

# Seguridad de las aplicaciones integrada en el desarrollo: Seguridad a la velocidad de DevOps



---

## Contenido

El problema actual de la seguridad de las aplicaciones	3
Estos problemas no harán más que agravarse	3
Por qué no darán los resultados esperados las prácticas tradicionales de seguridad de las aplicaciones	4
¿Qué es la seguridad de las aplicaciones integrada en el desarrollo?	4
La seguridad de las aplicaciones integrada en el desarrollo para su organización	4
Paso 1: Desarrollar pensando en la seguridad	5
Paso 2: Realizar pruebas pronto, a menudo y rápidamente	5
Paso 3: Aprovechar las integraciones para que la seguridad de las aplicaciones se convierta en una parte natural del ciclo de vida.	8
Paso 4: Automatizar la seguridad como parte de los procesos de desarrollo y pruebas	9
Paso 5: Pensar en el futuro	9
Primeros pasos	10
¿Por qué OpenText?	11

---

# 90 %

Porcentaje de los incidentes de seguridad que se deben a exploits contra defectos en el diseño o el código del software.

Fuente: Informe de riesgos de seguridad de las aplicaciones de 2019

## El problema actual de la seguridad de las aplicaciones

En los últimos 10 años, el software ha pasado de tener una función de apoyo a la empresa a ser a un centro de innovación, y se ha convertido en el diferenciador competitivo más importante para la mayoría de las empresas con independencia de su tamaño y sector vertical. Este cambio en el papel del software está llevando a las empresas a aumentar espectacularmente el número de aplicaciones y la frecuencia de sus versiones. Según [TechTarget](#), el 45 % de las organizaciones publican versiones una vez a la semana. Además, la complejidad del código sigue aumentando debido a que los desarrolladores intentan satisfacer la demanda empresarial aprovechando el código comercial y el código abierto, además de su código personalizado. Según el informe [Estado del sistema de suministro del software](#) de Sonatype de media, el **80 %** del código de una aplicación proviene de bibliotecas de código abierto. El informe también demostró que, de media, cada aplicación contiene 38 vulnerabilidades conocidas de código abierto. Esta situación tiene enormes implicaciones para los equipos de seguridad a la hora de buscar y gestionar las vulnerabilidades. En este sentido, varias de las brechas de seguridad más destacadas de los últimos años se debieron a vulnerabilidades en los componentes de código de otros fabricantes.

Con las necesidades empresariales como prioridad, proliferan las aplicaciones a través de sitios web, plataformas de redes sociales, y aplicaciones móviles y en la nube. Además, algunas aplicaciones son impulsadas por los equipos de marketing y creadas con software de otros fabricantes. Normalmente, a estas aplicaciones se les excluye de los procesos empresariales habituales y se las controla poco o nada.

Además de todos los desafíos que plantea el creciente número de aplicaciones, la mayor complejidad y la premura de los lanzamientos, las normativas como el RGPD y la recopilación de datos del cliente con fines comerciales se han convertido en la norma. La multiplicidad de instancias de datos de clientes incrementa la probabilidad y el impacto de las brechas. Esta situación resulta especialmente preocupante debido a que la mayoría de las brechas de seguridad actuales se deben a las vulnerabilidades de las aplicaciones. Según el [Informe de riesgos de seguridad de las aplicaciones de 2019](#) de nuestro equipo Software Security Research, el 80 % de las aplicaciones contiene al menos una vulnerabilidad crítica o alta, y el 90 % de los incidentes de seguridad se deben a exploits contra defectos en el diseño o el código del software.

## Estos problemas no harán más que agravarse

El tiempo necesario hasta la comercialización sigue siendo vital para las empresas, por lo que las organizaciones están adoptando DevOps u otras metodologías Agile similares para acelerar el desarrollo de su creciente éxito. Todo esto significa que, si la seguridad no pasa a ser una parte esencial del ciclo de vida del software, las organizaciones continuarán lanzando aplicaciones con más vulnerabilidades al mercado a una velocidad vertiginosa.

