

CURSO DE POSGRADO

# Técnicas y Gestión de las pruebas de software

---

María Elisa Presto

DOCENTE

Federico Orihuela

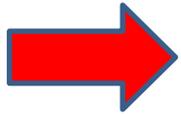
DOCENTE INVITADO

# Pruebas de características de calidad del Software

# Objetivos

- Presentar la importancia de las pruebas no funcionales
- Definir las principales características no funcionales
- Seleccionar las características a ser probadas
- Analizar en detalle los casos de prueba para algunas características
  - Usabilidad: Accesibilidad
  - Seguridad

# Índice



- Calidad de software
  - Estándar ISO 25000
  - Características no funcionales del software
    - Accesibilidad
    - Seguridad

# Calidad del software

- **Calidad:** “Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.”. RAE
- “**Grado** en el que un conjunto de características inherentes **cumple con los requisitos**”. ISO 9000.
- “Calidad es **cumplimiento de requisitos**”. Philip B. Crosby.  
“Calidad es satisfacción del **cliente**”. William E. Deming
- “Grado en que el producto software satisface las necesidades expresadas o implícitas, cuando es usado bajo **condiciones determinadas**”. ISO 25000.

# Falta de calidad

- Programas que no hacen exactamente lo que se espera
- Proyectos que no terminan nunca
- Sistemas informáticos que no se utilizan por la dificultad de su manejo
- Productos software que son imposibles de mantener cuando desaparece la persona o personas que lo desarrollaron
- Software poco seguro

# Puntos de vista

- Usuario
  - Desarrollador
  - Negocio
- 
- Interna
  - Externa
  - En uso: medible durante la utilización efectiva por parte del usuario (en un entorno de pre o producción).

Dada la complejidad de la calidad, es necesario utilizar un modelo que especifique las características de calidad

# Estándar

Del ingl. *standard*.

1. adj. Que sirve como tipo, modelo, norma, patrón o referencia.
2. m. Tipo, modelo, patrón, nivel.

Cumplir con un modelo de procesos no garantiza, por sí solo, que el producto final sea de calidad. ¿Por qué usar estándares, entonces?

# Estándar ISO 25000

SQuaRE (System and Software Quality Requirements and Evaluation)

En uso



Externa

Interna



# Familia de normas ISO/IEC 25000

- Conocida como SQuaRE (*System and Software Quality Requirements and Evaluation*)
- **Objetivo** la creación de un marco de trabajo común para evaluar la calidad del producto software
  - 25000 – Gestión de la Calidad
  - 25010 – Modelo de la Calidad
  - 25020 – Medida de la Calidad
  - 25030 – Requisitos de Calidad
  - 25040 – Evaluación de la Calidad



# Norma ISO 25010

Contiene un modelo de calidad de uso

- Compuesto de 5 características y 9 sub-características, vinculadas con el resultado de la interacción cuando se utiliza el producto

Un modelo de calidad del producto

- Compuesto por **9 características y 40 sub-características**, que son propiedades estáticas de software y propiedades dinámicas del sistema informático.

Ofrecen un conjunto coherente

- Para especificar

# Índice

- Calidad de software
- Estándar ISO 25000
- • Características no funcionales del software
  - Inclusividad (Accesibilidad)
  - Seguridad



