

#### Ingenieria de Software Aplicada

Noveno Semestre Ingenieria de Sistemas

**Docente: Sandra Marcela Guerrero** 



#### **UNIDAD 2. CALIDAD DE SOFTWARE**



- Conceptos Básicos
- Modelos de Calidad de software
- Estándares de calidad

## Conceptos básicos







## ¿Qué es Calidad del software?



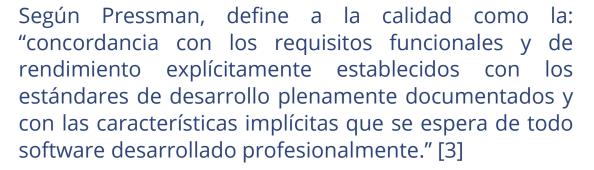
"La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad." [1].



"El término calidad de software se refiere al grado de desempeño de las principales características con las que debe cumplir un sistema computacional durante su ciclo de vida, dichas características de cierta manera garantizan que el cliente cuente con un sistema confiable, lo cual aumenta su satisfacción frente a la funcionalidad y eficiencia del sistema construido." [2]









Según IEEE, calidad es: "el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario"." [4]

• "La calidad en ingeniería del software es el cumplimiento de los requerimientos contractuales por parte del producto software desarrollado, así como durante el proceso de desarrollo. La calidad se obtiene mejorando día a día el proceso de producción, mantenimiento y gestión del software". [5]



La calidad del software puede referirse a: las características deseables de los productos de software, en la medida en que un producto de software en particular posea esas características, y los procesos, herramientas y técnicas utilizadas para lograr esas características. [6]



La calidad del software es medible y varía de un sistema a otro

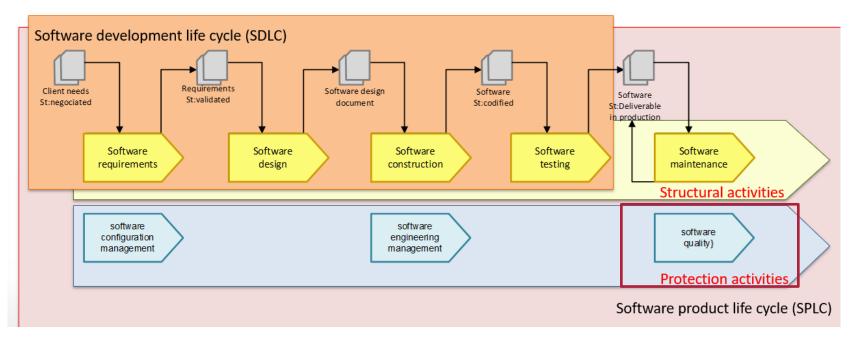






Software Cero Fallas

• La calidad como su control debe estar presente durante todas las etapas del ciclo de vida del software.



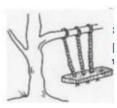
Créditos: Ph. Alexander Barón

### La importancia de los requisitos de software en los procesos de calidad

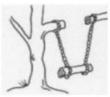
Esto es lo que pidió el usuario



El programador lo escribió así



El analista lo vio de esta forma



Esto es lo que quería el usuario

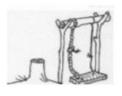


Los requisitos del software definen los atributos de calidad requeridos del software.

Así se diseñó el sistema



Así funciona el sistema en la actualidad

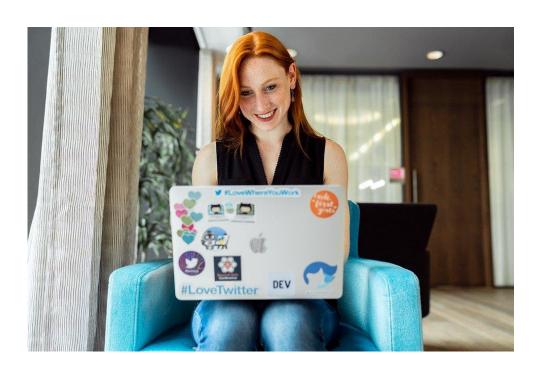


Extraído de: INGENIERÍA DE SOFTWARE I Tema 4: Ingeniería de Requisitos, Departamento de Informática y Automática Universidad de Salamanca

#### Mala calidad



- El producto de software no "satisface completamente las necesidades establecidas e implícitas"
- El producto software no cumple con los "requisitos establecidos



Comprendan conceptos de calidad, las características, los valores y su aplicación al software en desarrollo o mantenimiento.



