



# Lectura fundamental

## Modelos de madurez de ingeniería de software

Módulo: Temas avanzados en  
diseño de *software*

**Unidad 4**



# Contenido

- 1 Introducción
- 2 ¿Qué son los modelos de madurez del *software* y cuál es su utilidad?
- 3 Principales modelos de madurez en ingeniería de *software*
- 4 Conclusiones

## 1. Introducción

Los modelos de madurez del *software* (MM) son marcos que describen los niveles de madurez de los procesos de desarrollo de *software* de una organización. Estos modelos ofrecen una visión general de los procesos que deben implementarse para lograr una mayor eficiencia, productividad y calidad.

Los MM se basan en la idea de que las organizaciones pueden mejorar sus procesos de desarrollo de *software* a través de la implementación de buenas prácticas. Estas buenas prácticas se agrupan en áreas funcionales, como la gestión de requisitos, la gestión de la configuración o la gestión de pruebas.

Los MM más utilizados son el CMMI (*Capability Maturity Model Integration*) y el ISO 15504. El CMMI es un modelo más completo y detallado que el ISO 15504.

Es muy importante conocer estos modelos de madurez, de forma que se puedan aplicar y proponer mejoras en los procesos de las organizaciones con las que se interactúa. Toda empresa puede y debe buscar mejorar, por lo que poder aplicar estos modelos representa una gran oportunidad de mejora.

## 2. ¿Qué son los modelos de madurez de *software* y cuál es su utilidad?

Los modelos de madurez de *software* proporcionan una guía sistemática para evaluar, medir y mejorar la efectividad y eficiencia de los procesos de desarrollo de *software* (Sommerville, 2005). De acuerdo con Morales-Fernández *et al.* (2014), es una:

colección estructurada de elementos que describen características de procesos efectivos, este modelo proporciona entre otras cosas un lugar para comenzar, la ventaja de las experiencias previas de una comunidad y un modo de definir lo que la mejora quiere decir para su organización, y principalmente puede ser usado como una prueba patrón para evaluar organizaciones. (p. 15)

Su utilidad radica en varios aspectos:

### 2.1. Evaluación y medición

La evaluación y medición son aspectos fundamentales en los modelos de madurez de *software*. Estos modelos proporcionan una estructura para evaluar y medir la madurez de los procesos de desarrollo de *software* en una organización. Permiten a las organizaciones evaluar el estado actual de sus prácticas de desarrollo de *software* y medir la madurez de sus procesos. Los pasos que se pueden seguir para llevar a cabo la evaluación y medición son:

- Identificación de áreas a evaluar: por ejemplo, gestión de proyectos, ingeniería de *software*, calidad, etc.
- Definición de objetivos medibles: estos objetivos deben ser específicos, alcanzables, relevantes y medibles (SMART). Por ejemplo, un objetivo puede ser reducir la tasa de defectos en un 20 % en un periodo.
- Desarrollo de indicadores clave de rendimiento (KPI): los KPI deben reflejar la eficiencia y la efectividad de los procesos. Por ejemplo, un KPI puede ser la tasa de defectos por cada mil líneas de código.
- Recopilación de datos: los datos se pueden obtener a través de herramientas de seguimiento, revisiones de código, inspecciones, pruebas, etc.
- Análisis de datos: sirve para identificar tendencias, patrones y áreas de mejora.
- Identificación de áreas de mejora: puede servir para ajustar procesos, definir capacitaciones adicionales, implementar herramientas o definir otros cambios, según se necesite.
- Implementación de mejoras: una vez se han identificado los cambios, estos deben ser comunicados con claridad a todos los miembros del equipo.
- Monitoreo continuo: sirve para seguir ajustando procesos y detectar mejoras a implementar.