CALIDAD DEL PRODUCTO SOFTWARE								
ADECUACIÓN FUNCIONAL	EFICIENCIA DE DESEMPEÑO	COMPATIBILIDAD	CAPACIDAD DE INTERACCIÓN	FIABILIDAD	SEGURIDAD	MANTENIBILIDAD	FLEXIBILIDAD	PROTECCIÓN
COMPLETITUD FUNCIONAL	COMPORTAMIENTO TEMPORAL	COEXISTENCIA	RECONOCIBILIDAD DE ADECUACIÓN	AUSENCIA DE FALLOS	CONFIDENCIALIDAD	MODIFICABILIDAD	ESCALABILIDAD	RESTRICCIÓN OPERATIVA
CORRECCIÓN FUNCIONAL	UTILIZACIÓN DE RECURSOS	INTEROPERABILIDAD	APRENDIZABILIDAD	DISPONIBILIDAD	INTEGRIDAD	REUSABILIDAD	INSTALABILIDAD	IDENTIFICACIÓN DE RIESGOS
PERTINENCIA FUNCIONAL	CAPACIDAD		OPERABILIDAD	TOLERANCIA A FALLOS	NO-REPUDIO	ANALIZABILIDAD	REEMPLAZABILIDAD	PROTECCIÓN ANTE FALLOS
			PROTECCIÓN FRENTE A ERRORES DE USUARIO	RECUPERABILIDAD	RESPONSABILIDAD	CAPACIDAD DE SER MODIFICADO		ADVERTENCIA DE PELIGRO
			INVOLUCRACIÓN DEL USUARIO		AUTENTICIDAD	CAPACIDAD DE SER PROBADO		INTEGRACIÓN SEGURA
			INCLUSIVIDAD		RESISTENCIA			
			ASISTENCIA AL USUARIO					
			AUTO-DESCRIPTIVIDAD					

Características ISO 25010

# Características y sub-características

CALIDAD DEL PRODUCTO SOFTWARE								
ADECUACIÓN FUNCIONAL	EFICIENCIA DE DESEMPEÑO	COMPATIBILIDAD	CAPACIDAD DE INTERACCIÓN	FIABILIDAD	SEGURIDAD	MANTENIBILIDAD	FLEXIBILIDAD	PROTECCIÓN
COMPLETITUD FUNCIONAL	COMPORTAMIENTO TEMPORAL	COEXISTENCIA	RECONOCIBILIDAD DE ADECUACIÓN	AUSENCIA DE FALLOS	CONFIDENCIALIDAD	MOD	Características ISO 25010	
CORRECCIÓN FUNCIONAL	UTILIZACIÓN DE RECURSOS	INTEROPERABILIDAD	APRENDIZABILIDAD	DISPONIBILIDAD	INTEGRIDAD	REUSABILIDAD	ESCALABILIDAD	RESTRICCIÓN OPERATIVA
PERTINENCIA FUNCIONAL	CAPACIDAD		OPERABILIDAD	TOLERANCIA A FALLOS	NO-REPUDIO	ANALIZABILIDAD	INSTALABILIDAD	IDENTIFICACIÓN DE RIESGOS
			PROTECCIÓN FRENTE A ERRORES DE USUARIO	RECUPERABILIDAD	RESPONSABILIDAD	CAPACIDAD DE SER MODIFICADO	REEMPLAZABILIDAD	PROTECCIÓN ANTE FALLOS
			INVOLUCRACIÓN DEL USUARIO		AUTENTICIDAD	CAPACIDAD DE SER PROBADO		ADVERTENCIA DE PELIGRO
			INCLUSIVIDAD		RESISTENCIA			INTEGRACIÓN SEGURA
			ASISTENCIA AL USUARIO					
			AUTO-DESCRIPTIVIDAD					

iso25000.com

<https://iso25000.com/index.php/normas-iso-25000/iso-25010>

# Normas complementarias para la calidad en sistemas de IA

## **ISO/IEC 25059:2023 Modelo de calidad específico para IA**

- Adaptabilidad funcional
- Transparencia
- Controlabilidad por el usuario
- Intervenibilidad
- Robustez

## **ISO/IEC 42001:2023 – Gestión de sistemas de IA**

- Gobernanza
- Ética
- Transparencia
- Gestión de riesgos y ciclo de vida de IA

# Compatibilidad

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software. Se subdivide en dos subcaracterísticas:

- **Coexistencia.** Capacidad del producto para coexistir con otro software independiente, en un entorno común, **compartiendo recursos** comunes sin detrimento.
- **Interoperabilidad.** Capacidad de dos o más sistemas o componentes para **intercambiar información** y utilizar la información intercambiada.

# Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa **bajo unas condiciones y periodo de tiempo determinados**. Se subdivide a su vez en las siguientes subcaracterísticas:

- **Ausencia de fallos.** Capacidad del sistema de llevar a cabo sus funciones sin fallos bajo condiciones normales de operación.
- **Disponibilidad.** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- **Tolerancia a fallos.** Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- **Capacidad de recuperación.** Capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

# ¿Qué es la mantenibilidad?

Es la capacidad del producto software para ser modificado de manera efectiva y eficiente. Por diferentes razones se mantiene el software

- Correctivas (arreglar errores),
- Evolutivas (agregar o cambiar funcionalidades),
- Perfectivas (mejorar calidad interna).

# Características

- **Modularidad.** Capacidad de un producto para evitar que los cambios en un componente afecten a otros componentes.
- **Reusabilidad.** Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- **Analizabilidad.** Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- **Capacidad para ser modificado.** Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar su calidad.
- **Capacidad para ser probado.** Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

# ¿Qué es la flexibilidad?

- Es la capacidad del producto software para adaptarse a cambios en sus requisitos, contextos de uso o entornos técnicos (hardware, software, redes, etc.)

# Características

- **Adaptabilidad.** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- **Escalabilidad.** Capacidad del producto para gestionar cargas de trabajo crecientes o decrecientes y para adaptar su capacidad a la variabilidad.
- **Instalabilidad.** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- **Reemplazabilidad.** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

# Comparación

Característica	Compatibilidad	Flexibilidad
¿Qué evalúa?	Interacción con otros sistemas o versiones	Adaptabilidad del software a otros entornos
¿Palabras clave?	Interoperabilidad, coexistencia, estándares	Escalabilidad, instalación, reemplazo
¿Ejemplo clásico?	Funciona con otra app o API externa	Funciona en otro sistema operativo o entorno
¿Lo importante es...?	Que <i>conviva</i> sin conflicto	Que <i>migre, crezca o cambie</i> sin problema

# ¿Qué es la Protección?

- Es la capacidad del producto software para, bajo condiciones específicas, evitar situaciones que pongan en peligro la vida humana, la salud, la propiedad o el medio ambiente.

# Sub características

- **Restricción operativa**  
El software limita su funcionamiento a parámetros seguros cuando hay peligro.
- **Identificación de riesgos**  
El software puede detectar operaciones o situaciones que representan riesgos inaceptables.
- **Protección ante fallos**  
El producto se pone en un modo seguro automáticamente ante fallos o vuelve a una condición segura.
- **Advertencia de peligro**  
El sistema alerta sobre riesgos con tiempo para que se pueda actuar.
- **Integración segura**  
La seguridad se mantiene durante y después de la integración con otros sistemas o componentes.

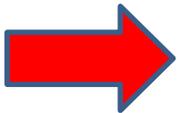
# Ejemplo

Imaginemos un software que controla un ascensor en un edificio:

- Restricción operativa: Si el ascensor detecta que la puerta no está bien cerrada, no permite que se mueva para evitar accidentes.
- Identificación de riesgos: El sistema detecta si hay una sobrecarga de peso o una falla en los frenos que pueda ser peligrosa.
- Protección ante fallos: Si el software nota un fallo, automáticamente detiene el ascensor y lo pone en modo seguro para que nadie salga lastimado.
- Advertencia de peligro: Cuando hay un problema, el sistema avisa con luces o sonidos para que las personas estén alertas.
- Integración segura: Si el ascensor se conecta con otros sistemas del edificio, como alarmas o centrales de emergencia, lo hace sin poner en riesgo la seguridad.

