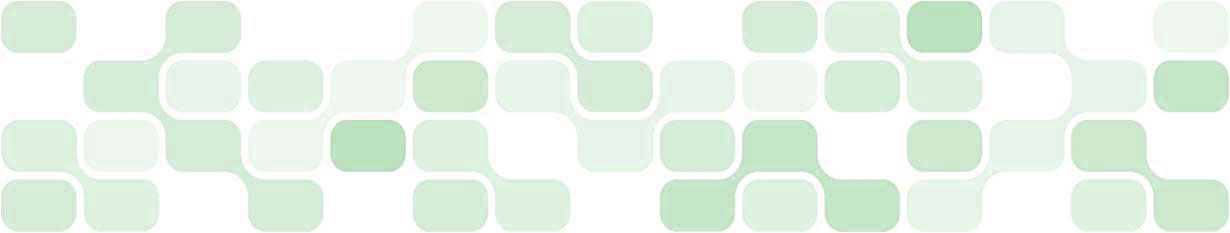


Implementación simplificada del proceso SDL de Microsoft



2 de febrero de 2010



Para obtener la información más actualizada, vea <http://www.microsoft.com/sdl>.

Este documento se proporciona “tal cual.” La información y opiniones expresadas en este documento, incluidas la URL y otras referencias a sitios Web de Internet, pueden cambiar sin previo aviso. Acepta el riesgo que pueda conllevar su utilización.

Este documento no le proporciona ningún derecho legal en relación a ninguna propiedad intelectual respecto de cualquier producto de Microsoft. Podrá copiar y utilizar este documento, a efectos de consulta internos.

© 2010 Microsoft Corporation. Todos los derechos reservados.

Con licencia bajo [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported](http://creativecommons.org/licenses/by-nc-sa/3.0/).

Argentina <http://creativecommons.org/licenses/by-nc-sa/2.5/ar/>, Chile <http://creativecommons.org/licenses/by-nc-sa/2.0/cl/>, Columbia <http://creativecommons.org/licenses/by-nc-sa/2.5/co/>, Ecuador <http://creativecommons.org/licenses/by-nc-sa/3.0/ec/>, Guatemala <http://creativecommons.org/licenses/by-nc-sa/3.0/gt/>, México <http://creativecommons.org/licenses/by-nc-sa/2.5/mx/>, Perú <http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>, Puerto Rico <http://creativecommons.org/licenses/by-nc-sa/3.0/pr/>, España <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>

**Contenido**

[Introducción 3](#_Toc264556234)

[Ciclo de vida de desarrollo de seguridad de Microsoft 3](#_Toc264556235)

[Desarrollo seguro en Microsoft 4](#_Toc264556236)

[Modelo de optimización del proceso SDL de Microsoft 4](#_Toc264556237)

[Aplicabilidad de SDL 5](#_Toc264556238)

[Roles, responsabilidades y cualificación del personal de seguridad 6](#_Toc264556239)

[Actividades de seguridad simplificadas de SDL 7](#_Toc264556240)

[Actividades de seguridad obligatorias 8](#_Toc264556241)

[Requisitos anteriores a SDL: formación en materia de seguridad 8](#_Toc264556242)

[Primera fase: requisitos 9](#_Toc264556243)

[Segunda fase: diseño 10](#_Toc264556244)

[Tercera fase: implementación 11](#_Toc264556245)

[Cuarta fase: comprobación 12](#_Toc264556246)

[Quinta fase: lanzamiento 13](#_Toc264556247)

[Actividades de seguridad opcionales 14](#_Toc264556248)

[Otros requisitos del proceso 15](#_Toc264556249)

[Análisis de causa raíz 15](#_Toc264556250)

[Actualizaciones periódicas del proceso 15](#_Toc264556251)

[Proceso de comprobación de la seguridad de las aplicaciones 15](#_Toc264556252)

[Conclusión 17](#_Toc264556253)

[Apéndice A: ilustración del proceso SDL 18](#_Toc264556254)

# Introducción

El presente documento tiene como objetivo ilustrar los conceptos básicos del ciclo de vida de desarrollo de seguridad (SDL, Security Development Lifecycle) de Microsoft y abordar cada una de las actividades de seguridad que deben llevarse a cabo para lograr la conformidad con el proceso SDL.