## 2.2. Mejora continua

La mejora continua es un concepto central en los modelos de madurez de *software*. Estos modelos no solo sirven para evaluar el estado actual de los procesos de desarrollo de *software* en una organización, sino que también proporcionan una estructura para la evolución constante y la mejora continua, ya que proporcionan una estructura al establecer niveles de madurez que van desde inicial hasta optimizado. Las organizaciones pueden identificar áreas de debilidad y trabajar en su mejora progresiva.

## 2.3. Estándares y buenas prácticas

Los modelos de madurez de *software* suelen incorporar estándares y buenas prácticas como parte integral de su enfoque para evaluar y mejorar los procesos de desarrollo de *software* en una organización. Estos estándares y buenas prácticas ofrecen orientación y criterios claros para ayudar a las organizaciones a alcanzar niveles superiores de madurez. Ofrecen estándares y buenas prácticas que las organizaciones pueden seguir para alcanzar niveles más altos de madurez en la gestión y ejecución de proyectos de *software*.

## 2.4. Enfoque sistemático

El enfoque sistemático en modelos de madurez de *software* implica una metodología estructurada y coherente para gestionar y mejorar los procesos de desarrollo de *software* en una organización. Este enfoque se basa en la idea de que los procesos de *software* son interdependientes y están interrelacionados y se centra en la implementación sistemática de mejores prácticas y estándares para lograr una mejora continua. Ayudan a establecer un enfoque sistemático para el desarrollo de *software*, lo que reduce la variabilidad y mejora la previsibilidad en la entrega de productos y servicios.

## 2.5. Gestión de riesgos

La gestión de riesgos es un componente crítico en los modelos de madurez de *software*, ya que contribuye significativamente a la eficiencia y eficacia de los procesos de desarrollo. Contribuyen a la gestión efectiva de riesgos al proporcionar una estructura para la identificación y mitigación de problemas en las fases tempranas del desarrollo.

## 2.6. Mejora de la calidad

La mejora de la calidad es un objetivo central en la mayoría de los modelos de madurez en *software*. Estos modelos proporcionan un marco estructurado para que las organizaciones evalúen y mejoren sus procesos de desarrollo de *software* con el fin de lograr una mayor calidad en los productos entregados. Facilitan la mejora de la calidad del *software* al enfocarse en la estandarización y optimización de los procesos, lo que da como resultado productos finales más confiables y de mayor calidad.

## 2.7. Orientación a objetivos

La orientación a objetivos en modelos de madurez de *software* se refiere a la alineación de los procesos y prácticas de desarrollo con los objetivos estratégicos y comerciales de la organización. Este enfoque busca garantizar que todas las actividades y decisiones en el desarrollo de *software* estén alineadas con metas claras y contribuyan al éxito general de la organización. Permiten a las organizaciones alinear sus objetivos comerciales con las prácticas de desarrollo de *software*, mejorando la coherencia y la eficacia de los esfuerzos de desarrollo.

## 2.8. Visibilidad y transparencia

Proporcionan visibilidad y transparencia en los procesos, lo que facilita la identificación de áreas de mejora y la toma de decisiones basada en datos.

Algunos de los modelos de madurez de *software*, como el *Capability Maturity Model* (CMM) y el *Capability Maturity Model Integration* (CMMI), han sido ampliamente adoptados y utilizados en la industria para mejorar la eficiencia y la calidad en el desarrollo de *software*. En resumen, estos modelos sirven como herramientas valiosas para impulsar la mejora continua y la excelencia en los procesos de ingeniería de *software*.

## 3. Principales modelos de madurez en ingeniería de *software*

En ingeniería de *software*, los modelos de madurez son marcos de referencia que describen la evolución y mejora de los procesos en una organización. Estos modelos proporcionan una estructura para evaluar la madurez de las prácticas de desarrollo de *software* y ofrecen pautas para mejorarlas. Y ¿por qué mejorar las prácticas de desarrollo de *software*? Porque los modelos que miden la madurez de las organizaciones usan una escala que permite medir la adherencia digital que tiene la empresa que se está estudiando. Esta escala ayuda a encontrar “fortalezas y debilidades en el uso de tecnologías de la información y permite desarrollar estrategias para mejorar su eficiencia y competitividad en el mercado” (Beltrán Saavedra, 2023, párr. 2).