# Índice

- Calidad de software
- Estándar ISO 25000
- Características no funcionales del software
  - Accesibilidad
  - Seguridad



# ¿Qué es la capacidad de interacción?

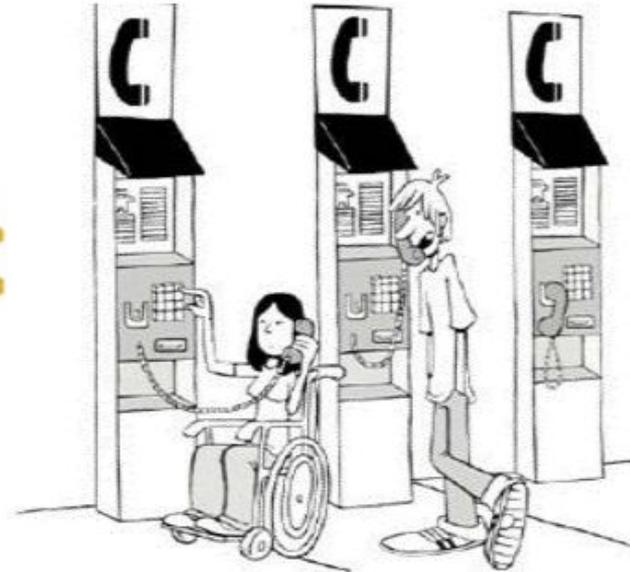
Capacidad del producto software para que el **usuario interactúe mediante su interfaz** intercambiando información para completar determinadas tareas.

La capacidad de interacción es fundamental porque hace que el software no solo funcione bien, sino que también sea fácil, agradable y seguro de usar. Abarca desde que el usuario pueda entender rápidamente si el software le sirve, hasta que se sienta motivado a usarlo sin frustraciones ni errores. Además, asegura que personas con diferentes habilidades o contextos puedan usarlo con éxito, y que no dependa de manuales complejos para ser aprovechado.

# Sub-características

- **Reconocibilidad de la adecuación** El usuario puede entender fácilmente si el software cumple con sus necesidades.
- **Aprendizabilidad** El usuario puede aprender a usar el software dentro de un tiempo determinado.
- **Operabilidad** El usuario puede manejar y controlar el software con facilidad.
- **Protección contra errores de usuario** El sistema previene que el usuario cometa errores durante la operación.
- **Involucración del usuario** El software presenta sus funciones e información de forma atractiva, fomentando la interacción continua.
- **Inclusividad** El producto puede ser usado por personas con distintos contextos (edad, habilidades, cultura, raza, lenguaje, género, etc.).
- **Asistencia al usuario** El producto ayuda a usuarios con diferentes características a lograr sus objetivos.
- **Auto-descriptividad** El software presenta la información y las funciones de manera que su uso sea evidente, sin requerir mucha ayuda externa.

# Inclusividad/ Accesibilidad Sub-característica



# Decreto 406/22

## Artículo 1

Se entenderá por accesibilidad digital, la posibilidad de que toda la información y contenidos disponibles a través de soluciones tecnológicas, independiente a su canal de implementación, ya sea tecnología web o móvil, en internet, intranets y/o cualquier tipo de redes informáticas, se hagan disponibles y utilizables por el usuario, mediante el uso de equipamiento adecuado, independientemente de su contexto y condiciones personales, contemplando especialmente a las personas con discapacidad.

## Artículo 3

En el marco del cumplimiento de las acciones referidas en el artículo anterior, corresponde a Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y del Conocimiento:

- a) actualizar en forma periódica, el elenco de requisitos mínimos que se anexa al presente Decreto, y la política de accesibilidad digital, considerando la evolución o adecuación de las buenas prácticas y recomendaciones internacionales, específicamente las recomendaciones del W3C - WAI (Web Accessibility Initiative del World Wide Web Consortium), lo que deberá publicar en su sitio web, junto con la fecha de cada actualización;
- b) colaborar para el cumplimiento de las acciones incluidas en el presente Decreto a través de la elaboración de recomendaciones y guías, talleres de sensibilización, cursos de capacitación y asesoramiento. Dicha colaboración podrá condicionarse a la presentación del plan de mejora previsto en el literal b del artículo

# Organización de las WCAG

En servicios y productos digitales quedan comprendidos los sitios web, intranets, servicios web y móviles, aplicaciones web y móviles, trámites, aplicaciones de uso interno y documentos en cualquier formato.

