Diseño de instrumentos que garanticen la calidad y mejores prácticas en el desarrollo de software para las pequeñas y medianas empresas de Bogotá

Elaborado por:

Jairo José Cetina Velandia

Nelson Gabriel Liberato Robayo

Jhon Carloc Tinjaca Garzón

Carlos Andrés Chaverra Córdoba

Universidad EAN

Grupo Tecnológico ONTARE

Seminario de Investigación de Pregrado

Bogotá

Noviembre de 2023

Diseño de instrumentos que garanticen la calidad y mejores prácticas en el desarrollo de software para las pequeñas y medianas empresas de Bogotá

Resumen

Esta investigación se realiza primordialmente en analizar el problema de la falta de concienciación y las vulnerabilidades en la seguridad cibernética en pequeñas y medianas empresas también llamadas PYMES en Bogotá, Colombia. Desde el año 2020 que comenzó la pandemia de SARS-CoV-2 o también llamado Covid-19, se ha observado un auge en la creación de compañías de tecnología de la información que se enfocan en desarrollar sistemas de información para pymes que previamente carecían de software en línea. Aunque esto ha impulsado el mercado de Tecnologías de la Información en Colombia, también ha aumentado la vulnerabilidad de estas empresas a los ataques cibernéticos debido a la falta de concienciación sobre lo crucial que es la ciberseguridad.

El problema radica en que las pymes a menudo priorizan la funcionalidad y la entrega rápida de software sobre la seguridad, lo que las convierte en objetivos atractivos para los ciberdelincuentes. Además, carecen de recursos financieros y talento humano para abordar la seguridad del software de manera adecuada. Los ataques cibernéticos exitosos pueden tener un efecto negativo en la reputación de estas pymes, erosionando la confianza de los clientes y afectando su viabilidad a largo plazo.

El estudio propone la creación de un marco de trabajo que incluya procesos esenciales para verificar la calidad y seguridad en los procesos de desarrollo de software en pymes de Bogotá. Los objetivos específicos incluyen identificar debilidades, crear un prototipo con certificaciones básicas, diseñar un documento con requisitos esenciales y presentar marcos globalmente reconocidos para pymes. La justificación se basa en la creciente importancia de la seguridad cibernética y la necesidad de difundir conocimientos accesibles sobre desarrollo seguro. Se argumenta que esto reducirá los riesgos, fortalecerá la credibilidad de las empresas de software y contribuirá a la comunidad de desarrollo.

En conclusión, el estudio tiene como objetivo abordar el problema relaciona con la baja calidad en el desarrollo de software en las pymes de Bogotá mediante la creación de un marco de trabajo y el fomento de prácticas de desarrollo seguro. Esto es esencial para proteger los datos de acuerdo con la norma ISO/IEC 27001:2022 “la integridad de los datos, la confidencialidad de la información y la disponibilidad en las operaciones comerciales en un entorno cada vez más digitalizado”.

Palabras clave: buenas prácticas, software, calidad, normas, ciberseguridad, metodologías de desarrollo, Scrum, Agile, CMMI, ITMark, arquitectura de software, repositorios de código, OWASP, medición de código.

Tabla de contenido

Problema de Investigación

Antecedentes del problema

En la actualidad, nos encontramos la era de la digitalización y dependemos cada vez más de la tecnología en nuestras operaciones empresariales. Las pequeñas y medianas empresas (pymes) en Bogotá, se enfrentan a desafíos significativos en la producción e implementación de desarrollo de software. En consecuencia, ante las exigencias de las tendencias del mercado, las pymes deben ajustar sus procesos para mejorar el ciclo de vida del software. (Limas Suárez, 2020)

A medida que estas pymes adoptan soluciones de software para mejorar su eficiencia y competitividad, también aumentan las vulnerabilidades, los riesgos, el mantenimiento, la escalabilidad, entre otros aspectos. Esto puede generar alertas de código malicioso debido al uso de librerías de terceros que hayan sido descargadas e integradas en el código del software en desarrollo, sin investigar el autor de esas librerías o paquetes.

Por tal motivo, las pymes, en su afán de liberar las implementaciones de software, evaden los procesos básicos o mínimos requeridos en el ciclo de vida del software. Esto conlleva a una deuda técnica a futuro, manifestada en una línea de tendencia ascendente en los reprocesos y la necesidad de reescribir el código realizado (SOFTWARE & TI, 2023).

No se han establecido metodologías claras en el proceso para el desarrollo de software, lo que resulta en una falta de estructura y coherencia en los proyectos. Además, la implementación de sistemas de control de versiones es inexistente, lo que dificulta el seguimiento y la gestión eficaz del código fuente. Asimismo, no se define una arquitectura acorde a las necesidades específicas de cada proyecto en desarrollo, lo que puede llevar a desafíos y dificultades en la fase de implementación y mantenimiento.

Descripción del problema

La presión por plazos es una de las causas más comunes que puede llevar a la falta de identificación y aplicación de prácticas adecuadas en el desarrollo de software en las empresas. En un entorno empresarial altamente competitivo, a menudo se enfrentan a plazos de entrega ajustados y demandas de los clientes para obtener productos o actualizaciones de software en un tiempo récord. Esta presión por cumplir fechas límite puede resultar en la omisión o el sacrificio de buenas prácticas que requieren tiempo adicional.

Teniendo en cuenta lo anteriormente mencionado la urgencia por entregar el producto puede llevar a la toma de decisiones apresuradas, como la omisión de pruebas exhaustivas, la reducción de la documentación o la implementación de soluciones de parcheo temporales en lugar de abordar de manera integral los problemas.

La consecuencia de esta presión por plazos de entrega genera problemas importantes, como lo son:

- Deficiencia en la calidad del software: Se vuelven más propensos a aparecer errores y defectos, lo que puede resultar en soluciones inestables y propensas a fallos.
- Retrasos y desviaciones en los proyectos: La ausencia de una gestión efectiva y la omisión de buenas prácticas de desarrollo pueden provocar retrasos en la finalización de los proyectos.
- Gastos altos: Corregir errores y defectos después del desarrollo inicial puede ser costoso.
- Desafíos en el mantenimiento: El software mal diseñado y documentado puede ser complicado de mantener.

- Inseguridad: La falta de buenas prácticas de seguridad puede exponer las aplicaciones a vulnerabilidades y ataques, poniendo en riesgo la seguridad de los datos.
- Baja productividad: La carencia de automatización y prácticas eficientes puede dar lugar a una baja productividad en el equipo de desarrollo.
- Conflictos y problemas de comunicación: La falta de buenas prácticas de comunicación y colaboración puede originar conflictos dentro del equipo y una comprensión insuficiente de los requisitos del cliente.
- Desmotivación del equipo de trabajo: La carencia de buenas prácticas en gestión y liderazgo puede desmotivar al equipo de desarrollo, afectando la calidad de cada componente construido.
- Incumplimiento de normativas y estándares: La falta de cumplimiento como aplicación de normas y estándares de la industria, puede derivar en problemas legales y de regulación.
- Insatisfacción del cliente: La entrega de software defectuoso o que no satisface los requisitos puede provocar la insatisfacción del cliente.
- Afectación negativa en la reputación de la empresa: La entrega de software de baja calidad puede perjudicar la reputación de la empresa y afectar las relaciones con los clientes.
- Cuestiones éticas y de privacidad: La falta de consideración de la ética y la privacidad en el desarrollo de software puede dar lugar a problemas legales y de privacidad.

La ausencia de buenas prácticas en el desarrollo de software puede tener un efecto alto en la calidad, eficiencia y satisfacción de todas las partes involucradas, incluyendo a los desarrolladores, los clientes y los usuarios finales. Por lo tanto, es esencial adoptar y fomentar

buenas prácticas en el desarrollo de software para mitigar estos problemas. Surge así el siguiente interrogante: ***¿Cuáles son las metodologías y herramientas básicas para garantizar la calidad en el desarrollo de software en las pequeñas y medianas empresas de Bogotá?***

Objetivos

Objetivo general

Presentar un diseño con los procesos fundamentales para la implementación de buenas prácticas de desarrollo, con el propósito de mejorar la calidad del software, la eficiencia de los equipos de desarrollo y satisfacer las necesidades al igual que los requerimientos de sus clientes.