La escala muestra niveles que van desde el 1, que es básico, hasta el 5, que es optimizado, y dentro de los elementos que está evaluando están: la gestión de los proyectos, la infraestructura, la cultura digital, la documentación de los sistemas, la arquitectura implementada, etc.

Algunos de los modelos de madurez más conocidos en ingeniería de *software* son:

## Recuerde que...



El uso de modelos de madurez en diversas áreas, incluida la ingeniería de *software*, ofrece varios beneficios que pueden contribuir al éxito y la mejora continua de una organización. Aquí se presentan algunas razones clave para utilizar modelos de madurez:

- Evaluación y *benchmarking*
- Identificación de mejoras
- Establecimiento de metas
- Mejora continua
- Gestión de riesgos
- Consistencia y repetibilidad
- Establecimiento de estándares
- Mejora de la eficiencia
- Toma de decisiones informadas
- Alineación con objetivos empresariales

### 3.1. Capability Maturity Model (CMM)

CMM (*Capability Maturity Model*), que se traduce como modelo de madurez de capacidad, es un conjunto de estándares y guías que describen las mejores prácticas clave asociadas con los procesos de desarrollo y mantenimiento de *software*. Fue originalmente desarrollado por el *Software Engineering Institute (SEI)* en la Universidad Carnegie Mellon con el propósito de ayudar a las organizaciones a mejorar la eficacia de sus procesos de desarrollo de *software* y, por ende, mejorar la calidad del *software* resultante.

- La idea central detrás del CMM es que la madurez en el desarrollo de *software* se puede medir y clasificar en diferentes niveles. Cada nivel representa un grado de madurez en la gestión y ejecución de los procesos. Hay cinco niveles en el modelo CMM, que se describen brevemente a continuación (Chacon, 2004):

**Nivel 1 - Inicial:** en este nivel, los procesos de desarrollo son generalmente caóticos, *ad hoc* y no están bien definidos. Los resultados pueden ser impredecibles y dependen en gran medida de la capacidad y experiencia individual de los desarrolladores.

- **Nivel 2 - Repetible:** las organizaciones en este nivel han establecido prácticas básicas de gestión de proyectos. Se centran en la estandarización de procesos y la repetición exitosa de enfoques que han funcionado en el pasado.
- **Nivel 3 - Definido:** en este nivel, las organizaciones han definido procesos estándar para el desarrollo y la gestión de proyectos. Existen procedimientos documentados y se enfatiza la consistencia en la ejecución.
- **Nivel 4 - Gestionado cuantitativamente:** en este nivel, las organizaciones utilizan mediciones cuantitativas para gestionar y controlar sus procesos. Se centran en la mejora continua y la gestión proactiva de la calidad.
- **Nivel 5 - Optimizado:** en el nivel más alto, las organizaciones optimizan continuamente sus procesos basándose en la retroalimentación cuantitativa y cualitativa. Se enfocan en la innovación y la mejora continua.

Es importante destacar que el CMM se ha convertido en una base para otros modelos más avanzados, como el *Capability Maturity Model Integration* (CMMI). El CMMI amplía la cobertura más allá del desarrollo de *software*, abarcando áreas como la gestión de proyectos, la gestión de la configuración y la gestión de la calidad, proporcionando una perspectiva más holística.