---

Las recomendaciones y las pruebas de seguridad de las aplicaciones deben integrarse en la cadena de herramientas del desarrollador.

## Por qué no darán los resultados esperados las prácticas tradicionales de seguridad de las aplicaciones

En muchas organizaciones, la seguridad de las aplicaciones se limita a un equipo específico que interviene en las etapas finales del desarrollo, y se percibe como un obstáculo para la rapidez. Estos equipos de seguridad no dan abasto debido a que los equipos de desarrollo crecen 80 veces más rápido que ellos. Cuando las vulnerabilidades de seguridad se detectan en fases avanzadas, las organizaciones se ven sometidas a una gran presión, lo que provoca fricciones entre los equipos, incumplimientos de los plazos de lanzamiento o incluso situaciones peores. Las versiones con defectos de seguridad conocidos también se están pasando a producción para cumplir los plazos del proyecto, en cuyo caso la empresa y sus clientes corren el riesgo de quedar expuestos a los atacantes.

Más allá de los plazos incumplidos y la dinámica del equipo, adoptar un enfoque reactivo en materia de seguridad de las aplicaciones resulta más costoso para las organizaciones. Según el NIST, el coste de subsanar los fallos de seguridad es 30 veces más caro en la fase de producción y 10 veces más en la fase de pruebas que si se detectaran en las primeras fases de desarrollo. Estos problemas y riesgos potenciales indican que la única forma de proteger las aplicaciones sin comprometer los costes es aumentar la seguridad en las primeras fases y adoptar un enfoque de seguridad de las aplicaciones integrado en el desarrollo.

## ¿Qué es la seguridad de las aplicaciones integrada en el desarrollo?

La seguridad de las aplicaciones integrada en el desarrollo consiste en hacer de la seguridad de las aplicaciones una parte integral del ciclo de vida del software sin crear una carga adicional para los participantes. Tanto si se trata de adoptar un enfoque DevSecOps como de crear un programa de seguridad más eficaz, lo primordial es pensar en la seguridad desde las primeras fases del ciclo de vida. Las recomendaciones y las pruebas de seguridad de las aplicaciones deben integrarse en la cadena de herramientas del desarrollador. Cuando se aplican correctamente, también significa que no es necesario comprometer la seguridad de las aplicaciones para lograr los ciclos de lanzamiento más rápidos que exige el mercado.

## La seguridad de las aplicaciones integrada en el desarrollo para su organización

Lograr integrar satisfactoriamente la seguridad en el desarrollo conlleva tiempo y esfuerzo, pero el mayor obstáculo en el camino es el cambio cultural necesario para incluir la seguridad a lo largo de todo el ciclo de vida del desarrollo del software. Es importante eliminar la fricción entre los equipos de seguridad y los desarrolladores. Muchas personas creen que los equipos de desarrollo y seguridad tienen prioridades contrapuestas, que a menudo se convierten en el mayor obstáculo para la consecución de un programa de seguridad de las aplicaciones. Los desarrolladores suelen mostrarse reacios a que su organización cree un programa de seguridad de las aplicaciones por temor a que se ralentice su trabajo. Esta mentalidad negativa en relación con la seguridad se debe a menudo a que los profesionales de la seguridad dictan reglas, flujos de trabajo y herramientas para los desarrolladores, en lugar de crear asociaciones sólidas, objetivos comunes y herramientas que se integren convenientemente en la cadena de herramientas de desarrollo.

---

# 1 de cada 8

descargas de fuentes presenta un riesgo conocido.