Objetivos específicos

- Identificar las debilidades, desventajas y barreras que impiden que las pymes apliquen buenas prácticas de desarrollo seguro.
- Realizar unas sugerencias y puntos claves con las certificaciones básicas que una pyme puede incorporar en las fases del desarrollo de software
- Realizar un cuadro comparativo con las arquitecturas más utilizadas en el mercado.
- Presentar los marcos o enfoques de seguridad globalmente conocidos, destacando los ítems relevantes que una Pyme debe tener en cuenta.

Justificación

Las fábricas de desarrollo de software, sin procesos para implementar buenas prácticas de desarrollo o metodologías, a menudo enfrentan problemas de reputación, desafíos legales, reprocesos e inversiones inesperadas. Estos inconvenientes están estrechamente relacionados

con la falta de personal debidamente capacitado, el desconocimiento de las metodologías más usadas en el mercado y la presión por las entregas del producto.

En este contexto, es fundamental difundir y proporcionar conocimientos accesibles sobre las metodologías de desarrollo existentes, ya que son cruciales para garantizar la calidad, eficiencia y sostenibilidad de los proyectos de desarrollo. (Carrizo & Alfaro, 2018; Carrizo & Alfaro, 2018)

Es importante que estas empresas tengan en cuenta las metodologías, técnicas, herramientas y las mejores prácticas en todo el ciclo de vida que involucra el desarrollo, así como el cumplimiento de la normativa legal y la evaluación constante de riesgos y amenazas.

El conjunto de estas metodologías, marcos de trabajo y buenas prácticas permite entregar el producto final con calidad, reducir los sobrecostos, optimizar los tiempos de desarrollo y hacer que el producto sea más fácil de mantener y escalar. Sin embargo, el punto más importante es cuando las empresas se vuelven más competitivas y satisfacen las expectativas de sus clientes.

En la actualidad ha surgido bastante preocupación en concerniente a la seguridad de los sistemas de información con la llegada de la WEB 3.0 y WEB 4.0, términos definidos por Zeldman (2006), que amplían el acceso a la Internet desde la IoT (Internet de las cosas) hasta la IA (inteligencia artificial) (Briceño, 2023). Esta evolución de la Internet conlleva a que en el desarrollo de software se deba garantizar la seguridad de los datos y la información de forma más estricta y restrictiva.

Marco Teórico

Modelos de mejoramiento de procesos.

a. CMMI

El *Capability Maturity Model Integration* (CMMI), es un grupo de prácticas y modelos que se utilizan para mejorar procesos de desarrollo y gestión de proyectos en las organizaciones (Isaca, 2023). CMMI provee un marco de referencia o entorno cuyo objetivo es, aportar a las organizaciones a valorar y perfeccionar su capacidad para desarrollar y mantener sus productos o servicios con una alta calidad.

Las empresas con actividades económicas, como la ingeniería de software y la tecnología de la información, por lo general requieren que sus procesos cumplan con ciertos requisitos de calidad para consolidarse como software de calidad y seguro. Estas buenas prácticas, reunidas, son la ayuda que brinda el CMMI para potenciar el rendimiento de las compañías.

El modelo CMMI está dividido en niveles o fases de madurez, representados en diferentes etapas de mejora en una organización (University, 2006) y son llamados así:

Niveles de capacidad

Nivel	Descripción
Nivel 0: Incompleto	Es un proceso que no se realiza o se lleva a cabo parcialmente. No se cumplen uno o varios de los objetivos específicos del área de proceso. En este nivel, no se satisfacen los objetivos genéricos, ya que no hay razón para institucionalizar un proceso que se realiza parcialmente.
Nivel 1: Realizado	Un proceso realizado es aquel que satisface los objetivos específicos del área de proceso y apoya el trabajo necesario para producir productos de trabajo. Aunque el nivel de capacidad 1 conlleva a mejoras significativas, estas mejoras pueden perderse con el tiempo si no se institucionalizan.
Nivel 2: Gestionado	Existe una definición y documentación de los procesos en la organización. Hay trazabilidad y recurrencia en la organización para cumplir con las buenas prácticas.
Nivel 3: Definido	Hay mediciones cuantitativas para la gestión y mejora de procesos dentro de la organización.
Nivel 4: Gestión cuantitativa	La organización busca la excelencia, innova y persigue la mejora continua de sus procesos mediante técnicas estadísticas y cuantitativas..
Nivel 5: Optimizado	El objetivo de un proceso de optimización se centra en la mejora continua del rendimiento mediante mejoras incrementales e innovadoras.

Fuente: propia

b. IT- MARK

Es un esquema de certificación de calidad en los procesos de trabajo técnico, de seguridad y de negocio, otorgado a empresas pequeñas y pymes. Cuenta con tres tipos de certificaciones.

- IT Mark: certifica a la empresa que cuenta con procedimientos habitualmente bajo control.
- IT Mark Premium: acredita a la empresa con buena madurez en sus procesos.
- IT Mark Elite: acredita a la empresa que tienen un nivel superior en la definición e institucionalización. ((ESI), 2017)

Este tipo de certificación es compatible y está alineado con el modelo CMMI V1.2, específicamente con CMMI DEV para los procesos de desarrollo y administración de software, y para los procesos de seguridad de la información está alineado con la norma ISO 27001:2005.

La metodología de certificación consta de 4 pasos:

1. Seminario: Introducción a la empresa en los conceptos y reglas necesarios para obtener la certificación.
2. Evaluación: evaluar la organización en los procesos actuales con el fin de obtener resultados e identificar las acciones de mejora.
3. Reevaluación: proceso que se realiza después de implementar las acciones de mejora reconocidas en el proceso de evaluación con el fin de volver a evaluar los procesos que no cumplieron con la calificación deseada.
4. Validación: Evaluación que se realiza un año después de haber obtenido la certificación, con el objetivo de asegurar el mantenimiento de los procesos. (Red Colombiana de Calidad del Software (RCCS), 2014)

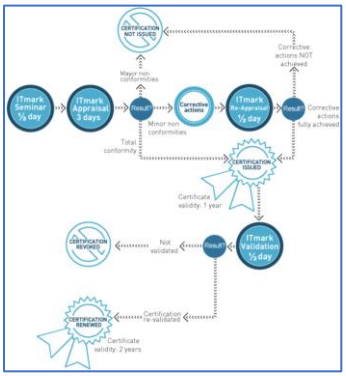


Ilustración 1 it-mark, 2023

c. Agile

Es un enfoque y un marco de referencia relacionado con la gestión de proyectos, adicionando elementos de agilismo al desarrollo de software, haciendo énfasis en:

- La entrega incremental.
- La colaboración
- La flexibilidad

Las metodologías ágiles generan en los equipos una adaptación a los cambios rápidamente y así adecuarse de manera más eficaz y efectiva a las necesidades demandantes por los usuarios The Agille Manifesto (2001)

Las fases de las metodologías agiles son:

- Iteración: corresponde a los períodos para desarrollo, avance y entrega de una funcionalidad dentro del desarrollo del software.
- Colaboración: los equipos son interdisciplinarios y se comunican entre sí para integrar los productos, cumpliendo con el objetivo general.
- Entrega incremental: se desarrolla en pequeñas partes funcionales que se entregan de manera incremental en lugar de esperar a tener el producto completo al final del proyecto.
- Flexibilidad: son permitidos los cambios en cualquier etapa del proyecto. Los requisitos pueden cambiar a medida que se obtiene retroalimentación del cliente o usuario final.
- Autoorganización: Los equipos Ágiles tienen un alto grado de autonomía y toman decisiones relacionadas con el desarrollo del producto.
- Adaptabilidad: La metodología Ágil permite a los equipos ajustar sus procesos y enfoques según las necesidades específicas de cada proyecto (proyecto, 2021)

Ciclo de vida del Software

El concepto de ciclo de vida del software abarca diversas etapas que un producto de software atraviesa, comenzando desde su concepción hasta su eventual retirada o cuando el desarrollo termina de ser útil para pasando a ser descontinuado. (Antonio Fernández-Medina, 2022)

Las fases del software:

- Planificación: Esta etapa concibe el alcance del proyecto, estableciendo los requisitos y necesidades del software y se desarrolla un plan de ejecución.
- Análisis: Realiza el análisis del problema como sus dominios con el fin de identificar los requisitos funcionales como no funcionales del software.
- Diseño: Fase en la cual se diseña la arquitectura del software y se crea la documentación del diseño.
- Implementación: Pasos y procesos donde se codifica el software y se realizan las pruebas unitarias.
- Pruebas: En esta fase, se realizan las pruebas de integración, las pruebas de sistema y las pruebas de aceptación.
- Implementación: se instala el software en el entorno de producción.
- Mantenimiento: se realizan las actualizaciones y correcciones del software.

