

1. Investigación de Modelos de Calidad

A continuación, se describen tres de los modelos más reconocidos en la industria del software para evaluar la calidad del producto.

A) Modelo ISO/IEC 25010

Es el estándar internacional actual para la evaluación de la calidad del software, sustituyendo a la norma ISO 9126. Se divide en dos modelos principales: Calidad del Producto y Calidad en Uso.

- **Características principales:** Define 8 características de calidad del producto: Adecuación funcional, Eficiencia de desempeño, Compatibilidad, Usabilidad, Fiabilidad, Seguridad, Mantenibilidad y Portabilidad.
- **Ventajas:** Es el más completo y actual. Incluye explícitamente la **Accesibilidad** (dentro de Usabilidad) y la **Compatibilidad**, aspectos vitales en el desarrollo moderno de apps.
- **Desventajas:** Puede ser complejo de implementar en su totalidad para proyectos muy pequeños debido a la gran cantidad de sub-características.
- **Áreas de aplicación:** Todo tipo de software moderno, especialmente aplicaciones móviles, web y sistemas críticos.

B) Modelo de McCall (1977)

Fue uno de los primeros modelos de calidad y se centra en tres perspectivas principales: Operación del producto, Revisión del producto y Transición del producto.

- **Características principales:** Descompone la calidad en 11 factores, como Corrección, Fiabilidad, Eficiencia, Facilidad de prueba, Flexibilidad y Portabilidad.
- **Ventajas:** Se enfoca mucho en la visión del usuario final y en la capacidad del sistema para adaptarse a cambios (mantenimiento).
- **Desventajas:** Es un modelo antiguo; algunos conceptos no se adaptan perfectamente a las arquitecturas modernas de aplicaciones móviles distribuidas.
- **Áreas de aplicación:** Sistemas militares, espaciales y de misión crítica donde la robustez es prioritaria.

C) Modelo de Boehm (1978)

Similar al modelo de McCall, pero añade una estructura jerárquica con énfasis en la mantenibilidad del software.

- **Características principales:** Se basa en niveles: utilidades generales, utilidades intermedias y características primitivas. Se centra fuertemente en que el software haga lo que debe hacer (Utilidad general).
- **Ventajas:** Excelente para asegurar que el mantenimiento del software sea económico y eficiente a largo plazo.
- **Desventajas:** La medición de sus atributos puede ser costosa y requiere mucho esfuerzo de gestión.
- **Áreas de aplicación:** Proyectos de gran escala donde el ciclo de vida será muy largo y el mantenimiento es el costo mayor.

2. Selección y Justificación del Modelo

Modelo Seleccionado: ISO/IEC 25010

Justificación

1. **Enfoque en la Accesibilidad:** El proyecto tiene como requisito no funcional clave (RNF-02) la inclusión de funciones de accesibilidad (narración, alto contraste). La ISO 25010 es uno de los pocos modelos que trata la "Accesibilidad" como una sub-característica explícita dentro de la Usabilidad.
2. **Compatibilidad y Portabilidad:** AlgeVisual debe funcionar tanto en Android como en iOS (RNF-01) y adaptarse a diferentes pantallas (RNF-05). Este modelo evalúa específicamente la "Coexistencia" y "Interoperabilidad" (Compatibilidad), así como la "Adaptabilidad" (Portabilidad).
3. **Calidad en Uso:** Al ser una herramienta educativa para estudiantes de nivel medio superior, la satisfacción del usuario y la ausencia de riesgos de uso son críticas, aspectos que este modelo cubre a profundidad.

Atributos de Calidad Prioritarios

Basado en los requisitos MOSCOW definidos previamente, los atributos prioritarios son:

- **Usabilidad (Accesibilidad y Operabilidad):** Esencial para que estudiantes con y sin discapacidad puedan usar la app.
- **Adecuación Funcional (Exactitud):** El sistema debe resolver ecuaciones algebraicas sin errores matemáticos.
- **Portabilidad (Adaptabilidad):** La app debe instalarse y verse bien en múltiples dispositivos móviles.


3. Propuesta de Métricas

Atributo (ISO 25010)	Métrica Propuesta	Descripción / Fórmula	Objetivo
Adecuación Funcional (Exactitud)	Tasa de error en resolución de ejercicios	$\frac{\text{Ejercicios con resultado incorrecto}}{\text{Total de ejercicios probados}} \times 100\%$	0% (El álgebra debe ser exacta).
Usabilidad (Accesibilidad)	Cobertura de etiquetas para lectores de pantalla	Porcentaje de elementos de interfaz (botones, inputs) que tienen etiquetas <code>aria-label</code> o descripciones de audio activas.	100% de elementos críticos etiquetados.
Usabilidad (Operabilidad)	Tiempo promedio de aprendizaje	Tiempo promedio que tarda un estudiante nuevo en completar su primer ejercicio sin ayuda externa.	Menor a 5 minutos.
Portabilidad (Adaptabilidad)	Tasa de éxito en instalación	$\frac{\text{Dispositivos donde instala y abre sin error}}{\text{Total de dispositivos de prueba (Android/iOS)}} \times 100\%$	100% en versiones soportadas.
Eficiencia de Desempeño	Tiempo de respuesta	Tiempo transcurrido desde que el alumno envía la respuesta hasta que recibe la retroalimentación ⁸ .	Menor a 2 segundos.

1. Tasa de error en resolución (Para "Adecuación Funcional")

- **¿Qué es?:** Básicamente, asegurarnos de que la app **no se equivoque en las matemáticas**.
- **La lógica:** Si un alumno pone $2+2=5$ y la app dice $5=5$, la calidad es basura.

2. Cobertura de etiquetas (Para "Accesibilidad")

- **¿Qué es?:** Esta es **vital** para el proyecto porque es inclusiva (con narración y para gente con debilidad visual)¹.
- **La lógica:** Los lectores de pantalla que usan las personas ciegas leen el código. Si un botón solo es un ícono de una flecha  pero no tiene una etiqueta de texto oculta que diga "Siguiente", el ciego escuchará "Botón sin etiqueta" y no sabrá qué hacer.
- **La métrica:** Se revisa todos los botones. Si tiene 10 botones y a los 10 se puso su etiqueta de texto, tiene el 100% de cobertura.

3. Tiempo promedio de aprendizaje (Para "Usabilidad")

- **¿Qué es?:** Qué tan intuitiva es la app.

- **La lógica:** Si un alumno nuevo descarga AlgeVisual y tarda 20 minutos en entender cómo resolver su primer ejercicio, tu diseño es complicado.
- **La métrica:** Se toma un cronómetro y a un usuario nuevo. Se mide cuánto tarda desde que abre la app hasta que termina el primer ejercicio. Si tarda menos de 5 minutos, se pasa la prueba.

4. Tasa de éxito en instalación (Para "Portabilidad")

- **¿Qué es?:** Si funciona en diferentes celulares.
- **La lógica:** El proyecto es para Android e iOS². A veces una app funciona en el iPhone 15 pero truena en un Samsung viejo.
- **La métrica:** Intenta instalar la app en 10 celulares diferentes. Si se instala bien en los 10, tiene 100% de éxito.

5. Tiempo de respuesta (Para "Eficiencia")

- **¿Qué es?:** Que no sea lenta (lag).
- **La lógica:** Cuando el alumno le da clic a "Revisar respuesta", ¿se queda congelada la pantalla?
- **La métrica:** Se mide los segundos que tarda la app en decir "Correcto" o "Incorrecto". Si es menos de 2 segundos, se siente rápida.