



**Asociación  
Internacional de  
Calidad de  
Software**

---

## Examen de Entrenamiento

---

# Accredited Software Testing Fundamentals Certification (AICS® ASTFC)

---

## Variantes A & B

---

Copyright Notice ©

AICS® es una marca registrada de la Asociación Internacional de Calidad de Software, LLC.  
Todos los derechos reservados.

Se permite el uso de este examen con fines no comerciales, siempre y cuando se cite la fuente.  
Nuestros *Accredited Training Partners* (AICS® ATP) pueden utilizar este material, sujeto al  
cumplimiento de las pautas establecidas por AICS®.

Cualquier otro uso requiere la aprobación escrita de AICS®.

---

# Tabla de Contenido

---

<b>Sobre AICS®</b> .....	<b>2</b>
<b>Propósito del Examen de Entrenamiento</b> .....	<b>3</b>
<b>Examen de Entrenamiento (Variante A)</b> .....	<b>4</b>
<b>Respuestas (Variante A)</b> .....	<b>11</b>
<b>Examen de Entrenamiento (Variante B)</b> .....	<b>15</b>
<b>Respuestas (Variante B)</b> .....	<b>23</b>

---

## Sobre AICS®

---

La Asociación Internacional de Calidad de Software (AICS®) se destaca como un referente líder en el desarrollo de software de calidad. Su enfoque principal radica en respaldar tecnologías, metodologías y buenas prácticas que promueven la excelencia en el desarrollo de software. Ofreciendo programas de certificación especializados, con énfasis en pruebas de software, AICS® trabaja incansablemente para convertirse en líder en estas áreas.

Su compromiso se centra en empoderar a profesionales y empresas en la búsqueda de la excelencia en el desarrollo de software. La AICS® se esfuerza por certificar las habilidades de los profesionales, asegurando que estén debidamente capacitados para crear software de calidad. Con una convicción sólida de que la calidad es prioridad en el ámbito del software, la asociación proporciona las herramientas necesarias para que los profesionales alcancen su máximo potencial y se destaquen en un mercado laboral cada vez más competitivo.

Las certificaciones internacionales ofrecidas por AICS® son reconocidas y respetadas, representando un estándar de excelencia en el desarrollo de software. En resumen, AICS® orienta sus objetivos a marcar el camino hacia el desarrollo de software de calidad y se compromete a impulsar la excelencia en este campo, tanto a nivel profesional como empresarial.

---

# Propósito del Examen de Entrenamiento

---

El examen de entrenamiento con preguntas y respuestas explicadas tiene como propósito principal servir como una herramienta de aprendizaje guiado para los candidatos que se preparan para obtener la certificación internacional AICS® ASTFC (Accredited Software Testing Fundamentals Certification). A diferencia de un examen formal de certificación, este recurso está diseñado con un enfoque pedagógico. Esto significa que no solo presenta preguntas tipo examen, sino que también ofrece explicaciones detalladas para cada respuesta, facilitando así la comprensión de los fundamentos teóricos y prácticos evaluados.

Este entrenamiento permite al candidato familiarizarse con el formato del examen de certificación, reducir la ansiedad y mejorar su preparación estratégica. Además, promueve el aprendizaje autónomo mediante la revisión activa de errores, con explicaciones que aclaran por qué una opción es correcta o incorrecta. También permite evaluar el nivel de preparación actual del candidato e identificar áreas de mejora antes de presentar el examen oficial.

Este examen forma parte del conjunto de materiales de apoyo provistos por AICS®. Su uso está orientado exclusivamente a fines formativos y no sustituye al examen de certificación oficial, sino que actúa como uno de los materiales clave de preparación.

Atentamente,

Lionel Baquero, M.S.

CEO & Founder at AICS®



---

# Examen de Entrenamiento (Variante A)

---

1. **¿Cuál es el principal beneficio de detectar defectos durante las etapas tempranas del desarrollo?**
  - A. Contribuye a reducir las fases del proceso de prueba.
  - B. Reduce levemente el tiempo de desarrollo del producto.
  - C. Disminuye de forma considerable el costo de corrección.
  - D. Aumenta la cantidad de funcionalidades implementadas.
  
2. **La paradoja del pesticida consiste en:**
  - A. Si se ejecutan siempre las mismas pruebas, se dejan de encontrar nuevos defectos.
  - B. Cuando se aplican muchas pruebas, el software se vuelve inmune a los errores.
  - C. Las pruebas automatizadas eliminan completamente la necesidad de pruebas manuales.
  - D. Los defectos detectados en etapas tempranas ya no necesitan corregirse.
  
3. **El hecho de no encontrar problemas o incidentes en el producto de software debido a que siempre se ejecutan las mismas pruebas de la misma forma también se conoce como:**
  - A. Paradoja del pesticida.
  - B. Principio de Pareto.
  - C. Criterios de salidas.
  - D. Cobertura de pruebas.
  
4. **¿Cuál de las siguientes afirmaciones refleja correctamente la relación entre error, defecto y falla?**
  - A. Un error causa una falla que genera un defecto.
  - B. Una falla causa un error que produce un defecto.
  - C. Un error causa un defecto que ocasiona una falla.
  - D. Un defecto causa un error y culmina con una falla.
  
5. **¿Por qué se considera que el testing y el debugging son actividades complementarias?**
  - A. Porque testing verifica resultados y debugging analiza las causas.
  - B. Porque testing detecta las fallas y debugging corrige los errores.
  - C. Porque testing revisa los requisitos y debugging el código fuente.
  - D. Porque testing evalúa la funcionalidad y debugging el código.

