

ISO 25000

# Introducción

- ISO/IEC 25000, conocido también como SQuaRE (System and Software Quality Requirements and Evaluation) es una familia de normas que evalúan la calidad de un producto de software
- Este estándar es el resultado de la evolución de normas anteriores:
  - ISO/IEC 9126, que define un modelo de calidad para la evaluación de software
  - ISO/IEC 14598, que define el proceso para la evaluación de software
- Cubre dos procesos principales:
  - Especificación de requisitos de calidad de software
  - Evaluación de la calidad de software, detallando el proceso para dicha evaluación

# Alcance

Va dirigido a empresas dedicadas al desarrollo de software, sin importar tamaño, presupuesto o alcance de la compañía.



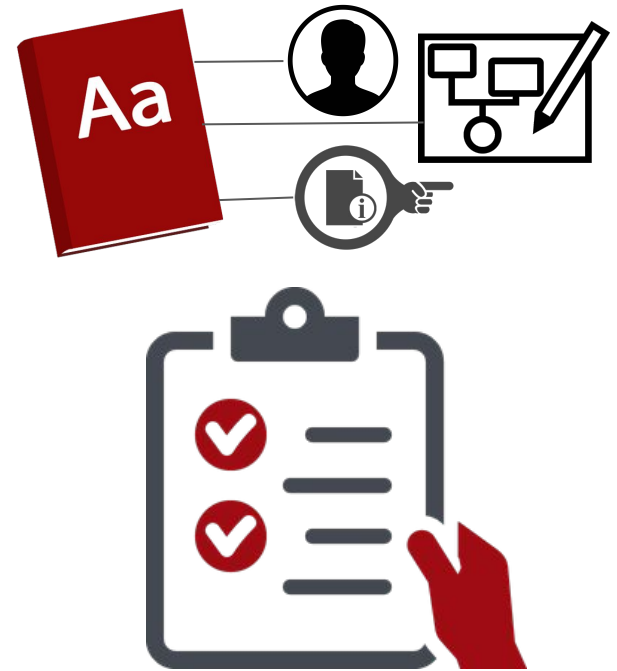
# Objetivo

Tener un marco de trabajo común para evaluar la calidad de un de producto software



# Objetivo: 2500n División de Gestión para la Calidad

- **ISO/IEC 25000** - *Guide to SQuaRE*
  - Describe:
    - La **arquitectura** del modelo SQuaRE.
    - La **terminología** de la familia.
    - Un **resumen**.
    - Los **usuarios** y partes asociadas.
    - Modelos de **referencia**.
- **ISO/IEC 25001** - *Planning and Management*
  - Establece los **requisitos** y **orientaciones** para gestionar la **evaluación** y **especificación** de los requisitos del producto software.



# Objetivo: 2501n División de Modelo de Calidad

- **ISO/IEC 25010** - *System and software quality models*
  - Describe el modelo de calidad para el producto software y para la calidad en uso. Esta Norma presenta las **características** y subcaracterísticas de **calidad** frente a las cuales **evaluar** el producto software.
- **ISO/IEC 25012** - *Data Quality model*:
  - Define un modelo general para la calidad de los datos, **aplicable a aquellos datos que se encuentran almacenados** de manera estructurada y forman parte de un Sistema de Información.



# Objetivo: 2502n División de Medición de Calidad

- **ISO/IEC 25020** - *Measurement reference model and guide*:
  - Proporciona una guía para que los **usuarios seleccionen** o desarrollen medidas propuestas por **normas ISO**.
- **ISO/IEC 25021** - *Quality measure elements*:
  - Especifica un **conjunto de métricas** para ser usadas a lo largo de todo el ciclo de vida del desarrollo de software.



# Objetivo: 2503n División de Requisitos de Calidad

- **ISO/IEC 25030** - *Quality requirements*:
  - Provee de un conjunto de recomendaciones para realizar la especificación de los requisitos de calidad del producto software.