El CMM y sus derivados han sido utilizados en la industria para evaluar y mejorar la capacidad y madurez de los procesos de desarrollo de *software*. Estos modelos proporcionan un marco estructurado para que las organizaciones avancen hacia la mejora continua y la excelencia en la ejecución de proyectos de *software*. Como se menciona en Comunicaciones (2019):

Aplicar este modelo en el desarrollo de *software* es importante, puesto que permite optimizar algunos procesos de negocio, desarrollar productos con calidad para satisfacer las necesidades del cliente y ayuda a cumplir de forma completa con los requerimientos de la norma ISO y crear una cultura de mejora continua. (párr. 4)

Construir un *software* dirigido bajo este modelo trae también unas ventajas, y esa utilización de buenas prácticas en el desarrollo de *software* trae los siguientes beneficios:

- Comunicación efectiva entre las partes: tanto entre los miembros del equipo como con los clientes, quienes se participan activamente en el proceso.
- *Software* más completos: el modelo permite tomar requisitos de una manera más acertada, por la cercanía con los clientes, una mejor capacitación del equipo de trabajo, aplicación de pruebas e inspección y buenas prácticas de construcción de *software*.



- *Software* entregado a tiempo: el modelo mejora predicciones de entrega debido a la realimentación constante.
- *Software* con menor cantidad de defectos: los errores son corregidos en las fases tempranas del proceso.

### ¿Sabía que...?

CMM y CMMI fueron diseñados por el SEI (*Software Engineering Institute*) en conjunto con la universidad de Carnegie Mellon con el objetivo de proporcionar un enfoque integral para mejorar la calidad y la eficacia de los procesos organizativos.



## 3.2. *Capability Maturity Model Integration (CMMI)*

CMMI (*Capability Maturity Model Integration*) es un modelo de madurez que integra diversas disciplinas de ingeniería y gestión en un único marco.

A diferencia del CMM, que se centraba en el desarrollo de *software*, el CMMI abarca múltiples disciplinas, incluyendo desarrollo de *software*, gestión de proyectos, ingeniería de sistemas, procesos de adquisición y gestión de servicios. El CMMI se organiza en niveles de madurez y áreas de proceso.

### • Niveles de madurez en CMMI:

1. **Nivel 1 - Inicial:** los procesos son impredecibles y *ad hoc*. Hay poca consistencia en la ejecución de procesos y dependen en gran medida de la capacidad individual.
2. **Nivel 2 - Gestionado:** los procesos básicos están establecidos y se gestionan de manera que se puedan repetir con éxito. Se enfoca en la planificación, el monitoreo y el control de proyectos.
3. **Nivel 3 - Definido:** se establecen procesos estándar organizacionales para gestionar proyectos y la ingeniería. La organización se centra en la mejora continua.
4. **Nivel 4 - Cuantitativamente gestionado:** la organización cuantifica la calidad y el rendimiento de los procesos. Se utilizan medidas cuantitativas para gestionar la variabilidad y mejorar la eficacia.

**5. Nivel 5 - Optimizado:** la mejora continua se institucionaliza en toda la organización. Se enfoca en la innovación y la optimización continua de procesos.

Además, CMMI cuenta con unas áreas de proceso, que son subgrupos que se enfocan en puntos específicos de los diferentes procesos industriales. Estas áreas de Proceso en CMMI se organizan en “constelaciones”, que agrupan áreas de proceso según áreas temáticas.

- Algunas de las constelaciones comunes, de acuerdo con Chacon (2004) son:

CMMI para desarrollo (CMMI-DEV): se centra en mejorar la calidad y la eficacia de los procesos de desarrollo.

CMMI para adquisición (CMMI-ACQ): dirigido a organizaciones que adquieren productos y servicios.

CMMI para servicios (CMMI-SVC): se aplica a organizaciones que proporcionan servicios en lugar de productos.

CMMI proporciona un conjunto de prácticas detalladas y áreas de proceso que las organizaciones pueden adoptar y adaptar según sus necesidades específicas. Al seguir las prácticas definidas en el marco CMMI, las organizaciones buscan mejorar la calidad, la eficiencia y la eficacia de sus procesos, lo que finalmente contribuye al logro de sus objetivos comerciales y estratégicos.

