modificar la arquitectura de automatización de la prueba (AAP) para mejorar la capacidad de ser probado del sistema sujeto a prueba (SSP). Estos cambios pueden realizarse en la arquitectura del sistema sujeto a prueba (SSP) y/o en la arquitectura de automatización de la prueba (AAP) del SAP. Esto puede proporcionar una mejora importante en la automatización de la prueba, pero puede requerir cambios significativos e inversiones en el sistema sujeto a prueba (SSP)/SAP. Por ejemplo, si se va a cambiar el sistema sujeto a prueba (SSP) para que proporcione interfaz de programación de aplicación (IPA) para las pruebas, también habrá que refactorizar la SAP en consecuencia. Añadir este tipo de prestaciones más adelante en el ciclo de vida de

desarrollo de software (CVDS) puede resultar bastante caro; es mucho mejor pensar en ello al inicio de la automatización de la prueba y en las primeras fases del ciclo de vida de desarrollo de software (CVDS) del sistema sujeto a prueba (SSP).

- **Marco de trabajo de automatización de la prueba (MTAP)**
  - A menudo, hay nuevas versiones de las librerías base utilizadas en un marco de trabajo de automatización de la prueba (MTAP). A veces se trata de actualizaciones importantes, y la última versión no se puede referenciar inmediatamente en la lista de dependencias del marco de trabajo de automatización de la prueba (MTAP), ya que provocaría una ruptura de las pruebas para muchos de los equipos que las utilizan. Por lo tanto, es mejor realizar primero un piloto y un análisis de impacto. Después, se puede crear un plan de adopción. O bien todos los equipos adoptan la nueva versión de las librerías base al mismo tiempo actualizando la dependencia en el archivo de construcción de la capa de librerías base, o bien cada equipo decide individualmente cuándo actualizarla en su capa de lógica de negocio. Finalmente, una vez que todos los equipos están preparados para aceptar la nueva versión de las librerías base, se pueden actualizar las dependencias en la capa de librerías base (ver sección 3.1.3).
- **Configurar y desmontar**
  - Las acciones y configuraciones que se repiten antes o después de cada guion de prueba o juego de prueba deberían moverse a los métodos de instalación o desmontaje. De este modo, cualquier cambio que afecte al código puede actualizarse en un solo lugar, lo que resulta en una disminución de los esfuerzos de mantenimiento. Por ejemplo, las llamadas a servicios web pueden utilizarse para cumplir precondiciones o postcondiciones para pruebas de interfaz de usuario (por ejemplo, registro de usuarios, limpieza de usuarios y configuración de perfiles).
- **Documentación**
  - Abarca todas las formas de documentación, desde la documentación de automatización de la prueba (por ejemplo, qué hace el código de automatización de la prueba y cómo debe utilizarse), la documentación de usuario para la SAP y los informes de prueba y registros en bitácora de prueba producidos por la solución de automatización de la prueba (SAP).
- **Prestaciones de la solución de automatización de la prueba (SAP)**
  - Inclusión de prestaciones y funciones a la SAP, como informes detallados de las pruebas, registros de pruebas e integración con otros sistemas. Añadir nuevas prestaciones sólo si se van a utilizar. Añadir prestaciones que no se utilicen sólo aumenta la complejidad y disminuye la fiabilidad y mantenibilidad.
- **Actualizaciones y mejoras del marco de trabajo de automatización de la prueba (MTAP)**
  - Al actualizar o pasar a nuevas versiones del marco de trabajo de automatización de la prueba (MTAP), es posible que se disponga de nuevas funciones que puedan ser utilizadas por los casos de prueba o que se corrijan fallos. El riesgo es que la actualización del MTAP mediante la mejora de las herramientas de prueba existentes o la introducción de otras nuevas pueda tener un impacto negativo en los casos de prueba existentes. Probar la última versión de la herramienta de prueba ejecutando pruebas de muestra antes de desplegar la nueva versión de la herramienta de prueba. Las pruebas de muestra deben ser representativas de las pruebas automatizadas de distintos sistemas sujetos a prueba (SSP), distintos tipos de prueba y, en su caso, distintos entornos de prueba.