Fuente: Sonatype, noveno informe anual sobre el Estado del sistema, 2023

Al igual que en DevOps, los equipos deben romper las barreras que les separa, adoptar la transparencia y colaborar. Aunque es más fácil decirlo que hacerlo, contar con el apoyo de la dirección y con algunos defensores clave de la seguridad dentro de la organización puede ayudar a impulsar esta iniciativa. Más allá del cambio cultural necesario, existen pasos importantes para avanzar en la transición hacia una seguridad de las aplicaciones integrada en el desarrollo sea un éxito. Entre ellos los siguientes:

## Paso 1: Desarrollar pensando en la seguridad

Con una proporción de desarrolladores y especialistas en seguridad de aproximadamente 80:1, es imprescindible proporcionar recursos a los desarrolladores para que asuman la responsabilidad de su propio código. Detectar y corregir los defectos de seguridad durante el proceso de codificación permitirá a los desarrolladores eliminar las posibles vulnerabilidades de seguridad antes de que lleguen a la fase de pruebas y producción, lo que ahorrará tiempo y dinero a la organización. Esta nueva perspectiva requiere formar a los desarrolladores para que codifiquen con la seguridad en mente y proporcionarles las herramientas adecuadas para que puedan recibir feedback en tiempo real sobre su código. Existen numerosas opciones para la formación en seguridad de los desarrolladores, pero las herramientas que proporcionan información sobre la seguridad del código en tiempo real (como el plugin OpenText™ Application Security Assistant de OpenText, que funciona de forma muy similar a un corrector ortográfico de seguridad y proporciona información sobre la seguridad del código en tiempo real mientras se desarrolla) o la formación integrada y gamificada para desarrolladores, como Secure Code Warrior, simplifican la adopción y aceleran la formación.

También es importante que los equipos de seguridad ayuden a los desarrolladores compartiendo con ellos información sobre amenazas conocidas, proporcionándoles comentarios, y adoptando la transparencia y la visibilidad en su trabajo. El hecho de que los responsables del desarrollo hayan recibido formación en materia de seguridad de las aplicaciones y se hayan asociado con ellos como expertos de seguridad proporciona resultados positivos. De esta forma, los responsables de desarrollo pueden aportar la perspectiva de seguridad durante las primeras fases del ciclo de vida del desarrollo, además de los aspectos funcionales y de calidad tradicionales.

## Paso 2: Realizar pruebas pronto, a menudo y rápidamente

Durante el ciclo de vida del desarrollo de software, existen varios enfoques que se pueden seguir para mantener la velocidad necesaria para estar al día con los lanzamientos actuales. Estas estrategias son realizar pruebas pronto, a menudo y rápidamente.

### Realizar pruebas pronto

Las pruebas estáticas de seguridad de la aplicación (SAST) identifican el origen de los problemas de seguridad y ayudan a solucionar los defectos de seguridad subyacentes desde las primeras fases del desarrollo. Para mantener la velocidad de los lanzamientos, los desarrolladores deben poder enviar código de forma rápida y sencilla, teniendo la inteligencia al alcance de la mano. [OpenText™ Static Application Security Testing \(Fortify SAST\)](#) lidera este método porque:

- Identifica y elimina las vulnerabilidades en código fuente, binario o de bytes.
- Abarca los lenguajes que utilizan los desarrolladores con compatibilidad para 27 lenguajes.
- Ofrece una detección y solución temprana de los defectos, lo que permite reducir los gastos de corrección.

---

## El análisis más inteligente se refiere a la validación de DAST de los hallazgos de SAST y la sintonización de DAST mediante los resultados de SAST.

- Revisa los resultados de los análisis en tiempo real gracias al acceso a recomendaciones y la navegación por las líneas de código para encontrar vulnerabilidades más rápidamente y permitir la auditoría colaborativa.
- Aumenta el enfoque en las primeras fases: disponer de análisis en todas partes, incluido el IDE del desarrollador y los procesos de CI/CD.

El plugin [Fortify Security Assistant by OpenText](#) va un paso más allá, ya que proporciona a los desarrolladores información y recomendaciones sobre las vulnerabilidades del código en tiempo real, mientras se escribe. Esto no solo sirve de "corrector ortográfico" de seguridad para los desarrolladores en lo que respecta a vulnerabilidades comunes conocidas, sino que, con el tiempo, les permite dejar de cometer esos errores desde el principio.