Herramientas para la medición de código en software:

El uso de herramientas de medición durante el desarrollo de software se realiza para mejorar la calidad del código, la productividad, durante la medición podemos implementar diversos tipos de métrica como de tamaño, complejidad y calidad (McDonald, 2022). Hay diversas herramientas para la medición de código como SonarQube, Codacy, Checkstyle, Cloc, Clang.

Arquitecturas de Software

(Kleppmann, 2022) explica los diferentes estilos arquitectónicos, sus ventajas y desventajas, y cómo se pueden aplicar en diferentes contextos. Las arquitecturas más utilizadas según el estudio realizado por (Gartner, 2023) a 3.000 profesionales, son: Arquitectura de microservicios,

Arquitectura basada en eventos y Arquitectura de contenedores, entre otros. (Ver imagen comparativa de arquitecturas actuales).

Tecnología	Características	Ventajas	Ventajas
Arquitectura monolítica	Una aplicación única, compuesta de un solo componente.	* Simplificación del desarrollo y mantenimiento. * Facilidad de comprensión y gestión	* Dificultad para escalar y mantener. * Mayor riesgo de errores
Arquitectura cliente-servidor	La aplicación se divide en dos componentes: el cliente y el servidor. El cliente proporciona la interfaz de usuario y el servidor proporciona la lógica de negocio	* Fácil de escalar * Mayor flexibilidad	* Mayor complejidad de desarrollo y mantenimiento * Mayor riesgo de errores
Arquitectura en capas	La aplicación se divide en capas, cada una con un propósito específico.	* Simplificación del desarrollo y mantenimiento. * Fácil de escalar y mantener.	* Dificultad para escalar y mantener. * Mayor complejidad de diseño
Arquitectura monolítica	Una aplicación única, compuesta de un solo componente.	Fácil de escalar y mantener.	Mayor complejidad de diseño.
Arquitectura orientada a servicios (SOA)	La aplicación se compone de servicios independientes que se comunican entre sí a través de una red.	Fácil de escalar y mantener.	* Mayor complejidad de diseño. * Mayor complejidad de diseño y desarrollo
Arquitectura monolítica	Una aplicación única, compuesta de un solo componente.	Fácil de escalar y mantener.	Mayor complejidad de diseño y desarrollo.
Arquitectura microservicios	La aplicación se ejecuta en la nube y no requiere servidores físicos.	Fácil de escalar y mantener.	Mayor complejidad de diseño y desarrollo.
Arquitectura serverless	La aplicación se ejecuta en la nube y no requiere servidores físicos.	Facilidad de desarrollo y mantenimiento	Mayor dependencia de la nube.

Ilustración 2 imagen comparativa de arquitecturas actuales

ISO/IEC 12207:2008 - ***Ingeniería de sistemas y software - Proceso del ciclo de vida del software***

Es un estándar internacional y un framework que define los procesos, actividades y tareas en el ciclo de vida del software, se establece una estructura y conjunto de procesos, desde los requisitos hasta la entrega del producto siendo también una herramienta para la gestión de los riegos y la mejora continua en el desarrollo del software. (Organización Internacional de Normalización (ISO) y Comisión Electrotécnica Internacional (IEC), 2008)

Frameworks de seguridad

Definición Ciberseguridad

La ciberseguridad engloba un conjunto diverso de herramientas, regulaciones, conceptos de seguridad, dispositivos especializados, directrices, estrategias de gestión de riesgos, acciones concretas, capacitación, mejores prácticas, garantías y tecnologías diseñadas para salvaguardar tanto el entorno digital como los valiosos activos de los usuarios y las organizaciones. Estos activos abarcan desde los equipos informáticos interconectados hasta el personal, la infraestructura, las aplicaciones, los servicios, los sistemas de telecomunicaciones, así como la totalidad de la información transmitida o almacenada en el ciberespacio. El esfuerzo en ciberseguridad tiene como objetivo fundamental asegurar el éxito y la integridad de los activos de los usuarios y de la organización, protegiéndolos contra los riesgos significativos en el ámbito de la ciberseguridad. (International Telecommunication Union (ITU), 2018)

Los objetivos generales de la seguridad en este contexto son:

- **Disponibilidad:** Asegurar que los sistemas y recursos críticos estén siempre accesibles y funcionando cuando se necesiten, evitando interrupciones no planificadas.
- **Integridad (incluyendo autenticidad):** Mantener la integridad de los datos y recursos, lo que significa que estos no deben ser alterados de manera no autorizada. La autenticidad se refiere a verificar que los datos provengan de fuentes confiables y sean genuinos.
- **Confidencialidad:** Proteger la información sensible y restringir el acceso solo a personas autorizadas, evitando cualquier divulgación no autorizada o filtración de datos. Estos objetivos constituyen pilares fundamentales en la estrategia de ciberseguridad, y su

consecución se traduce en un entorno digital más resiliente y seguro para todos los involucrados.

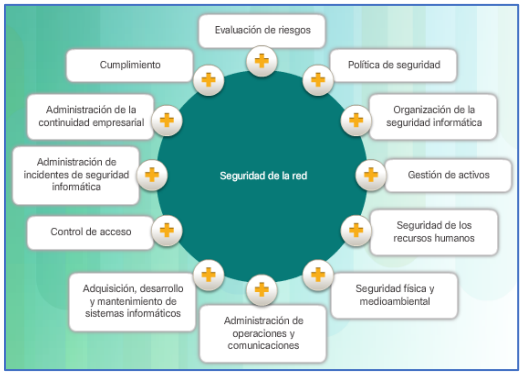


Ilustración 3 Dominio de Ciberseguridad ISO/IEC 27000

- **Valoración de riesgos:** Este constituye el punto de inicial en el proceso de gestión de riesgos. Consiste en determinar el valor, tanto cuantitativo como cualitativo, asociado a un riesgo específico o una amenaza identificada.
- **Parámetros de seguridad**: dentro de este documento establece las condiciones y comportamientos esperados de las partes de la organización, a menudo detallando cómo se accede a los datos y quiénes tienen permiso para acceder a información específica.
- **metodologías de seguridad la informática:** Se refiere al modelo de gestión implementado por una organización para garantizar la seguridad de la información.
- **Gestión de recursos:** Involucra la creación de un inventario y una estructura de categorización para los recursos de información.
- **Seguridad en el talento humano:** Este aspecto se centra en los procedimientos de seguridad relacionados con el ingreso, la movilidad y la salida de empleados dentro de la empresa.

- **Seguridad física y medioambiental:** Esta categoría aborda la protección de los sistemas físicos informáticos dentro de la corporación.
- **Gestión de operaciones y comunicaciones:** Se refiere a la administración de los controles de seguridad técnicos en sistemas software/hardware y redes.
- **Implantaciones, desarrollo y mantenimiento de sistemas de informática:** Describe cómo se integra la seguridad en las aplicaciones durante las fases del ciclo de vida, aplicado a los sistemas informáticos desarrollados.
- **Control de acceso:** Engloba la limitación de los derechos como permisos de acceso a la red, sistemas, software, funciones y datos.
- **Administración de incidentes de seguridad informática:** Este aspecto se enfoca en cómo crear procesos de atención, proactividad y respuesta a las violaciones de seguridad a los sistemas informáticos.
- **Administración de la continuidad empresarial:** Describe las acciones destinadas a resguardar, mantener y recuperar sistemas y procesos críticos para garantizar la continuidad de las operaciones.
- **Cumplimiento:** Se refiere al proceso de asegurarse de que se cumplan las políticas, normas estándares y regulaciones de seguridad de la información. (International Telecommunication Union (ITU), 2018)

A continuación, se encuentra la lista de normas y estándares más comunes:

a. ISO 27034

La norma ISO 27034 es un estándar internacional enfocado en la seguridad durante los procesos de desarrollo de software, ofrece directrices y buenas prácticas para la incorporación de medidas de seguridad durante las fases relacionadas con el ciclo de vida del software. La normativa consta de siete partes o capítulos que tratan diversos aspectos del desarrollo seguro de software. A continuación, se presenta una recapitulación de cada una de estas divisiones:

- **27034-1:** Esta parte, establece que no es un estándar para el desarrollo de software o de gestión para los proyectos de aplicaciones ni ciclo de desarrollo de software. Su objetivo es facilitar una guía general.
- **27034-2:** Explica la estructura del Marco Normativo de la Organización (ONF), que abarca políticas, herramientas y procedimientos relacionados con la seguridad del software desarrollado.
- **27034-3:** Describe el proceso general para gestionar la seguridad en aplicaciones específicas utilizadas por una organización.
- **27034-4:** Aunque inicialmente se planeó, este estándar se canceló y se reanudó como un nuevo trabajo. Se espera su publicación en 2021 y abordará la validación de la seguridad de aplicaciones.
- **27034-5:** Precisa la estructura de datos de Control de Seguridad del software (ASC), proporcionando requisitos y representaciones gráficas.