### 3.3. ISO/IEC 15504 (*SPICE - Software Process Improvement and Capability Determination*):

La norma ISO/IEC 15504, también conocida como SPICE (*Software Process Improvement and Capability Determination*), es un estándar internacional desarrollado para evaluar y mejorar los procesos de desarrollo y mantenimiento de *software*. Su objetivo principal es proporcionar un marco para la evaluación de la capacidad y la mejora de los procesos de *software* en una organización.

Aquí hay algunos puntos clave sobre ISO/IEC 15504 (SPICE):

1. Objetivo principal:
  - Evaluación de la capacidad del proceso:  
La norma se centra en la evaluación de la capacidad de los procesos de *software* en términos de su rendimiento, eficiencia y eficacia.
2. Principales componentes:

- Modelo de proceso:

SPICE utiliza un modelo de proceso que define un conjunto de procesos clave relacionados con el desarrollo y mantenimiento de *software*. Estos procesos se agrupan en categorías, como gestión de proyectos, ingeniería de *software* y soporte.

- Niveles de capacidad:

SPICE define niveles de capacidad que van desde 0 hasta 5, representando diferentes grados de madurez y capacidad en la ejecución de procesos. Estos niveles son similares a los niveles de madurez en otros modelos como CMMI.

- Atributos de proceso:

Cada proceso en el modelo SPICE se evalúa en función de varios atributos de proceso, que incluyen rendimiento, capacidad, disponibilidad y consistencia.

### 3. Proceso de evaluación:

- Evaluación por atributo:

La evaluación se realiza atributo por atributo para cada proceso, asignándole un nivel de capacidad específico. Esto proporciona una visión detallada del rendimiento de cada proceso en términos de los atributos seleccionados.

- Determinación de la capacidad global:

La capacidad global de la organización se determina a partir de la evaluación de los diferentes procesos. Se asigna un nivel de capacidad general basado en los resultados obtenidos.

### 4. Aplicaciones prácticas:

- Mejora continua:

La norma SPICE se utiliza para la mejora continua de los procesos de *software*. Al identificar áreas de mejora, las organizaciones pueden implementar acciones correctivas y preventivas.

- Comparación entre organizaciones:

SPICE permite comparar la capacidad de los procesos entre diferentes organizaciones, lo que facilita la identificación de mejores prácticas y áreas de excelencia.

- Aseguramiento de la calidad:

SPICE se utiliza como herramienta de aseguramiento de la calidad para garantizar que los procesos de *software* cumplan con los estándares y requisitos definidos.

- Soporte en adquisiciones:  
La norma SPICE también se utiliza en el proceso de adquisición de productos o servicios de *software*, proporcionando criterios objetivos para la evaluación de proveedores.

La aplicación de ISO/IEC 15504 (SPICE) ayuda a las organizaciones a comprender y mejorar sus procesos de *software*, lo que contribuye a la entrega de productos y servicios de *software* de alta calidad. Este estándar proporciona un enfoque sistemático y estructurado para la evaluación y mejora de la capacidad de los procesos de *software* en diversas organizaciones.

### 3.4. Agile Maturity Model (AMM)

Oficialmente, no existe un *Agile Maturity Model* universalmente aceptado como el CMMI (*Capability Maturity Model Integration*) para los enfoques ágiles (Humble y Russell, 2009). Sin embargo, hay varios modelos y marcos que se utilizan para evaluar y mejorar la madurez ágil en las organizaciones. Estos modelos a menudo se centran en la implementación y evolución de prácticas ágiles en lugar de seguir un enfoque de madurez similar al de CMMI.