Más allá del análisis estático, todavía existe una creciente preocupación sobre las vulnerabilidades conocidas dentro de los componentes de código abierto. Durante casi 10 años, el uso de componentes vulnerables conocidos ha formado parte de la lista OWASP Top 10. La comunidad DevSecOps ha descubierto recientemente que 1 de cada 10 descargas de componentes de código abierto contienen una vulnerabilidad de seguridad conocida. Se ha registrado un aumento del 71 % en las vulneraciones verificadas o sospechosas entre 2014 y 2020, y 1 de cada 5 organizaciones experimentó al menos una brecha de código abierto en los últimos 12 meses.<sup>1</sup>

Aunque este crecimiento es preocupante, muchas organizaciones han estado utilizando el análisis de la composición del software para contrarrestar estos riesgos. Sin embargo, priorizar los hallazgos de código abierto sigue presentando un reto importante con el análisis de la composición del software. Al igual que con los hallazgos de SAST, la auditoría manual de los descubrimientos es un proceso prolijo y que aumenta el tiempo para solucionar los problemas para los desarrolladores. Según un informe de Sonatype, las organizaciones dedicarán una media de 20 minutos a investigar manualmente un hallazgo de código abierto y la aplicación promedio contiene 38 problemas de código abierto. Debido a que la mayoría de las organizaciones cuentan con cientos o miles de aplicaciones, esto puede suponer miles de horas dedicadas a investigar hallazgos de código abierto que podrían no tener ningún impacto real en la seguridad de su aplicación. Los equipos deben poder centrarse en los problemas que no solo son vulnerables, sino también en aquellos susceptibles de ser vulnerado.

El análisis de susceptibilidad consiste en ilustrar rápidamente los componentes vulnerables que se invocan directa o indirectamente y que, por lo tanto, son explotables o "susceptibles". Ser capaz de priorizar los problemas de código abierto ahorra tiempo a la hora de investigar problemas conocidos y aún más tiempo dedicado a actualizar una biblioteca que tiene un beneficio de seguridad casi nulo.

En OpenText, nos asociamos con Sonatype para conseguirlo, recopilando los métodos y las firmas de las funciones en función de las solicitudes que se reciben para las indicaciones de Sonatype de componentes conocidos. Conforme Sonatype analiza varios componentes de código abierto, OpenText entiende que, para cualquiera de esas vulnerabilidades concretas conocidas que hayan sido actualizadas, es decir, que hayan sido parcheadas, OpenText genera una firma para esa función o método, de modo que podamos ver que la función se encuentra realmente en su propio código personalizado y que está utilizando ese componente vulnerable de la dependencia. Esto significa que los desarrolladores no solo saben que tienen la dependencia de su ruta de clase, sino que la utilizan de una manera que los hace susceptibles a esta vulnerabilidad en particular.

<sup>1</sup> Sonatype, encuesta de la comunidad DevSecOps 2020

---

Los análisis automatizados dinámicos o estáticos permiten identificar eficazmente las vulnerabilidades de seguridad en el código fuente y reducir la laboriosidad de las evaluaciones de seguridad.

## Realizar pruebas a menudo

Las pruebas dinámicas de seguridad de la aplicación (DAST) simulan ataques contra una aplicación web en funcionamiento para identificar las vulnerabilidades que los exploits podrían aprovechar. Esto proporciona una visión completa de la seguridad de las aplicaciones, centrándose en lo que es explotable y cubriendo todos los componentes (servidor, código personalizado, código abierto, servicios). Integrar las herramientas DAST en el desarrollo, el control de calidad y la producción permite obtener una visión integral de forma continuada. [OpenText™ Dynamic Application Security Testing \(Fortify DAST\)](#) ofrece una solución efectiva, ya que permite:

- Identificar rápidamente los riesgos en aplicaciones existentes.
- Automatizar las pruebas dinámicas de seguridad de las aplicaciones para cualquier tecnología, desde el desarrollo hasta la producción.
- Cumplir los estándares de conformidad y seguridad con las políticas y los informes preconfigurados para las principales normativas.
- Validar las vulnerabilidades en aplicaciones en funcionamiento, priorizando los problemas más graves en el análisis de causa principal.
- Rastrear los marcos de trabajo modernos y las API.

