

# Proceso de pruebas y evaluación para medir la madurez de un Software de Ventas para la Gestión y Control de Ventas del Minimarket WISA

## Testing and evaluation process to measure the maturity of a Sales Software for the Management and Control of Sales of the Minimarket WISA

Capacoila Apolinario Jhon Carlos <sup>1</sup>, Puchoc Laura Kenny William <sup>2</sup>, Torres Montes Alex Junior <sup>3</sup>

---

**RESUMEN**—El artículo destaca la importancia de la mejora de procesos en el desarrollo de software para la empresa WISA, con el objetivo de mejorar la calidad del producto final. Para lograr esta mejora, se utilizan herramientas como el modelo de proceso Capability Maturity Model Integration (CMMI). El artículo también describe el modelo Integración del Modelo de Madurez de Capacidad Técnica (TCMMI), que proporciona un enfoque centrado en la calidad del producto para evaluar y mejorar la calidad de los productos de software en la fase de desarrollo y lanzamiento, independientemente del proceso utilizado para desarrollar el software. El marco TCMMI se enfoca en la calidad del producto y permite medir la calidad desde la perspectiva de los desarrolladores y usuarios para evaluar y mejorar la madurez del producto de un software de ventas. El marco PMMI, que es una parte del TCMMI, se enfoca en la calidad del producto para evaluar y mejorar la calidad de los productos de software en la fase de desarrollo y lanzamiento, independientemente del proceso utilizado para desarrollar el software. Los autores proponen dos muestras para medir el grado de calidad del producto. El artículo señala que la producción de software de calidad se ha convertido en un factor fundamental para la competitividad de las empresas de América Latina en un mundo digital y globalizado, y se necesitan soluciones prácticas y eficaces para mejorar la calidad del software y aumentar la eficiencia en el proceso de desarrollo.

**Palabras clave**—Prueba de software, modelo de la calidad, verificación, validación.

**ABSTRACT**— The software industry is developing at an accelerated pace, and the competitiveness in the market requires company's high-quality processes and products. Quality is obtained by performing proper management, applying planning activities, quality assurance and control and improvements. The quality models propose a set of good practices focused on management processes. Cuba has a model applicable to the country, based on the best practices of the reference models and standards, named Quality Model for the Development of Computer Applications (MCDAI). In the case of quality control activities, the Verification and Validation processes ensure compliance with stakeholder requirements; software testing is the main dynamic technique of these processes. This research proposes a set of specific requirements that synthesize good software testing practices in a base process. It was designed to be part of the base processes of the MCDAI. It includes a graphic and textual description of the process, which facilitates compliance with the requirements and the roles involved. An indicator system is defined to provide the project and the management of an organization with an objective view of the processes and products of associated works. This process was submitted to focus groups, and an expert satisfaction survey to verify the usefulness of the proposal and to pilot projects to verify that with its application the effectiveness of the tests increases during software development.

**Keywords**— Testing of software, quality model, verification, validation.

---

## 1. Introducción

El desarrollo de software es una actividad de gran importancia y responsabilidad, especialmente cuando se trata de sistemas en tiempo real que pueden afectar la seguridad de las personas, además las pruebas de software son esenciales en el proceso de creación de sistemas informáticos para diversos contextos, incluyendo el de bodegas de tiendas. Por esta razón, algunas organizaciones de desarrollo de software están cada vez más interesadas en mejorar la calidad del software mediante la gestión efectiva del proceso de desarrollo. Para lograr este proceso, utilizan herramientas como el modelo de proceso Capability Maturity Model Integration (CMMI)[1][2], que se enfoca en la calidad del proceso utilizado para crear y mantener un producto o sistema. La filosofía detrás del modelo CMMI es que la calidad del producto final está estrechamente relacionada con la calidad del proceso utilizado para su desarrollo, sumado a esto la implementación del modelo CMMI implica documentar y evaluar los procesos utilizados por la organización para mejorar su madurez y, en consecuencia, mejorar el rendimiento general de la organización.

