



byond^{it} WEBINAR

DESCUBRE LAS BUENAS
PRÁCTICAS EN LAS PRUEBAS
DE SOFTWARE

Introducción y Buenas Prácticas en la Prueba de Software

Principios, tipos, herramientas y tendencias



Introducción





Hello Word!

Gypsissi Vizcaya

Ingeniera de Sistemas



Ingeniera de sistemas con experiencia en Desarrollo de software, soporte, gestión y procesos, con certificaciones en QA, agilidad y maestrías en Gestión de Proyectos y Alta Gerencia.

QA desde el 2016 y líder técnico, con el interés de darle a las pruebas la posición que se merecen en el ciclo de desarrollo de software.



Agenda

- ✓ Qué es el software testing
- ✓ Qué es QA
- ✓ Quién es un QA
- ✓ Qué cosas se deben atender en cada prueba
- ✓ Características de un QA
- ✓ Beneficios de las pruebas
- ✓ Responsabilidades legales
- ✓ Tipos de pruebas
- ✓ Pruebas Especializadas
- ✓ Ciclo de Vida de las pruebas
- ✓ Pruebas en entornos ágiles
- ✓ Cuáles son los retos más comunes
- ✓ Tendencias
- ✓ 3 Casos de famosos de error por falta de pruebas
- ✓ Caso exitoso
- ✓ Quiénes Certifican
- ✓ Preguntas



Qué es **El software testing?**

El testing como tal es un proceso de verificar y validar que el software cumpla con los requisitos, es quien asegura calidad, reduce costos y mejora la experiencia del usuario

el testing está en todas partes, no lo vemos, pero de alguna forma sabemos que todo paso por una prueba.



“

Desarrolla el éxito a partir de los fracasos

”

Dale Carnegie (escritor)

¿Qué es QA

Quality Assurance?

Tester:

Su función principal es realizar pruebas para identificar errores, fallas o cualquier comportamiento inesperado en el software.

Siguen los casos de prueba y buscan errores específicos, realizando pruebas manuales o automatizadas en función de escenarios ya definidos. Se enfocan en detectar fallas en el software antes de que llegue al usuario final.

Quality Assurance (QA):

También involucra pruebas, pero su enfoque es más amplio, centrándose en asegurar que los procesos de desarrollo de software sigan buenas prácticas para prevenir errores desde el principio.

Diseña e implementa procedimientos de calidad, como estrategias, métricas y estándares, revisión de requisitos de calidad a nivel de procesos y no solo de funcionamiento.

¿Quién es un QA? ∞

Es alguien que prueba y nos comparte el feedback, su objetivo principal es que el software cumpla con lo solicitado (requisitos) y aporta ideas de que podamos mejorar y ofrecer la mejor experiencia a los usuarios futuros. Interactúa con la herramienta y nos da su opinión sobre:

- ✓ Funcionamiento
- ✓ Diseño
- ✓ Usabilidad



¿Qué cosas debe atender en cada prueba?

¿Quién lo hace?



Explorar



Cuestionar



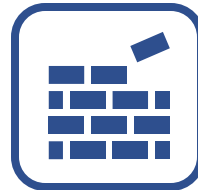
Transformar



Analizar el Impacto



**Identificar el
propósito**



Priorizar



Identificar Riesgos



Características de un QA



Atención al detalle

Minucioso, encontrar esos errores pequeños pero importantes que podrían afectar la calidad del producto



Curioso

Explorar el sistema, hacerse preguntas y siempre querer saber
CUESTIONAR TODO



Pensamiento crítico

Analizar cada aspecto de la aplicación, entender cómo debería funcionar y anticiparse a posibles problemas o fallos.



Características de un QA



Comunicación clara

Saber comunicar de manera precisa y detallada es fundamental para que el equipo entienda los problemas y pueda actuar.



Paciencia y persistencia

Testing implica probar, encontrar problemas y repetir procesos varias veces. La paciencia es clave para lograrlo de manera consistente



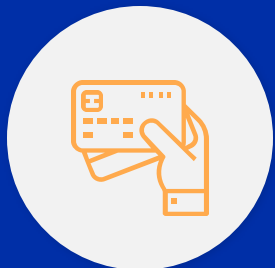
Aprendizaje continuo

Evolucionar y estar siempre dispuesto a aprender

Beneficios de Las Pruebas

Las pruebas de software son una inversión fundamental que contribuye a la creación de productos de software superiores, a la satisfacción de los usuarios y al éxito general del proyecto.





