

## Investigar 5 técnicas o métodos que permitan identificar riesgos en los proyectos de software.

En el mundo del desarrollo de software, la gestión de riesgos ha tomado mucha popularidad en los últimos años, según (Torres, 2022). Esto se debe a que algunas grandes compañías han presentado pérdidas económicas y reputacionales significativas por no haber realizado una buena gestión de riesgos en sus proyectos, lo cual, a largo plazo, provoca este tipo de incidentes. De aquí la importancia de llevar a cabo una adecuada gestión de riesgos para evitar estos problemas. Existen varias técnicas o métodos que ayudan a identificar riesgos futuros en el proyecto o incluso a encontrar riesgos ya presentes de manera más eficiente. A continuación, se presentan algunas de ellas:

**1. Análisis de Causa-Raíz:** Esta técnica se utiliza habitualmente para analizar la raíz del problema, riesgo o fallo en los proyectos. Se emplean diagramas como el de Ishikawa o espina de pescado, el diagrama de Pareto o la técnica de los 5 porqués. Se caracteriza por no solo identificar los riesgos, sino también las causas del problema. Sin embargo, requiere tiempo y esfuerzo para analizar cada situación en detalle (Borsalli, 2022).

### Diagrama de Ishikawa

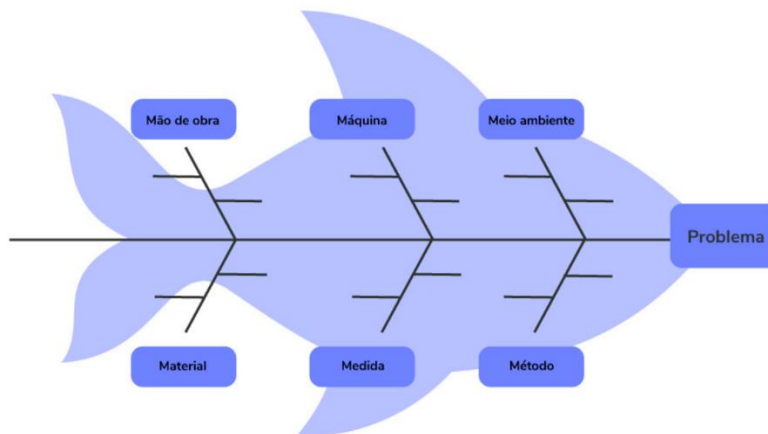


Imagen tomada de: [Diagrama-de-Ishikawa.png \(1097×705\)](#)  
([automacaoindustrial.info](#))

**2. Técnica Delphi:** Este método se centra en recolectar información u opiniones de especialistas para identificar posibles riesgos mediante rondas de preguntas, buscando así detectar los riesgos potenciales. La información se recolecta de manera anónima, lo que permite a los especialistas opinar libremente. No obstante, puede ser costoso si se involucran muchos expertos en el tema y requiere una buena coordinación para llevar a cabo las rondas de consultas de manera efectiva (EALDE, 2020).

**3. Análisis DOFA:** Es el acrónimo de Debilidades, Oportunidades, Fortalezas y Amenazas. Se utiliza principalmente para evaluar, de manera interna, las debilidades y fortalezas, y de manera externa, las amenazas y oportunidades que afectan al proyecto de software en su fase inicial. Suele combinarse con otras técnicas para aumentar su efectividad (Borsalli, 2022).