En el ámbito del desarrollo de software, la mejora de procesos es una práctica importante que puede generar beneficios significativos en términos de calidad del producto, productividad y reducción de costos y tiempo de

desarrollo. En este contexto, diversos autores han investigado las motivaciones y los resultados de la adopción de modelos de mejora de procesos, como Staples y Niazi [1], realizaron una revisión sistemática sobre las motivaciones organizacionales para adoptar la mejora de procesos de software (SPI) basado en CMM, mientras que Yamamura y Paulk, han explorado los efectos de la mejora de procesos en la satisfacción del cliente, la tasa de inyección de defectos y la correlación entre la madurez del software y diversos indicadores de desempeño del proyecto. Estos estudios muestran la importancia de implementar prácticas de mejora de procesos para mejorar la calidad del software y satisfacer las necesidades de los clientes [3] [4].

En la actualidad, la producción de software de calidad se ha convertido en un factor fundamental para la competitividad de las empresas de América Latina en un mundo digital y globalizado. A pesar de ello, muchas organizaciones todavía emplean un enfoque "artesanal" en lugar de técnicas de ingeniería de software, y se enfrentan a dificultades para implementarlas debido a recursos limitados y resistencia al cambio, como se observa la contratación de empresas de consultoría para mejorar la calidad del software resulta a menudo costosa y no viable para las pequeñas empresas de la región [5]. Por lo tanto, se hace necesario buscar soluciones prácticas y eficaces que permitan mejorar la calidad del software y aumentar la eficiencia en el proceso de desarrollo.

El propósito de este escrito es presentar un modelo que posibilite el proceso de pruebas y evaluación para medir la madurez de un software de ventas terminado sin importar el proceso empleado en su desarrollo. Por otro lado, en el desarrollo de software, la calidad del producto está determinada en gran parte por la calidad del proceso utilizado. Sin embargo, algunas organizaciones con un alto nivel de madurez en procesos aún presentan defectos en el producto final. Por lo tanto, es necesario evaluar la calidad del producto de software para garantizar la calidad del proceso. Este artículo propone un marco llamado Integración del Modelo de Madurez de Capacidad Técnica (TCMMI) para evaluar la madurez del producto final de software, independientemente del proceso utilizado para desarrollarlo [6]. El marco TCMMI consta de un modelo de referencia que es Product Maturity Model Integration (PMMI) y un método de evaluación (PMAM) y puede ser utilizado para evaluar la madurez de todo tipo de productos antes de su lanzamiento. El objetivo del texto es presentar el marco de referencia PMMI, el cual proporciona un enfoque centrado en la calidad del producto para evaluar y mejorar la calidad de los productos de software en la fase de desarrollo y lanzamiento, independientemente del proceso utilizado para desarrollar el software. Los autores proponen dos muestras para medir el grado de calidad del producto: el porcentaje de desviación de los valores reales de los atributos de calidad y el grado en que el producto satisface los requisitos funcionales y no funcionales. El marco TCMMI, basado en el marco PMMI [1], se enfoca en la calidad del producto y permite medir la calidad desde la perspectiva de los desarrolladores y usuarios, para evaluar y mejorar la madurez del producto de un software de ventas.

## **2. Materiales y métodos**

En la creación de TCMMI, nuestro enfoque se centró en garantizar que el marco fuera adaptable, fácilmente personalizable y, al mismo tiempo, compatible con CMMI-DEV. Para lograr esto, desglosamos el marco en diferentes componentes y etapas. Es importante señalar que el alcance de TCMMI abarca los requisitos hasta la etapa de lanzamiento del software en el ciclo de vida del desarrollo de software, pero no incluye etapas posteriores como el mantenimiento del software una vez que el producto ha sido liberado y se encuentra en funcionamiento [7]. El marco TCMMI consta de dos componentes: modelo de referencia (PMMI) y método de evaluación (PMAM). Las siguientes son las definiciones de PMMI y PMAM.