SAST y DAST realmente se complementan entre sí. Superponer el análisis dinámico al análisis estático permite a los clientes obtener una valiosa métrica de riesgo adicional que les ayuda a conseguir una imagen más completa del riesgo en el mundo real. Si bien es importante identificar las vulnerabilidades en las primeras etapas del SDLC mediante tecnologías como el análisis estático, es de suma importancia crear bucles de respuesta que puedan identificar cuándo surgen esos hallazgos en entornos de ejecución mediante una exploración DAST.

Una organización que identifica hallazgos como el XSS al principio del SDLC y continúa detectando esos problemas en la producción, puede centrar sus recursos de formación y desarrollo en abordar los problemas sistémicos.

La verdadera integración de SAST y DAST significa que las herramientas SAST y DAST se integran en una única plataforma centrada en el desarrollador con una consola de gestión única. La administración unificada de vulnerabilidades crea bucles de respuesta. Una plataforma unificada de administración de vulnerabilidades no solo es fundamental en cuanto a los flujos de trabajo simplificados de priorización y triaje que introduce, sino también por los patrones que se pueden derivar de los datos. El análisis más inteligente se refiere a la validación de DAST de los hallazgos de SAST y la sintonización de DAST mediante los resultados de SAST.

## Realizar pruebas rápidamente

Las pruebas interactivas de seguridad de la aplicación (IAST) son un tipo de pruebas de seguridad que combina las DAST con los comentarios del tiempo de ejecución de las aplicaciones probadas a la vez que se realizan las pruebas. Sin embargo, incluso con un enfoque IAST, buscar las vulnerabilidades solo es un tercio del trabajo. Los dos tercios restantes se invierten a menudo en la validación de falsos positivos y la corrección. Otro argumento en contra de las IAST es el hecho de que este método de prueba suele obviar verdaderos positivos debido a las limitaciones técnicas de este enfoque. Como método más eficaz, los algoritmos de aprendizaje automático aplicados y la automatización de auditorías pueden ahorrar tiempo y esfuerzo de auditoría, además de mejorar la precisión del análisis estático.

---

[Audit Assistant](#) es nuestra tecnología de aprendizaje automático. Disponible tanto in situ como en la nube, Audit Assistant utiliza los metadatos del resultado del análisis para predecir y eliminar falsos positivos, y reducir el tiempo de corrección hasta en un 50 %. Un cliente comprobó como 8000 problemas de Java se veían reducidos a 3000 gracias a esta tecnología. Nuestra última versión automatiza aún más el proceso para los clientes añadiendo la predicción automática en la versión de la aplicación para solicitar automáticamente predicciones automáticas cuando se añaden nuevos problemas.

Audit Assistant simplifica la fase de las pruebas de seguridad que consume más tiempo: la auditoría del análisis. Audit Assistant aplica amplios conocimientos sobre seguridad y aprendizaje automático para automatizar la eliminación de falsos positivos, priorizar los hallazgos e identificar las vulnerabilidades de seguridad relevantes para la organización. Esto significa que, tras iniciar un análisis estático, se pueden obtener resultados validados en cuestión de minutos y enviarlos al departamento de desarrollo para que aplique las correcciones oportunas.