### 8.1.3 Reestructurar el producto de prueba automatizado para alinearlos con las actualizaciones del sistema sujeto a prueba (SSP)

Tras un determinado conjunto de cambios en un sistema sujeto a prueba (SSP) existente, será necesario actualizar la SAP, incluidas el marco de trabajo de automatización de la prueba (MTAP) y las librerías de componentes. Cualquier cambio, por trivial que sea, puede repercutir negativamente en la fiabilidad y el rendimiento de la solución de automatización de la prueba (SAP).

- **Identificar cambios en los componentes del entorno de prueba**
  - Evaluar qué cambios y mejoras se necesitan. ¿Son necesarios cambios en el producto de prueba, en las librerías de funciones personalizadas o en el sistema operativo? Cada uno de ellos influye en el rendimiento de la solución de automatización de la prueba (SAP). El objetivo general es asegurar que las pruebas automatizadas sigan ejecutándose de manera eficiente. Los cambios deben realizarse de forma incremental, con una mentalidad de producto viable mínimo, de modo que el impacto en la SAP pueda medirse a través de una ejecución limitada de guiones de prueba. Una vez que se encuentre que no existen efectos secundarios, los cambios pueden implementarse por completo. El último paso para verificar que el cambio no ha afectado negativamente a los guiones de prueba automatizados es la realización de una regresión completa. Durante la ejecución de estos guiones de prueba de regresión, pueden encontrarse fallos. La identificación de la causa raíz de estos fallos (por ejemplo, a través de informes de prueba, registros de prueba y análisis de datos de prueba) proporcionará un medio para asegurar que no son el resultado de la actividad de mejora de la automatización de la prueba.
- **Aumentar la eficiencia y efectividad de las librerías de funciones básicas de la solución de automatización de la prueba (SAP)**
  - A medida que una solución de automatización de la prueba (SAP) madura, se descubren nuevas formas de realizar las tareas con mayor eficiencia. Estas nuevas técnicas (por ejemplo, la optimización del código en las funciones y el uso de librerías de sistemas operativos más recientes) necesitan incorporarse a las librerías de funciones básicas que utilizan el proyecto actual y los proyectos futuros.
- **Seleccionar las diferentes funciones que actúan sobre el mismo tipo de control para su consolidación**
  - Gran parte de lo que ocurre durante una ejecución de prueba automatizada es la interrogación de controles en la IGU (“Graphic User Interface - GUI”). Esta interrogación sirve para proporcionar información sobre un control (por ejemplo, visible/no visible, habilitado/no habilitado, tamaño y dimensiones, y datos). Con esta información, una prueba automatizada puede seleccionar un elemento de una lista desplegable, introducir datos en un campo y leer un valor de un campo. Hay varias funciones que pueden actuar sobre los controles para educir esta información. Algunas funciones son muy especializadas, mientras que otras son más generales. Por ejemplo, puede haber una función específica que sólo opere con listas desplegables. También puede haber una función que opere con varias funciones especificando una función como uno de sus parámetros. Por lo tanto, un Ingeniero de Automatización de Pruebas (IAP) puede utilizar varias funciones que pueden consolidarse en menos funciones, consiguiendo los mismos resultados y minimizando el mantenimiento.
- **Refactorizar la arquitectura de automatización de la prueba (AAP) para adaptarla a los cambios en el sistema sujeto a prueba (SSP)**
  - A lo largo del ciclo de vida de una SAP, será necesario realizar cambios para adaptarse a los cambios en el sistema sujeto a prueba (SSP). A medida que el sistema sujeto a prueba (SSP) evoluciona y madura, la arquitectura de automatización de la prueba (AAP)

subyacente tendrá que evolucionar también para asegurar que la capacidad está ahí para apoyar al sistema sujeto a prueba (SSP). Al ampliar las prestaciones, hay que tener cuidado de que no se implementen de forma improvisada, sino que se analicen y modifiquen en la arquitectura de automatización de la prueba (AAP). Esto asegurará que, a medida que las nuevas funcionalidades del sistema sujeto a prueba (SSP) requieran guiones de prueba adicionales, se disponga de componentes compatibles para acomodar estas nuevas pruebas automatizadas.