# Objetivo: 2504n División de Evaluación de Calidad

- **ISO/IEC 25040** - *Evaluation reference model and guide*:
  - Propone un **modelo de referencia** general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los **recursos necesarios** para obtener las correspondientes **salidas**.
- **ISO/IEC 25041** - *Evaluation guide for developers, acquirers and independent evaluators*:
  - Describe los **requisitos y recomendaciones** para la implementación práctica de la evaluación del producto software **desde el punto de vista de los desarrolladores**, de los adquirentes y de los evaluadores independientes.
- **ISO/IEC 25042** - *Evaluation modules*:
  - Define lo que la Norma considera un módulo de evaluación y la **documentación, estructura y contenido que se debe utilizar** a la hora de definir uno de estos **módulos**.
- **ISO/IEC 25045** - *Evaluation module for recoverability*:
  - Define un módulo para la evaluación de la sub-característica para la **recuperabilidad**

# ISO/IEC 25010

El modelo de calidad es pieza clave para la un modelo de evaluación de calidad. Dicho modelo determina qué características serán tomadas en cuenta al evaluar un producto de software.

El modelo definido en este estándar define 8 aspectos que serán evaluadas:

- adecuación funcional
- eficiencia de desempeño
- compatibilidad
- usabilidad
- fiabilidad
- seguridad
- mantenibilidad
- portabilidad

# Adecuación Funcional

Representa la capacidad del software para satisfacer necesidades cuando es usado en condiciones específicas. Se divide en:

- Completitud funcional: Describe qué tantas tareas y objetivos del usuario cubre el software.
- Corrección funcional: Es la capacidad de que el software tenga resultados correctos con un nivel de precisión alto.
- Pertinencia funcional: Capacidad para proporcionar funciones apropiadas para la resolución de tareas y objetivos del usuario

# Eficiencia de Desempeño

Trata sobre el desempeño relativo a los recursos utilizados. Se divide en:

- Comportamiento temporal: Mide los tiempos de respuesta bajo condiciones determinadas
- Utilización de recursos: Qué tantos recursos utiliza y de qué tipo
- Capacidad: Si los límites del producto cumplen con los requisitos

# Compatibilidad

Mide la capacidad de dos o más sistemas o componentes de intercambiar información o compartir recursos. Se divide en:

- Coexistencia: Capacidad para coexistir con otro software independiente, compartiendo recursos en un entorno común
- Interoperabilidad: Capacidad de dos o más componentes para intercambiar y utilizar información compartida.

# Usabilidad

Capacidad y facilidad del software para ser usado por el usuario. Se divide en:

- Capacidad para reconocer adecuación: Permite al usuario entender si el software es apto para sus necesidades
- Capacidad de aprendizaje: Facilidad de aprender cómo usar el software
- Capacidad para ser usado: Qué tan fácil puede el usuario operar o controlarlo
- Protección contra errores del usuario: Proteger al usuario de hacer errores
- Estética: Qué tan agradable es a la vista
- Accesibilidad: Qué tanto se adapta a las necesidades y discapacidades del usuario.

# Fiabilidad

Capacidad del sistema de realizar funciones bajo ciertas circunstancias. Se divide en:

- Madurez: Mantenerse fiable en condiciones de uso normales
- Disponibilidad: Capacidad de estar accesible para cuando se requiera
- Tolerancia a fallas: Capacidad para operar aún con fallas de software y/o hardware
- Capacidad de recuperación: Capacidad para recuperar datos y restablecer funcionamiento después de un fallo.

# Seguridad

Protección de los datos del usuario y del producto. Se divide en:

- Confidencialidad: Restringir el acceso de información no autorizado
- Integridad: Capacidad de prevenir acceso o modificaciones no autorizadas
- No repudio: Demostrar los eventos que han ocurrido para que no se repitan.
- Responsabilidad: Rastrear las acciones de una entidad
- Autenticidad: Demostrar la identidad y características de un sujeto o recurso.

# Mantenibilidad

Representa la capacidad para ser modificado efectivamente. Se divide en:

- Modularidad: Que un cambio en un componente no afecte a los demás
- Reusabilidad: Permite que un recurso sea usado en más de un sistema
- Analizabilidad: Facilidad con la que se pueden evaluar cambios al sistema
- Capacidad de modificación: Si permite que sea modificado de forma efectiva sin cambiar la integridad del sistema
- Capacidad para ser probado: Facilidad con la que se pueden establecer criterios de evaluación para un componente