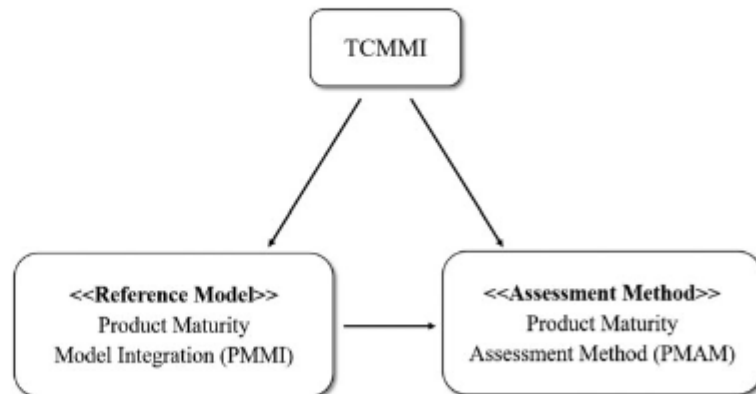
El PMMI es el modelo de referencia que sirve como plataforma para recopilar pruebas y determinar el grado de conformidad del producto con los indicadores de calidad utilizados para evaluar su nivel de madurez. Este modelo se complementa con una escala de cuatro niveles de madurez, cada uno de los cuales refleja un grado de conformidad del producto con sus atributos de calidad.

Por otro lado, el PMAM es el método de evaluación que ofrece una serie de pasos estándar para evaluar la capacidad o madurez del producto, mediante la evaluación del grado de conformidad del producto con los atributos de calidad requeridos por las partes interesadas.

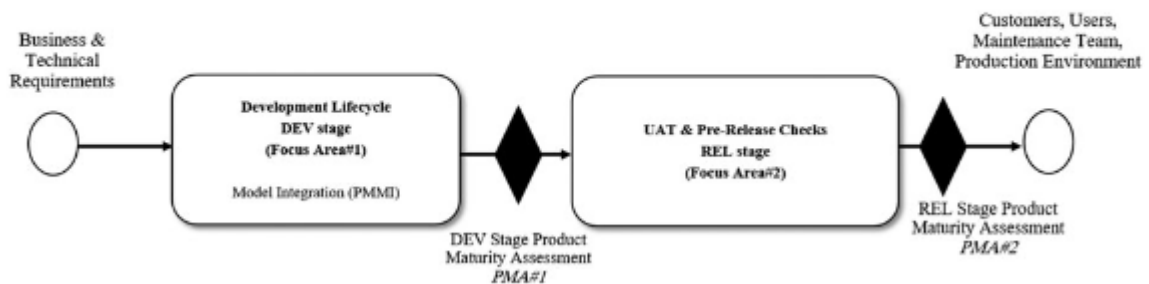
## 2.1 Arquitectura PMMI:

En esta sección, describimos la estructura del PMMI, que es el primer componente del marco TCMMI. Como se muestra PMMI tiene dos dimensiones:

- Dimensión de la calidad del producto, y
- Escala de capacidad del producto



## Arquitectura del marco TCMMI



## 2.2 Calidad, capacidad y madurez del producto:

En el marco de TCMMI, se define la calidad del producto como una medida de su conformidad con sus atributos de calidad internos definidos. Esto refleja en qué medida el producto cumple con los requisitos funcionales definidos por las partes interesadas relevantes. Normalmente, se realizan pruebas durante la etapa DEV en el ciclo de vida del desarrollo de software. El resultado de los diversos atributos funcionales refleja el nivel de madurez del producto al final de la etapa DEV.

En el contexto del producto de software, la calidad se refiere a dos nociones relacionadas pero distintas que cubren dos etapas principales en cualquier ciclo de vida de desarrollo: la etapa DEV para desarrollar el producto y la etapa REL para lanzarlo. Definimos una puerta de calidad al final de cada una de estas dos etapas: PMA#1 para verificar la calidad funcional del producto y PMA#2 para verificar su calidad no funcional y su grado de disponibilidad para ser entregado a los usuarios/clientes. Cuanto mayor sea la calidad del producto, mayores serán las capacidades que ofrece al usuario/cliente, y así se relaciona la calidad con la capacidad.

