



1. Datos Informativos

- 1.1. Módulo: 3
- 1.2. Nivel: Séptimo
- 1.3. Apellidos y Nombres: Aguas Buitron Katty Dalila

Benavides Muñoz Angelo Alexander

Chalacán Almeida Edwin Wladimir

- 1.4. Tema: Taller: Taller McCall
- 1.5. Fecha: 19/11/2025

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

Introducción

- ¿Qué entienden por calidad de software?

La calidad de software se refiere a cuán bien un sistema o producto satisface los requerimientos establecidos por el cliente y las demandas del usuario final. No se limita a "funcionar"; incluye ser seguro, eficiente, confiable, fácil de manejar y de mantener. Para nuestro proyecto, una aplicación de comercio electrónico de primera categoría significa que los usuarios pueden explorar, adquirir y administrar su cuenta sin inconvenientes, mientras que el equipo de desarrollo puede realizar actualizaciones y corregir fallos con facilidad (Miranda Realpe, 2025).

- ¿Por qué creen que es importante?

Es importante porque:

Aumenta la satisfacción del cliente: Si la aplicación es rápida, segura y accesible, los clientes la utilizarán nuevamente.

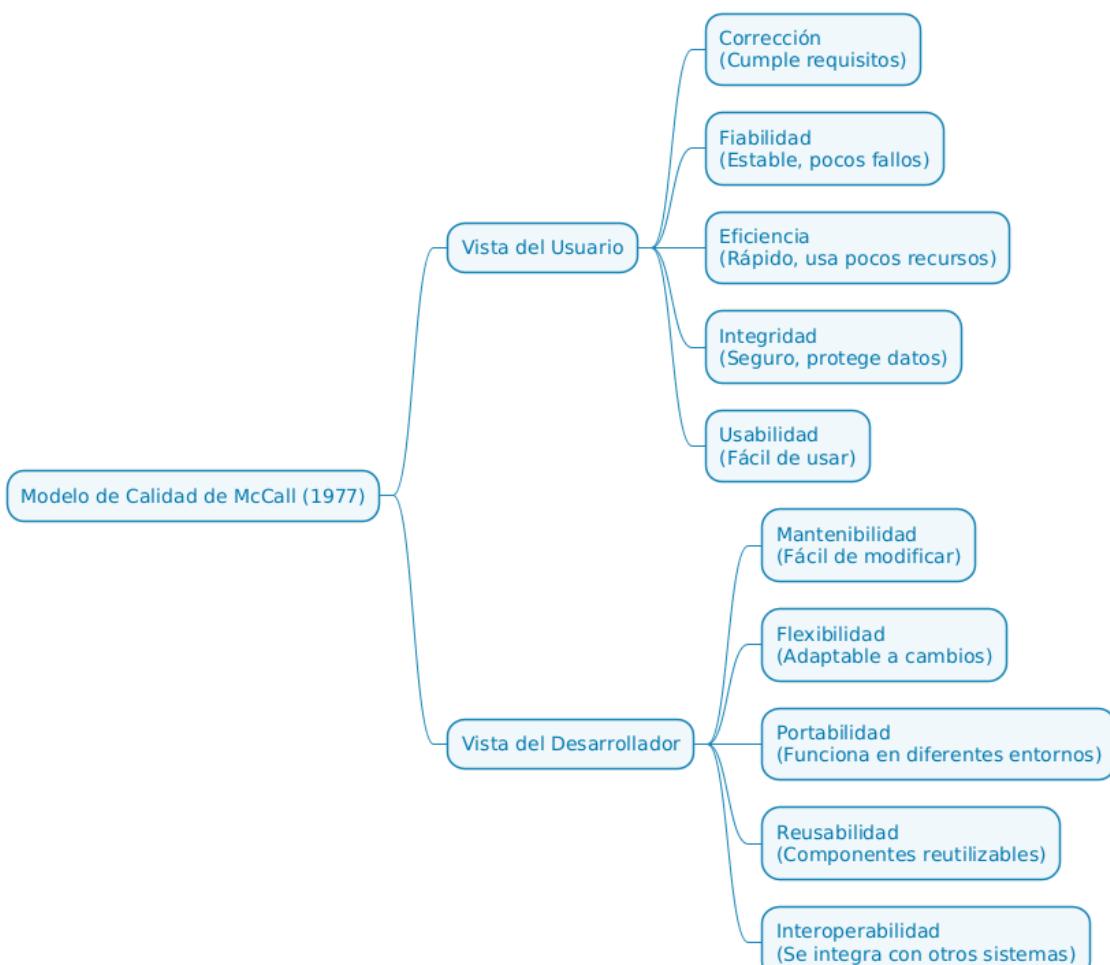
Disminuye costos a largo plazo: Un software de buena calidad presenta menos fallos, lo que se traduce en un menor gasto en soporte y soluciones.

Fortalece la competitividad: En un entorno lleno de tiendas en línea, contar con una experiencia del usuario sobresaliente es crucial para destacar.