Compartan compromisos a la calidad del software como parte de su cultura.



Deben poder utilizar métodos de Costos de Calidad del software para determinar los niveles de calidad del software.



Deben poder presentar alternativas de calidad y sus costos para que se puedan hacer compensaciones entre el costo, el cronograma y la entrega de valor para las partes interesadas.

#### ¿ Qué es el Triángulo de hierro?



#### Triangulo de hierro en la gestión de proyectos



- Todos los proyectos de software ágil tienen objetivos: qué es lo que el proyecto debe entregar, cuándo se debe entregar y con qué presupuesto.
- El triángulo de la gestión de proyectos representa el problema de la "triple restricción": la necesidad de equilibrar el alcance, el costo y el tiempo para mantener un producto final de alta calidad.
- Si el triángulo se rompe, es decir, si un punto se mueve sin modificar uno o ambos puntos restantes junto con él, la calidad del proyecto se verá afectada.

#### Costos de Evaluación de la Calidad

Costos de Evaluación	Descripción	Nivel
Prevención	Incluyen inversiones en esfuerzos de mejora de procesos de software, infraestructura de calidad, herramientas de calidad, capacitación, auditorías y revisiones de gestión.	Organizacional
Evaluación	Surgen de las actividades del proyecto que encuentran defectos. Estas actividades de evaluación se pueden clasificar en costos de revisión (diseño, pares) y costos de prueba (prueba de unidad de software, integración de software, prueba a nivel de sistema, prueba de aceptación)	Proyecto

#### Costos de Evaluación de la Calidad

Costos de Evaluación	Descripción	Nivel
Falla Interna	Se emplean para corregir los defectos encontrados durante las actividades de evaluación y descubiertos antes de la entrega del producto de software al cliente	
Falla Externa	Incluyen actividades para responder a problemas de software descubiertos después de la entrega al cliente.	Producto software después de la entrega al cliente

Fuente [6]

## Modelos de Calidad del Software



#### Modelos de Calidad del software

Un modelo de calidad debe ir enfocado a hacer seguimiento y evaluación a cada etapa de construcción del producto software.

"Los modelos de calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que

cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad." [7]



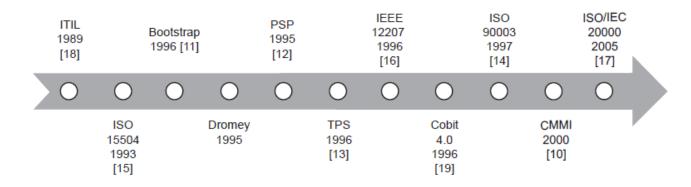
#### Modelos de Calidad del software

El modelo de calidad debe permitir evaluar el sistema, bien sea cualitativa o cuantitativamente, y de acuerdo con esta evaluación la organización podrá proponer e implementar estrategias que permitan la mejora del proceso dentro de las etapas de análisis, diseño, desarrollo y pruebas del software. [2]



#### Modelos de Calidad del software de Proceso

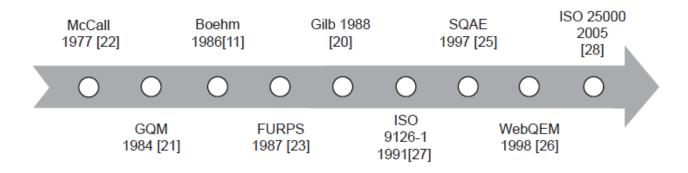
 La calidad de un sistema software debe ser programada desde el inicio del proyecto, y posteriormente en cada etapa del proceso de desarrollo se debe llevar a cabo el control y seguimiento de los aspectos de calidad. [2]



Fuente: [2]

#### Modelos de Calidad del software de Producto

 Permite especificar y evaluar el cumplimiento de criterios del producto, para lo cual se aplican medidas internas y/o medidas externas.



