

CARRERA DE COMPUTACIÓN

1. Datos Informativos

- 1.1. Módulo:**
- 1.2. Nivel:** Séptimo
- 1.3. Apellidos y Nombres:** Samira Narváez, Estiven Méndez, Almer
Paguay
- 1.4. Tema:** Factores de Calidad de McCall – Métricas de Software
- 1.5. Fecha:** 26-11-2025

2. Objetivo

Comprender y aplicar los factores de calidad de McCall y las métricas de calidad de software en la evaluación de un producto de software.

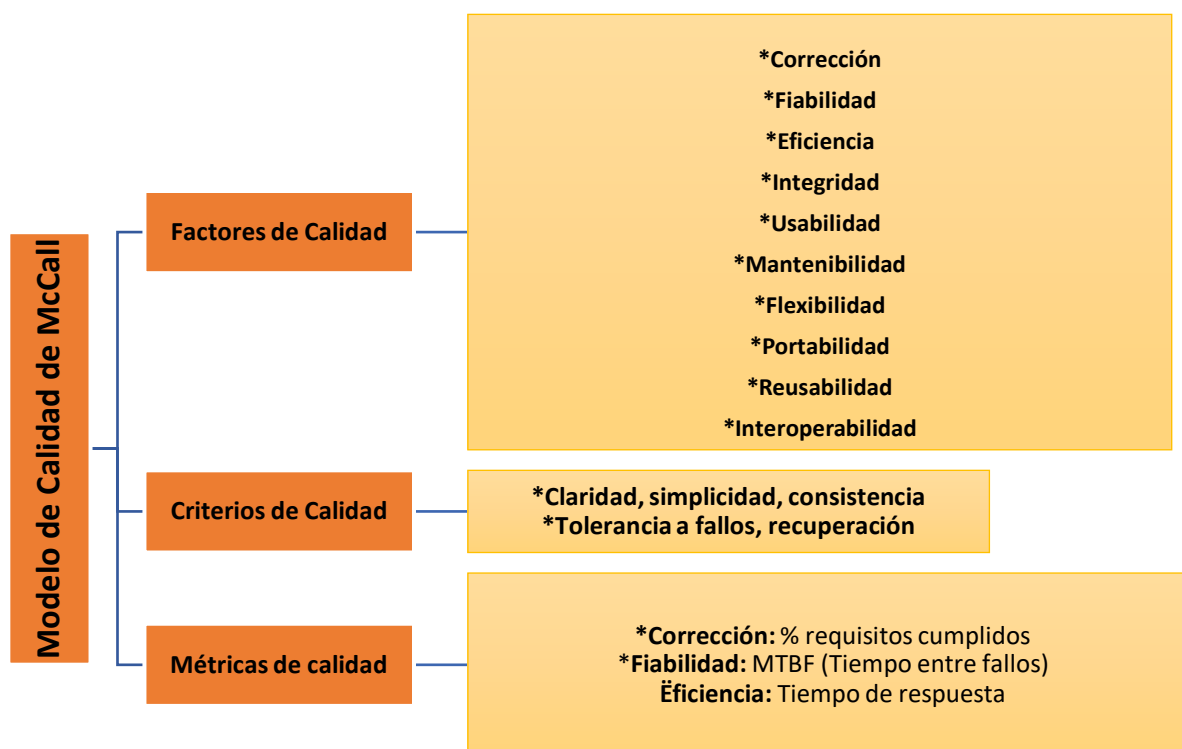
3. Contenido

1. Introducción

Nos referimos a la calidad del software cuando hablamos de qué tan bien un programa satisface lo que promete y qué tan satisfactorio es para sus usuarios. Un programa de buena calidad no solo sirve; además, es comprensible, confiable, seguro y tiene la capacidad de adaptarse cuando lo requiere.