- 
- 6. ¿Cuál es el propósito de la verificación en testing?**
- A. Confirmar los requisitos y las especificaciones definidas.
  - B. Validar la satisfacción de las necesidades del usuario final.
  - C. Detectar defectos observados en producción por usuarios.
  - D. Medir desempeño del sistema en escenarios operativos.
- 7. Durante el diseño de pruebas, ¿cuál es una tarea clave que debe realizar el analista?**
- A. Seleccionar la estrategia de prueba.
  - B. Identificar errores de programación.
  - C. Clasificar los riesgos y su impacto.
  - D. Elaborar matrices de trazabilidad.
- 8. El equipo define los datos de prueba necesarios, prepara el ambiente y detalla condiciones en casos específicos listos para su uso. ¿Qué actividad se está realizando?**
- A. Diseño de pruebas.
  - B. Planificación de pruebas.
  - C. Monitoreo de pruebas.
  - D. Cierre de pruebas.
- 9. Una persona afirma: “El testing retrasa la entrega del software”. ¿Cuál es una razón técnica que refuta este mito?**
- A. El testing limita la cantidad de código escrito y elimina los errores.
  - B. Las pruebas manuales eliminan la necesidad de documentación.
  - C. La detección de errores temprano ahorra tiempo y evita reprocesos.
  - D. Las pruebas unitarias son responsabilidad exclusiva de los tester.
- 10. Un equipo ágil comienza el testing desde la definición de criterios de aceptación. ¿Qué mito se evita con esta práctica?**
- A. Que el testing debe empezar cuando el desarrollo finaliza.
  - B. Que la validación depende únicamente del Product Owner.
  - C. Que la calidad se logra solo con más casos de automatización.
  - D. Que los defectos solo se detectan en ambientes de producción.
- 11. En el contexto de testing ágil, ¿por qué se dice que el proceso de feedback sobre el producto es más efectivo?**
- A. Porque las pruebas se realizan en ciclos cortos y continuos.
  - B. Porque se priorizan las pruebas de regresión sobre el resto.
  - C. Porque las pruebas se concentran después de cada release.
  - D. Porque se reduce el número total de casos de prueba.
-

**12. ¿Cuál es el rol del testing en la etapa de mantenimiento del ciclo de vida del software?**

- A. Realizar pruebas exclusivamente para las nuevas funcionalidades.
- B. Ejecutar pruebas de regresión ante cualquier cambio en el sistema.
- C. Hacer seguimiento de los defectos detectados por los usuarios finales.
- D. Detener el testing mientras no se encuentren fallas en producción.

**13. Las pruebas de aceptación tienen como objetivo principal:**

- A. Identificar defectos del código y aspectos técnicos de implementación.
- B. Validar que el sistema cumple con las expectativas del usuario final.
- C. Evaluar la cobertura de código lograda con la ejecución de pruebas.
- D. Asegurar que cada módulo aislado funcione correctamente por sí solo.

**14. Se añade una nueva regla visible para el usuario en la historia “Calcular envío”. ¿Qué nivel debería actualizar primero para reducir escapes a producción?**

- A. Pruebas unitarias sobre las funciones de cálculo internas.
- B. Pruebas de integración de componentes logísticos internos.
- C. Pruebas de sistema de los flujos end-to-end completos.
- D. Pruebas de aceptación frente a los criterios de la historia.

**15. ¿Qué describe mejor una prueba funcional?**

- A. Mide tiempos y uso de recursos en pruebas de estrés.
- B. Evalúa la mantenibilidad del código fuente.
- C. Comprueba la compatibilidad entre navegadores.
- D. Valida comportamientos contra requisitos definidos.

**16. Tras actualizar el driver SQL, aparecen timeouts en consultas específicas. Indica la prueba que mejor aísla el impacto del cambio.**

- A. Funcional: integración del repositorio con base de datos real.
- B. No funcional: resistencia de 24 h con tráfico sintético sostenido.
- C. Funcional: aceptación de pedidos con criterios UAT del negocio.
- D. No funcional: compatibilidad de interfaz en distintos navegadores.

**17. ¿Qué caracteriza a las pruebas de caja negra?**

- A. Evalúan la salida con entradas, sin ver el código.
- B. Analizan rutas internas y estructuras del código.
- C. Verifican estilo, formateo y reglas del código.
- D. Depuran fallos paso a paso en el IDE.

**18. Seleccione el tipo que corresponde a una prueba de caja negra:**

- A. Cobertura de condiciones y ramas del código.
- B. Revisión estática de estilo y convenciones.
- C. Partición de equivalencia sobre reglas de negocio.
- D. Trazado paso a paso de funciones en la herramienta.

**19. Se validan reglas de negocio de un formulario (obligatoriedad, formato, mensajes de error) usando entradas y salidas observables. ¿Cómo clasificas la prueba?**

- A. Caja negra no funcional.
- B. Caja negra funcional.
- C. Caja blanca funcional.
- D. Caja blanca no funcional.

**20. ¿Qué caracteriza a las pruebas de caja blanca?**

- A. Usan entradas y salidas sin ver código.
- B. Analizan estructura y flujo interno del código.
- C. Evalúan diseño visual y experiencia de uso.
- D. Miden tiempos y consumo de recursos.

**21. Se exige que cada rama tenga ambas opciones de resultados cubiertos. ¿Qué métrica usas?**

- A. Cobertura de decisiones.
- B. Cobertura por dispositivos.
- C. Cobertura de escenarios UAT.
- D. Cobertura de navegadores.

**22. Una función con condicionales anidados falla en los bordes. ¿Qué técnica aplicas?**

- A. Partición por clases de equivalencia.
- B. Cobertura de condiciones múltiples.
- C. Ensayo de usabilidad moderado.
- D. Validación de criterios con usuarios.

**23. Tienes que evaluar un producto que no conoces, no hay documentación y el tiempo es limitado; el equipo necesita riesgos iniciales y áreas críticas. Selecciona la opción más adecuada para cumplir este objetivo:**

- A. Iniciar sesiones exploratorias con charters breves.
- B. Esperar la documentación formal antes de probar.
- C. Automatizar flujos eze como primer paso.
- D. Redactar casos completos y seguir el guión.