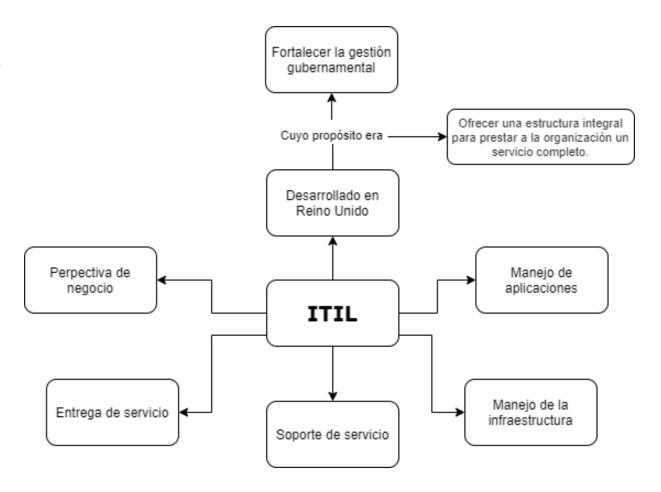
Fuente: [2]

# Actividad: Consultar sobre el Modelo de calidad asignado

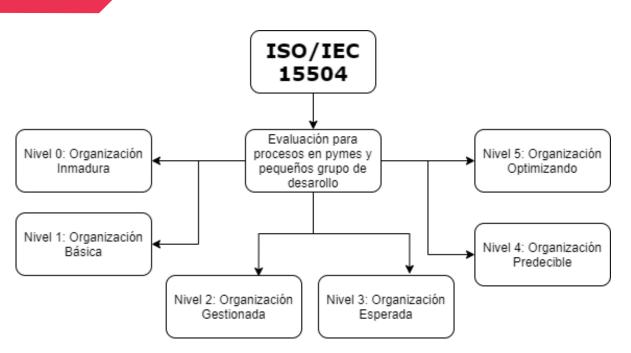
## Modelos de Calidad del Software







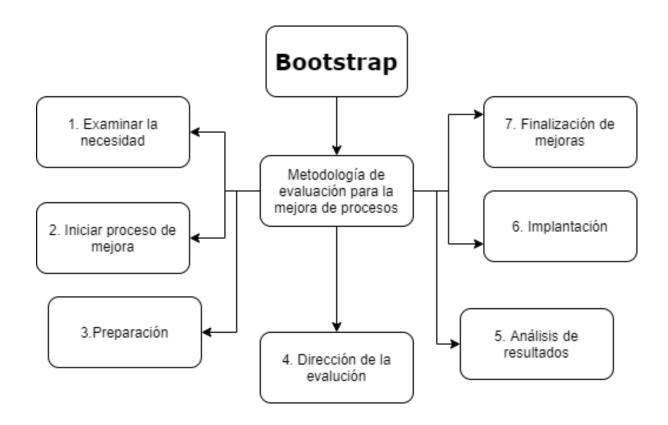
Elaboración propia basada en [2]







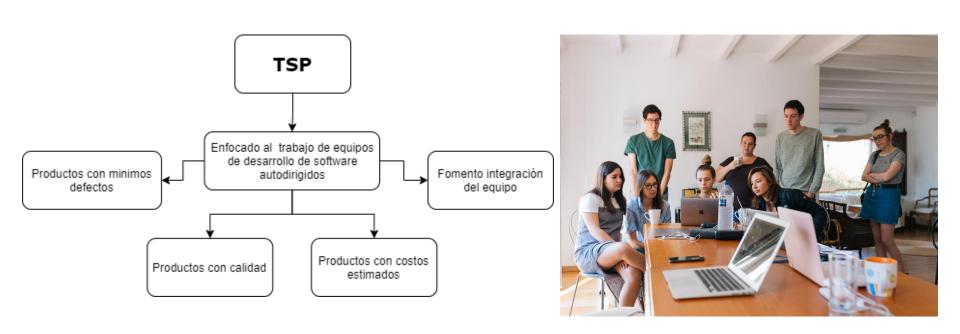
Elaboración propia basada en [2]



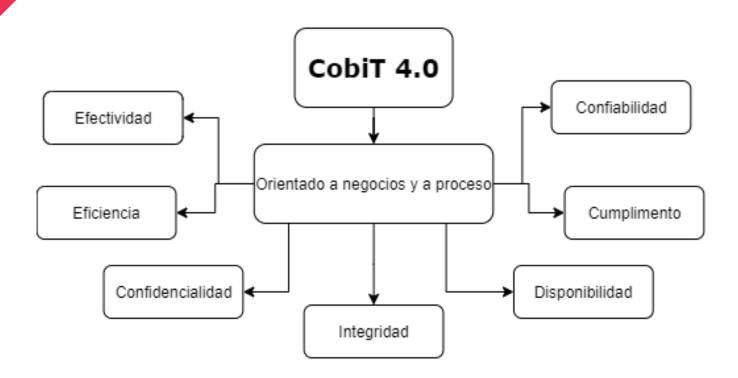
Elaboración propia basada en [2]