La calidad es relevante debido a que la gente depende cada vez más de la tecnología para trabajar, estudiar, comunicarse y solucionar problemas cotidianos. Un software que falla, es poco claro o no satisface las expectativas puede causar frustración, pérdida de tiempo e incluso peligros. Por lo tanto, es esencial garantizar que los programas que creamos realmente proporcionen una experiencia positiva y atiendan las necesidades verdaderas de los usuarios.

2. Modelo de Calidad de McCall



3. Actividad:

Factores de McCall aplicados al proyecto: Recetas Caseras en Línea

➤ Usabilidad

¿Cómo se aplica al proyecto?

La plataforma tiene que ser accesible para cualquier individuo, incluyendo aquellos no muy versados en tecnología. Esto implica contar con una interfaz clara, botones visibles, recetas agrupadas en categorías y un buscador fácil de usar.

Importancia

Si la aplicación resulta compleja, los usuarios se sienten frustrados y dejan de usarla. Por otra parte, una usabilidad óptima permite que los usuarios gocen explorando recetas, almacenándolas y regresando a la plataforma. Es fundamental para asegurar el éxito del proyecto y su accesibilidad para todos.

➤ **Fiabilidad**

¿Cómo se aplica al proyecto?

La aplicación tiene que cargar rápidamente, no tener fallos y mostrar las recetas correctamente en todo momento. Es igualmente importante que no cierre de manera imprevista ni muestre errores al abrir una receta.

Importancia

Cuando una persona está cocinando, necesita tener confianza en la información que tiene. Si la plataforma se cae justo en medio de la preparación de un platillo, se arruina la experiencia. La confiabilidad asegura que los usuarios puedan utilizar la aplicación sin preocupaciones.

➤ **Eficiencia**

¿Cómo se aplica al proyecto?

Como muchos usuarios ingresan desde teléfonos celulares, es importante que el sistema cargue las recetas rápidamente, utilice eficientemente los recursos del dispositivo y reduzca al mínimo el uso de datos.

Importancia

En la actualidad, a nadie le gusta esperar para consultar una receta o usar una aplicación que consuma muchos datos o batería. Una plataforma eficaz optimiza la experiencia del usuario, aumentando la probabilidad de que vuelva y sugiera el sitio.

4. Métricas de calidad de software

Las medidas de calidad son maneras de evaluar el estado de un programa informático. Son como señales que indican si el sistema es estable, rápido, si funciona correctamente o si requiere mejoras.

Podemos tomar decisiones concretas y no solamente "suponer" que el software es bueno gracias a estas métricas.

Tipos de Métricas de calidad

A) Métricas Internas

- Son las que se aplican al código sin necesidad de ejecutar el software.
- Se usan para revisar la estructura interna, organización del código y complejidad.

Ejemplos:

- Complejidad ciclomática: mide qué tan difícil es entender o probar el código.
- Líneas de código: ayuda a ver el tamaño del sistema.
- Cantidad de funciones o módulos: indica qué tan dividido está el software.

En pocas palabras: ayudan a saber si el software es fácil de mantener.

B) Métricas Externas

- Se aplican cuando el software está funcionando.
- Permiten evaluar el comportamiento real que ve el usuario.

Ejemplos:

- Tiempo de respuesta: cuánto tarda en cargar una receta o una página.
- Consumo de memoria: cuánta RAM necesita para funcionar bien.
- Tasa de fallos: cuántos errores aparecen cuando la gente lo usa.

En pocas palabras: permiten saber cómo "se siente" el software cuando alguien lo utiliza.

C) Métricas de calidad en uso

- Miden la experiencia final del usuario.
- Evalúan qué tan útil, fácil y agradable es el software.

Ejemplos:

- Errores cometidos por los usuarios: si la interfaz confunde o guía bien.
- Satisfacción del usuario: encuestas o comentarios positivos/negativos.
- Eficacia: si el usuario logra lo que busca sin perder tiempo.

En pocas palabras: nos muestran si el software realmente ayuda a las personas.