- **Convenciones de nomenclatura y estandarización**
  - A medida que se introduzcan cambios, las convenciones de nomenclatura para el nuevo código de automatización de la prueba y las librerías de funciones necesitarán ser consistentes con los estándares definidos previamente (véase la sección 4.3.1).
- **Evaluación de los guiones de prueba existentes para la revisión/eliminación del sistema sujeto a prueba (SSP)**
  - El proceso de cambio y mejora también incluye una evaluación de los guiones de prueba existentes, su uso y su valor continuado. Por ejemplo, si determinadas pruebas son complejas y su ejecución requiere mucho tiempo, descomponerlas en pruebas más pequeñas puede resultar más viable y eficiente. Si se eliminan las pruebas que se ejecutan con poca frecuencia o no se ejecutan en absoluto, se reducirá la complejidad de la solución de automatización de la prueba (SAP) y se aclarará lo que se necesita mantener.

#### 8.1.4 Resumir las oportunidades para el uso de herramientas de automatización de la prueba

Aparte de las pruebas propiamente dichas, la automatización de la prueba puede ayudar en actividades de prueba no específicas, tales como:

- **Configuración y control de entorno**
  - Algunos guiones de prueba (por ejemplo, la creación de datos de prueba) pueden aprovecharse en un método de configuración para crear diferentes datos de prueba en un nuevo entorno de prueba. En una situación en la que se necesitan crear usuarios con múltiples perfiles basados en diferentes entradas de datos, un equipo puede utilizar un guion de prueba automatizado para llamar a un punto final de servicio web que registre a estos usuarios. Los guiones de prueba pueden tener control sobre la configuración de la infraestructura de prueba y pueden aprovecharse en una limpieza posterior al proceso. Esto ayuda a ahorrar tiempo y a asegurar que los usuarios adecuados están presentes en cada nuevo entorno de prueba. Por ejemplo, los diferentes registros en bitácora de prueba y otros productos de prueba pueden eliminarse automáticamente de un entorno de prueba, haciendo que la limpieza y el uso del entorno de prueba sean más eficientes.
- **Envejecimiento de datos**
  - La automatización de la prueba puede utilizarse para manipular los datos de prueba en el entorno de prueba. Por ejemplo, en las bases de datos, se pueden comprobar y controlar los campos de fecha para mantenerlos actualizados desde el punto de vista del año.
- **Generación de capturas de pantalla y vídeos**
- La mayoría de las herramientas modernas de automatización de la prueba de interfaz de usuario tienen una capacidad incorporada para crear capturas de pantalla o vídeos en determinadas condiciones y almacenarlos. Con estas herramientas de prueba, los equipos pueden ayudar al negocio a crear capturas de pantalla y vídeos de uso real para la documentación de la entrega del software o con fines de marketing.