- 
- 24. ¿Qué técnica de pruebas se basa en el conocimiento de fallos recurrentes para diseñar casos de prueba más efectivos?**
- A. Técnicas fundamentadas en defectos conocidos.
  - B. Exámenes heurísticos con listas de comprobación.
  - C. Pruebas de sesgo humano centradas en preferencias.
  - D. Exámenes exploratorios con notas estructuradas.
- 25. ¿Cuál de las siguientes opciones representa un nivel de prueba?**
- A. Pruebas de seguridad.
  - B. Pruebas de regresión.
  - C. Pruebas de sistema.
  - D. Análisis de valores límite.
- 26. ¿Qué actividad caracteriza la fase de cierre de pruebas?**
- A. Documentar lecciones aprendidas y archivar artefactos.
  - B. Identificar condiciones de prueba para la siguiente etapa.
  - C. Automatizar las pruebas para integración continua.
  - D. Estimar el esfuerzo para las pruebas de aceptación.
- 27. En una auditoría se solicita evidencia de que las pruebas finalizaron cumpliendo los criterios de salida. ¿Qué documento es más adecuado?**
- A. Informe de ejecución con métricas de cobertura y estado de defectos.
  - B. Agenda de reuniones de seguimiento aprobada por el gerente de QA.
  - C. Cronograma del proyecto con fechas de inicio y fin de pruebas.
  - D. Manual de usuario entregado al área de soporte revisado por QA.
- 28. ¿Cuál es el propósito principal del análisis de riesgos en el proceso de pruebas?**
- A. Determinar qué módulos serán descartados por baja probabilidad.
  - B. Predecir el número total de fallas que ocurrirán y mitigarlas.
  - C. Priorizar el esfuerzo de pruebas en las áreas de mayor impacto.
  - D. Automatizar las pruebas más repetitivas para disminuir riesgos.
- 29. ¿Qué acción corresponde principalmente a la coordinación de actividades de testing dentro del equipo?**
- A. Asignar pruebas a los miembros del equipo y resolver bloqueos operativos.
  - B. Diseñar casos de prueba detallados para nuevos requisitos complejos.
  - C. Ejecutar pruebas de regresión automatizadas en cada integración.
  - D. Analizar la causa raíz de defectos críticos en el código y corregirlos.
-

**30. ¿Cuál es el propósito principal de la gestión de configuración en pruebas de software?**

- A. Garantizar que los cambios en código, datos y entornos sean controlados, versionados y rastreables.
- B. Asegurar que cada sprint entregue funcionalidades visibles al cliente, con objetivos y valor medibles.
- C. Validar la cobertura de casos de prueba contra requisitos y criterios de aceptación trazables.
- D. Reducir defectos en ejecución mediante prevención, revisión temprana y seguimiento sistemático.

**31. En el ciclo de vida de un defecto, ¿cuál es el primer estado habitual tras su registro?**

- A. Asignado.
- B. Cerrado.
- C. Abierto.
- D. Rechazado.

**32. ¿Cuál es la utilidad de adjuntar capturas de pantalla o logs al reporte de defectos?**

- A. Reducir la cantidad de pruebas necesarias en regresión.
- B. Dar evidencia objetiva para el análisis del defecto.
- C. Evitar que se registren defectos de baja prioridad.
- D. Automatizar la corrección del error en el código.

**33. La cobertura de código en pruebas automatizadas se refiere a:**

- A. Cantidad de requisitos validados por el usuario final.
- B. Porcentaje de líneas o funciones ejecutadas en pruebas.
- C. Número de casos de prueba ejecutados en un ciclo.
- D. Defectos reabiertos en un siguiente ciclo tras corrección.

**34. ¿Cuál es el propósito principal de QA (Quality Assurance) en el ciclo de vida del software?**

- A. Prevenir defectos mediante procesos y estándares de calidad.
- B. Detectar defectos durante la ejecución de pruebas funcionales.
- C. Identificar defectos post despliegue mediante reportes de usuarios.
- D. Evaluar defectos considerando revisiones técnicas de la arquitectura.

**35. ¿Cuál de las siguientes acciones corresponde a un enfoque correctivo?**

- A. Solventar defectos detectados durante la ejecución de pruebas.
- B. Definir criterios de aceptación antes de iniciar el desarrollo.
- C. Auditar procesos para garantizar cumplimiento de normas ISO.
- D. Establecer métricas de cobertura de requisitos por cada módulo.

**36. ¿Cuál es la función principal de un Test Manager o Líder de Pruebas?**

- A. Ejecutar casos de prueba manuales en cada ciclo de desarrollo.
- B. Planificar, coordinar y supervisar las actividades de pruebas.
- C. Redactar requisitos funcionales para el cliente del proyecto.
- D. Corregir defectos encontrados directamente en el código fuente.

**37. ¿Cuál es la función principal de herramientas de gestión de pruebas?**

- A. Automatizar la ejecución de pruebas unitarias.
- B. Organizar planes, casos y resultados de pruebas.
- C. Monitorear el rendimiento del sistema en test.
- D. Detectar y registrar vulnerabilidades de seguridad.

**38. ¿Cuál es la principal característica de una prueba automatizada?**

- A. Se ejecuta mediante scripts sin intervención humana directa.
- B. Solo puede usarse en pruebas exploratorias de usabilidad.
- C. Depende exclusivamente de la observación del responsable.
- D. No requiere mantenimiento a lo largo del ciclo de vida.

**39. ¿Cuál es la función de un marco de automatización (framework)?**

- A. Organizar scripts, datos de prueba, gestión de objetos y reportes.
- B. Sustituir la necesidad de herramientas de gestión de defectos.
- C. Garantizar que no haya defectos en el marco de producción.
- D. Automatizar pruebas manuales de una manera más eficiente.

**40. ¿Cuándo es recomendable priorizar la automatización en un proyecto?**

- A. Cuando existen funciones estables que cambian poco con el tiempo.
- B. Cuando se requieren únicamente validaciones manuales rápidas.
- C. Cuando las pruebas dependen de criterios definidos por el usuario.
- D. Cuando no existe tiempo para mantener ejecutar las pruebas manuales.

---

# Respuestas (Variante A)

---

- 1. Respuesta correcta: C.** Disminuye de forma considerable el costo de corrección. Arreglar un defecto es mucho más barato cuando se detecta temprano que cuando aparece en pruebas o producción.
- 2. Respuesta correcta: A.** Si se ejecutan siempre las mismas pruebas, se dejan de encontrar nuevos defectos. Repetir las mismas pruebas hace que ya no descubran fallos nuevos, por lo que deben actualizarse periódicamente.
- 3. Respuesta correcta: A.** Paradoja del pesticida. Ocurre cuando repetir siempre las mismas pruebas deja de revelar defectos nuevos.
- 4. Respuesta correcta: C.** Un error causa un defecto que ocasiona una falla. Un error humano introduce un defecto en el software, y cuando ese defecto se ejecuta, provoca una falla.
- 5. Respuesta correcta: B.** Porque testing detecta las fallas y debugging corrige los errores. El testing encuentra las fallas en el comportamiento del software; el debugging analiza y corrige el código que las causa.
- 6. Respuesta correcta: A.** Confirmar los requisitos y las especificaciones definidas. La verificación asegura que el producto se construya correctamente según requisitos, documentos y especificaciones.
- 7. Respuesta correcta: D.** Elaborar matrices de trazabilidad. En el diseño de pruebas, una tarea clave es asegurar que cada caso de prueba trace a un requisito, lo que se logra mediante la matriz de trazabilidad.
- 8. Respuesta correcta: A.** Diseño de pruebas. En el diseño de pruebas se preparan los datos, el ambiente y las condiciones necesarias para ejecutar los casos de prueba.
- 9. Respuesta correcta: C.** La detección de errores temprano ahorra tiempo y evita reprocesos. Encontrar errores antes reduce el retrabajo y evita fallas tardías, lo que acelera la entrega del software en lugar de retrasarla.