Ejemplo de como medir la calidad

- **Porcentaje de requisitos cumplidos (Corrección)**

¿Qué mide?

Qué tanto del proyecto se logró correctamente.

Ejemplo:

El proyecto tiene 20 requisitos.

Si 18 están funcionando bien:

$$(18/20) \times 100 = 90\% \quad (18 / 20) \times 100 = 90\%$$

Entre más cerca del 100%, mejor.

Actividad Corta:

Caso propuesto:

Una app llamada “Mi Cocina Fácil”, donde los usuarios buscan recetas, guardan favoritas y ven los pasos para cocinar.

Actividad:

El grupo debe elegir qué métricas usaría para evaluar la app y explicar brevemente por qué.

Ejemplos de métricas (guía):

- **Tiempo de respuesta:** qué tan rápido carga una receta.
- **Tasa de fallos:** cuántas veces la app se cierra o muestra errores.
- **Satisfacción del usuario:** calificaciones o comentarios de quienes usan la app.

La idea es elegir métricas que ayuden a saber si la app es rápida, confiable y agradable para los usuarios.

5. Caso Práctico Integrado

Planteamiento del caso

Estamos evaluando la calidad de la aplicación de nuestro proyecto:

“Recetas Caseras en Línea”, una app donde los usuarios buscan recetas, guardas favoritas y cocinan paso a paso desde su celular.

1. Cinco factores de calidad clave y su métrica correspondiente

➤ Usabilidad

Métrica: número de errores cometidos por el usuario al navegar (por ejemplo, clics equivocados).

Tiene sentido porque queremos que cualquier persona, incluso sin experiencia tecnológica, pueda usar la app sin confundirse.

➤ Fiabilidad

Métrica: tasa de fallos (cuántas veces la app se cierra o muestra errores).

Es esencial que la app no falle en pleno proceso de una receta.

➤ Eficiencia

Métrica: tiempo de respuesta (segundos que tarda en cargar una receta o categoría).

Si las recetas tardan en abrirse, el usuario pierde paciencia y se va.

➤ Corrección

Métrica: porcentaje de requisitos cumplidos correctamente.

Nos ayuda a saber si lo que prometimos realmente está funcionando.

➤ Satisfacción del usuario (Calidad en uso)

Métrica: calificación promedio (estrellas o puntuación que dejan los usuarios).

Es la forma más directa de saber si la app les gusta o no.

2. ¿Qué factor es más difícil de medir y por qué?

El factor más difícil de medir suele ser la Usabilidad.

¿Por qué?

Porque depende de personas, y cada persona usa la app de manera diferente.

- Lo que es fácil para un usuario, puede ser complicado para otro.
- Requiere pruebas con personas reales, observar su comportamiento, analizar errores, tiempos y frustraciones.

No se puede medir solo con números; también importa la experiencia y las emociones.

En pocas palabras: medir cómo se siente un usuario usando la app es más complejo que medir un tiempo de carga o un fallo.

4. Conclusiones

Es fundamental medir la calidad del software, ya que esto nos posibilita observar más allá de lo que "pensamos" que funciona. Podemos comprender qué elementos de la aplicación realmente benefician al usuario y cuáles requieren mejoras gracias a las métricas y a los factores de calidad. En lo que respecta a nuestra aplicación de recetas caseras en línea, medirnos posibilita garantizar que sea confiable, segura, rápida y sencilla de manejar. En resumen, la medición es fundamental para mejorar, brindar una mejor experiencia y desarrollar un software que realmente acompañe a las personas en su vida cotidiana.

5. Referencia Bibliografía

- Galin, D. (2004). *Software Quality Assurance: From Theory to Implementation*. Pearson Education.
- Florac, W. A., & Carleton, A. D. (1999). *Measuring the Software Process: Statistical Process Control for Software Process Improvement*. Addison-Wesley.

6. Referencia linkografía