Algunos de los marcos y modelos relacionados con la madurez ágil incluyen:

- *Spotify's Squad Health Check*: Spotify ha compartido su *Squad Health Check* como una forma de evaluar la salud y madurez de los equipos ágiles. Este enfoque incluye una serie de dimensiones que los equipos pueden revisar y discutir para evaluar su rendimiento.
- *SAFe (Scaled Agile Framework) - SAFe Big Picture*: SAFe es un marco ágil escalado que proporciona un enfoque estructurado para implementar ágil a nivel empresarial. Si bien no es un modelo de madurez en el sentido clásico, incluye una representación visual llamada *Big Picture* que proporciona una visión integral de los roles, equipos y prácticas ágiles.
- *Agile Fluency Model*: el *Agile Fluency Model*, desarrollado por Diana Larsen y James Shore, describe diferentes etapas de madurez ágil basadas en cómo una organización utiliza y se beneficia de las prácticas ágiles. Estas etapas son “*Focusing*,” “*Delivering*,” “*Optimizing*,” y “*Strengthening*”.
- *Agile Capability Model (ACM)*: el *Agile Capability Model* es un modelo de madurez ágil que se centra en la capacidad de las organizaciones para adoptar prácticas ágiles. Proporciona una estructura para evaluar y mejorar la madurez ágil en diferentes áreas.
- *Agile Maturity Matrix*: algunas organizaciones han desarrollado sus propias matrices de madurez ágil para evaluar su progreso en la adopción de prácticas ágiles. Estas matrices pueden incluir criterios específicos para las prácticas ágiles y se utilizan internamente para la autoevaluación.

Es importante tener en cuenta que la agilidad es un concepto multifacético, y la madurez ágil puede variar según la organización, el equipo y el contexto. Además, la comunidad ágil tiende a enfocarse más en la mejora continua y la adaptación que en la adhesión rígida a un modelo de madurez específico. Como resultado, las organizaciones a menudo adaptan y combinan múltiples enfoques para satisfacer sus necesidades específicas en su viaje hacia la madurez ágil.

### 3.5. *Lean Software Development Maturity Model (LSDMM)*

Tampoco existe un modelo de madurez con *Lean* oficialmente reconocido. Sin embargo, es posible que haya enfoques y modelos específicos desarrollados por organizaciones o expertos en *Lean* y *Agile* que se centren en la madurez del desarrollo de *software* desde una perspectiva *Lean*.

El pensamiento *Lean* se origina en la fabricación, pero sus principios se han aplicado con éxito en el desarrollo de *software* para mejorar la eficiencia y la calidad. Aquí hay algunas áreas clave que podrían ser consideradas en la evaluación de la madurez *Lean* en el contexto del desarrollo de *software*:

- Eliminación de desperdicios: evaluar cómo se están abordando y eliminando los desperdicios en el proceso de desarrollo de *software*. Los desperdicios en este contexto pueden incluir tiempos de espera, trabajo innecesario y otros elementos no productivos.
- Entrega continua: evaluar la capacidad de la organización para implementar prácticas de entrega continua, lo que implica la capacidad de entregar *software* de manera rápida y eficiente.
- Enfoque en el valor para el cliente: evaluar cómo se comprenden y se priorizan las características y funcionalidades del *software* en función del valor que aportan al cliente.
- Colaboración y comunicación: evaluar la eficacia de la colaboración y la comunicación en el equipo de desarrollo. Los principios *Lean* a menudo destacan la importancia de la comunicación efectiva y la colaboración cercana.
- Mejora continua: evaluar la presencia de prácticas y procesos que fomenten la mejora continua en el desarrollo de *software*. Esto puede incluir retroalimentación regular, revisiones retrospectivas y ajustes constantes en el proceso.

Hay que tener en cuenta que, si bien no existe un *Lean Software Maturity Model* específico, muchos principios *lean* se han incorporado a marcos y modelos ágiles, como *Scrum* y *Kanban*. Además, algunas organizaciones pueden haber adaptado prácticas *lean* en sus propias evaluaciones internas de madurez.

Si se está interesado en evaluar la madurez *Lean* en la organización, se podría considerar trabajar con expertos en *Lean* o *Agile* para adaptar principios *lean* al contexto específico y desarrollar un marco de evaluación de madurez personalizado.