## Estándar W3C-WCAG 2.1

W3C es el consorcio internacional que define los estándares web, tales como HTML, CSS, entre otros.

El grupo de trabajo que se ocupa de la accesibilidad dentro de W3C se denomina Iniciativa de Accesibilidad Web (WAI) y ha publicado las Pautas de Accesibilidad al Contenido Web (WCAG por su sigla en inglés) las cuales son periódicamente revisadas, actualmente la última versión aprobada es la versión WCAG 2.1. Cada una de estas pautas tiene criterios de conformidad verificables, el listado completo de criterios puede ser consultado en la web de W3C - WAI Pautas de accesibilidad para el contenido web 2.1

A los efectos de comprender los requisitos de accesibilidad es necesario conocer la definición de algunos términos específicos tales como productos de apoyo y agentes de usuario:

- \* Los productos de apoyo o ayudas técnicas: son programas o dispositivos que permiten a los usuarios con discapacidad interactuar con los

<https://www.w3.org/>

# Principios

1. **Perceptible:** La información debe presentarse de manera que los usuarios puedan percibirla.
2. **Operable:** Los componentes de la interfaz de usuario y la navegación deben ser operables.
3. **Entendible:** La información y el funcionamiento de la interfaz de usuario deben ser comprensibles.
4. **Robusto:** El contenido debe ser lo suficientemente robusto como para que pueda ser interpretado por una amplia variedad de agentes de usuario, incluidas las tecnologías de asistencia.

# Nivel de conformidad

1. Nivel A: el más bajo. Debe cumplir
  - a. 30 criterios de conformidad de nivel A.
2. Nivel AA: nivel medio. Debe cumplir 50 criterios de conformidad:
  - a. 30 del nivel A más
  - b. 20 del nivel AA.
3. Nivel AAA: el más alto. Debe cumplir 78 criterios de conformidad:
  - a. 30 del nivel A más
  - b. 20 del nivel AA más
  - c. 28 del nivel AAA.

# Requisitos de accesibilidad digital:

## Estado Uruguayo

1. Alcanzar los niveles A y AA.
2. Cumplir con los siguientes criterios de conformidad del nivel AAA
  - a. Contraste aumentado - criterio 1.4.6.
  - b. Presentación visual - criterio 1.4.8.
  - c. Límites de tiempo - criterio 2.2.6.
  - d. Ubicación - criterio 2.4.8.
  - e. Encabezados de sección - criterio 2.4.10.
  - f. Tamaño del área de interacción - criterio 2.5.5.
  - g. Cambio a petición - criterio 3.2.5.

# Herramientas

- [aXe Dev Tools](#)
- [TAW](#)
- [WAVE](#)

# Pruebas con usuarios

Técnicas de filtrado

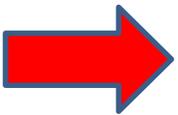
Test formal

- laboratorio aislado
- evaluadores especializados
- Software que registre las acciones del usuario
- Guión orientativo

# Preguntas

# Índice

- Calidad de software
- Estándar ISO 25000
- Características no funcionales del software
  - Accesibilidad
  - Seguridad



# Seguridad



# Seguridad

## ¿Qué es?

- Característica que indica que un sistema está libre de todo peligro, daño o riesgo (*diccionario*).
- Deseo de proteger un elemento (un activo).
- Preservación de la confidencialidad, integridad y disponibilidad de la información (*seguridad de la información*).
- Una utopía, una característica a la que aspirar, aunque nunca podamos llegar a afirmar que nuestro sistema es completamente seguro.