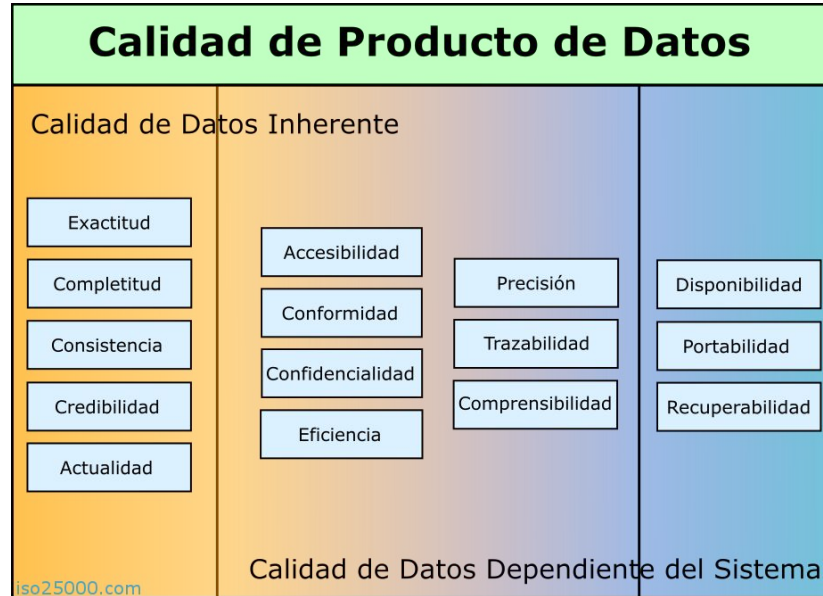
# Portabilidad

Capacidad de ser transferido de forma efectiva a otro entorno. Se divide en:

- Adaptabilidad: Capacidad para adaptarse de forma efectiva a diferente hardware, software o entorno operacional.
- Capacidad para ser instalado: Facilidad con la que se pueda instalar o desinstalar en un entorno determinado.
- Capacidad para ser reemplazado: La capacidad de que sea usado en lugar de otro software con el mismo propósito y el mismo entorno.

# ISO/IEC 25012

Este estándar representa la base sobre la cual se construye un sistema para la evaluación de software. En un modelo de Calidad de Datos se establecen las características que se deben tener en cuenta a la hora de evaluar dicho sistema. Está compuesto por 15 características.



# Calidad de Datos Inherente

## Exactitud

- Representan correctamente el verdadero valor del atributo.
- Representan correctamente el verdadero valor del atributo en un contexto.

## Compleitud

- Los datos asociados con una entidad tienen valores para todo los atributos esperados.

## Consistencia

- Los datos asociados son libres de contradicción y son coherentes con otros datos en el contexto.



# Calidad de Datos Inherente

## Credibilidad

- Los datos son ciertos y creíbles en un contexto dado.

## Actualidad

- Los datos son actualizados dependiendo de su uso y si sí deben ser actualizados.

# Calidad de Datos Inherente y Dependiente del Sistema

## Accesibilidad

- Todos los datos pueden ser accedidos en un contexto para aquellas personas o usuarios que necesiten de ellos.

## Conformidad

- Los datos tienen atributos que se adhieren a estándares, convenciones o normativas vigentes y reglas similares.

## Confidencialidad

- Los datos son sólo accedidos o interpretados por aquellos usuarios autorizados.

## Eficiencia

- Los datos pueden ser procesados y proporcionados con los niveles de rendimiento esperados.

# Calidad de Datos Inherente y Dependiente del Sistema

## Presición

- Los datos son exactos.

## Trazabilidad

- Los datos proporcionan información de si fueron modificados, creados, etc.

## Comprensibilidad

- Los datos tienen atributos que permiten ser leídos e interpretados por los usuarios y son expresados utilizando lenguajes, símbolos, etc.

# Calidad de Datos Dependiente del Sistema

## Disponibilidad

- Los datos pueden ser obtenidos por usuarios y/o aplicaciones autorizados.

## Portabilidad

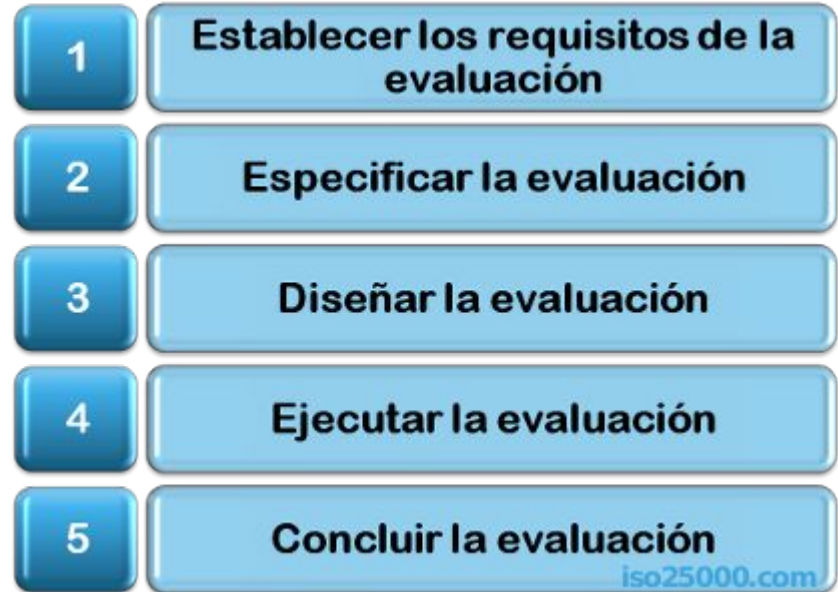
- Los datos pueden ser movidos de sistema a otro preservando el nivel de calidad.

