

**UNIVERSIDAD TECNOLÓGICA DE**

**SAN LUIS RIO COLORADO**

**Practice 3**

**MTRO. RICARDO ALEJANDRO SOTO MORALES**

**ALUMNO: VICTOR MANUEL GALVAN COVARRUBIAS**

**FACTOR DE CALIDAD NO. 7**

**CALIDAD EN USO**

[**VIDEO**](https://youtu.be/AfT0PYbgMXk)

San Luis Rio Colorado, Sonora Febrero, 2021

**1. FUNCIONALIDAD**

Cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.

**a. Adecuación.**

Atributos del software relacionados con la presencia y aptitud de un conjunto de funciones para tareas específicas.

Completitud funcional. Revisa el grado de las funcionalidades al cumplir con las tareas.

Corresión funcional. Capacidad que tiene para proporcionar elementos.

**b. Exactitud:**

Evaluar el software para saber si cumple las necesidades por las que fue creado.

**c. Interoperabilidad**

Capacidad de comunicar, ejecutar programas, o transferir datos entre varias unidades.

Semántica. Los datos que se intercambian son comprensibles para los dos o mas sistemas.

Tecnica. Los diferentes componentes de software cooperan incluso cuando la interfaz y el lenguaje de programación no sea el mismo

Maneja tres niveles:

Vertical. Comunicación jerárquica.

Horizontal. En el mismo nivel.

Longitudinal. El ciclo de vida de la comunicación.

**d. Seguridad**

La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados.

**e. Conformidad de la funcionalidad**

La capacidad del software de cumplir los estándares referentes a la funcionalidad. Siendo estos adecuación, exactitud e interoperabilidad. Ajustándose a los es tandeares convenciones y regulaciones. Estará completa una vez cumpla con las tareas.

**2. FIABILIDAD**

La capacidad del producto de software para mantener un nivel especifico de rendimiento cuando se utiliza en condiciones específicas. Un software es confiable cuando realiza lo que el usuario desea.

**a. Madurez**

El punto hasta cual un determinado proceso es explícitamente definido, administrado, medido, controlado y efectivo.

Un software maduro es definido, documentado, entrenado, practicado, soportado, mantenido, controlado, verificado, validado, medido y capaz de mejorar.

**b. Tolerancia a errores**

Garantizar que el sistema continúe funcionando correctamente, incluso en la presencia de fallas.

**c. Recuperabilidad**

Capacidad del software para encontrarse en un estado apto para desempeñar una función requerida.

**d. Conformidad de la fiabilidad**

Señales de que un software es confiable:

Muestra errores

Es capaz de evitar fallos

Evita operaciones incorrectas

Tiene un tiempo medido de recuperación

Tiene la habilidad de reiniciar

Es capaz de evitar errores

Es capaz de restaurarse

Es robusto

**3. USABILIDAD**

Ser entendido, aprendido, y usado en forma fácil y atractiva. Está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido.

**a. Entendimiento**

La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.

**b. Aprendizaje**

La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.

**c. Operabilidad**

La manera como el software permite al usuario operarlo y controlarlo.

**d. Atracción**

La presentación del software debe ser atractiva al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.

**e. Conformidad de uso**

La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

**4. EFICIENCIA**

La forma del desempeño adecuado, de acuerdo a al número recursos utilizados según las condiciones planteadas. Se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo, entre otros.

**a. Comportamiento de tiempos**

Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas. Ejemplo, ejecutar el procedimiento más complejo del software y esperar su tiempo de respuesta, realizar la misma función, pero con más cantidad de registros.

**b. Utilización de recursos**

La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas. Ejemplo, los recursos humanos, el hardware, dispositivos externos.

**c. Conformidad de eficiencia**

La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

**5. MANTENIBILDAD**

La cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

**a. Capacidad de ser analizado**

La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.

**b. Cambiabilidad**

La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.

**c. Estabilidad**

La forma como el software evita efectos inesperados para modificaciones del mismo.

**d. Facilidad de prueba**

La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

**e. Conformidad de facilidad de mantenimiento**

La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

**6. PORTABILIDAD**

Ser trasladado de un entorno a otro.

**a. Adaptabilidad**

Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

**b. Facilidad de instalación**

La facilidad del software para ser instalado en un entorno específico o por el usuario final.

**c. Coexistencia**

La capacidad que tiene el software para coexistir con otro o varios softwares, la forma de compartir recursos comunes con otro software o dispositivo.

**d. Reemplazabilidad**

La capacidad que tiene el software para ser remplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la remplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.

**e. Conformidad de portabilidad**

La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

**7. CALIDAD EN USO**

Calidad del software que el usuario final refleja, la forma como el usuario realiza los procesos con satisfacción, eficiencia y exactitud.

**a. Eficacia**

La capacidad del software para permitir a los usuarios finales realizar los procesos con exactitud e integridad.

**b. Productividad**

La relación entre la cantidad de productos obtenida por un sistema productivo y los recursos como tiempo, esfuerzo, materiales y costos, utilizados para obtener dicha producción.

**c. Seguridad**

Seguridad: grado en el que puede controlarse el acceso al software y a los datos o capacidad para alcanzar niveles aceptables de riesgo de daño para las personas, software, equipos o entorno en un contexto especificado de uso.

**d. Satisfacción**Esta característica se refiere a la capacidad del producto de software para satisfacer a los usuarios en un marco de trabajo particular, entendiendo por satisfacción como una respuesta de los usuarios a la interacción con el producto, incluyendo las actitudes producidas frente a su uso.