- **27034-6:** Ofrece ejemplos sobre cómo desplegar y documentar todos los controles en funciona de mejorar la seguridad sobre las aplicaciones, al igual parámetros de cómo gestionar la seguridad de la información durante el desarrollo de software.
- **ISO 27034-7:** Nos propone un marco para garantizar de manera optima la seguridad en los programas de software que dependen de otros para funciones críticas de seguridad.

En resumen, ISO 27034 ofrece un enfoque integral para el desarrollo seguro de software, desde la explicación de términos y conceptos hasta la descripción de procesos y métricas para garantizar la seguridad en todas las etapas del ciclo de vida del software. Esta norma es de gran utilidad para organizaciones que buscan mejorar la seguridad de cada una de las aplicaciones y sistemas de software desarrollados.

b. OWASP (top 10, cada cuanto publican el top 10)

Open Worldwide Application Security Project (OWASP) por más de 20 años es una fundación con el objetivo de perfeccionar la seguridad del software, la cual cuenta con una comunidad abierta dedicada y respaldo de corporaciones, fundaciones, desarrolladores y voluntarios para permitir que las organizaciones conciban, desarrollen, adquieran, operen y mantengan aplicaciones confiables (Owasp, s.f.). Su relevancia se fundamenta en varios aspectos esenciales:

Concientización sobre Seguridad: OWASP desempeña un rol crucial al crear conciencia dentro de la comunidad de desarrolladores, expertos en seguridad y empresas, acerca de las amenazas y retos que involucra la seguridad de las aplicaciones.

Recursos Abiertos: La fundación pone a disposición de toda una amplia gama de recursos, herramientas y guías de seguridad de aplicaciones de código abierto y de acceso gratuito. Esto facilita que tanto los desarrolladores como los profesionales de seguridad accedan a conocimientos y utilidades para fortalecer la seguridad de sus aplicaciones.

Estandarización: OWASP ha contribuido a establecer estándares y prácticas recomendadas en el campo de la seguridad de aplicaciones, como el reconocido "OWASP Top Ten", que enumera las principales vulnerabilidades de seguridad en el desarrollo de software.

Comunidad Activa: OWASP cuenta con una comunidad activa formada por profesionales de seguridad y desarrolladores de diversas partes del mundo. Esto fomenta el intercambio de experiencias, conocimientos y la colaboración en proyectos de seguridad.

Eventos y Conferencias: La organización realiza conferencias y eventos educativos que reúnen a expertos en seguridad y desarrolladores para analizar, compartir información sobre las tendencias e identificar amenazas actuales en el campo de la ciberseguridad.

La creación más relevante de OWASP, es el "OWASP Top Ten". Se trata de una lista que se actualiza en periodos de 4 años con el fin de identifica las diez (10) vulnerabilidades principales de seguridad en aplicaciones de software. Este catálogo es esencial para que los desarrolladores y profesionales de seguridad prioricen sus esfuerzos en la mitigación de riesgos.

Cabe mencionar que OWASP también desarrolla muchos otros proyectos y recursos valiosos, como herramientas de seguridad de tipo open source, directrices de mejores prácticas y documentos de referencia, que abordan diversos aspectos concernientes con la seguridad de las aplicaciones web. OWASP como organización sigue desempeñando un papel muy

transcendental en el desarrollo de la seguridad aplicado al desarrollo de software y en la protección contra las amenazas cibernéticas.

El ultimo "OWASP Top Ten" fue actualizado en 2021, en la imagen a continuación se observan los riesgos organizados desde el relevante:

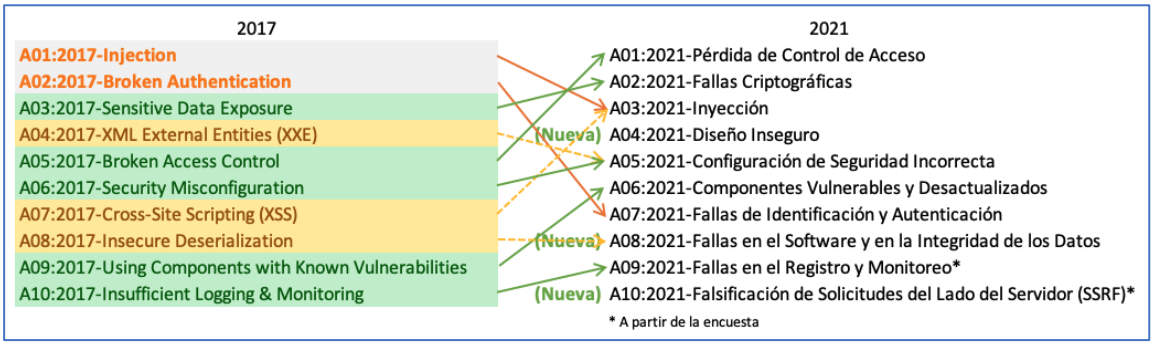


Ilustración 4 Owasp Top 10 for 2021

Dentro de la última revisión de "OWASP Top Ten" destaca que está encaminada en la protección de los datos, ocho de sus categorías corresponde a datos analizados y dos ítems restantes corresponde a la encuesta realizada a la comunidad.

Para seleccionar las 10 categorías más importantes se analizan las siguientes situaciones:

- CWEs mapeadas
- Tasa de incidencia de CWE
- Cobertura (de pruebas)
- Explotabilidad ponderada de las CVEs.
- Impacto ponderado de las CWEs.
- Total de ocurrencias de una categoría.

- Total de CVEs de una categoría

OWASP ha desarrollado un modelo de madures denominado OWASP SAMM (Software Assurance Maturity Model) está enfocado principalmente en la evaluación y el fortalecimiento de la madurez de la seguridad cibernética en el proceso ligados al desarrollo de software. Ofrece un conjunto organizado de pautas para reconocer, cuantificar y perfeccionar las prácticas de seguridad en durante todas las etapas del ciclo de vida del software.

El modelo está representado bajo los pilares Gobernanza, Diseño, Implementación, verificación y Operaciones los cuales tienen una mejor practica definida para cada área, como se ilustrada en la siguiente imagen:

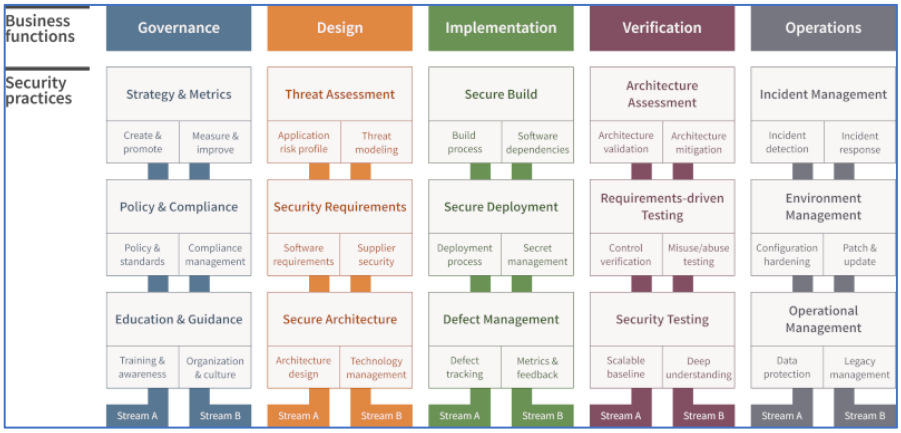


Ilustración 5 Owasp SAMM

c. NIST APPLICATION SECURITY

El NIST Secure Software Development Framework (SSDF) es un conjunto de prácticas de desarrollo de software seguras que se fundamentan en las mejores prácticas. El SSDF tiene como objetivo apoyar a las organizaciones a desarrollar software seguro y confiable.