La capacidad de un producto se define por su capacidad para entregar la funcionalidad especificada en los requisitos funcionales, que generalmente son especificados por el cliente que necesita estas funciones. El grado de conformidad del producto con los requisitos funcionales refleja su capacidad, y cuanto más capaz sea el producto, mayor será su nivel de madurez.

Se adoptó la definición de madurez del Diccionario Inglés Collins: "la madurez es el estado o cualidad de ser maduro; pleno desarrollo". Además, se puede definir como la madurez de la calidad del producto de software. La hipótesis es que cuanto mayor sea el nivel de calidad del producto, mayor será su capacidad y mayor será su nivel de madurez. Los modelos de capacidad y madurez se refieren generalmente al proceso, pero también se usan para referirse al producto, que es el foco de este trabajo[8].

### **2.3 Dimensión de la calidad del producto:**

La dimensión de calidad del producto tiene como objetivo establecer los atributos de calidad del producto que se utilizarán para calcular su nivel de madurez. Cada atributo de calidad contribuye a la capacidad del producto, y el grado de conformidad del producto con cada atributo es un indicador de su capacidad adicional. Los resultados de las pruebas de cada atributo de calidad reflejan sus indicadores, y el grado de conformidad de esos resultados con los esperados por las partes interesadas define el nivel de capacidad de cada atributo. La calibración de ese grado de conformidad define el nivel de capacidad de cada atributo dentro de la dimensión de capacidad.

A diferencia de los modelos de madurez del proceso, los modelos de madurez del producto no tienen un conjunto fijo de atributos de madurez aplicables a todos los tipos de productos. Debido a que los diferentes dominios de productos varían en su naturaleza, no se fija un conjunto de atributos de calidad para todo tipo de productos. En cambio, el modelo permanece abierto para que las partes interesadas del producto decidan sobre los atributos de calidad que se medirán y que servirán de base para la evaluación de la madurez del producto. Es responsabilidad del patrocinador decidir la lista de atributos de calidad de las partes interesadas que se medirán y que servirán de base para el proceso de madurez del producto.

### **2.4 Dimensión de la capacidad del producto:**

La dimensión de la capacidad del producto tiene como objetivo principal calcular el nivel de madurez del producto. Esta dimensión establece la escala que se utilizará para medir el grado de capacidad adicional que el producto obtiene al cumplir con un atributo de calidad específico. Cada atributo de calidad contribuye a la capacidad y madurez del producto, y el nivel de madurez se calcula como el promedio de todos los niveles de capacidad de cada atributo de calidad individual. Las partes interesadas deben especificar las prioridades de cada atributo de calidad, que se convertirán en factores de ponderación para influir en el resultado final en función de la importancia asignada a cada atributo. Por ejemplo, si la seguridad del producto es más importante que la modularidad, se asignará una ponderación más alta al atributo de seguridad. El equipo de evaluación recopila los resultados de las métricas seleccionadas para cada atributo de calidad y calcula el nivel de capacidad para cada uno. Para ello, el equipo de pruebas debe evaluar cada atributo de calidad utilizando las métricas asociadas con ese atributo y capturar los resultados reales de las pruebas, que se comparan con los resultados esperados/objetivos especificados. El nivel de conformidad se calcula como un porcentaje, que determinará el nivel de madurez que ese atributo de calidad contribuye al nivel general de madurez del producto. Los porcentajes se acuerdan con las partes interesadas y el equipo de pruebas y se basan en la comparación de los resultados reales con los objetivos. Se proporcionan ejemplos de cifras de una escala de madurez para atributos de calidad individuales[9].