Detección temprana de errores, seguridad y confiabilidad

Las pruebas ayudan a identificar y corregir errores y defectos antes de que el software sea liberado al usuario final. Esto resulta en un producto más estable, funcional y con menos fallos.



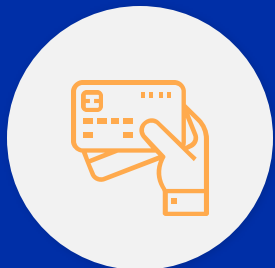
Mejorar la calidad del producto final

Las pruebas de seguridad identifican vulnerabilidades que podrían ser explotadas, protegiendo así la información de los usuarios y la integridad del sistema.



Reducción de riesgos y costos

Detectar y solucionar problemas en las etapas tempranas del desarrollo es mucho más económico que hacerlo una vez que el software está en producción. Los errores tardíos pueden generar costos significativos en correcciones, soporte técnico e incluso daños a la reputación.



Eficiencia en los procesos

Aunque pueda parecer contradictorio, una buena estrategia de pruebas, especialmente si se automatiza, puede acelerar el ciclo de desarrollo al detectar problemas rápidamente y evitar la necesidad de extensas reestructuraciones posteriores.



Satisfacción del cliente

Un software de alta calidad y con pocos errores conduce a una mejor experiencia del usuario, lo que incrementa la satisfacción y la confianza en la empresa.



Mejor colaboración

El proceso de pruebas a menudo requiere una comunicación constante entre los miembros del equipo. Los testers necesitan informar los errores a los desarrolladores, quienes a su vez pueden necesitar más información para reproducirlos y corregirlos. Esta interacción promueve el diálogo y el intercambio de conocimientos.



El más Importante Beneficio

Permite una mejor toma de decisiones

La prueba de software no es solo una actividad técnica, sino una función empresarial crítica que proporciona información valiosa para la toma de decisiones estratégicas. Al comprender el estado de calidad del software, los riesgos asociados y el impacto en los clientes, los líderes empresariales pueden tomar decisiones más informadas que conduzcan al éxito del producto y al crecimiento de la empresa:

- Reduce el riesgo y la incertidumbre.
- Impacta en la satisfacción del cliente.
- Optimiza la asignación de recursos.
- Proporciona información para la planificación estratégica.
- Evalúa el retorno de la inversión (ROI).

Responsabilidad ∞



Probar también tiene

Responsabilidades legales



Confidencialidad

Deben proteger la información confidencial de la empresa y sus clientes.



Condiciones de trabajo

Los testers tienen los mismos derechos laborales que cualquier otro trabajador.



Protección de datos

Los testers manejan información sensible y deben cumplir con leyes como el RGPD (Reglamento General de Protección de Datos, es una ley de la UE).



Propiedad intelectual

No deben infringir derechos de autor del software que prueban.



Responsabilidad

Dependiendo de la gravedad del defecto, podría surgir responsabilidad legal por defectos no detectados.



Tipos de Pruebas

Tipos de pruebas

Según su objetivo o propósito

FUNCIÓN

Pruebas Funcionales:

Verifican que el software cumpla con los requisitos funcionales definidos

Pruebas No Funcionales:

- Rendimiento
- Usabilidad
- Seguridad del software
- Compatibilidad
- Recuperación
- Entre otras

ENFOQUE

Pruebas de Caja Blanca:

Se ve la lógica de programación.

Pruebas de Caja Negra

No se ve el código

Pruebas de Caja Gris

Combina varios aspectos

AUTOMATIZACIÓN

Manuales:

Las realiza una persona

Automatizadas:

Un Script

Otros

Tipos de pruebas

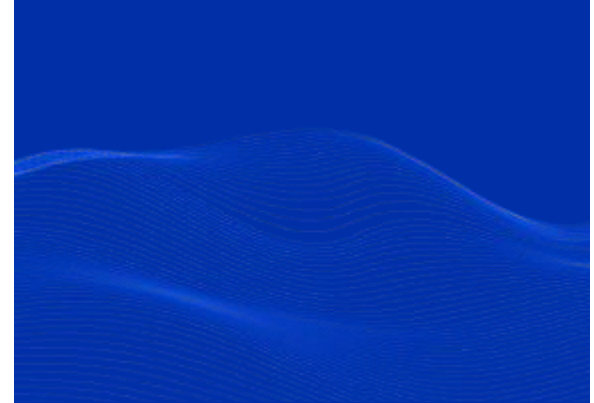
Pruebas estáticas:

Se realizan sin ejecutar el software, revisando documentos, código fuente o diagramas.