El SSDF se divide en cuatro áreas principales:

- **Integración de la seguridad en el desarrollo:** El SSDF proporciona orientación sobre cómo integrar la seguridad durante las fases y actividades de desarrollo de software. Esto incluye actividades como la evaluación de riesgos, la planificación de la seguridad, la implementación de controles de seguridad y examinar la seguridad.
- **Protección del software:** El SSDF proporciona orientación sobre cómo proteger el software de amenazas y vulnerabilidades. Esto incluye actividades como la gestión de configuración, la seguridad de las aplicaciones web y la seguridad de los datos.
- **Gestión de la seguridad del software:** El SSDF proporciona orientación sobre cómo gestionar la seguridad del software durante todos los procesos relacionados con el ciclo de vida. Incluyendo lo relacionado con las actividades de auditoría de seguridad, la capacitación de seguridad y la gestión de incidentes de seguridad.
- **Mejoramiento de la seguridad del software:** El SSDF proporciona orientación sobre cómo mejorar la seguridad del software de forma continua. Esto incluye actividades como la investigación al igual que el desarrollo de nuevas técnicas de seguridad, la adopción de nuevas tecnologías y la colaboración con la comunidad de seguridad.

Ilustración 6 SSDF NIST

Ejemplos de cómo las organizaciones han utilizado el SSDF para mejorar la seguridad del software:

- Una organización que utilizó el SSDF para reducir el número de vulnerabilidades de seguridad en su software de un 50% a un 20%.
- Una organización que utilizó el SSDF para mejorar la seguridad de su software web al implementar controles de seguridad estandarizados de la industria.
- Una organización que utilizó el SSDF para optimizar la seguridad de sus datos al implementar políticas y procedimientos de gestión de datos sólidos.

Metodología

Primer Nivel
Enfoque, alcance y diseño de la investigación

Este proyecto relacionado con investigación busca obtener datos objetivos, medibles y generalizables, por ende, es necesario realizar un análisis estadístico, riguroso en los resultados. Consecuentemente, este caso de estudio es enfocado a un pequeño sector de las Pymes de Bogotá, durante el último semestre del año 2023. Por consiguiente, esta es una investigación causal comparativa.

Definición de Variables

Esta investigación centra su esfuerzo en recolectar datos de entrevistas a miembros del área de la tecnología de las pymes. Los entrevistados, generalmente están involucrados en el análisis, diseño, implementación y pruebas de las pymes en Bogotá. A continuación, se muestran la definición y la categorización de las variables:

Variables independientes

- **Metodologías de desarrollo de software.**

Es el enfoque, marco referencial o metodológico utilizado en la pyme consultada.

- **Métodos de gestión de proyectos**

Son los procesos que se aplican durante la gestión de proyectos concernientes con el desarrollo de software, que incluyen planificación, control y seguimientos.

- **Técnicas procedimentales de desarrollo de Software**

Los procesos utilizados en el desarrollo de software; diseño, implementación pruebas unitarias, y documentación.

- **Procesos de garantía de calidad**

Se refiere a los procesos, implementaciones y certificaciones que se le realizan al software implementado.

Variables dependientes

- **Índice de calidad del software**

Corresponde al software producido, teniendo como referencia los indicadores tales como: cantidad de errores reportados, retroalimentaciones, eficiencia.

- **Madurez en el desarrollo de software.**

Se refiere a la madurez en la que se encuentra la compañía en la implementación de procesos, mejorando la calidad al igual que la eficiencia durante el desarrollo del software y su mantenimiento post-implementacion.

Población y Muestra

En Bogotá existen 4.014 empresas de desarrollo de software conforme con la información estadística del DANE, bajo la Encuesta Anual Manufacturera.

Utilizando muestreo probabilístico se realizará una encuesta a 25 empresas en la ciudad de Bogotá. Este estudio incluye solo empresas que pertenezcan al a la actividad económica CIIU 6201 (Actividades de desarrollo de sistemas informáticos) de conformidad a los códigos de clasificación de actividades económicas definitivas en la Cámara de Comercio de Bogotá.

Segundo nivel

Selección de métodos o instrumentos para recolección de información

Para este estudio se realizará la correlación de los datos utilizando una encuesta una persona por empresa; esta encuesta se enviará por el canal de correo electrónico para ser diligenciada en línea desde cualquier computador o teléfono móvil con el apoyo de Microsoft Froms. Permitiendo entregar la encuesta de manera más ágil a la muestra de la población, donde cada individuo puede diligenciar el cuestionario en cualquier momento y lugar.

Para la presente investigación se ejecutará las siguientes etapas:

- Selección de empresas a entrevistar.
- Diseño del instrumento que diligenciara la muestra para recolectar los datos.
- Envió de la encuesta a la muestra poblacional través de medios digitales.
- Tabulación, creación de representaciones de los datos y análisis de resultados.
- Generación de conclusiones relacionadas con los resultados obtenidos.

Técnicas de análisis de datos

Para poder comprender las encuestas realizadas es indispensable analizar los datos recopilados de la encuesta ejecutada a 25 organizaciones de desarrollo de software, esto permitirá una mejor comprensión e identificación de los datos. Para cumplir con este objetivo se utilizaran el análisis con orientación cuantitativa permitiendo sintetizar y visualizar los datos en graficas que permitan mostrar la información de un modo claro y sencilla; verificar la hipótesis y responder a la pregunta de la investigación, también ser elaborara el análisis de frecuencia para evaluar la distribución de respuestas categóricas y la periodicidad cada respuesta, permitiendo identificar patrones y tendencias en los resultados obtenidos, sin olvidar verificar la correlación de la variables para evaluar relaciones entre ellas y elementos que pueden tener causalidad.

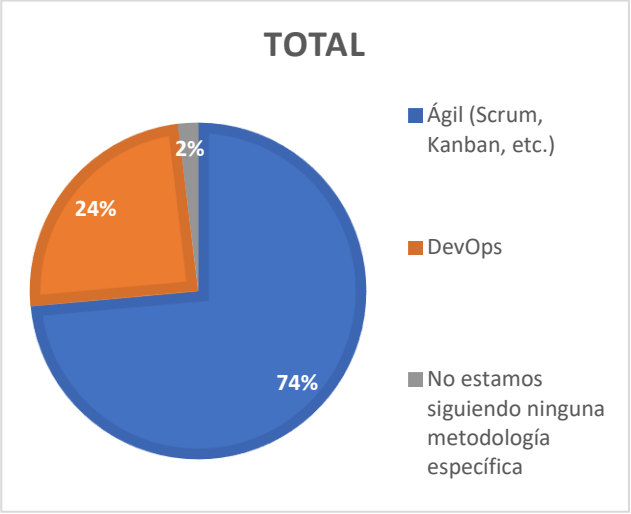
Las técnicas de análisis de datos serán una parte esencial de esta investigación, estas permiten transformar datos crudos en información significativa y útil para poder tomar decisiones al igual que generare conocimiento para la comunidad de desarrollo de software.

Análisis y discusión de los resultados

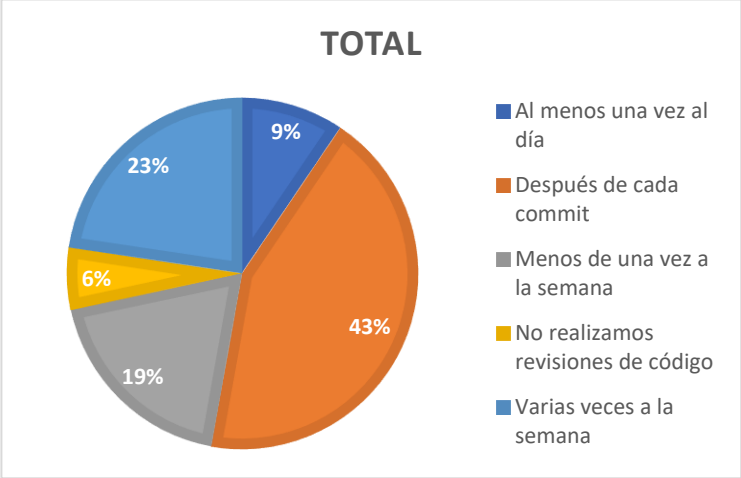
De acuerdo con la rapidez exponencial en la que crecen los avances tecnológicos, las empresas se ven obligadas a mejorar, reinventar y reformular sus procesos para adaptarse a las necesidades y herramientas tecnológicas, integrarse a los entornos y plataformas tecnológicas y poder certificar sus procesos con calidad, para hacer de sus productos confiables y escalables.