### **Paso 3: Aprovechar las integraciones para que la seguridad de las aplicaciones se convierta en una parte natural del ciclo de vida.**

La seguridad de las aplicaciones debe integrarse sin problemas a su canal de SDLC y CI/CD para lograr el éxito. La integración en las herramientas que utilizan su organización y los desarrolladores para desarrollar y probar sus aplicaciones le permite encontrar problemas con antelación y de manera frecuente, y solucionarlos como parte de los ciclos de pruebas de desarrollo. OpenText cuenta con un ecosistema de integración que a los desarrolladores les resulta fácil de usar, aprovecha su inversión en las herramientas actuales y reduce la fricción al integrar la seguridad en sus procesos. La seguridad de las aplicaciones de OpenText se incorpora a su proceso de DevOps. La velocidad de DevOps a escala empresarial no significa sacrificar la seguridad y poner en riesgo su negocio.

OpenText aprovecha Swagger en todas nuestras API para proporcionar documentación/referencia automática de la API. La página de [seguridad de las aplicaciones de OpenText](#) tiene varios proyectos con ejemplos de cómo aprovechar nuestras diversas API para realizar las tareas más solicitadas. La referencia API está incorporada en los productos y se puede acceder a ella a través de la interfaz web de los respectivos productos.

### **Implantación de software más rápida**

Gracias a las opciones de automatización para análisis estáticos y dinámicos, y a la disponibilidad de integraciones para las herramientas de desarrollo más populares (como Visual Studio, Eclipse y Jenkins), los equipos de desarrollo pueden ahorrar tiempo y reducir los malentendidos. Las integraciones con sistemas de gestión de defectos, como JIRA o BugZilla, mejoran el manejo y la resolución de problemas de seguridad y garantizan que se puedan gestionar de la misma manera que la organización gestiona los problemas funcionales. Este enfoque eficaz conlleva un desarrollo y una implantación de software más rápidos, que satisfacen las necesidades empresariales en cuanto a velocidad.

---

## Riesgos reducidos

Aumentar la seguridad en las primeras fases para cubrir el ciclo de vida de desarrollo del software de forma integrada y automatizada permite a las organizaciones reducir sus riesgos y gastos asociados, ya que se reduce el coste de corregir las vulnerabilidades en las primeras fases del proceso. El plugin Fortify Security Assistant y la automatización de los análisis de seguridad basados en Jenkins o Azure DevOps ayudan a la organización de desarrollo a realizar las pruebas de seguridad en una etapa temprana del proceso, así como en cualquier momento posterior.

## Mejor retorno de la inversión

Fortify trabaja con las herramientas de desarrollo existentes para proteger su inversión y permite a los equipos de desarrollo usar de forma continuada sus herramientas favoritas. Gracias al plugin Fortify Security Assistant, por ejemplo, los desarrolladores no necesitan aprender a usar una herramienta distinta para llevar a cabo análisis de seguridad del código, ya que funciona dentro de su IDE actual. O con las integraciones de análisis estático, los análisis de seguridad se realizan como parte del proceso de compilación y los desarrolladores reciben los problemas de seguridad dentro del sistema de gestión de defectos, sin introducir ninguna complejidad en las herramientas y procesos existentes.

## Paso 4: Automatizar la seguridad como parte de los procesos de desarrollo y pruebas

La automatización del desarrollo, los procesos, el aprovisionamiento de servidores y la implementación de aplicaciones es la clave para ser eficiente con la iniciativa DevOps. La automatización permite a las organizaciones desarrollar y lanzar aplicaciones de alta calidad de manera más rápida. Para conseguir integrar la seguridad de las aplicaciones en el desarrollo, se puede utilizar la automatización del mismo modo con las pruebas de seguridad para mantener la misma calidad a una velocidad mayor. La automatización consiste en incluir la seguridad como parte de las cadenas de herramientas de DevOps. Esto puede ocurrir en el IDE durante la codificación, en las fases de confirmación, compilación y prueba. Se trata de un aspecto fundamental de todos los programas de seguridad de las aplicaciones. Automatizar las pruebas de seguridad permite crear y ejecutar pruebas de seguridad automatizadas, del mismo modo que lo haría con pruebas de unidades o de integración.