Este documento incluye:

* Una breve información general sobre el proceso SDL de Microsoft.
* Información general sobre el modelo de optimización del proceso SDL de Microsoft, con especial hincapié en cómo SDL encaja en el modelo de optimización.
* Una descripción de cada uno de los procedimientos de desarrollo de seguridad de Microsoft, incluidos:

Roles y responsabilidades de las personas implicadas en el proceso de desarrollo de aplicaciones.

Actividades de seguridad obligatorias.

Actividades de seguridad opcionales.

Proceso de comprobación de la seguridad de las aplicaciones.

El proceso que se describe en el presente documento establece un umbral *mínimo* para la conformidad con SDL. Dicho eso, las organizaciones no son uniformes: los equipos de desarrollo deben aplicar SDL teniendo en cuenta las capacidades y los recursos humanos disponibles, pero sin poner en peligro los objetivos de la organización en materia de seguridad.

Cabe observar que el presente documento aborda únicamente las metodologías de seguridad para el desarrollo de aplicaciones de software y de servicios en línea tanto para uso externo como interno. En las organizaciones, hay otras metodologías (por ejemplo, procedimientos de seguridad de TI) orientadas a los riesgos de seguridad “operativos”; si bien los grupos encargados de administrar estos procesos pueden estar sujetos a contextos tecnológicos y normativos similares a los de los grupos de desarrollo de software, normalmente no crean software de aplicaciones para entornos con riesgos significativos para la seguridad.

En la medida de lo posible, se proporcionan referencias a fuentes de información públicas. Asimismo, el documento incluye vínculos web a información específica de procesos, herramientas y otro tipo de información complementaria.

# Ciclo de vida de desarrollo de seguridad de Microsoft

El ciclo de vida de desarrollo de seguridad (SDL) es un proceso de control de seguridad orientado al desarrollo de software. Siendo una iniciativa corporativa y una directiva obligatoria desde el año 2004, SDL ha desempeñado un papel muy importante en la integración de la seguridad y de la privacidad en el software y la cultura de Microsoft. Con un enfoque tanto holístico como práctico, SDL tiene como objetivo reducir el número y la gravedad de las vulnerabilidades en el software. SDL introduce la seguridad y la privacidad en todas las fases del proceso de desarrollo.

El proceso SDL de Microsoft se basa en tres conceptos básicos: *formación, mejora continua de los procesos y responsabilidad*. La formación continua de los roles técnicos dentro de un grupo de desarrollo de software es fundamental. Si se invierte de manera apropiada en la transferencia de los conocimientos, las organizaciones podrán responder adecuadamente a los cambios que sufren las tecnologías y las amenazas. Dado que los riesgos para la seguridad no son estáticos, SDL destaca especialmente la importancia de comprender la causa y el efecto de las vulnerabilidades de seguridad y *requiere* una evaluación periódica de los procesos de SDL y la introducción de cambios como respuesta a los avances tecnológicos o las nuevas amenazas. Se recopilan datos para evaluar la eficacia de la formación, se usan métricas de procesos para documentar su conformidad y métricas posteriores al lanzamiento ayudan a definir los futuros cambios. Por último, SDL requiere el archivado de todos los datos necesarios para realizar el mantenimiento de una aplicación en caso de que surjan problemas. Si lo combinan con detallados planes de comunicación y de respuesta en materia de seguridad, las organizaciones podrán orientar de manera concisa y contundente a todas las partes implicadas.

# Desarrollo seguro en Microsoft

*El proceso SDL de Microsoft que se describe en el presente documento va orientado a la aplicación de procedimientos de seguridad de eficacia probada en distintos momentos a lo largo del ciclo de vida de desarrollo de software. Este proceso es válido para diversas tecnologías y no se limita a las grandes empresas, por lo que es fácil de aplicar en organizaciones de cualquier tamaño.*

El presente documento se centra en la aplicación del proceso SDL de Microsoft a los procedimientos de desarrollo que suelen seguir los grupos externos a Microsoft. Existe una estrecha correlación entre el presente documento y el proceso publicado que se aplica al desarrollo de los productos y servicios de Microsoft. No obstante, cabe observar que Microsoft ha ampliado los conceptos básicos abordados en el presente documento y aplica SDL como un proceso que refleja los contextos empresarial y técnico de la organización. En Microsoft, SDL ha evolucionado con respecto a los conceptos básicos descritos en este documento y se ha aplicado a cientos de productos de Microsoft que se ejecutan en diversos sistemas operativos y plataformas de hardware. Son estos mismos conceptos básicos los que usan los equipos de Microsoft en más de cien países del mundo para hacer frente a los retos de seguridad y privacidad específicos que plantea ese amplio abanico de tecnologías.