- 
- 10. Respuesta correcta: A.** Que el testing debe empezar cuando el desarrollo finaliza.  
Al definir criterios de aceptación desde el inicio, el equipo demuestra que el testing comienza temprano, no al final del desarrollo.
- 11. Respuesta correcta: A.** Porque las pruebas se realizan en ciclos cortos y continuos.  
En métodos ágiles el testing ocurre en iteraciones frecuentes, lo que permite recibir y aplicar feedback rápidamente.
- 12. Respuesta correcta: B.** Ejecutar pruebas de regresión ante cualquier cambio en el sistema.  
En mantenimiento, cualquier corrección o ajuste puede afectar funcionalidades existentes, por lo que el testing se enfoca en regresión para asegurar que nada se rompa.
- 13. Respuesta correcta: B.** Validar que el sistema cumple con las expectativas del usuario final.  
Las pruebas de aceptación buscan confirmar que el producto satisface las necesidades y criterios esperados por el usuario o cliente.
- 14. Respuesta correcta: D.** Pruebas de aceptación frente a los criterios de la historia.  
Como la regla es visible para el usuario y está ligada a una historia (“Calcular envío”), primero deben actualizarse las pruebas de aceptación, alineadas a los criterios de esa historia, para asegurar el comportamiento esperado de negocio en producción.
- 15. Respuesta correcta: D.** Valida comportamientos contra requisitos definidos.  
Las pruebas funcionales se enfocan en verificar que el sistema haga lo que debe hacer según los requisitos.
- 16. Respuesta correcta: A.** Funcional: integración del repositorio con base de datos real.  
El problema surge tras cambiar el driver SQL, por lo que la mejor forma de aislar el impacto es probar la integración directa con la base de datos, donde ocurren los timeouts.
- 17. Respuesta correcta: A.** Evalúan la salida con entradas, sin ver el código.  
En caja negra se prueban funciones desde afuera: se dan entradas y se observan salidas, sin analizar el código interno.
- 18. Respuesta correcta: C.** Partición de equivalencia sobre reglas de negocio.  
La partición de equivalencia es una técnica de caja negra, porque se basa en entradas y salidas sin revisar el código interno.
- 19. Respuesta correcta: B.** Caja negra funcional.  
Se prueban reglas de negocio mediante entradas y salidas visibles, sin revisar código, por lo que es funcional y de caja negra.

- 
- 20. Respuesta correcta: B.** Analizan estructura y flujo interno del código.  
La caja blanca se basa en conocer y probar la lógica interna del código, como rutas, condiciones y estructuras.
- 21. Respuesta correcta: A.** Cobertura de decisiones.  
La cobertura de decisiones exige validar tanto el resultado verdadero como falso de cada rama (if/else).
- 22. Respuesta correcta: B.** Cobertura de condiciones múltiples.  
Con condicionales anidados y fallos en bordes, se necesita cubrir todas las combinaciones de condiciones, lo que se logra con condiciones múltiples.
- 23. Respuesta correcta: A.** Iniciar sesiones exploratorias con charters breves.  
Cuando no hay documentación y el tiempo es poco, la exploración estructurada con charters permite descubrir rápido riesgos, comportamientos críticos y áreas problemáticas.
- 24. Respuesta correcta: A.** Técnicas fundamentadas en defectos conocidos.  
Esta técnica usa fallos recurrentes o patrones de defectos previos para diseñar pruebas que apunten a áreas donde suelen aparecer errores.
- 25. Respuesta correcta: C.** Pruebas de sistema  
Las pruebas de sistema son un nivel de prueba. Las otras opciones son tipos o técnicas de prueba, no niveles.
- 26. Respuesta correcta: A.** Documentar lecciones aprendidas y archivar artefactos.  
En el cierre de pruebas se recopilan resultados, se documentan aprendizajes y se archivan los artefactos del ciclo de prueba.
- 27. Respuesta correcta: A.** Informe de ejecución con métricas de cobertura y estado de defectos.  
El informe de ejecución demuestra que se cumplieron los criterios de salida, mostrando métricas, cobertura y estado final de los defectos.
- 28. Respuesta correcta: C.** Priorizar el esfuerzo de pruebas en las áreas de mayor impacto.  
El análisis de riesgos sirve para enfocar el testing donde el impacto y la probabilidad de fallas son mayores, optimizando el esfuerzo y reduciendo escapes críticos.
- 29. Respuesta correcta: A.** Asignar pruebas a los miembros del equipo y resolver bloqueos operativos.  
La coordinación del testing implica organizar el trabajo, asignar tareas y destrabar impedimentos, no diseñar, ejecutar ni corregir código.
-

**30. Respuesta correcta: A.** Garantizar que los cambios en código, datos y entornos sean controlados, versionados y rastreables.

La gestión de configuración asegura control y trazabilidad sobre versiones de código, datos, entornos y artefactos de prueba para evitar inconsistencias.

**31. Respuesta correcta: C.** Abierto.

Tras registrar un defecto, su primer estado habitual es Abierto, indicando que fue creado y espera análisis o asignación.

**32. Respuesta correcta: B.** Dar evidencia objetiva para el análisis del defecto.

Capturas y logs brindan evidencia clara que facilita reproducir, entender y corregir el defecto.

**33. Respuesta correcta: B.** Porcentaje de líneas o funciones ejecutadas en pruebas.

La cobertura de código mide qué parte del código se ejecuta durante las pruebas automatizadas.