Ejemplo: Revisiones de código y análisis estático.

Pruebas dinámicas:

Implican la ejecución del software para validar su comportamiento en tiempo real.



Otros

Tipos de pruebas

Pruebas en desarrollo (Dev):

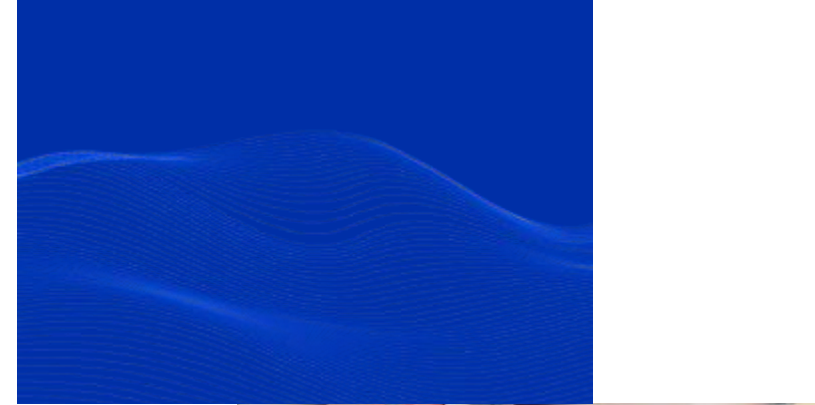
Realizadas durante la fase de desarrollo por los desarrolladores o un equipo interno.

Pruebas en staging:

Validan el software en un entorno casi idéntico al de producción, simulando el uso real antes del lanzamiento.

Pruebas en producción:

Se ejecutan después del despliegue para monitorear y detectar problemas en el entorno real.



Pruebas Especializadas

Pruebas regresivas:

Verifican que los cambios recientes (correcciones o nuevas funciones) no hayan introducido errores en funcionalidades existentes.

Pruebas exploratorias:

No siguen un guion fijo, investigan el software de manera creativa para descubrir problemas inesperados.

Pruebas de humo:

Se realizan para comprobar que las funcionalidades básicas del software están operativas antes de iniciar pruebas exhaustivas.

Pruebas A/B:

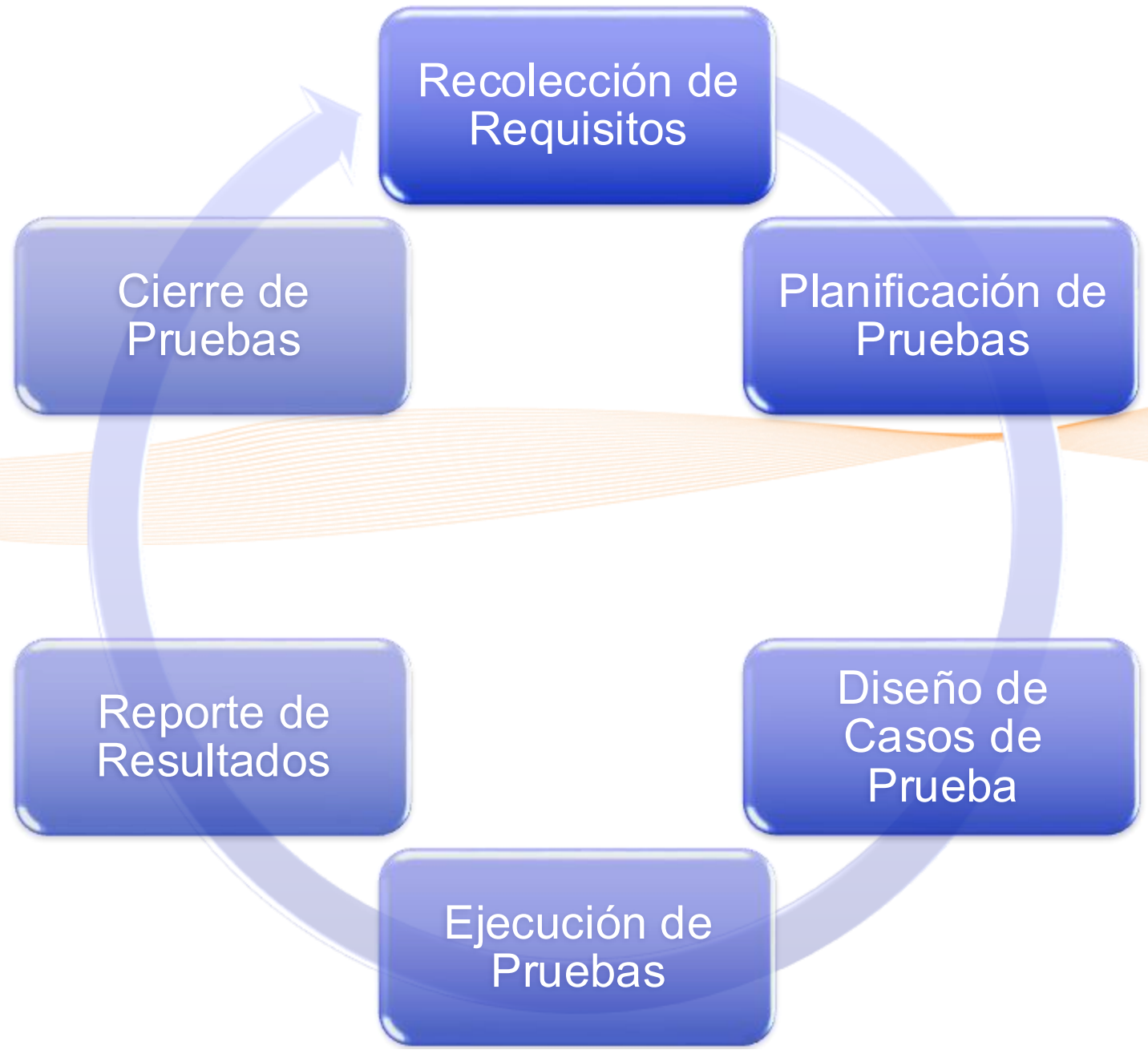
Usadas en productos digitales para comparar dos versiones de una funcionalidad o interfaz y determinar cuál funciona mejor.



Ciclo de vida de una prueba



Ciclo de Vida de las pruebas



El ciclo de vida de las pruebas de software (Software Testing Life Cycle, STLC) describe las etapas sistemáticas que se siguen para garantizar la calidad del software mediante pruebas bien planificadas y estructuradas

Pruebas en Entornos Ágiles



Pruebas en Entornos ágiles



Integración de pruebas en cada sprint

La integración de pruebas en un sprint ágil sirve para asegurar que el software se prueba continuamente a medida que se desarrolla, lo que permite detectar errores temprano, mejorar la calidad del producto y entregar software funcional al final de cada sprint. Esto ayuda a mantener el ritmo ágil sin acumular defectos y facilita una entrega continua y confiable



DEVOPS: Testing continuo y CI/CD

DevOps es clave en el testing ágil porque automatiza y acelera las pruebas, integrándolas en todo el ciclo de desarrollo. Esto permite detectar errores rápidamente, garantizar entregas continuas y mejorar la calidad del software con mayor eficiencia y colaboración entre equipos



Priorización y deuda técnica

permite mantener la salud del código a largo plazo, evitar que pequeños problemas crezcan y se vuelvan costosos, y asegurar que el equipo pueda seguir desarrollando nuevas funciones de forma ágil y sostenible. Ignorarla puede ralentizar el desarrollo, aumentar errores y afectar la calidad del producto.

Retos 





Retos

Más comunes



Manejo de los datos de prueba

Es vital tener buen contexto de las prueba, escenarios y consistencia de datos.



Ambientes de prueba

Los ambientes de prueba deben ser consistentes, deben ser una buena simulación del ambiente real.