Microsoft cree firmemente en la transparencia de los procesos de seguridad y privacidad. Como respuesta a la solicitud por parte de los usuarios de más información sobre cómo Microsoft aplica SDL para proteger sus productos y servicios, se ha publicado una descripción detallada del proceso SDL de Microsoft en [www.microsoft.com/sdl](http://www.microsoft.com/sdl).

# Modelo de optimización del proceso SDL de Microsoft

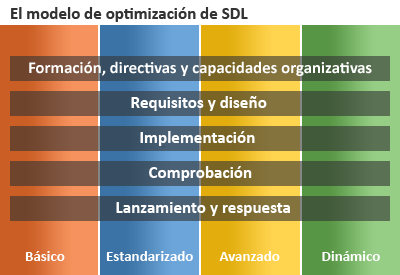
La integración de los conceptos de desarrollo seguro en un proceso de desarrollo existente puede resultar intimidante y costosa si no se procede de manera adecuada. Muchas veces, el éxito o el fracaso depende de variables como el tamaño de la organización, los recursos (tiempo, talento y presupuestos) y el respaldo de los directivos. Para poder controlar el impacto de estos intangibles, hace falta comprender los elementos que componen procedimientos de desarrollo de seguridad eficaces y establecer prioridades de implementación según el nivel de madurez del equipo de desarrollo. Microsoft ha creado el modelo de optimización de SDL como ayuda para abordar estas cuestiones.

El modelo de optimización de SDL está estructurado en torno a cinco áreas de capacidades que, en líneas generales, se corresponden con las fases del ciclo de vida de desarrollo de software:

* Formación, directivas y capacidades organizativas
* Requisitos y diseño
* Implementación
* Comprobación
* Lanzamiento y respuesta

Además, el modelo de optimización de SDL define cuatro niveles de madurez para los procedimientos y capacidades de estas áreas: básico, estandarizado, avanzado y dinámico.

El modelo de optimización de SDL comienza en el nivel de madurez básico, con pocos o sin procesos, cursos de formación y herramientas, y evoluciona hacia el nivel dinámico, que se caracteriza por la total conformidad con SDL en toda la organización. La total conformidad con SDL incluye procesos eficaces, personas altamente cualificadas, herramientas especializadas y un alto grado de responsabilidad por parte de las personas tanto internas como externas a la organización.



El presente documento se centra en las tareas y los procesos necesarios para lograr el nivel de madurez “avanzado”. Es decir, el nivel en que una organización que ha demostrado su competencia en cada una de las cinco áreas de capacidades anteriormente mencionadas puede afirmar razonablemente que sigue los procedimientos de SDL de Microsoft.

A diferencia de otros modelos de madurez de software, el modelo de optimización de SDL de Microsoft se centra estrictamente en las mejoras de los procesos de desarrollo. Proporciona instrucciones prácticas sobre cómo pasar de un nivel de madurez de procesos más bajo a un nivel de madurez más alto y evita el enfoque “lista de listas” de los otros modelos de optimización.

*Figura 1. Modelo de optimización de SDL con los niveles de capacidades  
y madurez*

# Aplicabilidad de SDL

Es de vital importancia definir claramente las expectativas de una organización con respecto a los tipos de proyectos que se van a someter a los controles impuestos por el proceso SDL de Microsoft. Por experiencia, se recomienda someter a SDL las aplicaciones con una o varias de las siguientes características:

* Aplicaciones implementadas en un entorno empresarial
* Aplicaciones que procesan información de identificación personal (PII) u otro tipo de información confidencial
* Aplicaciones que se comunican frecuentemente a través de Internet u otras redes

Teniendo en cuenta la propagación de las tecnologías informáticas y la evolución del entorno de amenazas, quizás sea más fácil identificar los proyectos de desarrollo de aplicaciones que *no estén* sujetos a los controles de seguridad como los de SDL.

# Roles, responsabilidades y cualificación del personal de seguridad

SDL incluye criterios generales y descripciones de trabajo de los roles de seguridad y privacidad. Estos roles se asignan durante la fase de requisitos del proceso SDL. Son de índole consultiva y proporcionan la estructura organizativa necesaria para identificar, catalogar y mitigar los problemas de seguridad y privacidad de un proyecto de desarrollo de software. Entre estos roles, se incluyen:

* **Roles de revisión o asesoramiento.** Estos roles se han diseñado para supervisar la seguridad   
  y la privacidad de los proyectos y tener autoridad para aceptar o rechazar los planes de seguridad y privacidad de un equipo de proyecto.