- Nivel 4: Cumplimiento Total (FC) (Más del 90% de cumplimiento)
- Nivel 3: Cumplimiento Mayoritario (LC) (60% a 90% de cumplimiento)
- Nivel 2: Cumplimiento Parcial (PC) (20% a 60% de cumplimiento)
- Nivel 1: No Cumple (NC) (Menos del 20% de cumplimiento)

### **3. Proceso base de Pruebas de software**

#### **3.1 Procedimientos PMAM**

Hemos adoptado los modelos de referencia CMMI y SCAMPI junto con todos sus procesos y pasos en nuestro método propuesto. Para estar en línea con CMMI, hemos propuesto el método de evaluación PMAM para TCMMI en lugar de SCAMPI para CMMI. PMAM se enfoca en evaluar la conformidad de los productos de software con los atributos de calidad acordados con las partes interesadas y el patrocinador de la evaluación, mientras que SCAMPI se centra en la evaluación de procesos de software. En el marco de TCMMI, se realiza una evaluación PMAM en dos etapas del ciclo de vida del desarrollo del producto: PMA#1 al final de la etapa de desarrollo, antes de la integración y prelanzamiento, y PMA#2 al final de la Prueba del Sistema y UAT. Ambas evaluaciones cubren un enfoque integral del ciclo de vida del desarrollo del producto y cumplen con el marco ISO 15504-3 para la evaluación del software[10]. El propósito del PMAM es proporcionar un marco estándar para evaluar el nivel de madurez/capacidad del producto y su conformidad con los atributos de calidad requeridos por las partes interesadas:

1. Entrada de evaluación.
2. Proceso de Evaluación.
3. Resultado de la evaluación.
4. Identificación de los patrocinadores de la evaluación.
5. Identificación de los Evaluadores.
6. Responsabilidades de cada Integrante del Equipo PMAM.
7. Resultados esperados de la evaluación y datos mínimos que deben incluirse en el informe final de la evaluación.

#### **3.2 Proceso de fases:**

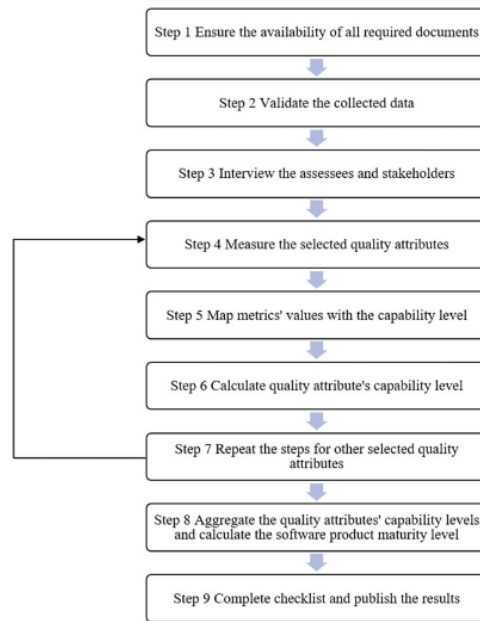
Siguiendo el modelo ISO “Orientación para realizar una evaluación”,<sup>47</sup>PMAM garantiza que los resultados de la evaluación sean confiables, repetibles y representativos. Para mantener la coherencia con el modelo ISO “Orientación para realizar una evaluación”, el PMAM se divide en tres fases:

1. Fase de entrada de PMA.
2. Fase de Procesos del PMA.
3. Fase de salida de PMA.

##### **3.2.1 En el PMA, el equipo de evaluación sigue estos pasos para la etapa DEV y la etapa REL:**

1. Identificar stakeholders para cada etapa del PMMI (DEV y REL).
2. Definir el alcance y los stakeholders de cada etapa del PMMI.
3. Enumerar los atributos de calidad que se medirán (según lo acordado por las partes interesadas de DEV y REL)
4. Definir los resultados de las pruebas objetivo esperados por las partes interesadas.
5. Seleccionar métricas para medir y probar los atributos de calidad identificados.
6. Pruebe el producto y compare los resultados reales de la prueba con los resultados previstos.
7. Calcule la variación entre los resultados de prueba reales y esperados.
8. Identifique el nivel de capacidad de cada atributo de calidad como se especifica en la escala de capacidad de PMMI.
9. Repita todos los pasos anteriores para todos los atributos de calidad.
10. Agregue los niveles de capacidad de todos los atributos de calidad (teniendo en cuenta cualquier ponderación) y calcule el promedio, que refleja el nivel general de madurez del producto.
11. Publicar los resultados de la evaluación para el Patrocinador y otras partes interesadas.