# Seguridad

## Motivación

- Calidad del software (filosóficamente).
- Brindar un servicio adecuado.
- Mejora la imagen de la empresa (o al menos no empeora al no sufrir ataques).
- Existen personas o grupos de personas dispuestas a explotar los problemas de seguridad.
- Dinero.



# Seguridad

## Características

- **Confidencialidad**

- Propiedad de la información de no ponerse a disposición o ser revelada a individuos, entidades o procesos no autorizados. (ISO27000)

- **Integridad**

- Propiedad de la información relativa a su exactitud y completitud. (ISO27000)

- **Disponibilidad**

- Propiedad de la información de estar accesible y utilizable cuando lo requiera una entidad autorizada. (ISO27000)

# Seguridad

## Otras características

- **Autenticidad**

- Propiedad de que una entidad es lo que afirma ser. (ISO27000).
- La información no fue creada o transmitida por un tercero.
- Permite identificar el origen de la información.

- **No repudio**

- El no repudio o irrenunciabilidad es un servicio de seguridad que permite probar la participación de las partes en una comunicación. (ISO27000)
- No repudio en origen (enviado por el autor)
- No repudio en destino (recibido por el destinatario)
- El no repudio evita que el emisor o el receptor nieguen la transmisión de un mensaje.

# Seguridad

## Información sensible

- Información sensible es aquella información, así definida por su propietario, cuya revelación, alteración, pérdida o destrucción puede producir daños importantes a la organización o persona propietaria de la misma.
  - Ej. Datos personales, en Uruguay protegidos por la ley 18.331
  - Ej. Otros datos protegidos por ley como el secreto bancario.
  - Ej. Datos confidenciales de una empresa
  - Ej. Información sobre la infraestructura o tecnología subyacente de un software o sistema de información

# Testing de Seguridad

- El Testing de Seguridad es un proceso que tiene como objetivo evaluar la resistencia y la robustez de una aplicación o sistema informático ante posibles amenazas y vulnerabilidades de seguridad.
- El objetivo es garantizar la protección de la información confidencial, la integridad de los datos y la disponibilidad de los sistemas, previniendo y mitigando riesgos y posibles ataques malintencionados.

# Testing de Seguridad

## Metodologías

- OWASP Testing Guides (Guías de Pruebas OWASP)
  - Web, Mobile, Firmware
- Penetration Testing Execution Standard (PTES)
  - Define etapas y herramientas a utilizar para el testing de penetración
- PCI Penetration Testing Guide
  - Definido por el estándar de seguridad de la industria de pago con tarjeta (o tarjetas de pago)
- Penetration Testing Framework
  - Guía práctica con herramientas, ejemplos concretos, y tecnologías específicas.
- OSSTMM (Open Source Security Testing Methodology Manual)
  - Más orientada a seguridad operativa

# Testing de Seguridad

## Guía OWASP

### Pruebas de intrusión de aplicaciones Web

- Recopilación de información
  - Ej. Enumeración (descubrimiento) de aplicaciones
  - Ej. Análisis de códigos de error
- Pruebas de gestión de configuración de la infraestructura
  - Ej. Pruebas de SSL/TLS
  - Interfaces de administración de la infraestructura y de la aplicación
- Comprobación del sistema de autenticación
  - Ej. Transmisión de credenciales a través de un canal cifrado
  - Ej. Saltarse el sistema de autenticación

# Testing de Seguridad

## Guía OWASP

- Pruebas de gestión de sesiones
  - Pruebas para el esquema de gestión de sesiones
  - Pruebas para CSRF
- Pruebas de autorización
  - Ej. Pruebas de ruta transversal
  - Ej. Pruebas de escalación de privilegios
- Comprobación de la lógica de negocio
- Pruebas de validación de datos
  - Ej. Inyecciones (XSS, SQL, LDAP, XML, etc.)
  - Ej. Desbordamientos de búfer