*Asesor de seguridad/Asesor de privacidad.* Estos roles se asignan   
a expertos en la materia ajenos al equipo de proyecto. Puede tratarse de un miembro cualificado de un grupo centralizado independiente dentro de la organización que se ha constituido específicamente para este tipo de revisiones o puede tratarse de un experto ajeno a la organización. La persona elegida para esta tarea debe desempeñar dos subroles:

*Es preferible recurrir a un grupo de asesoramiento interno y centralizado; proporciona un contexto organizativo y conocimientos de los procesos más allá de las capacidades de expertos externos.*

* + - * Auditor. Esta persona debe supervisar cada fase del proceso de desarrollo de software y certificar que se cumple cada uno de los requisitos de seguridad. El asesor debe tener la libertad de certificar el cumplimiento (o incumplimiento) de los requisitos de seguridad y privacidad sin que interfiera el equipo de proyecto.
      * Experto. La persona elegida para el rol de asesor debe disponer de conocimientos contrastados en materia de seguridad.

*Combinación de los roles de asesoramiento.* El rol de asesor de seguridad puede combinarse con el rol de asesor de privacidad si se logra identificar a una persona con los conocimientos y la experiencia apropiados.

* **Líderes del equipo.** Estos roles se deben asignar a los expertos en la materia del equipo de proyecto. Se encargan de la negociación, la aceptación y el seguimiento de los requisitos mínimos de seguridad y privacidad, además de mantener una comunicación transparente con los asesores y responsables de la toma de decisiones a lo largo de un proyecto de desarrollo de software.

*Líder de seguridad/Líder de privacidad.* Esta persona (o grupo de personas) no solo es responsable de asegurar que se han resuelto todos los problemas de seguridad de una versión de software sino que también se encarga de coordinar y realizar un seguimiento de los problemas de seguridad del proyecto. Este rol también notifica el estado al asesor de seguridad y a otros componentes pertinentes del equipo de proyecto (por ejemplo, los responsables de desarrollo y de pruebas).

*Combinación de roles.* Al igual que en el caso de los roles de asesor de seguridad y asesor de privacidad, las responsabilidades atribuidas al rol de líder pueden combinarse si se logra identificar a una persona con los conocimientos y la experiencia apropiados.

# Actividades de seguridad simplificadas de SDL

*El énfasis debe recaer en la exactitud de los resultados en cada una de las fases. Sofisticados informes con información incompleta o incorrecta significan una pérdida de tiempo   
y de recursos.*

En términos sencillos, el proceso SDL de Microsoft es un conjunto de actividades de seguridad obligatorias, que se presentan en el orden en que deben llevarse a cabo y se agrupan por cada una de las fases de un ciclo de vida de desarrollo de software tradicional. Muchas de las actividades descritas aportarían *ciertos* beneficios en materia de seguridad si se implementaran de manera independiente. Sin embargo, la experiencia en Microsoft demuestra que las actividades de seguridad realizadas como parte de un proceso de desarrollo de software aportan mayores beneficios que las actividades implementadas de manera poco sistemática o de modo ad hoc.

Es posible agregar actividades de seguridad opcionales a discreción del equipo de proyecto o del asesor de seguridad a fin de alcanzar los objetivos deseados en el ámbito de la seguridad y la privacidad. Para mayor brevedad, no se incluyen descripciones detalladas de las actividades de seguridad.



*Figura 2. Ciclo de vida de desarrollo de software de Microsoft: simplificado*

Como concepto fundamental, cabe observar que una organización debe enfocarse en la calidad y la integridad de los resultados obtenidos en cada fase. De las organizaciones situadas en los niveles Avanzado y Dinámico del modelo de optimización de SDL, se espera que los procesos de seguridad tengan cierto grado de sofisticación. Dicho eso, da igual que un modelo de riesgos sea generado, por ejemplo, como resultado de una sesión de pizarra con el equipo de desarrollo, se redacte como un relato en un documento de Microsoft Word o se elabore con ayuda de una herramienta especializada, como la herramienta de creación de modelos de riesgos de SDL. Si bien el proceso SDL se beneficia de las inversiones en herramientas y procesos de automatización eficaces, su verdadero valor reside en los resultados exhaustivos y exactos.