Los análisis automatizados dinámicos o estáticos permiten identificar eficazmente las vulnerabilidades de seguridad en el código fuente y reducir la laboriosidad de las evaluaciones de seguridad. El hecho de contar con un análisis automatizado del código no solo reduce los tiempos de revisión, evaluación de seguridad y pruebas, sino que conlleva un ahorro de gastos de corrección al encontrar antes las vulnerabilidades.

## Paso 5: Pensar en el futuro

El cambio actual en el que el desarrollo moderno es más dinámico que nunca, con mayor velocidad y complejidad, lo que se traduce en una migración continua a las API, los microservicios, laC y más. Garantizar la seguridad de este panorama cambiante será cada vez más crucial en los próximos meses y años.

---

## Primeros pasos

La seguridad de las aplicaciones integrada en el desarrollo e incorporada a lo largo de todo el ciclo de vida de desarrollo del software crea procesos controlados y de riesgo reducido, lo que, en última instancia, reduce los gastos, mejora el tiempo de comercialización y optimiza el esfuerzo. Disponer de un itinerario claro hacia la integración y automatización de la seguridad de las aplicaciones con KPI cuantificables aumentará las probabilidades de éxito de su organización. La seguridad de las aplicaciones proporciona beneficios más fáciles de demostrar en comparación con otras inversiones en ciberseguridad. La evidencia del progreso realizado y del retorno de la inversión garantizará una inversión continuada en seguridad de las aplicaciones.

Estas son algunas de las consideraciones importantes que debe tener en cuenta a la hora de trazar la hoja de ruta de ese proceso:

- Identifique a sus expertos en seguridad de las aplicaciones.
- Desarrolle su estrategia y sus procesos principales antes de la implantación.
  - Defina el alcance inicial y las métricas clave, como: Las aplicaciones y los equipos de desarrollo con los que empezar
  - La posibilidad de emplear las SAST, DAST o ambas
  - Qué integraciones aprovechar
  - La posibilidad de usar herramientas de seguridad in situ, a pedido, o un enfoque híbrido
  - Cuáles son las mejoras esperadas en un plazo de 12 meses respecto a la línea de base.
- Encuentre las herramientas adecuadas para su organización.

Así, medir su éxito es crucial. Los KPI adecuados permiten a su organización no solo medir eficazmente su postura de seguridad, sino también justificar el gasto y la inversión continua en su programa de seguridad. Los KPI deben ajustarse a los objetivos de la empresa/programa. A continuación, se indican algunos que se deben tener en cuenta:

- **Tendencia de riesgo ponderado:** Representación del riesgo basada en el negocio de los defectos de seguridad de las aplicaciones web evaluados durante un periodo de tiempo especificado o casos repetidos del desarrollo de aplicaciones.
- **Ventana de corrección de defectos de seguridad:** Tiempo transcurrido desde que se identifica un defecto de seguridad en una aplicación web verificada hasta que se verifica su cierre. Se puede hacer referencia a este indicador como tiempo medio de espera hasta la corrección (MTTR).
- **Tasa de recurrencia de defectos de seguridad:** Tasa, a lo largo del tiempo, en la que los defectos de seguridad de las aplicaciones web previamente cerrados se vuelven a introducir en una aplicación, organización u otra unidad lógica determinada.
- **Relación de seguridad respecto a los defectos de calidad:** Relación entre los defectos de seguridad y el número total de defectos de calidad del software que se generan (funcional + rendimiento + seguridad).

---

## ¿Por qué OpenText?

Las personas, los procesos y la tecnología son los componentes esenciales de la seguridad de las aplicaciones por los desarrolladores. OpenText tiene la experiencia y los recursos, así como la tecnología, las personas y los procesos (a través de [OpenText™ Core Application Security \(Fortify\)](#) para ayudarle en cada paso del proceso.