Balance adecuado

La automatización excesiva puede resultar al final ineficaz

Tendencias ∞



Tendencias En el futuro



Inteligencia Artificial y Machine learning



Testing en la Nube y blockchain

Flexibilidad y escalabilidad



Robots



Shift-Left Testing

** pruebas en requisitos*



Pruebas de IoT

(Internet of Things)

Dispositivos interconectados que recolectan, transmiten y procesan datos mediante internet, como sensores, cámaras, dispositivos inteligentes, vehículos conectados, entre otros.

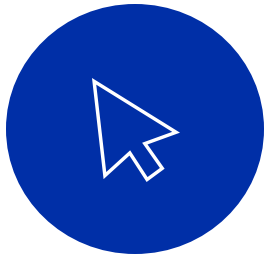


Aprendamos Historia



3 Casos de falta de pruebas

1962



Mariner 1

La Nasa se disponía a realizar el lanzamiento al espacio de la misión Mariner 1, con el fin de navegar la órbita de Venus. fue creada para recabar datos sobre la temperatura y atmósfera de Venus, pero no logró salir de la atmósfera de la tierra.

La catástrofe se originó a solo 5 minutos del despegue debido a la mala transcripción de un código y la omisión de un guion “-”.

El fracaso de la Mariner 1 produjo la pérdida de 18,5 millones de dólares a la Nasa.

1982



Therac-25

Una máquina diseñada para la administración de radioterapia a pacientes con cáncer que era controlada exclusivamente por un computador sin tener sistemas de protección mecánicos para evitar irradiar a pacientes con dosis muy altas.

Por un error de software, que ocasionó la sobredosis de radiación a 6 pacientes, 100 veces más elevadas a las que exigía su tratamiento ocasionando la muerte de al menos 3 de ellos.

2000



Y 2 k

Los programadores trataban de optimizar el espacio para guardar información, utilizando como truco guardar fechas como: **dd/mm/aa en lugar de dd/mm/aaaa**

Para el cambio del siglo XX al siglo XXI se desconfiaba en esta forma de programación, debido a que esta podría provocar un caos mundial cuando la fecha realizara el cambio de 31/12/1999 al 01/01/2000, porque para el software existente sería 01/01/00, es decir, del año 1900 atrasando 100 a todos los artefactos tecnológicos produciendo en ellos errores fatales.



Caso De Éxito ∞

Space X

SpaceX haciendo historia en la industria aeroespacial. completó su quinta prueba del Starship, el mayor cohete espacial jamás construido. El test fue todo un éxito y en su primer intento consiguió recuperar la primera fase de la nave con una grúa en la misma plataforma de lanzamiento, un hito de la ingeniería espacial.

www.elmundo.es/ciencia-y-salud/ciencia/2024/10/13/670bcca021efa0e16e8b45ab.html

Certificaciones



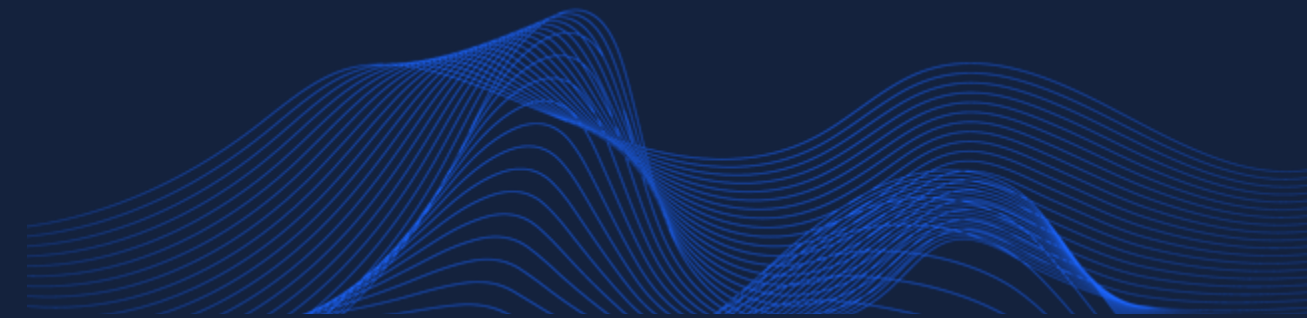
Quiénes Certifican?