Facilita el mantenimiento y la adaptación: Un código bien organizado permite incorporar nuevas características (como métodos de pago adicionales o informes) sin afectar lo que ya funciona.

Modelo de Calidad de McCall

Mapa Conceptual sobre el modelo



Actividad

Proyecto: E-commerce Web App (con módulos de autenticación, catálogo, carrito, checkout y panel de administración).

- Elegir 3 factores de McCall

Las métricas que se han elegido para el proyecto son las siguientes:

- Usabilidad
- Fiabilidad
- Mantenibilidad

- ¿Como se aplicaría al proyecto escogido?

- La interfaz debe ser fácil de usar para que cualquier persona, desde un niño hasta un anciano, pueda explorar, buscar artículos, añadirlos al carrito y finalizar la compra sin asistencia. El panel administrativo también debe ser simple para que el vendedor pueda manejar pedidos y productos sin complicaciones.
- La aplicación debe estar operativa las 24 horas del día, los 7 días de la semana, y no debe experimentar fallos en momentos críticos como el proceso de pago.
- El código necesita estar adecuadamente diseñado y segmentado (por ejemplo, separando el módulo que maneja la autenticación del que se encarga de los pagos). Esto facilitará la rápida solución de errores (como problemas en el sistema de pago) o la adición de nuevas características (como la integración con PayPal) sin interferir con otros componentes del sistema.

- ¿cuál sería su importancia?

Es fundamental para la conversión de ventas debido a que, si nuestros potenciales clientes se sienten frustrados, podrían tomar la decisión de abandonar el proceso de compra.

También debemos recalcar que si tomamos en cuenta estos tres factores vamos a asegurar la confianza del cliente y de esta manera se previene pérdidas económicas debido a transacciones que no se concretan.

Disminuye el tiempo y el costo de las actualizaciones y soluciones, algo esencial para un negocio en expansión.

Métricas de Calidad de Software

- **Mostrar ejemplos de cómo medir la calidad (puedes usar ejemplos simples como tasa de fallos o tiempo de respuesta).**

Las métricas son criterios objetivos que nos van a ayudar a juzgar la calidad del software que se está diseñando para ello proponemos el uso de estas métricas que creemos necesarias para la evaluación de este:

Métricas Internas: Se cuantifican en el código fuente (antes de su ejecución).

- **Ejemplo:** Para este tipo de métricas se podría comenzar con la cantidad de líneas de código por función, complejidad ciclomática, cobertura de pruebas unitarias.

Métricas Externas: Se evalúan mientras se está ejecutando el software.

- **Ejemplo:** En este aspecto tenemos que realizar la comprobación del retardo en la respuesta de la página y de igual manera verificación de la cantidad de fallos por hora que se generen, uso de memoria/CPU.

Métricas de Calidad en Uso: Se analizan desde la óptica del usuario final.

- **Ejemplo:** como esta métrica esta netamente ligada al usuario final el análisis y evaluación se centraría en el tiempo medio utilizado para finalizar una compra, de igual manera se podría emplear un análisis del porcentaje de abandono del carrito, y finalmente la evaluación en cuanto a la satisfacción del usuario (a través de encuestas).

Actividad corta:

- **Proponer un caso (una app del proyecto escogido) y que los grupos propongan qué métricas usarían para evaluarla.**

Evaluamos la funcionalidad de “Página de Detalle de Producto” en la app LexBen.

¿Por qué es relevante este caso?

Se trata de una de las secciones más esenciales para aumentar la tasa de conversión. El visitante accede a este punto tras realizar una búsqueda o explorar categorías, y aquí es donde toma la decisión de realizar una compra. La efectividad de esta página influye de manera directa en las ventas.

Objetivo	Métrica	Como medir	Meta
Flexibilidad	Porcentaje de errores al cargar la imagen principal del artículo.	Registros del servidor que documentan errores 404 o 500 al intentar mostrar la imagen.	0% (la imagen siempre debe cargar sin fallos).
Usabilidad	Promedio de tiempo que un visitante permanece en la página del producto antes de hacer clic en "Comprar en Amazon" o "Consultar a LexBen Agent".	Utilizando herramientas de análisis web como Google Analytics o Hotjar, se cuantifica el tiempo desde que el usuario abre la página hasta que ejecuta una acción.	Menos de 45 segundos (lo que indica que la información es fácilmente comprensible y el proceso es sencillo).
Integridad	Cantidad de intentos de acceso no autorizado a la API que proporciona los datos del producto.	Supervisión de registros del backend.	0 intentos exitosos.
Eficiencia	Tiempo requerido para cargar la página del producto (Largest Contentful Paint - LCP)	Se evalúa con Lighthouse (una herramienta de Chrome DevTools).	Menos de 2.5 segundos para asegurar una experiencia de usuario fluida.

Caso Práctico Integrado

a) Planteamiento del caso:

- "Evaluamos la app del proyecto que se escogió".