En la fase de salida de PMA, los resultados se documentarán y se informarán a los patrocinadores de la evaluación. El propósito de la evaluación está alineado con los objetivos comerciales de la organización que desarrolló el software y con los clientes del software. La perspectiva de la organización permitirá a la organización de desarrollo garantizar que el producto de software sea de alta calidad y pueda alcanzar el nivel de madurez objetivo, lo que permitirá a la organización comercializar su producto de software final como un producto de alta calidad.



### 3. Validación

En este artículo se usó un conjunto de experimentos que fueron llevados a cabo para validar el marco TCMMI. TCMMI es un acrónimo que significa "Capability Maturity Model Integration for Testing, Verificación y Validación de Sistemas Software", y se trata de un marco de referencia que se utiliza para evaluar la madurez del software; el objetivo principal de estos experimentos de validación fue demostrar que TCMMI es capaz de evaluar de manera sistemática la madurez del software, es decir, de manera estructurada y rigurosa. La validación experimental se llevó a cabo a través de diferentes pruebas y evaluaciones para comprobar la efectividad del marco en diferentes escenarios y situaciones.

El resultado de estos experimentos permitió validar la eficacia del marco TCMMI como una herramienta valiosa para evaluar la madurez del software. Esto es importante para las organizaciones que desarrollan software, ya que les permite mejorar su proceso de desarrollo y aumentar la calidad de sus productos. En donde se plantea la siguiente pregunta de investigación "¿Puede el marco TCMMI medir la madurez del software de una manera que esté en línea con la evaluación de la opinión de los profesionales?".

Para la selección de variables se evaluaron la aplicación de software con respecto a tres atributos de calidad, es decir, mantenibilidad, reutilización y complejidad con el objetivo de comparar los resultados obtenidos mediante la evaluación de la calidad del software a través de TCMMI y el juicio de los profesionales, se seleccionaron tres atributos de calidad como variables independientes. Estos atributos incluyen la mantenibilidad, que se refiere al esfuerzo requerido para encontrar y corregir un error en un programa; la reutilización, que mide la capacidad de un programa para ser utilizado en otras aplicaciones en relación con el paquete y el alcance de las funciones que realiza; y la complejidad, que se refiere al esfuerzo necesario para desarrollar software en un entorno específico. La variable dependiente en este experimento fue la calidad o nivel de madurez de dos aplicaciones de software. De acuerdo con TCMMI, la madurez se define como el grado de conformidad del producto con los atributos de calidad requeridos por las partes interesadas.

El experimento fue diseñado para probar si la evaluación del marco TCMMI se alinea con el juicio de los profesionales. Con el fin de determinar la eficacia, formulamos la siguiente hipótesis: La evaluación de la madurez del software a través del marco TCMMI se correlaciona con la evaluación realizada por un profesional. Si se encuentra una correlación significativa en todos los atributos de calidad seleccionados, se aceptará la hipótesis propuesta.

**3.1.Nivel de madurez deseado para las etapas DEV y REL.**

Esta sección presenta un ejemplo detallado sobre cómo PMMI y PMAM se han utilizado para evaluar la madurez de un producto de software en un caso de estudio. En este ejemplo, se asume que un gerente de una organización desea evaluar un proyecto en particular para determinar si cumple con el nivel de capacidad deseado para un conjunto de atributos de calidad y el nivel de madurez deseado para las etapas de desarrollo (DEV) y de liberación (REL). El gerente tiene la intención de lanzar el software solo si satisface las expectativas gerenciales. Los siguientes son los pasos seguidos en el ejemplo:

El gerente (patrocinador de las etapas DEV y REL) selecciona el proyecto X para su evaluación.

El patrocinador selecciona un evaluador competente que será responsable de la evaluación del proyecto X.

El patrocinador y el asesor competente seleccionan el alcance predeterminado de PMMI del Área de enfoque, los asesores y las partes interesadas para las etapas DEV y REL de la lista recomendada de partes interesadas de PMMI, según se muestra en las Tablas 1 y 2.