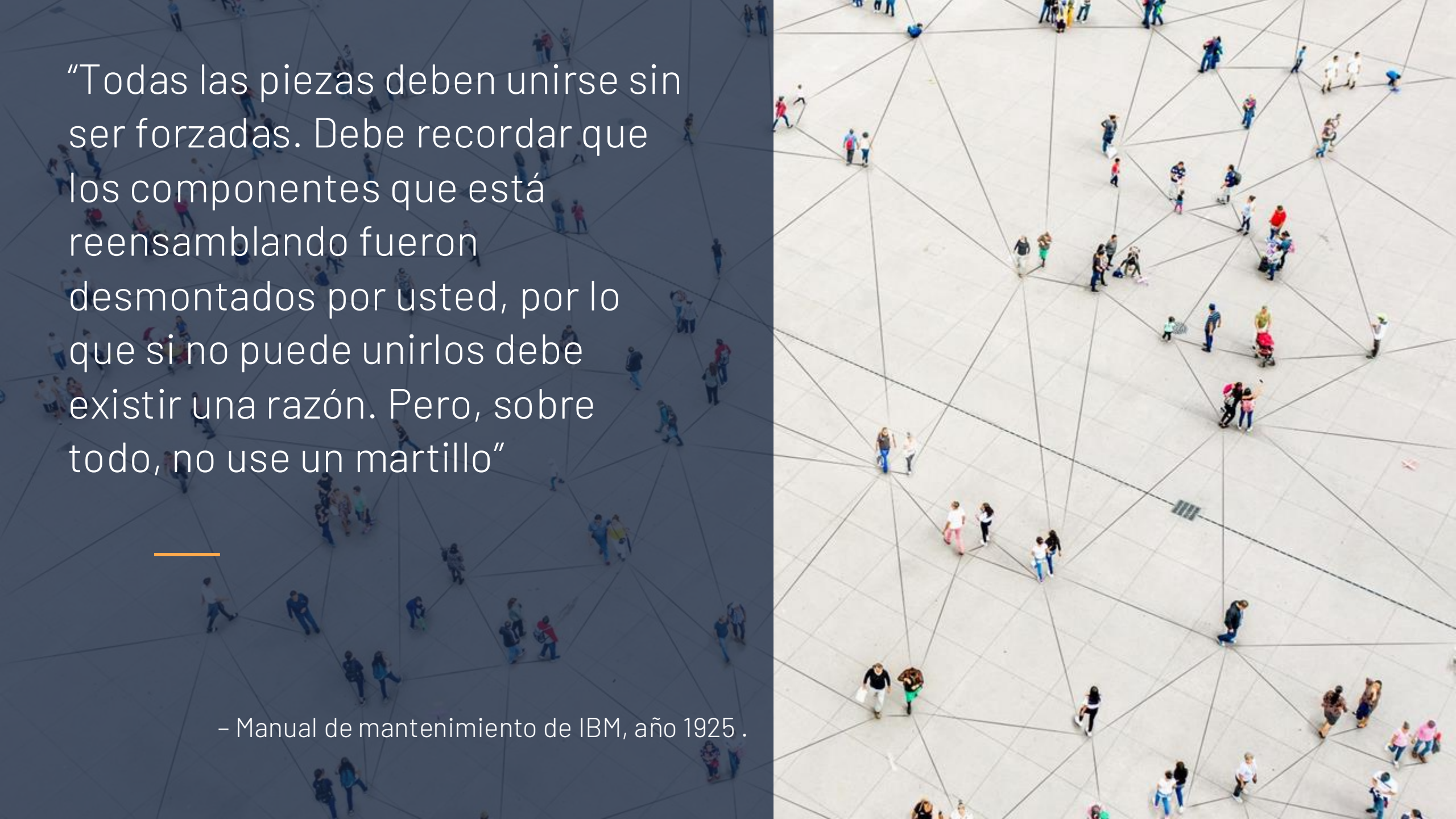
La más conocida y reconocida a nivel mundial es la certificación ISTQB (international software testing qualification board)

Certificaciones iSQI
(International Software Quality Institute):

CAT (certified agile tester)
QAMP (Quality Assurance Management Professional)

AICS: Asociación internacional de Calidad del software





“Todas las piezas deben unirse sin ser forzadas. Debe recordar que los componentes que está reensamblando fueron desmontados por usted, por lo que si no puede unirlos debe existir una razón. Pero, sobre todo, no use un martillo”

– Manual de mantenimiento de IBM, año 1925 .

Gracias!
Vamos por más!



byond^{it}