"Analizamos la aplicación LexBen, un asistente digital enfocado en productos, diseñado para ayudar a los usuarios a seleccionar artículos de acuerdo con sus gustos y necesidades. La herramienta integra un chatbot avanzado (Agente LexBen) con una plataforma web que presenta descripciones de productos, críticas y enlaces directos a compras como Amazon.

Dado que LexBen funciona en un entorno de negocio y de interacción directa con el usuario final, es esencial asegurar que se mantenga altos niveles de calidad en diversas áreas: debe ser confiable en sus sugerencias, veloz en la carga de información, seguro en el tratamiento de las interacciones del usuario. De igual forma este debe ser accesible para individuos sin habilidades técnicas y fácil de mantener para permitir actualizaciones frecuentes de productos o de los algoritmos del chatbot.

Por lo tanto, utilizaremos el modelo de calidad de McCall para identificar los factores más importantes en esta situación y establecer métricas específicas que nos ayuden a medir, de manera objetiva y continua, la calidad del software proporcionado."

b) Se tiene:

- Elegir 5 factores de calidad que consideran clave.

Los 5 factores que hemos escogido son:

- Usabilidad
- Fiabilidad
- Eficiencia
- Integridad
- Mantenibilidad

- Definir una métrica específica para medir cada uno.

Las métricas seleccionadas para poder medir cada factor elegido son las siguientes.

Factor	Métrica	Justificación
Usabilidad	% de rebote en la página de detalles del producto. Proporción de visitantes que acceden a la página de detalles, pero se van sin realizar ninguna acción (ni comprar ni hacer preguntas)	Un porcentaje elevado sugiere deficiencias en la usabilidad (información poco clara, diseño no atractivo, etc.).
Fiabilidad	MTBF (Tiempo Medio Entre Fallos) en la integración con Amazon. Intervalo de tiempo promedio entre fallos en el botón "Comprar en Amazon".	Si ocurre un fallo, el usuario pierde la posibilidad de realizar una compra, lo que impacta directamente en las ventas.
Eficiencia	Consumo medio de ancho de banda por sesión en la página de categorías. Esto es lo que la aplicación utiliza al presentar varios productos.	Un uso excesivo puede hacer que la navegación sea más lenta en conexiones de baja velocidad, perjudicando la experiencia del usuario.
Integridad	Cantidad de vulnerabilidades de seguridad encontradas en el código del chatbot "LexBen Agent".	Dado que el chatbot tiene la capacidad de recopilar información del usuario (preguntas, posibles intereses), garantizar su seguridad es esencial. Esto se evalúa mediante herramientas de análisis estático (SAST).
Mantenibilidad	Promedio de líneas de código alteradas por cada defecto solucionado en el módulo de búsqueda. Esto refleja la complejidad del código.	Si se requieren muchas líneas para corregir un pequeño fallo, queda demostrado que el código es complicado de mantener. Meta: menor de 50 líneas por defecto.

c) ¿Qué factor consideran más difícil de medir en un software? ¿Por qué?

A nuestra consideración el elemento más complicado de evaluar es la usabilidad porque por naturaleza es subjetiva. Aunque existen métricas indirectas que podemos observar, como la tasa de abandono o el tiempo que pasa el usuario en una página, estos datos no aclaran las razones del abandono. Tal vez la página se cargó rápidamente, pero el contenido era poco claro, o el botón para comprar no se encontraba a la vista. Para profundizar en esto, se requieren enfoques cualitativos, tales como:

Análisis de cómo utilizan la aplicación, cuestionarios de satisfacción del cliente: preguntar directamente qué aspectos les agradaron y cuales no, identificar dónde dirigen su atención y dónde encuentran problemas.

Estos métodos suelen ser considerablemente más lentos y difíciles de sintetizar a diferencia de métricas como la fiabilidad, eficiencia, que pueden ser medidas de manera automatizada y repetitiva.

4. Conclusiones

La calidad del software no es un elemento opcional, más bien es uno de los pilares esenciales en la creación de productos tecnológicos. Durante este taller, aplicar el modelo de calidad de McCall a la aplicación LexBen, se puede identificar de forma organizada cuáles dimensiones de calidad más cruciales para facilitar la interacción de cualquier usuario y salvaguardar datos que permitan generar confianza en el cliente, así se asegura la estabilidad del sistema y el crecimiento futuro.

Además, al establecer métricas concretas y cuantificables para los aspectos a evaluar, como tiempo de carga de la página, tasa de errores en la integración con Amazon, cobertura de pruebas del chatbot, se puede transformar conceptos indeterminados en parámetros tangibles fáciles de monitorear, comparar y mejorar de forma continua.

Por último, el hecho de que algunos aspectos, como usabilidad, sean más complicados de medir nos enseña que la calidad del software necesita una dirección equilibrada. En iniciativas como LexBen, la experiencia del cliente determina el valor del software, es por ello que el uso de métricas de calidad es fundamental para crear un producto funcional y valioso.

5. Referencia Bibliografía

Miranda Realpe, J. H. (2025). *Factores de Calidad de McCall y Métricas de Calidad de Software*. Obtenido de ¿Qué es la Calidad de Software?

6. Referencia Linkografía