### 3.6. People Capability Maturity Model (People CMM)

El *People Capability Maturity Model* (PCMM) es un modelo desarrollado por el *Software Engineering Institute* (SEI) en la Universidad Carnegie Mellon. Al igual que el CMMI (*Capability Maturity Model Integration*) se enfoca en la mejora de procesos, el PCMM se centra específicamente en la gestión y el desarrollo de las capacidades de las personas en una organización (Uriona Herrera, 2007). Este modelo está diseñado para ayudar a las organizaciones a mejorar la gestión de su talento humano y maximizar el rendimiento individual y colectivo.

A continuación, se describen los cinco niveles de madurez en el *People Capability Maturity Model*:

1. **Nivel 1 - Inicial:** en este nivel, las prácticas de gestión de personas son informales y en gran medida reactivas. No hay un enfoque sistemático para gestionar el rendimiento y el desarrollo del personal.
2. **Nivel 2 - Repetible:** se establecen prácticas básicas de gestión de personal. Se definen y documentan algunos procesos, y la organización comienza a adoptar un enfoque más proactivo para la gestión de recursos humanos.
3. **Nivel 3 - definido:** en este nivel, se establecen prácticas formales y bien definidas para la gestión de personas. Se enfatiza la consistencia en la aplicación de procesos y políticas de gestión de recursos humanos.
4. **Nivel 4 - Gestionado cuantitativamente:** se introducen medidas cuantitativas y técnicas de gestión del rendimiento. La organización se enfoca en la mejora continua y utiliza datos para la toma de decisiones relacionadas con el personal.
5. **Nivel 5 - Optimizado:** en el nivel más alto, la organización se centra en la mejora continua a través de la innovación y la optimización. Se busca la excelencia en la gestión de personas y en la creación de un entorno que fomente el desarrollo y la retención del talento.

Áreas de proceso en el PCMM: El PCMM aborda diversas áreas de proceso, que incluyen:

- Selección: definición de prácticas efectivas para la selección de personal.
- Compensación: establecimiento de prácticas de compensación equitativas y basadas en el rendimiento.
- Desarrollo de competencias: desarrollo de planes de formación y desarrollo personalizados.
- Desempeño y reconocimiento: establecimiento de prácticas para evaluar y reconocer el rendimiento.

- Cultura: fomento de una cultura que apoye el desarrollo del personal y el rendimiento excepcional.

El PCMM es aplicable a organizaciones de diversos sectores y tamaños, y su objetivo es ayudar a las organizaciones a mejorar la gestión de su capital humano. Proporciona un marco sistemático para la mejora continua y la optimización de las capacidades individuales y colectivas dentro de la organización.

## 4. Conclusiones

Los MM son una herramienta valiosa para las organizaciones que desean mejorar sus procesos de desarrollo de *software*. Los MM ofrecen una visión general de las buenas prácticas que deben implementarse para lograr una mayor eficiencia, productividad y calidad.

La evaluación y medición en modelos de madurez de *software* son herramientas clave para que las organizaciones comprendan su nivel de madurez en el desarrollo de *software*, identifiquen áreas de mejora y establezcan un camino claro hacia la mejora continua. Estas prácticas contribuyen a la eficiencia, calidad y éxito general de los proyectos de *software* en una organización.

La mejora continua en modelos de madurez de *software* implica un enfoque sistemático y progresivo para elevar constantemente la madurez de los procesos y lograr un rendimiento óptimo en el desarrollo de *software*. Este enfoque contribuye significativamente a la eficiencia, calidad y éxito a largo plazo de las operaciones de desarrollo de *software* de una organización.

La integración de estándares y buenas prácticas en los modelos de madurez de *software* proporciona a las organizaciones una base sólida para mejorar sus procesos, aumentar la calidad del *software* y alcanzar niveles más altos de madurez en la gestión y ejecución de proyectos. Estos estándares y buenas prácticas ofrecen una guía valiosa para las organizaciones que buscan optimizar sus procesos de desarrollo de *software* y alcanzar la excelencia en la entrega de productos y servicios.