BORRADOR



## 9 Referencias

### 9.1 Estándares

Entre los estándares para la automatización de la prueba se incluyen:

The Automatic Test Markup Language (ATML) by IEEE (Institute of Electrical and Electronics Engineers) consisting of

- IEEE Std 1671.1: Test Description
- IEEE Std 1671.2: Instrument Description
- IEEE Std 1671.3: UUT Description
- IEEE Std 1671.4: Test Configuration Description
- IEEE Std 1671.5: Test Adaptor Description
- IEEE Std 1671.6: Test Station Description
- IEEE Std 1641: Signal and Test Definition

IEEE Std 1636.1: Test Results

ISO/IEC/IEEE 29119-5 (2016) Software and systems engineering – Software testing – Part 5: Keyword- Driven Testing

ISO/IEC 30130:2016 (E) Software engineering — Capabilities of software testing tools

The Testing and Test Control Notation (TTCN-3) by ETSI (European Telecommunication Standards Institute) and ITU (International Telecommunication Union) consisting of

- ES 201 873-1: TTCN-3 Core Language
- ES 201 873-2: TTCN-3 Tabular Presentation Format (TFT)
- ES 201 873-3: TTCN-3 Graphical Presentation Format (GFT)
- ES 201 873-4: TTCN-3 Operational Semantics
- ES 201 873-5: TTCN-3 Runtime Interface (TRI)
- ES 201 873-6: TTCN-3 Control Interface (TCI)
- ES 201 873-7: Using ASN.1 with TTCN-3
- ES 201 873-8: Using IDL with TTCN-3
- ES 201 873-9: Using XML with TTCN-3
- ES 201 873-10: TTCN-3 Documentation
- ES 202 781: Extensions: Configuration and Deployment Support
- ES 202 782: Extensions: TTCN-3 Performance and Real-Time Testing
- ES 202 784: Extensions: Advanced Parameterization
- ES 202 785: Extensions: Behaviour Types
- ES 202 786: Extensions: Support of interfaces with continuous signals
- ES 202 789: Extensions: Extended TRI

The UML Testing Profile (UTP) by OMG (Object Management Group) specifying test specification concepts for

- Test Architecture
- Test Data

- Test Behavior
- Test Logging
- Test Management

## 9.2 Documentos de ISTQB®

ISTQB-AL-TTA	ISTQB Certified Tester, Advanced Level Syllabus, Technical Test Analyst, Version 4.0, June 2021, available from [ISTQB-Web]
ISTQB-FL	ISTQB Certified Tester, Foundation Level Syllabus, Version 4.0, April 2023, available from [ISTQB-Web]
ISTQB-MBT	ISTQB Certified Tester, Model-Based Testing Syllabus, Version 1.0, October 2015, available from [ISTQB-Web]
ISTQB-PT	ISTQB Certified Tester, Performance Testing Syllabus, December 2018, available from [ISTQB-Web]
ISTQB-SEC	ISTQB Certified Tester, Security Tester Syllabus, Version 1.0, March 2016, available from [ISTQB-Web]
ISTQB-TAS	ISTQB Certified Tester, Test Automation Strategy Syllabus, February 2024, available from [ISTQB-Web]
ISTQB-Glossary	ISTQB Glossary of Terms, available online from [ISTQB-Web]

## 9.3 Libros

- Paul Baker, Zhen Ru Dai, Jens Grabowski, and Ina Schieferdecker, “Model-Driven Testing: Using the UML Testing Profile”, Springer 2008 edition, ISBN-10: 3540725628, ISBN-13: 978-3540725626
- Efriede Dustin, Thom Garrett, Bernie Gauf, “Implementing Automated Software Testing: how to save time and lower costs while raising quality”, Addison-Wesley, 2009, ISBN 0-321-58051-6
- Efriede Dustin, Jeff Rashka, John Paul, “Automated Software Testing: introduction, management, and performance”, Addison-Wesley, 1999, ISBN-10: 0201432870, ISBN-13: 9780201432879
- Mark Fewster, Dorothy Graham, “Experiences of Test Automation: Case Studies of Software Test Automation”, Addison-Wesley, 2012
- Mark Fewster, Dorothy Graham, “Software Test Automation: Effective use of test execution tools”, ACM Press Books, 1999, ISBN-10: 0201331403, ISBN-13: 9780201331400
- Robert C Martin, “Clean Code: A Handbook of Agile Software Craftsmanship”, 2008, ISBN-10: 9780132350884