OpenText proporciona una solución de seguridad de las aplicaciones integral y flexible con modelos in situ, a pedido e híbridos. Con beneficios cuantificables, como el tiempo de comercialización un 30 % más rápido, un 95 % menos de positivos, análisis entre un 10 y un 15 % más rápidos, soluciones 10 veces más rápidas y la detección del doble de vulnerabilidades, OpenText sigue siendo líder en el sector de las herramientas de seguridad de las aplicaciones.

### Elija OpenText para:

**Inicios sencillos:** Puede empezar en un día con [OpenText Core Application Security](#).

**Integración intuitiva en los procesos existentes:** La seguridad de las aplicaciones de OpenText se integra fácilmente en el entorno que a sus desarrolladores les resulta familiar, lo que convierte a la seguridad en un componente natural para sus herramientas y procesos.

**Automatización y escalabilidad rápidas:** La mayoría de los análisis de OpenText se completan en cuestión de minutos. Además, puede procesar los datos brutos de los análisis para obtener resultados de auditoría con funciones asistidas por ordenador también en pocos minutos. Los análisis automatizados pueden iniciarse como parte de las comprobaciones de código, las comprobaciones, las compilaciones, las actualizaciones u otros componentes de los canales de la integración continua y la implantación continua (CI/CD). Los clientes de OpenText disfrutan de una gran facilidad de ampliación in situ al usar técnicas de análisis centralizado, OpenText Core Application Security o un enfoque híbrido.

**Resultados precisos y cobertura en lenguajes de programación:** Los clientes de OpenText experimentan un número mayor de verdaderos positivos (más hallazgos validados) y menos falsos positivos (menos ruido) en comparación con otros productos. OpenText también admite más de 27 lenguajes de programación, la cobertura más amplia disponible.

**Reconocimiento unánime en el sector:** OpenText (anteriormente Fortify) se considera una de las mejores herramientas en materia de seguridad para las aplicaciones de los últimos 15 años. Ha sido reconocida como líder en el Gartner Magic Quadrant for Application Security por octavo año consecutivo. OpenText cuenta con la confianza de las empresas más importantes de numerosos mercados verticales de todo el mundo.

### Cree software seguro con rapidez con la ayuda de OpenText con estas características clave:

- El **plugin Security Assistant** proporciona un análisis de seguridad del código en tiempo real. Solucione cada problema con la confianza de saber que solo se marcan los problemas importantes.
- Las **plantillas GitHub Actions y GitLab CI** permiten integrar y automatizar las pruebas de seguridad de las aplicaciones estáticas (SAST) en sus flujos de trabajo de CI/CD.

---

## Recursos

Póngase en contacto con nosotros

[opentext.com](https://opentext.com)

[LinkedIn](#) ›

[X](#) ›

- El **análisis de susceptibilidad** permite a los desarrolladores o profesionales de la seguridad comprobar si alguien ha invocado una vulnerabilidad en su código personalizado. Lo que es más importante, pueden comprobar si la entrada que controla el atacante alcanza la función del código.
- **Speed Dial** en OpenText Static Application Security Testing permite a los desarrolladores controlar mejor la profundidad y la velocidad de sus análisis estáticos.
- **Commit Scan** ofrece a los desarrolladores análisis automatizados y ligeros en su flujo de trabajo, integrando pruebas estáticas en el proceso de confirmación de Git, proporcionándoles información inmediata sobre el código que se está comprobando en GitHub, GitLab y Bitbucket.
- **Fortify Audit Assistant** minimiza la carga de trabajo del auditor con el aprendizaje automático para identificar las vulnerabilidades a partir de los resultados de OpenText Static Application Security Testing. De este modo, se reducen los problemas que requieren un examen manual profundo.
- **Smart View** en Audit Workbench ayuda a los desarrolladores a comprender rápidamente cómo se relacionan varios problemas desde la perspectiva del flujo de datos, con la posibilidad de clasificar los problemas de seguridad y después solucionarlos en el punto más eficiente.