**34. Respuesta correcta: A.** Prevenir defectos mediante procesos y estándares de calidad.

QA se enfoca en prevenir defectos asegurando que existan procesos, prácticas y estándares de calidad antes y durante el desarrollo.

**35. Respuesta correcta: A.** Solventar defectos detectados durante la ejecución de pruebas.

Un enfoque correctivo actúa después de que ocurre un problema, corrigiendo defectos ya detectados.

**36. Respuesta correcta: B.** Planificar, coordinar y supervisar las actividades de pruebas.

El Test Manager se encarga de organizar y dirigir todo el proceso de pruebas, no de ejecutar pruebas ni corregir código.

**37. Respuesta correcta: B.** Organizar planes, casos y resultados de pruebas.

Las herramientas de gestión permiten centralizar y controlar planes, casos, ejecuciones y resultados de pruebas.

**38. Respuesta correcta: A.** Se ejecuta mediante scripts sin intervención humana directa.

La automatización permite que las pruebas se ejecuten solas, usando scripts sin interacción manual durante la ejecución.

**39. Respuesta correcta: A.** Organizar scripts, datos de prueba, gestión de objetos y reportes.

Un framework da estructura y organización a la automatización, facilitando mantenimiento, reutilización y estandarización.

**40. Respuesta correcta: A.** Cuando existen funciones estables que cambian poco con el tiempo.

La automatización es más eficiente cuando la funcionalidad es estable, porque reduce retrabajo y mantenimiento.

---

# Examen de Entrenamiento (Variante B)

---

1. **¿Por qué es relevante incluir actividades de testing desde el inicio del ciclo de desarrollo?**
  - A. Para validar los requisitos funcionales y no funcionales.
  - B. Para asegurar la calidad del producto en todo el proceso.
  - C. Para cumplir con la correcta documentación del proyecto.
  - D. Para desarrollar la cantidad de funcionalidades justas.
  
2. **La calidad de un sistema se puede asegurar y garantizar por tiempo indeterminado.**
  - A. Verdadero
  - B. Falso
  
3. **Se suele afirmar que la ejecución de pruebas exhaustivas obteniendo una cobertura del 100%:**
  - A. Es imposible en la práctica por limitaciones de tiempo y recursos.
  - B. Garantiza que el software esté completamente libre de defectos.
  - C. Es viable únicamente en proyectos pequeños con pocos requisitos.
  - D. Sustituye la necesidad de aplicar técnicas de diseño de pruebas.
  
4. **¿Qué ocurre cuando al ejecutar un código se produce un comportamiento inesperado en el sistema?**
  - A. Error
  - B. Falla
  - C. Anomalía
  - D. Defecto
  
5. **Un equipo observa que, tras ejecutar pruebas, se genera un reporte con varios fallos funcionales. Los desarrolladores analizan el código, identifican la causa y aplican una solución. ¿Qué conclusión describe mejor la relación entre testing y debugging?**
  - A. El testing detecta los problemas y el debugging los soluciona.
  - B. El testing corrige defectos y el debugging valida resultados.
  - C. El testing se aplica en producción y el debugging en desarrollo
  - D. El testing y el debugging son sinónimos de verificación de errores.

- 
- 6. Usuarios realizan pruebas de aceptación al sistema para confirmar utilidad real. ¿Qué actividad corresponde a esta situación?**
- A. Verificación, por revisar trazabilidad de requisitos y pruebas.
  - B. Validación, por confirmar satisfacción de necesidades de usuarios.
  - C. Verificación, por analizar el cumplimiento de normas del proceso.
  - D. Validación, por ejecutar pruebas controladas en laboratorio técnico.
- 7. ¿Qué elemento debe estar presente en la documentación de un caso de prueba para facilitar su ejecución?**
- A. Objetivo de negocio del proyecto.
  - B. Registro de versiones del producto.
  - C. Pasos detallados para la ejecución.
  - D. Historial de cambios del equipo.
- 8. Durante una revisión de requisitos, un tester identifica ambigüedades y propone criterios de aceptación más claros antes de diseñar pruebas. ¿Qué tarea está realizando principalmente?**
- A. Planificación y análisis de pruebas.
  - B. Ejecución de pruebas de regresión.
  - C. Monitoreo y control de pruebas.
  - D. Cierre del ciclo de pruebas.
- 9. ¿Cuál es un error común sobre el rol de los testers en la calidad del producto?**
- A. Asumir que los testers son los únicos responsables de la calidad.
  - B. Considerar que los testers contribuyen a la detección de defectos.
  - C. Pensar que los testers aportan información sobre el estado del producto.
  - D. Entender que los testers colaboran con desarrollo y negocio.
- 10. Un tester observa que los defectos frecuentes provienen de código deficiente y requisitos ambiguos. ¿Qué debería hacer comprender a su equipo?**
- A. La calidad es responsabilidad compartida entre todos los miembros.
  - B. La calidad consiste en automatizar solamente los escenarios críticos.
  - C. La calidad se logra con un buen analista y mejores desarrolladores.
  - D. La calidad mejora cuando se reducen los tiempos de entrega.

**11. En el Modelo V, ¿qué expresa la relación entre ambos lados de la “V”?**

- A. Las pruebas se diseñan al final y se ejecutan sin relación directa con las fases anteriores.
- B. Las pruebas unitarias sustituyen integración, sistema y aceptación en todo el ciclo.
- C. La validación se realiza solo con exploración, sin usar requisitos ni artefactos de diseño.
- D. Cada fase de desarrollo se empareja con una prueba que valida sus resultados previstos.

**12. En un modelo de desarrollo ágil, ¿cuál es la forma más común de integrar el testing durante todo el ciclo de vida del software?**

- A. Ejecutar todas las pruebas al finalizar cada sprint.
- B. Incluir pruebas en las historias de usuario prioritarias.
- C. Aplicar testing continuo e involucrar a todo el equipo.
- D. Delegar la validación final al PO antes de la entrega.