# Testing de Seguridad

## Guía OWASP

- Pruebas de denegación de servicio
  - Ej. Bloqueando cuentas de usuario
  - Ej. ataques CON Wildcards SQL
- Comprobación de servicios web
  - Ej. Comprobación de XML a nivel de contenido
  - Ej. Adjuntos SOAP maliciosos.
- Pruebas de Ajax
  - Ej. Vulnerabilidades Ajax

# Testing de Seguridad

## OWASP Top 10 (2021)

- A01:2021 - Pérdida de Control de Acceso
  - Ej. Exposición de información sensible a un actor no autorizado
  - Ej. CSRF
- A02:2021 - Fallas Criptográficas
  - Ej. Uso de contraseñas en código fuente
  - Ej. Uso de algoritmo criptográfico vulnerado o inseguro
- A03:2021 - Inyección
  - Ej. Inyecciones SQL
  - Ej. XSS
- A04:2021 - Diseño Inseguro
  - Ej. Generación de mensaje de error que contiene información sensible
  - Ej. Almacenamiento de credenciales sin protección o con protección insuficiente

# Testing de Seguridad

## OWASP Top 10 (2021)

- A05:2021 - Configuración de Seguridad Incorrecta
  - Ej. Configuración
  - Ej. Restricción incorrecta entidades externas referenciadas de XML
- A06:2021 - Componentes Vulnerables y Desactualizados
  - Ej. Uso de componentes de terceros no mantenidos
- A07:2021 - Fallas de Identificación y Autenticación
  - Ej. Autenticación incorrecta
  - Ej. Fijación de sesiones
- A08:2021 - Fallas en el Software y en la Integridad de los Datos
  - Ej. Desererialización de datos no confiables.
  - Ej. Inclusión de funcionalidades provenientes de fuera de la zona de confianza

# Testing de Seguridad

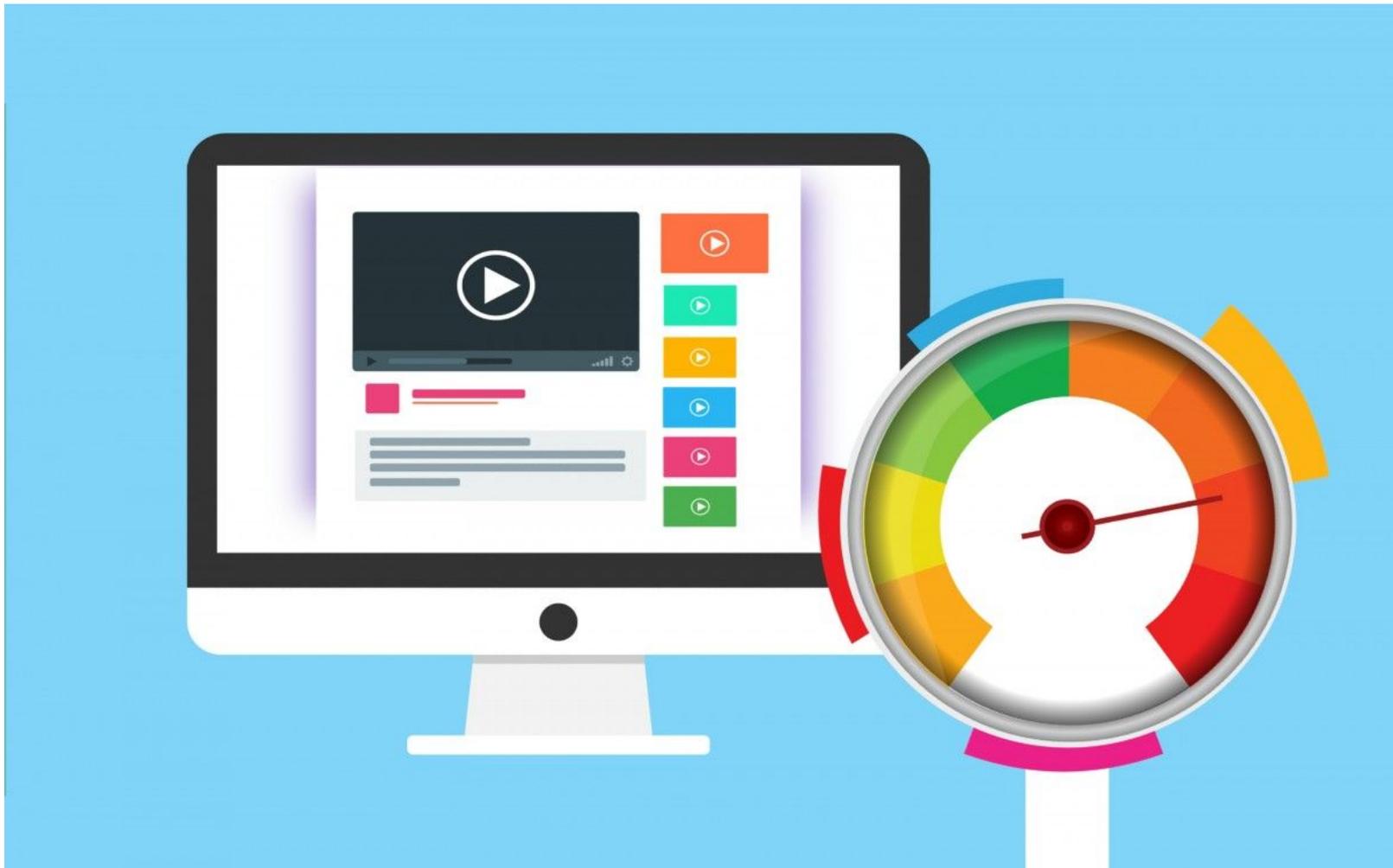
## OWASP Top 10 (2021)