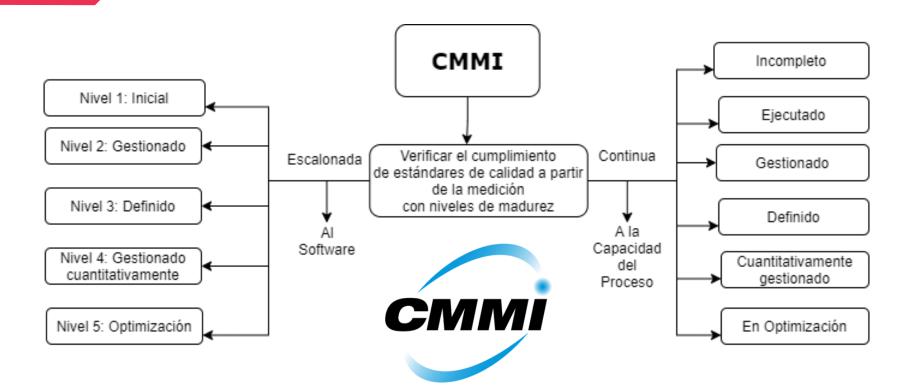


Elaboración propia basada en [2]



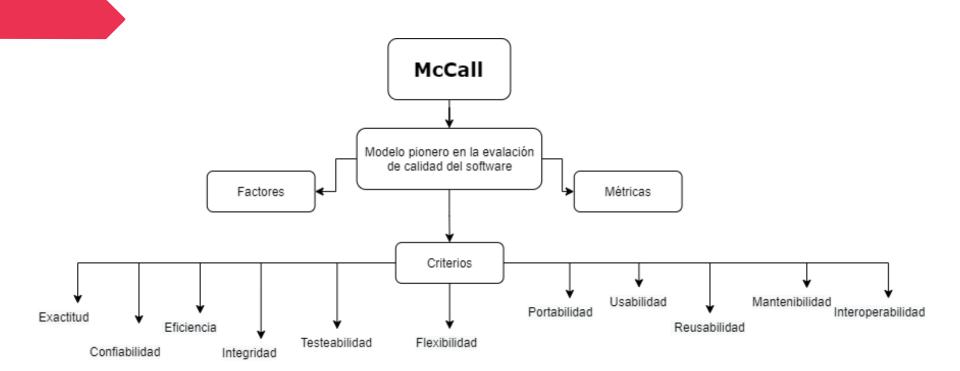
Elaboración propia basada en [2]