Partes interadas	Seleccionado
Jefes de proyecto y jefes de equipo	✓
Ingenieros de requisitos	✓
Diseñadores/arquitectos de soluciones	
Desarrolladores	✓
Equipos de integración y probadores de integración	✓
Equipos de mantenimiento	✓
Equipos de seguridad TI	✓

Tabla 1: Lista de verificación de las partes interesadas de PMMI para la etapa DEV del proyecto.

Partes interadas	Seleccionado
Usuarios del producto	✓
Clientes de productos	
Equipos de ventas y marketing	
Vendedores de productos	
Equipos de seguridad TI	
Equipos de mantenimiento	✓
Revendedores de productos	
Equipos de aceptación de usuarios	✓
Equipos de lanzamiento	✓

Tabla 2: Lista de verificación de las partes interesadas de PMMI para la etapa REL del proyecto X

El evaluador competente debe tener acceso a todos los documentos necesarios para realizar la evaluación del proyecto. Luego, el patrocinador y el evaluador deciden qué atributos de calidad serán medidos utilizando la lista recomendada de PMMI, como se detalla en la Tabla 3. Algunos atributos pueden estar relacionados o superponerse, como la complejidad y la mantenibilidad en este caso. A continuación, el patrocinador y el evaluador asignan pesos y niveles de capacidad a cada atributo de calidad utilizando las Tablas 4 y 5 para DEV y REL, respectivamente. En estas tablas se muestra que la suma de los pesos de los atributos de calidad para cada fase es 100. El patrocinador y el evaluador competente eligen seguir los niveles de madurez recomendados por PMMI y también determinan el nivel de madurez del producto para la etapa DEV y la etapa REL, que son 2 (PC) y 3 (LC), respectivamente.

Atributos de calidad	Seleccionado	Etapas
Consistencia		
Eficiencia		
Mantenibilidad	✓	NIR
Fiabilidad	✓	NIR
Testabilidad	✓	DEV
Comprensibilidad		
Usabilidad	✓	NIR
Seguridad		
Extensibilidad		
Seguridad		
Lo completo		
Concisión		
Legibilidad		
Reutilización		
Modularidad		
Complejidad	✓	

Tabla 3: Lista de verificación de atributos de calidad de PMMI.

Atributos de calidad	Valor	Nivel de capacidad deseado
Testabilidad	30%	2 ordenadores personales
Complejidad	70%	2 ordenadores personales

Tabla 4: Atributos de calidad DEV con sus pesos.

Atributos de calidad	Valor	Nivel de capacidad deseado
Mantenibilidad	30%	3 ordenadores personales
Usabilidad	50%	4 FC
Fiabilidad	20%	3 LC

Tabla 5: Atributos de calidad REL con sus pesos.



## 4. Resultados del experimento y discusión

Una encuesta fue completada por profesionales de un software para evaluar la eficacia de TCMMI en la evaluación de la calidad del producto de software. Los resultados mostraron que el 90% de los participantes estuvo de acuerdo en que TCMMI se ajusta a su propósito y el 72% indicó que se puede aplicar a diferentes dominios. Además, el 64% evaluó la estructura de TCMMI como fácil de entender, y el 72% la encontró fácil de usar. En cuanto a la usabilidad de la herramienta, el 64% de los participantes estuvo de acuerdo en que las herramientas de apoyo proporcionadas son fáciles de aprender y usar. Sin embargo, algunos participantes indicaron que la personalización de TCMMI para diferentes dominios puede ser difícil debido a la necesidad de identificar diferentes entradas y tener suficiente experiencia y conocimiento del contexto del software y los objetivos de medición. Algunos participantes también indicaron que la estructura de TCMMI puede ser difícil de comprender y usar, y algunos tuvieron opiniones neutrales o en desacuerdo con respecto a la facilidad de uso y el soporte de la herramienta. A pesar de estos desafíos, una vez que los usuarios aprendieron a usar TCMMI, encontraron que era fácil de usar debido a las diferentes listas de verificación y el sitio web automatizado que proporciona información sobre las métricas y cómo calcular los niveles de capacidad y madurez. En un minimarket, se podría encuestar a los empleados para evaluar la eficacia de una herramienta de gestión de inventario y ventas, y se podrían identificar desafíos similares en cuanto a la comprensión y el uso de la herramienta, así como la necesidad de personalización para diferentes contextos y objetivos.