- James D. McCaffrey, “.NET Test Automation Recipes: A Problem-Solution Approach”, APRESS, 2006 ISBN-13:978-1-59059-663-3, ISBN-10:1-59059-663-3
- Daniel J. Mosley, Bruce A. Posey, “Just Enough Software Test Automation”, Prentice Hall, 2002, ISBN- 10: 0130084689, ISBN-13: 9780130084682
- Manikandan Sambamurthy, “Test Automation Engineering Handbook”, January 2023, ISBN: 9781804615492
- Colin Willcock, Thomas Deiß, Stephan Tobies and Stefan Keil, “An Introduction to TTCN-3” Wiley, 2nd edition 2011, ISBN-10: 0470663065, ISBN-13: 978-0470663066

## 9.4 Artículos

- Robert V. Binder, Suzanne Miller, “Five Keys to Effective Agile Test Automation for Government Programs” August 24, 2017, Software Engineering Institute, Carnegie Mellon University, [https://resources.sei.cmu.edu/asset\\_files/Webinar/2017\\_018\\_101\\_503516.pdf](https://resources.sei.cmu.edu/asset_files/Webinar/2017_018_101_503516.pdf)
- DoD CIO, Modern Software Practices “DevSecOps Fundamentals Guidebook: Activities & Tools”, Version 2.2, May 2023,  
[https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsActivitesToolsGuidebookTables.pdf?ver=\\_Sylg1WJB9K0Jxb2XTvzDQ%3d%3d](https://dodcio.defense.gov/Portals/0/Documents/Library/DevSecOpsActivitesToolsGuidebookTables.pdf?ver=_Sylg1WJB9K0Jxb2XTvzDQ%3d%3d)
- Péter Földházi, “Tri-Layer Testing Architecture”, <https://www.pnsc.org/docs/PROP53522057-FoldhaziDraftFinal.pdf>
- Thomas Pestak, William Rowell, PhD, “Automated Software Testing Practices and Pitfalls”, September 2018,  
[https://www.afit.edu/stat/statcoe\\_files/Automated%20Software%20Testing%20Practices%20and%20Pitfalls%20Rev%201.pdf](https://www.afit.edu/stat/statcoe_files/Automated%20Software%20Testing%20Practices%20and%20Pitfalls%20Rev%201.pdf)
- Andrew Pollner, Jim Simpson, Jim Wisnowski, “Automated Software Testing Implementation Guide for Managers and Practitioners”, October 2018,  
[https://www.afit.edu/stat/statcoe\\_files/0214simp%20%20AST%20IG%20for%20Managers%20and%20Practitioners.pdf](https://www.afit.edu/stat/statcoe_files/0214simp%20%20AST%20IG%20for%20Managers%20and%20Practitioners.pdf)
- The Selenium Project, “Page object models”,  
[https://www.selenium.dev/documentation/test\\_practices/encouraged/page\\_object\\_models/](https://www.selenium.dev/documentation/test_practices/encouraged/page_object_models/)

## Anexo A - Objetivos de aprendizaje/nivel cognitivo de conocimiento

Los siguientes objetivos de aprendizaje se definen como aplicables a este programa de estudio. Cada tema del programa de estudio se examinará de acuerdo con el objetivo de aprendizaje correspondiente.

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

### Nivel 2: Comprender (K2)

El candidato puede seleccionar las razones o explicaciones de los enunciados relacionados 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 herramientas de prueba en función de su finalidad y de las actividades de prueba a las que dan soporte.	
Comparar los distintos niveles de prueba.	Puede utilizarse para buscar similitudes, diferencias o ambas.
Diferencie la prueba de la depuración.	Busca diferencias entre conceptos.
Distinga entre riesgos de proyecto y de producto.	Permite clasificar por separado dos (o más) conceptos.
Explicar el impacto del contexto en el proceso de prueba.	
Aportar ejemplos de por qué es necesaria la prueba.	
Inferir la causa raíz de defectos a partir de un perfil dado de fallos.	
Resumir las actividades del proceso de revisión del producto de trabajo.	