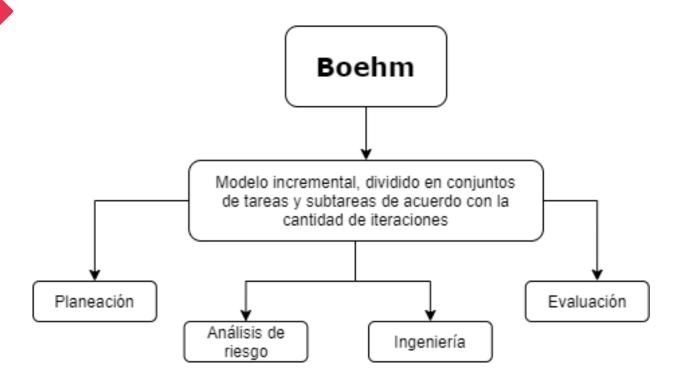


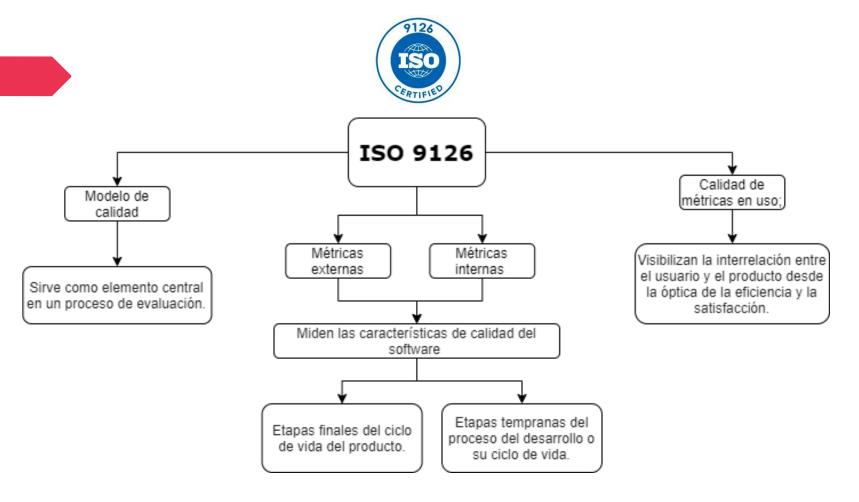


Elaboración propia basada en [2]

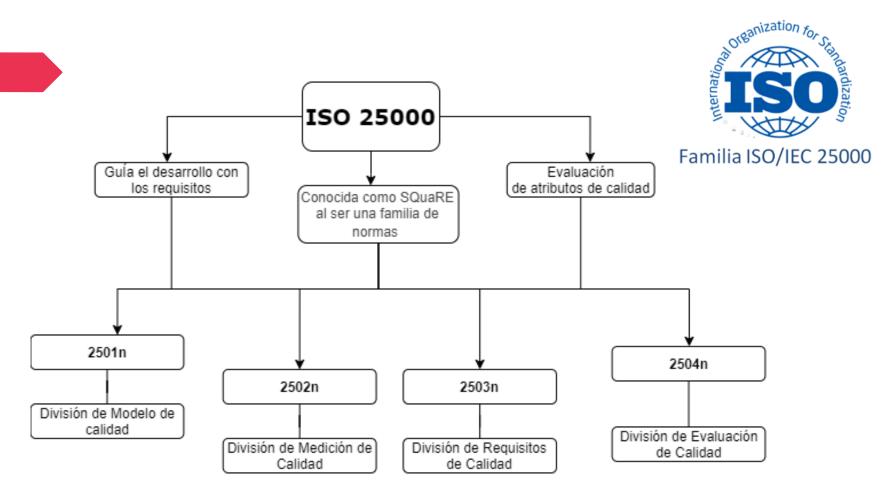








Elaboración propia basada en [2]



Elaboración propia basada en [2]

# Estándares de Calidad del Software



### **ISO 25000 SQUARE**

Fue desarrollado en el seno del proyecto SQUARE (Software Quality Requirements and Evaluation) el cual tiene como objetivo fundamental guiar el desarrollo de los productos de software con la especificación y evaluación de requisitos de calidad.



### ISO 25000

#### **Antecedentes:**

 ISO/IEC 9126: Evaluación de la calidad de productos de software descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

Características	Pregunta central	
Funcionalidad	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué ?	
Confiabilidad	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?	
Usabilidad	¿El software es fácil de usar y de aprender?	
Eficiencia	¿Es rápido y minimalista en cuanto al uso de recursos?	
Mantenibilidad	¿Es fácil de modificar y verificar?	
Portatilidad	¿Es fácil de transferir de un ambiente a otro?	

Tabla 1. Características de ISO-9126 y aspecto que atiende cada una.

Fuente [8]

### ISO 25000

#### **Antecedentes:**

 ISO/IEC 14598: Proporciona métodos para las mediciones y evaluaciones de la calidad de un producto de software. Entre las principales características del proceso de evaluación del proceso están: Repetitividad, Reproducibilidad, Imparcialidad, Objetividad.



# **ISO 25000 SQUARE**

SQUARE es básicamente una revisión de la norma ISO 9126 que hereda sus características, pero se centra más del lado del producto software contemplando tres fases principales en su ciclo de vida: producto bajo desarrollo, producto en operación y producto en uso.

El modelo de calidad SQUARE categoriza también la calidad del software en características, las cuales están subdivididas en subcaracterísticas y atributos de calidad.



### ISO/IEC 2501n — División de Modelo de Calidad

Las normas de este grupo presentan modelos de calidad detallados incluyendo características para calidad interna, externa y en uso del producto software. Actualmente esta división se encuentra formada por:

- **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.