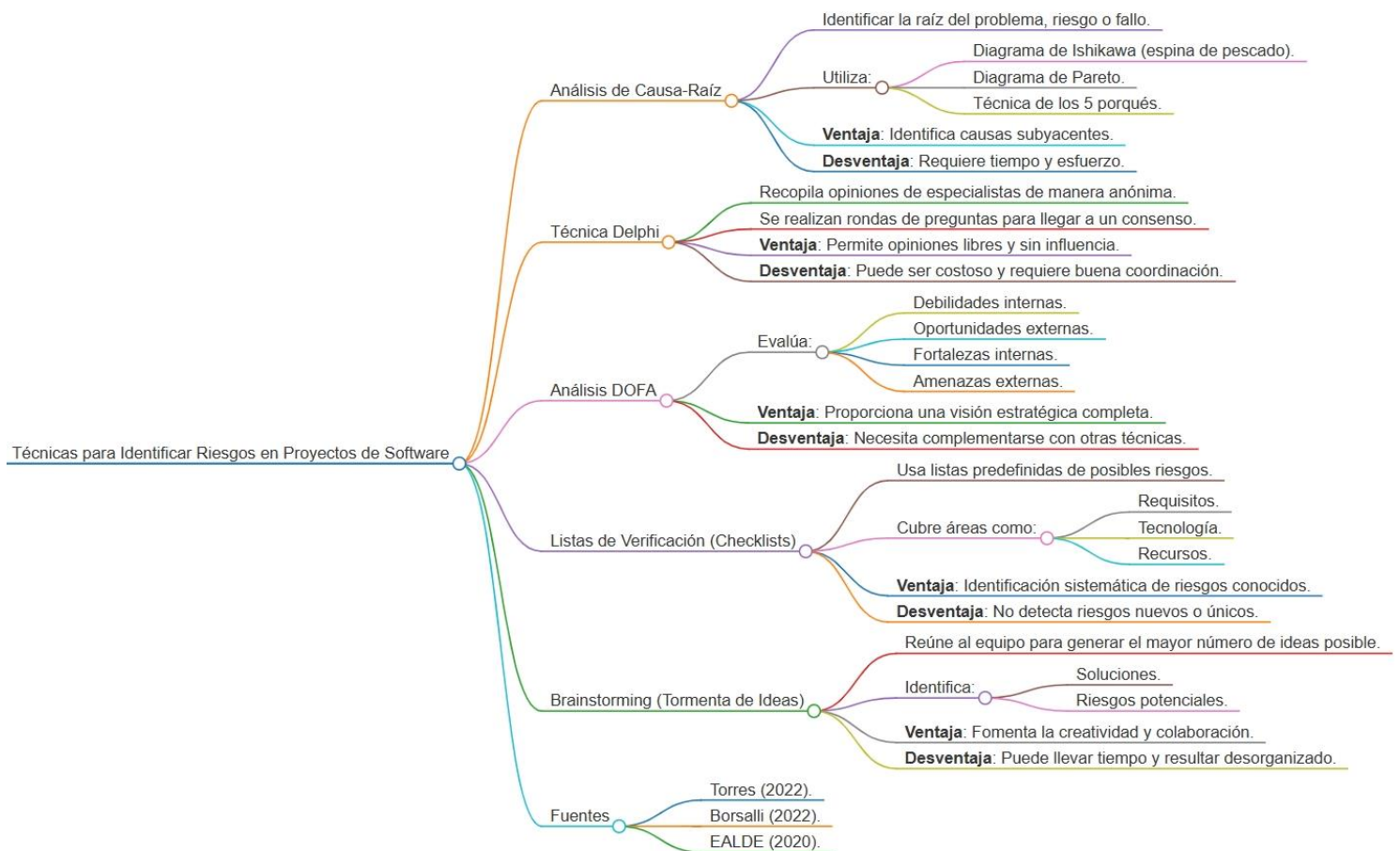


Imagen recuperada de: [7 métodos y herramientas para la identificación de riesgos](#)  
([softexpert.com](#)) (labitstudio, s.f.)

**4. Listas de Verificación (Checklists):** Esta técnica consiste en utilizar listas con los posibles riesgos y verificarlos en el proyecto de software, cubriendo áreas como requisitos, tecnología, etc. Facilita la identificación de riesgos conocidos, pero al basarse en riesgos predefinidos, puede no ser útil para identificar riesgos nuevos o únicos.

**5. Brainstorming (Tormenta de Ideas):** Se basa en reunir a los miembros del equipo para que aporten una lluvia de ideas con el mayor número posible de propuestas. De esta manera, se busca encontrar posibles soluciones o riesgos en el proyecto, aprovechando la experiencia y conocimientos de cada miembro del equipo. Esta técnica fomenta la creatividad y la colaboración (Borsalli, 2022).

### Mapa Mental



**Investigar 2 estándares que contenga métricas para evaluar la calidad de un producto de software, mencione las métricas, cómo se implementan y qué evalúan.**

En el ámbito de TI y desarrollo de software, es fundamental seguir estándares de medición para asegurar la calidad del producto final. La calidad del software es medible y requiere de un marco de referencia, apoyándose en estándares de calidad reconocidos internacionalmente.

Según (Domínguez, 2017), “los estándares son normas y protocolos internacionales que deben cumplir los productos de cualquier índole para su distribución”, lo que implica que deben satisfacer las necesidades del cliente.

**Los estándares se pueden clasificar en tres grupos principales:**

Estándares que miden el producto (ej. ISO 25000 o SQUARE)

Estándares que miden el proceso (ej. CMMI)

Estándares específicos para TI (ej. ISO/IEC 9126)

Hablemos de dos de estos estándares:

**ISO 25000(SQUARE):** Este estándar está más enfocado a la evaluación y mejora de calidad del software incluyendo mantenibilidad y usabilidad (Medina, 2022).

De acuerdo con (Medina, 2022) esta se compone de las siguientes divisiones

- Gestión de calidad

- Modelo de calidad
- Mediciones de calidad
- Requisitos de calidad
- Evaluación de la calidad
- Estándares de extensión

Este evalúa las siguientes características del proyecto de software

- Adecuación funcional
- Fiabilidad
- Usabilidad
- Eficiencia
- Compatibilidad
- Seguridad
- Mantenibilidad
- Portabilidad