### Nivel 3: Aplicar (K3)

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

**Verbos de acción:** aplicar, implementar, preparar, utilizar.

Ejemplos	Notas
Aplicar el análisis del valor frontera para obtener casos de prueba a partir de unos requisitos dados.	Debería referirse a un procedimiento / técnica / proceso, etc.
Implementar métodos para la recopilación de métricas que respalden los requisitos técnicos y de gestión.	
Preparar pruebas de instalabilidad para aplicaciones móviles.	

Ejemplos	Notas
Utilizar la trazabilidad para monitorizar el avance de la prueba en cuanto a compleción y consistencia con los objetivos de prueba, la estrategia de prueba y el plan de prueba.	Podría utilizarse en un Objetivo de Aprendizaje que quiera que el candidato sea capaz de utilizar una técnica o procedimiento. Similar a «aplicar».

#### Nivel 4: Analizar (K4)

El candidato puede separar la información relacionada con un procedimiento o técnica en sus partes constituyentes para una mejor comprensión y puede distinguir entre hechos e inferencias. Una aplicación típica es analizar la situación de un documento, software o proyecto y proponer acciones adecuadas para resolver un problema o tarea.

**Verbos de acción:** Analizar, deconstruir, esquematizar, priorizar, seleccionar.

Ejemplos	Notas
Analizar una situación concreta de un proyecto para determinar qué técnicas de prueba de caja negra o basadas en la experiencia deben aplicarse para alcanzar unos objetivos específicos.	Evaluable sólo en combinación con un objetivo medible del análisis. Debe tener la forma 'Analizar xxxx a xxxx' (o similar).
Dar prioridad a los casos de prueba de un determinado juego de prueba para su ejecución basándose en los riesgos de producto relacionados.	
Seleccionar los niveles de prueba y los tipos de prueba adecuados para verificar un conjunto de requisitos dado.	Necesario cuando la selección necesita análisis.

Referencias para los niveles cognitivos de los objetivos de aprendizaje

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



Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12
TAE - 2.1.2	Explicar cómo se utiliza la automatización de la prueba en diferentes entornos	2			X									
2.2	Proceso de evaluación para seleccionar las herramientas y estrategias adecuadas													
TAE - 2.2.1	Analizar un sistema sujeto a prueba para determinar la solución de automatización de la prueba adecuada.	4				X								
TAE - 2.2.2	Aportar ejemplos de hallazgos técnicos de la evaluación de una herramienta.	4				X								
3	Arquitectura de automatización de la prueba													
3.1	Conceptos de diseño utilizados en la automatización de la prueba													
TAE - 3.1.1	Explicar las principales capacidades en una arquitectura de automatización de la prueba.	2					X							
TAE - 3.1.2	Explicar cómo diseñar una solución de automatización de la prueba.	2					X							
TAE - 3.1.3	Aplicar la estructura por capas de los marcos de automatización de pruebas.	3					X							
TAE - 3.1.4	Aplicar diferentes enfoques para automatizar casos de prueba.	3					X							
TAE - 3.1.5	Aplicar principios de diseño y patrones de diseño en la automatización de pruebas.	3					X							
4	Implementación de la automatización de la prueba													