# ISO 25010



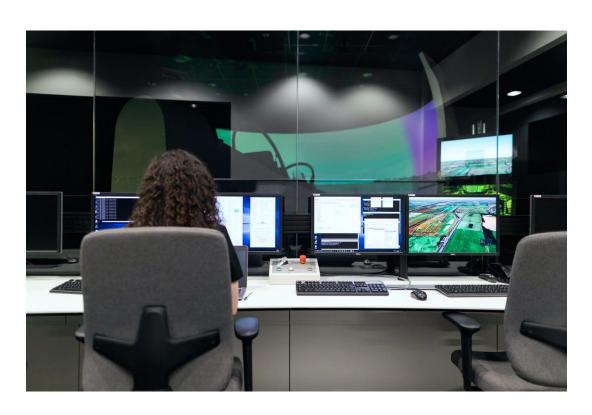
Fuente [9]

### Adecuación Funcional

Capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.

Subcaracteristica	Descripción
Completitud funcional.	Capacidad del sistema para proporcionar todas las funciones especificadas por el usuario.
Corrección funcional.	Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
Pertinencia funcional.	Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

### Adecuación Funcional

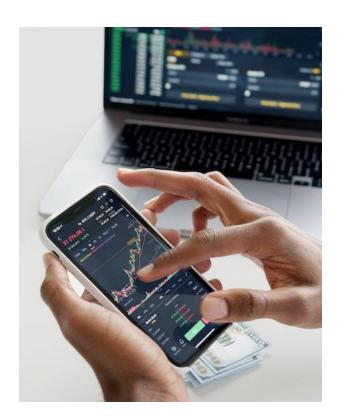


### Eficiencia de desempeño

Representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones.

Subcaracteristica	Descripción
Comportamiento temporal	Los tiempos de respuesta y procesamiento y el nivel de rendimiento de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas establecido.
Utilización de recursos	Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
Capacidad	Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

### Eficiencia de desempeño



"El sistema debe responder a una petición en un tiempo máximo de 5 segundos".

### 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.

Subcaracteristica	Descripción
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.

### Compatibilidad



#### **Ejemplo:**

• "El sistema debe utilizar el motor de base de datos PostgreSQL".



**Interoperabilidad.** Es la habilidad que debe tener el sistema para interactuar con uno o más sistemas.

### Usabilidad

 Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

Subcaracteristica	Descripción
Capacidad para reconocer su adecuación.	Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
Capacidad de aprendizaje.	Capacidad del producto que permite al usuario aprender su aplicación.
Capacidad para ser usado.	Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

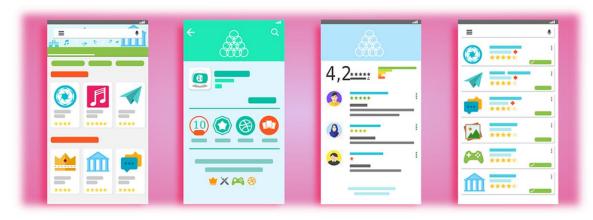
### Usabilidad

 Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

Subcaracteristica	Descripción
Protección contra errores de usuario.	Capacidad del sistema para proteger a los usuarios de hacer errores.
Estética de la interfaz de usuario.	Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
Accesibilidad.	Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

### Usabilidad





#### **Ejemplo:**

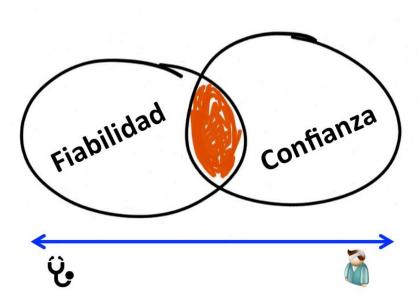
"El sistema debe permitir que los usuarios aprendan a utilizarlo en un tiempo máximo de 1 hora".

### Fiabilidad

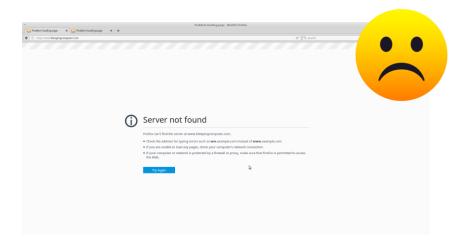
 Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados.

Subcaracteristica	Descripción
Madurez	Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
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.

### Fiabilidad



Tomado: https://saludconcosas.es/wp-ontent/uploads/2015/05/Diapositiva1-1.png



#### **Ejemplo:**

"El sistema debe estar disponible el 99,5 % de las veces que uno de los usuarios lo solicite". [4]

### Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.