La encuesta realizada permitió identificar cuales procesos implementados en la actualidad en cuanto a la calidad y mejores prácticas para el desarrollo de software, en la siguiente tabla mostramos los resultados obtenidos por cada empresa:

Tabla 1 Resultados obtenidos por cada pregunta

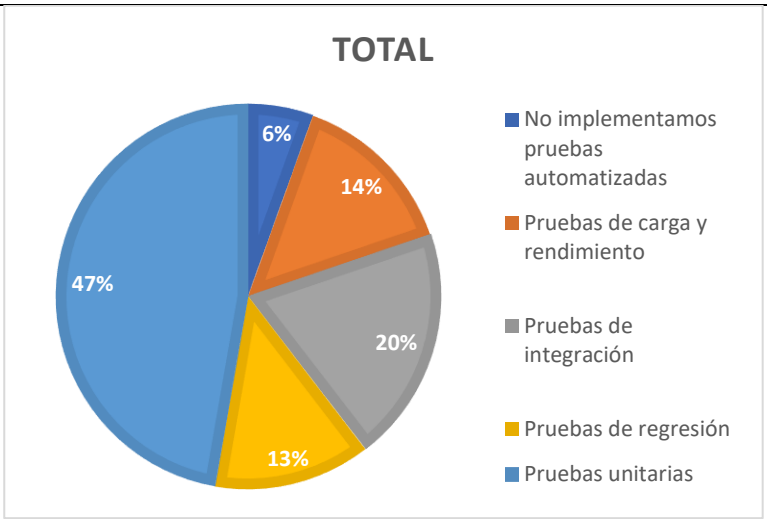
<table>
<tr><th>Pregunta</th><th>Resultado</th></tr>
<tr><td>1. ¿Qué metodología de desarrollo de software se utiliza en su empresa?</td><td>Ilustración 7 Resultado de la pregunta 1
 

Fuente: elaboración propia</td></tr>
<tr><td colspan="2">La selección de una metodología de desarrollo es uno de los procesos importantes para el desarrollo de un proyecto (Blandón-Jaramillo, 2023), el 74% de las empresas encuestadas usa la metodología ágil en sus proyectos de desarrollo de software, el 24% de las empresas usa el conjunto de prácticas de DevOps y el 2% no usa una metodología. Vemos que al menos más de la mitad de las empresas usan una metodología ágil.</td></tr>
</table>

Pregunta	Resultado
2. **¿Con qué frecuencia se realizan revisiones de código en su equipo?**	*Ilustración 8 Resultado de la pregunta 2*  *Fuente: Elaboración propia*

En del desarrollo de software requerimos del control, uso y administración de repositorios para guardar el código fuente de una aplicación. El 43% de las empresas acostumbran a realizar la revisión del código después de cada commit captura instantánea de los cambios realizados en un script de un proyecto (Atlassian, 2023)) y el 6% de estas empresas no realiza revisiones de código.

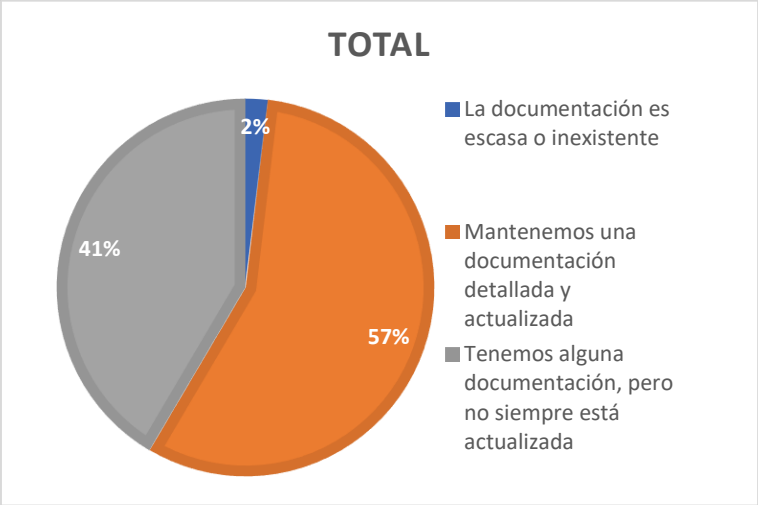
Pregunta	Resultado
3. **¿Qué tipo de pruebas implementa**	*Ilustración 9 Resultado de pregunta 3*

en su proceso de entrega desarrollo?	 *Fuente: Elaboración propia*

El proceso de calidad del software es una fase importante en el ciclo de vida del software, hay variedad de tipos de pruebas como pruebas funcionales y no funcionales (buscar referencia).

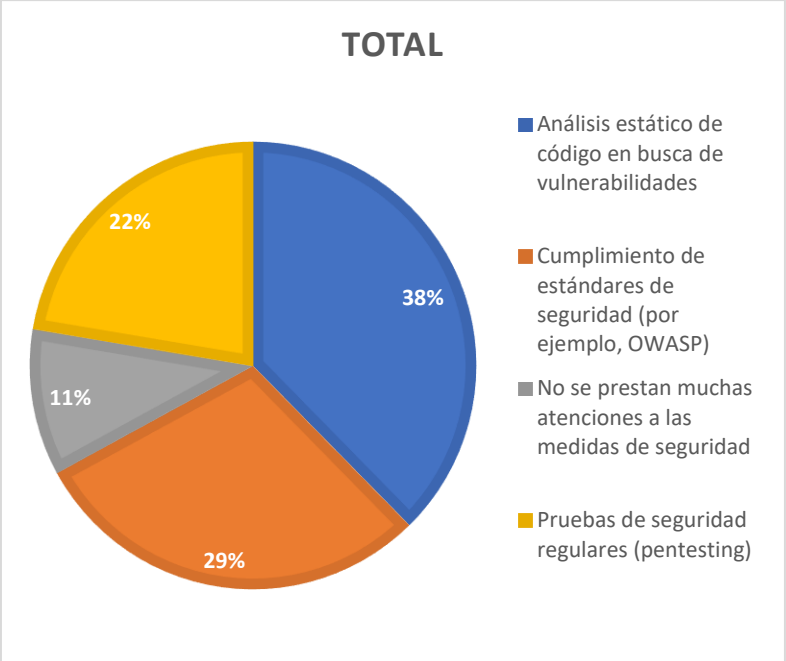
El 49% representa el uso de las pruebas unitarias incluidas en la fase de desarrollo (buscar referencia), el 14% realiza pruebas de carga y rendimiento (buscar referencia), el 20% hacen pruebas de rendimiento y carga por último el 6% no realiza ningún tipo de prueba al producto.

Pregunta	Resultado
4. ¿Cómo manejan la documentación del	*Ilustración 10 Resultado pregunta 4*

código y los requisitos?	 *Fuente: Elaboración propia}*

La documentación del software en es un proceso que también está incluido en el ciclo de vida del software, vemos que el 57% de las empresas manejas esta documentación sin embargo el 41% que tiene alguna documentación, pero no siempre está actualizada y puede representar un riesgo para el entendimiento del código, y el 2% cuenta con documentación escasa o inexistente.

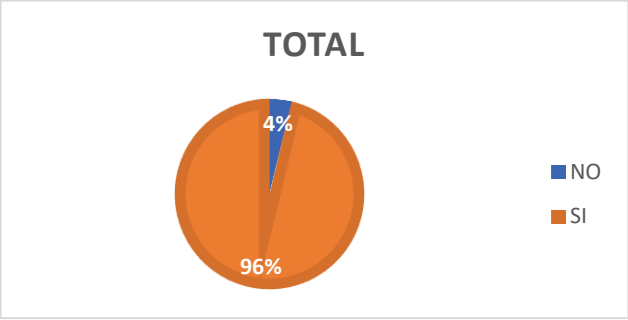
Pregunta	Resultado
5. ¿Qué medidas de seguridad se implementan durante el desarrollo del software?	*Ilustración 11 Resultado pregunta 5*



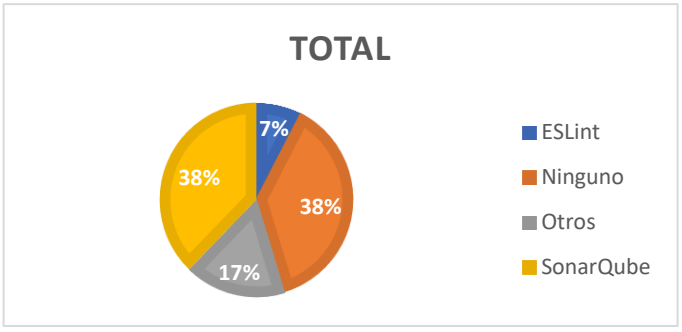
Fuente: Elaboración propia

Las medidas de seguridad deben ser contempladas en el proceso de desarrollo del software (Anderson, 2023), el 38% de las empresas realizan un análisis estático de código (Hidalgo, 2023) , el 29% realizan seguimiento de estándares de seguridad con respecto a páginas oficiales como OWASP, el 22% realizan pruebas de seguridad regulares y por últimos el 11% de las empresas encuestadas no prestan mucha atención a las medidas de seguridad.