Para facilitar la revisión, se incluye un diagrama detallado del flujo de procesos de SDL en el [Apéndice A](#_Apéndice_A:_ilustración). En este diagrama, se muestran las actividades de seguridad que se llevan a cabo en un hipotético proyecto, desde la formación de los empleados hasta el lanzamiento de la aplicación. Este ejemplo incluye tanto las tareas obligatorias como las opcionales.

## Actividades de seguridad obligatorias

Si se determina que un proyecto de desarrollo de software debe someterse al proceso SDL (vea la sección [Aplicabilidad de SDL](#_Aplicabilidad_de_SDL)), el equipo de desarrollo deberá realizar correctamente dieciséis actividades de seguridad obligatorias para lograr la conformidad con el proceso SDL de Microsoft. Expertos en seguridad y privacidad han reconocido la eficacia de estas actividades obligatorias, que se revisan constantemente como parte de un riguroso proceso de evaluación anual. Tal y como se menciona anteriormente, los equipos de desarrollo deben mantener la flexibilidad para especificar otras actividades de seguridad según sean necesarias, pero en la lista de procedimientos de “ejecución obligatoria” siempre deben aparecer las dieciséis actividades incluidas en el presente documento.

### Requisitos anteriores a SDL: formación en materia de seguridad

#### Procedimiento 1 de SDL: requisitos de formación

Todos los miembros de un equipo de desarrollo de software deben recibir una formación apropiada con el fin de mantenerse al corriente de los conceptos básicos y últimas tendencias en el ámbito de la seguridad y privacidad. Las personas con roles técnicos (desarrolladores, evaluadores y administradores de programas) que están directamente implicadas en el desarrollo de programas de software deben asistir como mínimo una vez al año a una clase de formación en materia de seguridad.

La formación básica en materia de seguridad de software debe abordar conceptos fundamentales como:

* Diseño seguro, incluidos los siguientes temas:
  + Reducción de la superficie de ataques
  + Defensa en profundidad
  + Principio de privilegios mínimos
  + Configuraciones predeterminadas seguras
* Modelos de riesgos, incluidos los siguientes temas:
  + Información general sobre los modelos de riesgos
  + Implicaciones de diseño de un modelo de riesgos
  + Restricciones de codificación basadas en un modelo de riesgos
* Codificación segura, incluidos los siguientes temas:
  + Saturaciones de búfer (para aplicaciones que usen C y C++)
  + Errores aritméticos de enteros (para aplicaciones que usen C y C++)
  + XSS (para código administrado y aplicaciones web)
  + Inyección de código SQL (para código administrado y aplicaciones web)
  + Criptografía débil
* Pruebas de seguridad, incluidos los siguientes temas:
  + Diferencias entre pruebas de seguridad y pruebas funcionales
  + Evaluación de riesgos
  + Métodos para poner a prueba la seguridad
* Privacidad, incluidos los siguientes temas:
  + Tipos de datos confidenciales
  + Procedimientos recomendados de diseño de privacidad
  + Evaluación de riesgos
  + Procedimientos recomendados de desarrollo de privacidad
  + Procedimientos recomendados de pruebas de privacidad

La formación anterior establece una línea base adecuada para los conocimientos del personal técnico. Según lo permitan el tiempo y los recursos, quizás sea necesario proporcionar formación en materia de conceptos avanzados. Algunos ejemplos son los siguientes:

* Diseño y arquitectura de seguridad avanzados
* Diseño de interfaces de usuario de confianza
* Información detallada sobre vulnerabilidades de seguridad
* Implementar mitigaciones de amenazas personalizadas

### Primera fase: requisitos

#### Procedimiento 2 de SDL: requisitos de seguridad

La necesidad de considerar la seguridad y la privacidad “desde el primer instante” es un aspecto fundamental del desarrollo de un sistema seguro. La fase de planificación inicial es el momento más apropiado para definir los requisitos de fiabilidad de un proyecto de software. Al definir los requisitos en una fase temprana, los equipos de desarrollo podrán identificar los principales hitos y resultados, y es posible integrar la seguridad y la privacidad de modo que los planes y programaciones se alteren lo menos posible. Al principio de un proyecto, se realiza un análisis de los requisitos de seguridad y privacidad, el cual incluye la especificación de los requisitos de seguridad mínimos de la aplicación en el entorno operativo previsto, así como la especificación y la implementación de un sistema de seguimiento de los elementos de trabajo y de las vulnerabilidades de seguridad.