**Implementación:** Se aplica en todo el ciclo de vida del software desde que se planifica hasta que se entrega.

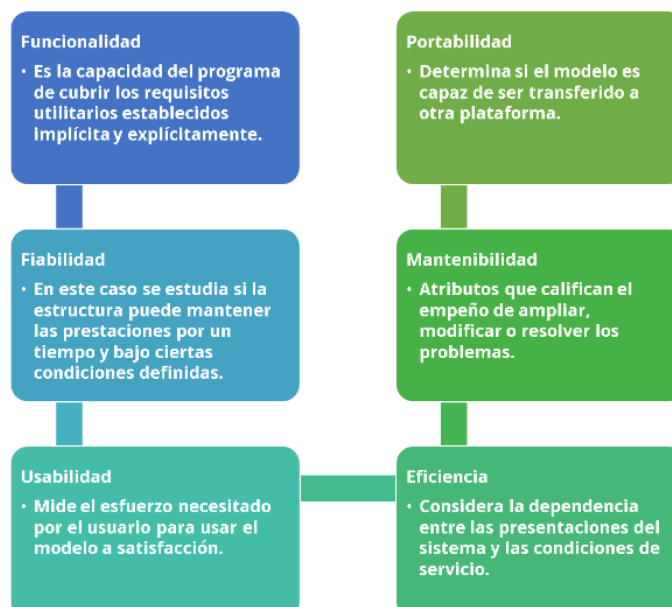
### **Métricas**

- Tiempo medio de detección: Indica cuánto tiempo se necesita para detectar errores.

- Tiempo medio de preparación: Mide el tiempo necesario para corregir defectos
- Prueba de confiabilidad: Evalúan la robustez y estabilidad del software.
- Cobertura de ejecución de pruebas: Cuantifica qué porcentaje del código se ha ejecutado durante las pruebas.
- Cobertura de código: Analiza la cantidad de código fuente cubierto por pruebas.

**ISO 9126:** Es uno de los estándares que nos permite realizar la medición de la calidad del software, implementando siete características claves: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad y satisfacción cada una facilitan la medición de los atributos internos y externos del software. Lo que hace que esta norma sea crucial ya que nos proporciona métricas que permiten cuantificar y calificar la calidad del software, permitiendo que las empresas puedan estar a la vanguardia de la transformación digital asegurando que el software sea muy excelente y así solventar la satisfacción del cliente.

Pero según (GRUPO DIGITAL 360°, 2018) estas métricas están siendo remplazadas por el proyecto ISO 25000, el cual tiene características y conceptos similares



## Métricas

1. **Valoración de requerimientos:** permite asegurarse de que en cada fase se ha cumplido con las exigencias correspondientes. (GRUPO DIGITAL 360°, 2018)
2. **Integridad del sistema:** analiza las amenazas potenciales, el número de ocasiones que se presentan y cuántas veces la página se logra recuperar.
3. **Madurez del software:** indica el nivel de mantenimiento considerando la cantidad de módulos actuales, modificados, añadidos y eliminados. El número se aproxima a 1 cuando aumenta la estabilidad. En cuyo caso se necesitará menos tiempo en la corrección de errores o implementación de mejoras.
4. **Eficacia de detección y eliminación de errores:** las fallas se analizan por separado

## Herramientas útiles para la medición o detección de errores

**SonarQube:** SonarQube es una herramienta de análisis on-premise diseñada para detectar problemas de codificación en [+30 idiomas, frameworks y plataformas IaC](#). Al integrarse directamente con su [canalización de CI](#) (SonarQube, s.f.)

Documentación de SonarQube 10.6: [SonarQube 10.6 \(sonarsource.com\)](#)

## Referencias

- Borsalli, B. (28 de 02 de 2022). *SoftExpert Blog*. Obtenido de SoftExpert Blog:  
<https://blog.softexpert.com/es/identificacion-riesgos/>
- Domínguez, I. R. (08 de 03 de 2017). *Calidad del software | MODELOS Y ESTÁNDARES DE UN PRODUCTO*. Obtenido de Youtube:  
<https://www.youtube.com/watch?v=2lyCHFg-co>
- EALDE. (10 de 11 de 2020). *EALDE*. Obtenido de EALDE:  
<https://cursos.ealde.es/blogs/gestion-de-riesgos-direccion-de-proyectos-online/tecnicas-y-herramientas-para-la-identificacion-de-riesgos-en-proyectos#:~:text=Algunos%20ejemplos%20de%20t%C3%A9cnicas%20de%20reco%20pilaci%C3%B3n%20de%20informaci%C3%B3n,de%20deb>
- GRUPO DIGITAL 360°. (13 de 12 de 2018). *ISO 9126*. Obtenido de GRUPO DIGITAL 360°:  
<https://grupodigital360.com/iso-9126-metricas-para-calidad-web/>
- labitstudio. (s.f.). *labitstudio*. Obtenido de labitstudio: <https://labitstudio.com/diferencias-sistemas-control-de-versiones/>
- Medina, I. F. (05 de 12 de 2022). *Los estándares de calidad del software más importantes*. Obtenido de hiberus blog: <https://www.hiberus.com/crecemos-contigo/los-estandares-de-calidad-del-software-mas-importantes/>



SonarQube. (s.f.). *Documentación de SonarQube 10.6*. Obtenido de SonarQube:  
<https://docs.sonarsource.com/sonarqube/latest/>

Torres, O. P. (14 de 10 de 2022). *pirani*. Obtenido de pirani:  
<https://www.piranirisk.com/es/blog/gestion-de-riesgos-proyectos-de-software>