## Recuperabilidad

- Incluso en caso de fallo, los datos permiten mantener y preservar un nivel de operaciones y calidad.

# ISO/IEC 25040

Define el proceso para llevar a cabo la evaluación de la calidad del software.  
Consiste de cinco actividades



# 1. Establecer requisitos de evaluación

Consiste de 4 tareas:

1. Establecer el propósito de la evaluación
2. Obtener los requisitos de calidad del producto
3. Identificar las partes del producto que se van a evaluar
4. Definir el rigor de la evaluación

## 2. Especificar la evaluación

Consiste de 3 tareas:

### 1. Seleccionar los módulos de evaluación

- Deben cubrir todos los requerimientos de evaluación de calidad. El evaluador puede usar el estándar ISO/IEC 2502n como ayuda para esta tarea.

### 2. Definir los criterios de decisión para las métricas

- Las métricas son límites u objetivos numéricos que determinan la necesidad de describir el nivel de confianza o de investigar más a profundidad cierto resultado

### 3. Definir los criterios de decisión de la evaluación

- El evaluador debe preparar un procedimiento para la abstracción y resumen de los datos de evaluación, que se usará como base para la evaluación final de la calidad del software.

### 3. Diseñar la evaluación

Consiste de una sola tarea: Planear las actividades de evaluación.

Se deben tomar en cuenta la disponibilidad de recursos que serían necesarios para la evaluación, así como el presupuesto, métodos de evaluación y estándares adaptados.



## 4. Ejecutar la evaluación

Se subdivide en 3 tareas:

1. Realizar las mediciones
2. Aplicar los criterios de decisión para las métricas
3. Aplicar los criterios de decisión de la evaluación

## 5. Concluir la evaluación

Consta de 4 tareas:

1. Revisar los resultados de la evaluación
  - De esta manera se detectan errores y se interpretan resultados
2. Crear informe de evaluación
3. Revisar la calidad de evaluación y obtener feedback
  - El feedback sirve para mejorar el proceso y técnicas de evaluación usadas.
4. Tratar los datos de la evaluación
  - El evaluador debe, de acuerdo con lo acordado antes de realizar las pruebas, entregar, devolver, archivar o eliminar los datos resultantes.

# Ventajas

## Para la organización:

- Alinea los objetivos del software con las necesidades reales que se le demandan.
- Evitan ineficiencias y maximiza la rentabilidad y calidad
- El proceso de evaluaciones periódicas ayuda a supervisar continuamente el rendimiento y la mejora.
- Mejora la imagen de la empresa.



## Para los clientes:

- Al demostrar el compromiso de la organización con la calidad del software
- Certificar el software aumenta la satisfacción del cliente.



# Normas Mexicanas

Son regulaciones técnicas de aplicación voluntaria expedidas por Economía, las cuales prevén para un uso común, aplicables a un producto, proceso, actividad, servicio o método de producción u operación.


**NMX**  
Normas Mexicanas



**NOM**



En el caso de software, la NOM define los métodos de prueba y evaluación de software usados en alguna actividad, así como los instrumentos que se utilizarán para su evaluación.

# Referencias

- "ISO/IEC 2503n – División De Requisitos De Calidad.", iso25000.com/index.php/normas-iso-25000?limit=4&start=4.
- "Tecnología Investigación y Academia." *Revista Científica*, revistas.udistrital.edu.co/ojs/index.php/tia.
- <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>
- <https://iso25000.com/index.php/normas-iso-25000/iso-25040>
- <https://iso25000.com/index.php/en/iso-25000-standards/iso-25012>
- <http://www.economia-nmx.gob.mx/normasmx/consulta.nmx>
-