**13. ¿Cuál de las siguientes opciones representa correctamente el orden secuencial típico de los niveles de prueba?**

- A. Aceptación » Sistema » Integración » Unidad
- B. Unidad » Integración » Sistema » Aceptación
- C. Integración » Sistema » Unidad » Aceptación
- D. Sistema » Aceptación » Unidad » Integración

**14. Un flujo de pagos falla cuando el servicio Checkout llama a Pagos; en pruebas locales cada servicio funciona. ¿Qué nivel de prueba priorizas para detectar el problema antes?**

- A. Pruebas unitarias en cada función del servicio Checkout.
- B. Pruebas de integración entre Checkout y Pagos.
- C. Pruebas de sistema sobre todo el producto en entorno QA.
- D. Pruebas de aceptación con usuarios y casos reales.

**15. ¿Qué describe mejor una prueba no funcional?**

- A. Verifica reglas de negocio y flujos.
- B. Evalúa rendimiento, seguridad y usabilidad.
- C. Repite casos unitarios con datos sintéticos.
- D. Confirma criterios de aceptación con usuarios.

- 
- 16. Se ajusta la regla de cálculo de impuestos con topes y redondeos. Elige la prueba que primero reduce escapes a producción.**
- A. Funcional: casos de borde y reglas de negocio del cálculo.
  - B. No funcional: estrés del servicio con picos de tráfico.
  - C. Funcional: UAT de flujos completos con usuarios clave.
  - D. No funcional: compatibilidad entre navegadores y dispositivos.
- 17. Sin acceso al código, se prueban entradas límite y reglas visibles de una API. ¿Qué errores es más probable identificar con caja negra?**
- A. Desajustes entre requisitos y comportamiento.
  - B. Defectos en la complejidad ciclomática del módulo.
  - C. Violaciones de formateo y errores de sintaxis.
  - D. Errores de acoplamiento entre clases internas.
- 18. Las pruebas funcionales pueden diseñarse como caja negra o como caja blanca, según acceso e información disponible.**
- A. Verdadero
  - B. Falso
- 19. Se comprueba la compatibilidad de la interfaz en Chrome, Firefox y Safari móviles, verificando comportamiento consistente. ¿Cómo clasificas la prueba?**
- A. Caja negra funcional.
  - B. Caja blanca funcional.
  - C. Caja blanca no funcional.
  - D. Caja negra no funcional.
- 20. ¿Qué técnica es propia de las pruebas de caja blanca?**
- A. Partición de equivalencia por requisitos.
  - B. Cobertura de decisiones y ramas del código.
  - C. Ensayos de compatibilidad entre navegadores.
  - D. Pruebas de usabilidad con usuarios reales.
- 21. Una función con retornos tempranos debe cubrir todos los caminos. ¿Qué técnica aplicas primero?**
- A. Partición de equivalencia.
  - B. Pruebas A/B de interfaz.
  - C. UAT con usuarios finales.
  - D. Camino básico.
-

- 
- 22. Tras refactorizar un módulo, quieres evitar zonas no ejercitadas. ¿Qué harías primero?**
- A. Medir y elevar cobertura de sentencias y ramas.
  - B. Ejecutar pruebas exploratorias con los usuarios.
  - C. Repetir pruebas de compatibilidad de UI.
  - D. Hacer pruebas de estrés de 24 horas continuas.
- 23. Debes investigar comportamientos no registrados en un producto poco conocido y con poca documentación. ¿Qué enfoque usas primero?**
- A. Exámenes exploratorios con charters breves y notas estructuradas.
  - B. Pruebas de sesgo humano centradas en preferencias del usuario.
  - C. Exámenes heurísticos basados solo en listas de comprobación.
  - D. Pruebas de ataques enfocadas en explotación inmediata.
- 24. Se necesita examinar la robustez ante condiciones excepcionales y hallar puntos potencialmente peligrosos. ¿Qué técnica priorizas?**
- A. Exámenes exploratorios orientados a rutas críticas del sistema.
  - B. Exámenes heurísticos centrados en estética y presentación.
  - C. Pruebas de sesgo humano para medir preferencias personales.
  - D. Pruebas de ataques con escenarios de abuso y entradas maliciosas.
- 25. ¿Cuál de los siguientes enunciados es correcto sobre las técnicas de diseño de pruebas?**
- A. Indican la fase del ciclo para ejecutar las pruebas del sistema correctamente.
  - B. Priorizan los tipos de prueba según impacto y riesgo del sistema total.
  - C. Generan casos de prueba a partir de condiciones del sistema bien definidas.
  - D. Determinan si los usuarios finales aceptan la solución propuesta en producción.
- 26. Durante la fase de implementación de pruebas, ¿qué tarea debe realizarse?**
- A. Reportar defectos encontrados en producción.
  - B. Ejecutar revisiones estáticas sobre código fuente.
  - C. Preparar datos, entorno y secuencias de ejecución.
  - D. Validar la satisfacción del cliente con el sistema.
- 27. Durante la planificación de pruebas de un sistema de reservas, el equipo debe priorizar los escenarios más críticos. ¿Qué criterio otorga mejor solución a la situación planteada?**
- A. Anteponer los casos que sean más fáciles de automatizar.
  - B. Priorizar escenarios con alto impacto en el usuario o alto riesgo.
  - C. Seleccionar los flujos que tengan más pasos de ejecución.
  - D. Ordenar las pruebas según el orden de importancia de los módulos.
-

- 
- 28. ¿Cuál de los siguientes sería un ejemplo de riesgo del proyecto en el contexto de testing?**
- A. Un módulo crítico presenta errores en validaciones lógicas.
  - B. El rendimiento del sistema bajo carga es deficiente.
  - C. El equipo de testing tiene una alta rotación de personal.
  - D. El login del sistema no cumple con los requisitos de seguridad.
- 29. ¿Cuál de los siguientes aspectos es clave para supervisar la calidad del trabajo de testing en un equipo?**
- A. Evaluar solo el número de pruebas ejecutadas.
  - B. Medir defectos encontrados y cobertura de pruebas.
  - C. Calcular el tiempo invertido en cada caso de prueba.
  - D. Contar con mayor cantidad de pruebas automatizadas.
- 30. ¿Qué elemento se gestiona habitualmente bajo la gestión de configuración en pruebas?**
- A. Historias de usuario documentadas en backlog.
  - B. Casos de pruebas, datos de prueba y versiones.
  - C. Registros de métricas de productividad del equipo.
  - D. Resultados de encuestas de satisfacción del cliente.
- 31. ¿Cuál es el propósito principal de un registro en un sistema de gestión de defectos?**
- A. Documentar el error para permitir su posterior análisis.
  - B. Eliminar automáticamente el defecto del código fuente.
  - C. Reducir el número de pruebas necesarias en el futuro.
  - D. Llevar control de las métricas de defectos encontrados.
- 32. ¿Cuál es un campo básico que debe estar presente al reportar un defecto para garantizar su solución?**
- A. Pasos para reproducir el defecto.
  - B. Opinión del tester sobre el defecto.
  - C. Nombre o identificador del tester.
  - D. Resumen de la funcionalidad.
- 33. ¿Qué refleja la métrica tasa de éxito de casos de prueba?**
- A. Porcentaje de casos ejecutados respecto al plan previsto.
  - B. Proporción de pruebas aprobadas sobre el total ejecutado.
  - C. Número de defectos cerrados durante el ciclo de pruebas.
  - D. Severidad promedio de los defectos detectados en el ciclo.