- A09:2021 - Fallas en el Registro y Monitoreo
  - Neutralización de salida incorrecta para registros
  - Omisión de información relevante para la seguridad
  - Inserción de información sensible en archivo de registro
- A10:2021 - Falsificación de Solicitudes del Lado del Servidor
  - SSRF (Server side request forgery)

# Índice

- Calidad de software
- Estándar ISO 25000
- • Características no funcionales del software
  - Accesibilidad
  - Seguridad

# Testing de Performance



# Testing de Performance

- **Performance (Eficiencia o Desempeño)**
  - Velocidad a la cual un sistema de información procesa transacciones.
  - En general se miden tiempos.
  - Calidad del software.
  - Requerimiento no funcional del sistema.
  - Es muy importante desde la usabilidad del sistema.
  - Aprovechar los recursos de forma apropiada.



# Testing de Performance

- Pruebas (testing) realizadas para conocer el desempeño de un sistema.
- Permiten observar o medir tiempos de respuesta, productividad (Throughput), capacidad, eficiencia, concurrencia, escalabilidad.
- Para optimizar las aplicaciones, lo ideal es integrarlo al ciclo de desarrollo luego del testing funcional. Para otros objetivos, es necesario ejecutar “al final”.

# Testing de Performance

## ¿Por qué?

- Conocer la performance con la que el usuario percibe al sistema
- Encontrar punto de quiebre
- Detectar los cuellos de botella
- Analizar la estabilidad de la aplicación
- Dimensionar el sistema
- “Tunear” el Hardware o Software de Base
- Mitigar riesgos

# Testing de Performance

## ¿Cómo?

- En general se simula la cantidad de usuarios que se desea modelar. Estos usuarios "realizan" distintas transacciones (funcionalidades) sobre el sistema.
- Se utilizan herramientas para generar carga que simulan los usuarios (se les llama usuarios virtuales). Cada usuario ejecuta el flujo deseado.

# Preguntas

Muchas gracias

# Bibliografía

Testing Funcional, Centro de Ensayos de Software  
Calidad de procesos y productos de Software, ISO 25000