Subcaracteristica	Descripción
Confidencialidad.	Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
Integridad.	Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
No repudio.	Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
Responsabilidad.	Capacidad de rastrear de forma inequívoca las acciones de una entidad.
Autenticidad.	Capacidad de demostrar la identidad de un sujeto o un recurso.

Fuente [9]

### Seguridad



#### **Ejemplo:**

"El sistema debe cerrar automáticamente la sesión de un usuario cuando esté inactivo por un tiempo igual o superior a 10 minutos"

### Mantenibilidad

• Representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

Subcaracteristica	Descripción	
Modularidad.	Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.	
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.	

### Mantenibilidad

 Representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

Subcaracteristica		1	Descripción
Capacidad modificado	para	ser	Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
Capacidad probado.	para	ser	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.

### Mantenibilidad



```
GregorianCalendar GregorianCalendar::NthXday(int n, int x, int month, int year, int day) {
        if (day == 0) {
            day = 1;
        GregorianCalendar calendar =
                GregorianCalendar(month, day, year);
                GregorianCalendar
                         ((7 * (n - 1)) + General::XdayOnOrBefore(6 + calendar, x));
    else {
        if (day == 0) {
                                                      Declared In: General.h
            day = LastMonthDay(month, year);
        GregorianCalendar calendar =
                                                      static int General::XdayOnOrBefore(int d, int x)
                GregorianCalendar(month, day, year
                                                      XdayOnOrBefore returns the date of the weekday before the given date.
        int date = (7 * (n + 1)) +
                General::XdayOnOrBefore(calendar.
        return GregorianCalendar(date);
                                                            d Absolute date.
                                                            x The weekday (0 through 6, with 0 meaning Sunday).
                                                      Returns
                                                            The date of the last weekday of type X on or before date D.
```

Fuente: https://www.jetbrains.com/help/clion/creating-and-viewing-doxygen-documentation.html

#### **Ejemplo:**

"El código fuente de la aplicación debe ser documentado con ayuda de la herramienta Doxygen"

### Portabilidad

 Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro.

Subcaracteristica	Descripción
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.
Capacidad para sei instalado	Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
Capacidad para ser reemplazado	Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

### **Portabilidad**







"El sistema debe poder ejecutarse bajo la plataforma de MacOS".

# REFERENCIAS



- 1. Fernández Carrasco, Oscar M, García León, Delba, & Beltrán Benavides, Alfa. (1995). Un enfoque actual sobre la calidad del software. ACIMED, 3(3), 40-42. Recuperado en 03 de agosto de 2022, de <a href="http://scielo.sld.cu/scielo.php?script=sci">http://scielo.sld.cu/scielo.php?script=sci</a> arttext&pid=S1024-94351995000300005&lng=es&tlng=es.
- 2. CALLEJAS-CUERVO, Mauro ; ALARCÓN-ALDANA, Andrea Catherine; ÁLVAREZ-CARREÑO, Ana María. Modelos de calidad del software, un estado del arte. En: Entramado. Enero Junio, 2017. vol. 13, no. 1, p. 236-250, http://dx.doi.org/10.18041/entramado.2017v13n1.25125
- رك. Ingeniería de software un enfoque practico. Roger Pressman, 2010
- 4. Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), 1990
- 5. Carrizo, Dante, & Alfaro, Andres. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. Ingeniare. Revista chilena de ingeniería, 26(1), 114-129. <a href="https://dx.doi.org/10.4067/S0718-33052018000100114">https://dx.doi.org/10.4067/S0718-33052018000100114</a>
- **6.** Swebok Guide, Version 3.0
- /. SCALONE. Fernanda. Estudio comparativo de los modelos y estándares de calidad del software. Tesis Ingeniería de Calidad. Buenos Aires: Universidad Tecnológica Nacional Regional de Buenos Aires. 2006. 488 p.
- 8. Calidad en la Industria del Software. La Norma ISO-9126, María Antonieta Abud Figueroa. Online, Available <a href="https://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf">https://www.nacionmulticultural.unam.mx/empresasindigenas/docs/2094.pdf</a>
- 9. ISO 25000. <a href="https://iso25000.com/index.php/normas-iso-25000">https://iso25000.com/index.php/normas-iso-25000</a>

# Gracias

Ingeniería de Software Aplicada Programa de Ingeniería de Sistemas Universidad de Nariño Noveno Semestre

**CREDITS**: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik** 