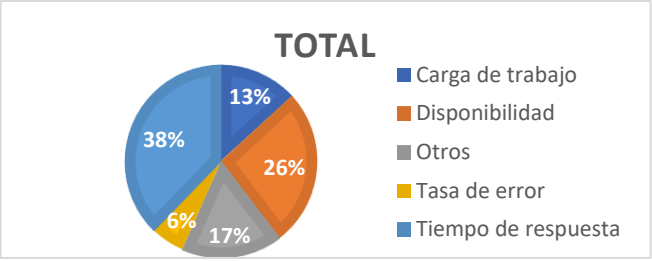
Pregunta	Resultado
6. **¿La compañía tiene identificados roles y responsabilidades**	*Ilustración 12 Resultado pregunta 6*

claras para la implementación y desarrollo de Software?	 *Fuente. Elaboración propia*

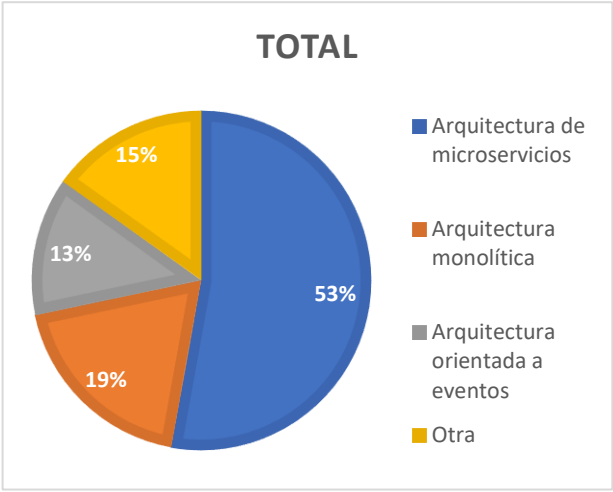
La identificación de roles y responsabilidades es fundamental para el éxito de cualquier proyecto de software (José Manuel Arévalo, 2023), el 96% de las empresas son demuestra que este proceso es importante puesto que estos roles apoyan en la comunicación efectiva, reducen riesgos de errores y aumentan la eficiencia en el equipo, y solo el 4% no tienen identificados los roles necesarios.

Pregunta	Resultado
7. ¿Cuál de las siguientes herramientas para la medición de código realizado utiliza con frecuencia?	*Ilustración 13 Resultado pregunta 7*  *Fuente: Elaboración propia*

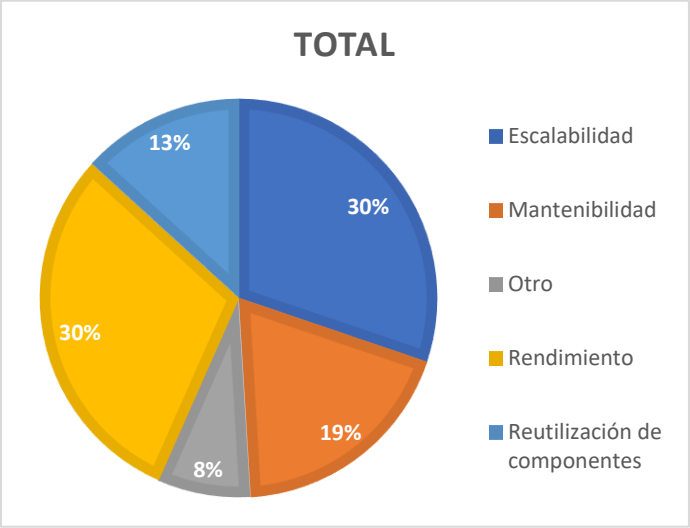
La afirmación la realiza (Díaz, 2023) puesto que es un beneficio en la medición del código, existen varias herramientas para la medición de código, desde herramientas de pago y otras gratis o licencia abierta, el 39% de las empresas no usa al menos una herramienta para la medición del código, el 38% de las empresas usan Sonar Qube para llevar a cabo la medición, el 7% usan ESlint y el 17% usa otra herramienta, en total, el 62% de las empresas cumple con medir el código aportando a la calidad del producto.

Pregunta	Resultado
8. ¿Su compañía genera métricas de desempeño de acuerdo con uno o más de los siguientes ítems?	*Ilustración 14 Resultado pregunta 8*  *Fuente: Elaboración propia*

Las métricas de desempeño según los encuestados a los perfiles en esta encuesta, se tiene que el ítem Carga de Trabajo, es más relevante para las compañías con un 38%, junto con 26% en la disponibilidad del colaborador. Las métricas de desempeño son esenciales para el éxito en un proyecto y como estándar de procesos, porque las gráficas de rendimiento de los colaboradores indican la velocidad del equipo. (Rigby, Elk, & Berez, 2020)

Pregunta	Resultado
9. **¿Qué arquitectura de software usa con mayor frecuencia su compañía?**	*Ilustración 15 Resultado pregunta 9*  *Fuente: Elaboración propia*

En la actualidad, las empresas han tenido que reformarse y usar las arquitecturas que exigen en el mercado. De los perfiles encuestados, el 53% manifiesta que tienen acercamiento con la arquitectura de microservicios. La segunda arquitectura que aún se usa es la monolíticas con un 19%. Por último, se encuentran la arquitectura orientada a servicios con 19% y otras sin definir en un 13%. Por lo anterior, se infiere que aún falta evolución y/o actualización en las arquitecturas de las empresas, para que en sus procesos se aproximen a una certificación. (Cervantes & Kazman, 2016)

Pregunta	Resultado
10. ¿Cuál de los siguientes factores en el más influyente en su compañía para la elección de una arquitectura?	*Ilustración 16 Resultado pregunta 10*  Fuente: Elaboración propia

En este ítem se analizan los factores relevantes para tener en cuenta en la elección de una arquitectura de software. En la encuesta realizada, de las 25 empresas escogieron como los factores más relevantes e imprescindibles, el Rendimiento y la Escalabilidad con un 30%. La Mantenibilidad, un factor transcendental, obtuvo un 19%, lo que significa que se tiene un balance en peso para la elección de los factores. Por último, los otros factores que en menor medida se escogen es la reutilización de componentes con un 8%.

Es fundamental encontrar un equilibrio entre los factores, teniendo un alcance claro y conciso de los requisitos específicos del proyecto, para que en la elección de la arquitectura los aspectos conformen un total exitoso y sostenible. (Bass, Clements, & Kazman, 2012)

Fuente: Elaboración propia

Conclusiones

Durante la pandemia, en las empresas con productos tecnológicos existía una brecha notable en cuanto a la definición de procesos y la era digital, como explica (Laura Isabel Rojas de Francisco, 2023) en su capítulo 4. En contexto, las empresas con poca frecuencia definían sus procesos en el ciclo de vida del software y los estandarizaban, se evidenciaba desorden en la documentación y las pruebas de aseguramiento de calidad eran insuficientes. Por tal motivo, el hecho de entrar en la virtualidad formó conciencia en las empresas creando así la necesidad de certificar sus procesos y adaptarse al mundo tecnológico.

Considerando los niveles de certificación vigente, se pueden tener puntos o aspectos básicos para que una empresa pueda acceder y garantizar que sus operaciones y productos tecnológicos cumplen con los requisitos mínimos que los guiarán hacia la excelencia.

La empresa debe contar con las herramientas y tecnologías necesarias para desarrollar software de calidad. Esto incluye IDEs, Frameworks, bases de datos y herramientas de pruebas.

Además de estos aspectos mínimos, las empresas que desean obtener una certificación como empresa de desarrollo de software pueden optar por cumplir con los requisitos de una norma específica. Las normas más comunes para la certificación de empresas de desarrollo de software son las siguientes:

- ISO 9001: esta norma establece los requisitos para un sistema de gestión de la calidad.
- CMMI: esta norma establece los niveles de madurez de los procesos de desarrollo de software.
- IEEE 12207: esta norma establece los requisitos para el proceso de desarrollo de software.

En este trabajo se recopilan los puntos básicos, necesarios, relevantes y fundamentales para prepararse y adaptarse a las certificaciones para las medianas y pequeñas empresas. A continuación, mencionaremos los puntos importantes:

- Metodología o marco de trabajo.