**34. ¿Qué describe mejor el alcance de QC (Quality Control)?**

- A. Supervisar que los procesos de desarrollo cumplan estándares de calidad.
- B. Documentar los requisitos de negocio entregados por los interesados.
- C. Medir la productividad del equipo de pruebas en proyectos anteriores.
- D. Detectar fallos durante la ejecución de pruebas funcionales del sistema.

**35. ¿Qué caracteriza a un enfoque preventivo en gestión de calidad?**

- A. Consiste en corregir defectos encontrados tras las pruebas ejecutadas.
- B. Evalúa el número de incidentes reportados en producción operativa.
- C. Garantiza la trazabilidad de requisitos y pruebas documentadas formalmente.
- D. Busca evitar defectos mediante procesos, estándares y buenas prácticas.

**36. ¿Cuál es la función del Usuario o Cliente dentro del proceso de pruebas?**

- A. Configurar los entornos de ejecución para la ejecución de las pruebas planificadas.
- B. Revisar y validar que el software cumpla sus necesidades mediante pruebas de aceptación.
- C. Redactar los planes de prueba del sistema con los objetivos y alcance definidos.
- D. Reportar defectos directamente en el código fuente sin pasar por la herramienta de gestión.

**37. ¿Qué permiten las herramientas de gestión de defectos?**

- A. Identificar defectos del código no cubiertos por pruebas.
- B. Registrar, asignar y dar seguimiento a defectos.
- C. Automatizar la generación de scripts de pruebas.
- D. Validar las soluciones de los defectos encontrados.

**38. ¿Cuándo resulta más conveniente aplicar pruebas manuales en lugar de automatizadas?**

- A. En validaciones exploratorias donde importa la percepción humana.
- B. En ejecuciones repetitivas de regresión sobre grandes volúmenes de datos.
- C. En pruebas de carga y estrés con múltiples usuarios virtuales.
- D. En integración continua con despliegues automáticos en distintos entornos.

**39. ¿Cuál es el propósito de comenzar con un programa de prueba en automatización?**

- A. Automatizar todas las funciones de manera inmediata.
- B. Probar en un área pequeña y confiable antes de expandirse.
- C. Reducir la necesidad de diseñar casos de prueba manuales.
- D. Evitar la integración de pruebas con el proceso de desarrollo.

**40. ¿Qué tipo de casos de prueba son los más recomendados para automatizar?**

- A. Pruebas de regresión y funcionalidades poco variables.
- B. Pruebas exploratorias y validaciones de usabilidad.
- C. Escenarios que requieren interacción visual compleja.
- D. Validaciones manuales realizadas por los usuarios finales.

---

# Respuestas (Variante B)

---

- 1. Respuesta correcta: B.** Para asegurar la calidad del producto en todo el proceso. Iniciar el testing desde el principio permite prevenir errores y controlar la calidad en cada etapa del desarrollo.
- 2. Respuesta correcta: B.** Falso  
La calidad no se puede garantizar indefinidamente porque el sistema cambia, el contexto evoluciona y pueden surgir nuevos riesgos o defectos.
- 3. Respuesta correcta: A.** Es imposible en la práctica por limitaciones de tiempo y recursos. Probar absolutamente todo es inviable porque implicaría combinaciones infinitas de datos, flujos y condiciones. Por eso no se alcanza el 100% real.
- 4. Respuesta correcta: B.** Falla  
Cuando el sistema se comporta de manera diferente a lo esperado durante la ejecución, se produce una falla.
- 5. Respuesta correcta: A.** El testing detecta los problemas y el debugging los soluciona. El testing encuentra fallas en el comportamiento del sistema, mientras que el debugging analiza y corrige la causa en el código.
- 6. Respuesta correcta: B.** Validación, por confirmar satisfacción de necesidades de usuarios. Las pruebas de aceptación realizadas por usuarios verifican si el sistema satisface sus necesidades reales, lo que corresponde a validación.
- 7. Respuesta correcta: C.** Pasos detallados para la ejecución.  
Los pasos de ejecución permiten que cualquier persona pueda reproducir la prueba de forma clara y consistente.
- 8. Respuesta correcta: A.** Planificación y análisis de pruebas.  
Revisar requisitos, detectar ambigüedades y proponer criterios claros forma parte del análisis y planificación, antes de diseñar o ejecutar pruebas.
- 9. Respuesta correcta: A.** Asumir que los testers son los únicos responsables de la calidad.  
La calidad es responsabilidad de todo el equipo, no solo de los testers. Ellos aportan información, pero no pueden garantizarla por sí solos.

**10. Respuesta correcta: A.** La calidad es responsabilidad compartida entre todos los miembros.

Los defectos surgen de varias fuentes (requisitos, diseño, código), por lo que la calidad depende del trabajo conjunto de todo el equipo.

**11. Respuesta correcta: D.** Cada fase de desarrollo se empareja con una prueba que valida sus resultados previstos.

En el Modelo V, cada etapa del desarrollo tiene una actividad de prueba asociada que valida lo producido en esa fase.

**12. Respuesta correcta: C.** Aplicar testing continuo e involucrar a todo el equipo.

En entornos ágiles el testing se integra desde el inicio y durante todo el ciclo mediante testing continuo, con participación de todo el equipo, no solo testers.