## 5. Conclusiones

- La implementación de un software de ventas puede ser beneficiosa para un minimarket, ya que puede mejorar la eficiencia en la gestión de inventarios, el seguimiento de las ventas y la elaboración de informes financieros.
- La elección de un software de ventas adecuado es fundamental para el éxito de la implementación. Debe seleccionarse un software que cumpla con las necesidades específicas del minimarket, sea fácil de usar y ofrezca soporte técnico confiable.
- Es importante que todos los empleados del minimarket reciban una capacitación adecuada en el uso del software de ventas. Esto garantizará que el software se utilice de manera efectiva y que se eviten errores que puedan afectar la precisión de los registros de ventas.
- La implementación del software de ventas debe ser vista como una inversión a largo plazo, ya que los beneficios no se verán inmediatamente, sino que se acumularán con el tiempo. Por lo tanto, es necesario tener una perspectiva a largo plazo y ser paciente.
- Es importante que se realice un seguimiento constante de la implementación del software de ventas para asegurarse de que se está utilizando de manera efectiva y para identificar posibles problemas y solucionarlos de manera oportuna.

En general, la implementación de un software de ventas puede ser una herramienta valiosa para mejorar la eficiencia y la rentabilidad de un minimarket. Sin embargo, se requiere una planificación cuidadosa, una selección adecuada del software, una capacitación adecuada de los empleados y un seguimiento constante para asegurarse de que se está utilizando de manera efectiva.

## REFERENCIAS

- [1] Staples M, Niazi M. *Systematic review of organizational motivations for adopting CMM-based SPI*. Inform Softw Tech J. 2008;50(7-8):605-620.

- [2] CMMI® for Development, Version 2.0, 2018.
- [3] Proceso de Yamamura G. *Software empleados satisfechos*. *Software IEEE*, septiembre/octubre. 1999;16(5):83-85
- [4] Paulk MC. The Capability Maturity Model: Guidelines for Improving the Software Process (SEI Series in Software Engineering). USA: Addison-Wesley Pub. Co.; 1995.
- [5] Bush, M. (2015). *CMM, el modelo de madurez de la capacidad. Pautas para mejorar el proceso de software*, Universidad Carnegie Mellon, Instituto de Ingeniería de Software, Serie SEI en Ingeniería de Software, Addison-Wesley.
- [6] Alshayeb M, Abdellatif AK, Zahran S, Niazi M. *Hacia un marco para la medición de la madurez del producto de software*. En: La Décima Conferencia Internacional sobre Avances en Ingeniería de Software. Barcelona, España: Academia Internacional, Asociación de Investigación e Industria (IARIA); 2015.
- [7] Alshayeb M, Abdellatif AK, Zahran S, Niazi M. Hacia un marco para la medición de la madurez del producto de software. En: La Décima Conferencia Internacional sobre Avances en Ingeniería de Software. Barcelona, España: Academia Internacional, Asociación de Investigación e Industria (IARIA); 2015.
- [8] ISO/IEC, "15504-3: Tecnología de la información—Evaluación de procesos—Parte 3—Orientación para realizar una evaluación No. 15504-3", 2004
- [9] . Abreu FB, Goulão M, Esteves R. Hacia la evaluación de la calidad del diseño de sistemas de software orientados a objetos. En: 5ta Conferencia Internacional sobre Calidad de Software, Austin, Texas, EE.UU. Austin, Texas: EUA, Sociedad Estadounidense para la Calidad; 1995: 44-5
- [10] Alvaro A, Almeida E, Meira S. Atributos de calidad para un modelo de calidad de componentes. En: 10° WCOP/19° ECCOP. Glasgow, Escocia: Springer; 2005