Para realizar un proyecto debemos identificar el conjunto de principios, técnicas y procedimientos que se utilizan para planificar, ejecutar y controlar un proyecto. La metodología de trabajo debe adaptarse en cuanto a las necesidades específicas del proyecto a ejecutar, sugerimos alguna de estas metodologías:

- Metodología en cascada: Esta metodología es lineal, es decir, los pasos se llevan a cabo secuencialmente.
- Metodología ágil: Esta metodología es iterativa, es decir, el proyecto se divide en fases cortas llamadas iteraciones.
- Metodología híbrida: Esta metodología combina elementos de las metodologías en cascada y ágiles. Es una buena opción para proyectos que requieren una combinación de flexibilidad y estructura.

- Definición de roles y responsabilidades.

La definición de roles y responsabilidades en el desarrollo del software es un proceso esencial que establece claramente las funciones y tareas asignadas a cada miembro del equipo durante el ciclo de vida del proyecto. Esta práctica organizativa es fundamental para la eficacia, la eficiencia y calidad del desarrollo del software.

La claridad en los roles garantiza que cada individuo comprenda sus habilidades específicas y contribuciones al proyecto, reduciendo la posibilidad de malentendidos y conflictos.

Además, esta estructura organizativa optimiza la utilización de recursos, asignando tareas según habilidades y experiencias de los miembros del equipo, lo que conduce a una distribución equitativa del trabajo.

- Documentación de requisitos y código.

Es importante para garantizar que el software cumpla con las necesidades de los clientes. La documentación de requisitos debe incluir una descripción completa de los requisitos del software, incluidos los requisitos funcionales y no funcionales. En cuanto La documentación

debe comenzarse temprano en el proceso de desarrollo de software. Esto ayudará a garantizar que la documentación sea completa y precisa.

- Definición de arquitectura de software y lineamientos de desarrollo.

La arquitectura de software es el diseño estructural y organizativo de un sistema de software. Incluye decisiones clave sobre la interacción de componentes, patrones de diseño, capas del sistema, gestión de datos y seguridad, estableciendo la base para el desarrollo del software.
Los lineamientos de desarrollo son reglas y directrices que guían el proceso de desarrollo de software. Proporcionan un marco de trabajo para prácticas coherentes en codificación, documentación, pruebas y colaboración. Mejoran la calidad y coherencia del software, asegurando estándares compartidos entre el equipo de desarrollo.

- Automatización de procesos técnicos.

En este apartado se hace énfasis en la eliminación de algunas tareas operativas o repetitivas de gran esfuerzo humano, comprendiendo que para las empresas corresponde en aumento de la eficiencia, productividad y la seguridad.

  - Pruebas automatizadas

Son herramientas y/o scripts para realizar pruebas de un software sin necesidad de que una persona deba implementar tiempo para realizarlas, las pruebas suelen ser actividades repetitivas, pero con la automatización podemos optimizar estos tiempos aumentando la eficiencia y precisión.

- CI/CD

Es la automatización del proceso de integración y entrega de código, se basa en la idea de que el código debe ser integrado y entregado de forma continua, es decir, a menudo y de forma automática, reduciendo el tiempo de entrega y mejora la productividad.

- Definición de métricas de rendimiento y código.

Las métricas de rendimiento miden cómo se desempeña un software en analizando y cuantificando: la disponibilidad, carga de trabajo, tiempos de respuesta, tasa de error entre otros. Estas métricas son importantes para garantizar que el software cumpla con los requisitos de rendimiento establecidos por los usuarios y los clientes.

Las métricas de código miden la calidad del código fuente, incluyendo su complejidad, legibilidad y mantenibilidad. Estas métricas son importantes para garantizar que el código sea fácil de entender, modificar y depurar en el tiempo.

Bibliografía[1]

(ESI), E. S. (2017). *IT-MARK: Certificación para PYMEs del sector TI*. Obtenido de www.esi.es: https://www.esi.es/es/it-mark

Anderson, R. J. (2023). *Principles of Software Security.* Wiley.

Antonio Fernández-Medina, P. G.-P.-M. (2022). *Ciclo de vida del software: Metodologías, modelos y prácticas.* RA-Ma.

Atlassian. (2023). *Atlassian*. Obtenido de https://www.atlassian.com: https://www.atlassian.com/es/git/tutorials/saving-changes/git-commit

Bass, L., Clements, P., & Kazman, R. (2012). *Arquitectura de software: principios, patrones y práctica.* Boston, MA: Addison-Wesley Professional.

Blandón-Jaramillo, C. A.-B. (2023). Calidad del software y seguridad de aplicaciones a partir del proceso de desarrollo de software AGILISO y el estándar OWASP. *Revista Tecnología En Marcha*, 3-10.

Briceño, I. (2023). *LinkedIn*. Obtenido de LinkedIn: https://www.linkedin.com/pulse/web-30-ivan-brice%C3%B1o

Carrizo, D., & Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare. Revista chilena de ingeniería*. Obtenido de scielo: https://scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052018000100114#B1

Cervantes, H., & Kazman, R. (2016). *Designing Software Architectures: A Practical Approach (SEI Series in Software Engineering).* Crawfordsville, Indiana.: Pearson Education, Inc.

Chaudhary. (2017). *CMMI for Development Implementation Guide (2017)*. Obtenido de CMMI: https://doi.org/10.1007/978-1-4842-2529-5

Díaz, L. J. (2023). *Métricas de software.* Ra-Ma.

Gartner. (2023). *The Future of Software Architecture.* Gartner.

Hidalgo, R. J. (2023). *Static Analysis Techniques for Software Fault Detection and Prevention.* Springer.

International Telecommunication Union (ITU). (2018). *www.itu.int.* Obtenido de www.itu.int: https://www.itu.int/en/ITU-D/Cybersecurity/Documents/Cybersecurity_Definitions_Concepts_and_Areas_of_Application.pdf

International Telecommunication Union (ITU). (2018). *www.itu.int.* Obtenido de www.itu.int: https://www.itu.int/en/ITU-D/Cybersecurity/Documents/Cybersecurity_Definitions_Concepts_and_Areas_of_Application.pdf

Isaca. (2023). *ISACA CMMI Perfomance solutions*. Obtenido de Isaca: https://cmmiinstitute.com/

José Manuel Arévalo, J. C. (2023). *Seguridad en el código de desarrollo de software.* O'Reilly.

Kleppmann, P. S. (2022). *Arquitectura de software: las partes difíciles. Análisis moderno de ventajas y desventajas para arquitecturas distribuidas.* O'Reilly.

Laura Isabel Rojas de Francisco, T. O. (2023). *Experiencias innovadoras en la educación superior colombiana.* Ediciones Universidad Cooperativa de Colombia.

Limas Suárez, S. (2020). El comercio electrónico (e-commerce) un aliado estratégico para las empresas en Colombia. *Revista ibérica de sistemas e tecnologias de informação*, 235-251.

McDonald, S. y. (2022). *Ingeniería de software: Un enfoque práctico.* Pearson.

Organización Internacional de Normalización (ISO) y Comisión Electrotécnica Internacional (IEC). (2008). *ISO/IEC 12207:2008 - Ingeniería de sistemas y software - Procesos del ciclo de vida del software*. Obtenido de www.iso.org: https://www.iso.org/standard/43447.html

Owasp. (s.f.). *Owasp*. Obtenido de Owasp: https://owasp.org/Top10/

proyecto, M. A. (2021). *Metodología Agile: qué es y cómo aplicarla a tu proyecto.* Obtenido de blog.hubspot.es: https://blog.hubspot.es/marketing/metodologia-agile

Red Colombiana de Calidad del Software (RCCS). (2014). *IT-MARK: Certificación para PYMEs del sector TI.* Obtenido de www.rccs.org.co: https://www.rccs.org.co/it-mark

Rigby, D., Elk, S., & Berez, S. (2020). *Doing Agile Right: Transformation Without Chaos.* Harvard Business Review Press.

Sdv.gov. (2021). *Sdv.gov*. Obtenido de Transformación digital del tejido empresarial de la ciudad de Bogotá para afrontar los efectos de la pandemia.

Suarez. (01 de 04 de 2020). *Revista ibérica de sistemas e tecnologias de informação*.

University, C. M. (2006). *CMMI® for Development, Version 1.2* . Carnegie Mellon University. : CMMI-DEV, V1.2 .

Ilustraciones[2]