**13. Respuesta correcta: B.** Unidad » Integración » Sistema » Aceptación

El orden estándar va desde probar las partes más pequeñas (unidad), luego cómo interactúan (integración), después el sistema completo, y finalmente la aceptación por parte del usuario.

**14. Respuesta correcta: B.** Pruebas de integración entre Checkout y Pagos.

El error aparece cuando los servicios interactúan, por lo que deben priorizarse las pruebas de integración para detectar fallas en la comunicación entre componentes.

**15. Respuesta correcta: B.** Evalúa rendimiento, seguridad y usabilidad.

Las pruebas no funcionales miden cómo se comporta el sistema (rendimiento, seguridad, usabilidad), no qué hace.

**16. Respuesta correcta: A.** Funcional: casos de borde y reglas de negocio del cálculo.

Cuando cambia una regla de cálculo, los errores más probables aparecen en bordes, límites y redondeos, por lo que las pruebas funcionales del cálculo son las primeras que reducen escapes a producción.

**17. Respuesta correcta: A.** Desajustes entre requisitos y comportamiento.

Con caja negra se validan entradas y salidas sin ver el código, por lo que es más probable detectar comportamientos que no cumplen los requisitos.

**18. Respuesta correcta: A.** Verdadero

Las pruebas funcionales validan qué hace el sistema, pero pueden diseñarse como caja negra (sin ver código) o caja blanca (si se revisa la lógica interna).

**19. Respuesta correcta: D.** Caja negra no funcional.

La compatibilidad entre navegadores evalúa cómo se comporta la interfaz en distintos entornos, lo que es no funcional, y se hace sin ver código, por lo que es caja negra.

**20. Respuesta correcta: B.** Cobertura de decisiones y ramas del código.

La caja blanca se basa en analizar el código interno, por lo que técnicas como la cobertura de decisiones y ramas son propias de este enfoque.

**21. Respuesta correcta: D.** Camino básico.

La técnica de camino básico permite identificar y cubrir todos los caminos independientes de una función, incluyendo retornos tempranos.

**22. Respuesta correcta: A.** Medir y elevar cobertura de sentencias y ramas.

Tras una refactorización, lo primero es verificar que el código nuevo o modificado esté realmente ejercitado, elevando la cobertura de sentencias y ramas para evitar zonas sin probar.

**23. Respuesta correcta: A.** Exámenes exploratorios con charters breves y notas estructuradas.

Cuando el producto es poco conocido y no hay documentación, la exploración guiada con charters permite descubrir rápidamente comportamientos, riesgos y áreas críticas.

**24. Respuesta correcta: D.** Pruebas de ataques con escenarios de abuso y entradas maliciosas.

Para evaluar robustez ante condiciones extremas y detectar puntos peligrosos, se priorizan pruebas de ataque, usando entradas maliciosas o abusivas que fuerzan el sistema al límite.

**25. Respuesta correcta: C.** Generan casos de prueba a partir de condiciones del sistema bien definidas.

Las técnicas de diseño permiten crear casos de prueba estructurados basados en reglas, requisitos o condiciones específicas del sistema.

**26. Respuesta correcta: C.** Preparar datos, entorno y secuencias de ejecución.

En la implementación de pruebas se alistan los datos, el ambiente y las condiciones necesarias para poder ejecutar los casos de prueba.

**27. Respuesta correcta: B.** Priorizar escenarios con alto impacto en el usuario o alto riesgo.

En planificación, se priorizan primero los escenarios que implican mayor riesgo o impacto para asegurar que las áreas más críticas se validen antes.

**28. Respuesta correcta: C.** El equipo de testing tiene una alta rotación de personal.

Un riesgo del proyecto es algo externo al producto que afecta el proceso (como la rotación del equipo). Los demás son riesgos del producto.

**29. Respuesta correcta: B.** Medir defectos encontrados y cobertura de pruebas.

La supervisión de la calidad del testing requiere métricas relevantes, como defectos detectados y cobertura, no solo cantidad de pruebas o tiempo invertido.

**30. Respuesta correcta: B.** Casos de pruebas, datos de prueba y versiones.

La gestión de configuración controla y versiona artefactos de prueba, como casos, datos y configuraciones, para asegurar consistencia entre ejecuciones.

**31. Respuesta correcta: A.** Documentar el error para permitir su posterior análisis.

El registro de defectos existe para documentar claramente el problema, facilitando su análisis, reproducción y corrección posterior.

**32. Respuesta correcta: A.** Pasos para reproducir el defecto.

Sin pasos claros para reproducir el problema, el equipo no puede analizarlo ni corregirlo adecuadamente.

**33. Respuesta correcta: B.** Proporción de pruebas aprobadas sobre el total ejecutado.

La tasa de éxito indica qué porcentaje de casos ejecutados resultaron aprobados, mostrando estabilidad del producto.

**34. Respuesta correcta: D.** Detectar fallos durante la ejecución de pruebas funcionales del sistema.

QC se enfoca en identificar defectos en el producto mediante actividades de prueba, verificando que el software funcione correctamente.

**35. Respuesta correcta: D.** Busca evitar defectos mediante procesos, estándares y buenas prácticas.

El enfoque preventivo se centra en evitar que los defectos ocurran, aplicando prácticas y procesos de calidad desde el inicio.

**36. Respuesta correcta: B.** Revisar y validar que el software cumpla sus necesidades mediante pruebas de aceptación.

El usuario o cliente participa validando que el producto cumpla sus necesidades reales, generalmente a través de pruebas de aceptación.

**37. Respuesta correcta: B.** Registrar, asignar y dar seguimiento a defectos.

Estas herramientas permiten documentar, gestionar y monitorear cada defecto durante todo su ciclo de vida.

**38. Respuesta correcta: A.** En validaciones exploratorias donde importa la percepción humana.

La prueba manual es ideal cuando se necesita criterio humano, exploración, percepción visual o análisis contextual que no puede automatizarse.

**39. Respuesta correcta: B.** Probar en un área pequeña y confiable antes de expandirse.

Se inicia en un área acotada y estable para validar el enfoque, evitar desperdicio y asegurar una base sólida antes de ampliar la automatización.

**40. Respuesta correcta: A.** Pruebas de regresión y funcionalidades poco variables.

La automatización es más efectiva en escenarios repetitivos y estables, como regresiones, donde el mantenimiento es mínimo y el beneficio es mayor.