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12
4.1	Desarrollo de la automatización de la prueba													
TAE - 4.1.1	Aplicar directrices que soporten actividades efectivas de piloto y despliegue de automatización de la prueba.	3						X						
4.2	Riesgos asociados al desarrollo de la automatización de la prueba													
TAE - 4.2.1	Analizar los riesgos del despliegue y planificar estrategias de mitigación para la automatización de la prueba.	4							X					
4.3	Mantenibilidad de la solución de automatización de la prueba													
TAE - 4.3.1	Explicar los factores que soportan y afectan la mantenibilidad de la solución de automatización de la prueba.	2								X				
5	Estrategias de implementación y despliegue para la automatización de la prueba													
5.1	Integración en canalizaciones de integración continua/entrega continua (IC/EC)													
TAE - 5.1.1	Aplicar la automatización de la prueba en diferentes niveles de prueba dentro de las canalizaciones.	3									X			
TAE - 5.1.2	Explicar la gestión de la configuración para el producto de prueba.	2									X			
TAE - 5.1.3	Explicar las dependencias de automatización de pruebas para la infraestructura de una interfaz de programación de aplicación (IPA).	2									X			



Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12
6	Suministro de información de la automatización de la prueba y métricas													
6.1	Recopilación, análisis y suministro de información de los datos de automatización de la prueba													
TAE - 6.1.1	Aplicar métodos de recopilación de datos de la solución de automatización de la prueba y del sistema sujeto a prueba.	3										X		
TAE - 6.1.2	Analizar los datos de la solución de automatización de la prueba y del sistema sujeto a prueba para comprender mejor los resultados.	4										X		
TAE - 6.1.3	Explicar cómo se elabora y publica un informe del avance de la prueba.	2										X		
7	Verificación de la solución de automatización de la prueba													
	Verificación de la infraestructura de automatización de la prueba													
TAE - 7.1.1	Planificar la verificación del entorno de automatización de la prueba, incluyendo la configuración de la herramienta de prueba.	3											X	
TAE - 7.1.2	Explicar el comportamiento correcto para un guion de prueba automatizado y/o un juego de prueba dado.	2											X	
TAE - 7.1.3	Identificar dónde la automatización de la prueba produce resultados inesperados	2											X	
TAE - 7.1.4	Explicar cómo el análisis estático puede ayudar a la calidad del código de automatización de pruebas.	2											X	

Capítulo/ sección/ subsección	Objetivos de Aprendizaje	Nivel - K	B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12
8	Mejora continua													
8.1	Oportunidades de mejora continua para la automatización de la prueba													
TAE - 8.1.1	Descubrir oportunidades para mejorar casos de prueba a través de la recopilación y análisis de datos.	3												X
TAE - 8.1.2	Analizar los aspectos técnicos de una solución de automatización de la prueba implementada y proporcionar recomendaciones para la mejora.	4												X
TAE - 8.1.3	Reestructurar el producto de prueba automatizado para alinearlos con las actualizaciones del sistema sujeto a prueba (SSP)	3												X
TAE - 8.1.4	Resumir las oportunidades para el uso de herramientas de automatización de la prueba.	2												X

## **Apéndice C - Notas de la entrega**

“ISTQB® Test Automation Engineering Syllabus 2024” es una importante actualización y reescritura de la entrega 2016. Por esta razón, no hay notas de la publicación detalladas por capítulo y sección. El contenido se ha mejorado significativamente para el rol de ingeniero de automatización de la prueba, mientras que el contenido estratégico se ha trasladado a un nuevo “ISTQB® Test Automation Strategy Syllabus 2024”.

BORRADOR

## Apéndice D - Términos específicos del dominio

Término en español	Término en inglés	Definición
DevSecOps	DevSecOps	Una metodología que combina desarrollo, seguridad y operaciones y que utiliza un alto grado de automatización.
patrón de modelo de flujo	flow model pattern	Una visión de alto nivel del dominio de trabajo, sus componentes y las interconexiones entre ellos.
recorrido de usuario	user journey	Una serie de pasos que muestran, desde la perspectiva de la experiencia de la persona, cómo interactúa un usuario con un sistema sujeto a prueba.
patrón de objeto de página	page object pattern	Un patrón de diseño en la automatización de pruebas para mejorar el mantenimiento de las pruebas y reducir la duplicación de código.
control de versión	version control	Un proceso para comprobar y almacenar versiones específicas del código fuente.