Al adoptar un enfoque sistemático en modelos de madurez de *software*, las organizaciones pueden establecer una base sólida para la mejora continua y la optimización de sus procesos de desarrollo de *software*. Este enfoque proporciona una estructura coherente para gestionar la complejidad de los proyectos y mejorar la eficacia y calidad de los resultados.

La gestión de riesgos en modelos de madurez de *software* contribuye a la mitigación de problemas potenciales, mejora la toma de decisiones y aumenta la probabilidad de éxito en los proyectos de desarrollo de *software*. Integrar prácticas efectivas de gestión de riesgos es esencial para lograr niveles más altos de madurez y garantizar la entrega exitosa de productos y servicios de *software*.

La mejora de la calidad en modelos de madurez en *software* es un proceso integral que abarca todos los aspectos del desarrollo de *software*. Al adoptar las prácticas recomendadas y los estándares establecidos por estos modelos, las organizaciones pueden elevar sus niveles de madurez y, por ende, mejorar la calidad de sus productos y servicios de *software*.

Al adoptar una orientación a objetivos en modelos de madurez de *software*, las organizaciones pueden garantizar que sus esfuerzos de desarrollo estén alineados con las metas más amplias de la empresa. Esto contribuye a la eficacia general de la organización y ayuda a garantizar que el desarrollo de *software* aporte un valor significativo y sostenible.



# Referencias

- Beltrán Saavedra, G. (2023). Qué son los modelos de madurez y cómo mueven tu transformación digital. *Pragma*. <https://www.pragma.co/es/blog/modelos-de-madurez-y-como-mueven-tu-transformacion-digital>
- Chacon, W. (2004). *Modelo de capacidad de madurez del software y su influencia en las mejoras de calidad del software* [tesis de pregrado en ingeniería en ciencias y sistemas, Universidad de San Carlos de Guatemala]. [http://biblioteca.usac.edu.gt/tesis/08/08\\_0213\\_CS.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0213_CS.pdf).
- Comunicaciones. (2019). Modelo CMMI, calidad y buenas prácticas en el desarrollo de software. *Globalbit*. <https://www.globalbit.co/2019/07/22/modelo-cmmi-calidad-y-buenas-practicas-en-el-desarrollo-de-software/>.
- Humble, J., y Russell, R. (2009). *The Agile Maturity Model pplied to Building and A Releasing Software*. [https://info.thoughtworks.com/rs/thoughtworks2/images/agile\\_maturity\\_model.pdf](https://info.thoughtworks.com/rs/thoughtworks2/images/agile_maturity_model.pdf).
- Morales-Fernández, R., Brito-Rojas, J., Villaseñor Marcial, J. (2014). Capability Maturity Model Integration (CMMI). *RITI Journal*, 2(3), 15-20. <https://dialnet.unirioja.es/descarga/articulo/7242755.pdf>.
- Sommerville I. (2005). *Ingeniería del software*. <https://tognuhome.files.wordpress.com/2020/10/ingenierc3ada-del-software-ian-sommerville.pdf>.
- Uriona Herrera, C. (2007). People capability maturity model p-cmm. Modelo para la madurez de las capacidades de la gente un modelo para gestionar recursos humanos. *Perspectivas*, (19), 93-106. <https://www.redalyc.org/pdf/4259/425942453008.pdf>.

# Glosario

*Ad hoc*: para referirse a lo que se dice o hace solo para un fin determinado.



## Información técnica

**Módulo:** Temas avanzados en Diseño de *Software*

**Unidad 4:** Modelos de madurez de ingeniería de software

**Autor:** Yamid E. Ramírez Sánchez

**Asesora Pedagógica:** Jessica Katherin Pinzón Arias

**Diseñador Gráfico:** Nicolás Jiménez Osorio

Este material pertenece al Politécnico Grancolombiano.  
Prohibida su reproducción total o parcial.