#### Procedimiento 3 de SDL: umbrales de calidad y límites de errores

Se usan umbrales de calidad y *límites de errores* para establecer niveles mínimos aceptables de calidad en materia de seguridad y privacidad. Al definir estos criterios al comienzo de un proyecto, se comprenderán mejor los riesgos asociados a los problemas de seguridad y los equipos podrán identificar y corregir los errores de seguridad durante el desarrollo. El equipo de proyecto debe negociar los umbrales de calidad (por ejemplo, todas las advertencias del compilador se deben evaluar y corregir antes de proteger el código) para cada fase de desarrollo; a continuación, el asesor de seguridad debe aprobarlos y puede agregar aclaraciones específicas del proyecto así como requisitos de seguridad más estrictos según proceda. El equipo de proyecto también debe ilustrar la conformidad con los umbrales de calidad negociados para cumplimentar la [revisión de seguridad final.](#_Procedimiento_15_de)

Un límite de errores es un umbral de calidad que se aplica a todo el proyecto de desarrollo de software. Se usa para definir los umbrales de gravedad de las vulnerabilidades de seguridad; por ejemplo, en la aplicación no hay vulnerabilidades conocidas que estén clasificadas como “críticas” o “importantes” en el momento de su lanzamiento. Una vez definido, el límite de errores no debe bajarse nunca. Un límite de errores dinámico es un objetivo no fijo que probablemente no se entienda muy bien en el ámbito del desarrollo.

#### Procedimiento 4 de SDL: evaluación de los riesgos de seguridad y privacidad

Las evaluaciones de los riesgos de seguridad (SRA) y de privacidad (PRA) son procesos obligatorios que identifican los aspectos funcionales del software que requieren una revisión exhaustiva. Dichas evaluaciones deben incluir la siguiente información:

1. (Seguridad) Partes del proyecto que van a requerir [modelos de riesgos](#_Procedimiento_7_de) antes del lanzamiento.
2. (Seguridad) Las partes del proyecto que van a requerir revisiones del diseño de seguridad antes del lanzamiento.
3. (Seguridad) Partes del proyecto (si procede) que van a requerir pruebas de penetración por parte de un grupo externo al equipo de proyecto y acordado mutuamente.
4. (Seguridad) Requisitos de análisis o de pruebas adicionales que el asesor de seguridad considera necesarios para mitigar los riesgos de seguridad.
5. (Seguridad) Ámbito específico de los [requisitos en materia de pruebas de exploración de vulnerabilidades mediante datos aleatorios](#_Procedimiento_12_de).
6. (Privacidad) ¿Cuál es el nivel de impacto sobre la privacidad? La respuesta a esta pregunta se basa en las siguientes pautas:

* P1 Alto riesgo para la privacidad. La característica, el producto o el servicio almacena o transfiere datos de identificación personal, cambia la configuración o las asociaciones de los tipos de archivo, o instala software.
* P2 Riesgo moderado para la privacidad. El único comportamiento que afecta a la privacidad en la característica, el producto o el servicio es una transferencia de datos anónimos única e iniciada por el usuario (por ejemplo, el usuario hace clic en un vínculo y el software abre un sitio web).
* P3 Bajo riesgo para la privacidad. La característica, el producto o el servicio no tiene ningún comportamiento que afecte a la privacidad. No hay ninguna transferencia de datos personales ni anónimos, no se almacenan datos de identificación personal en el equipo, no se modifica ninguna configuración en nombre del usuario y no se instala ningún software.

### Segunda fase: diseño

#### Procedimiento 5 de SDL: requisitos de diseño

El momento más oportuno para influir en la fiabilidad del diseño de un proyecto es la etapa inicial de su ciclo de vida. Es muy importante que se estudien detenidamente las cuestiones de seguridad y de privacidad durante la fase de desarrollo. La mitigación de los problemas de seguridad y de privacidad es mucho menos costosa si se realiza en la etapa inicial del ciclo de vida de un proyecto. Los equipos de proyecto deben evitar abordar deprisa y corriendo las características de seguridad y de privacidad y las mitigaciones cerca del final del desarrollo de un proyecto.

Además, es de suma importancia que los equipos de proyecto entiendan la diferencia entre “características seguras” y “características de seguridad”. Es perfectamente posible que se implementen características de seguridad que, en realidad, son inseguras. Las *características seguras* se definen como características cuya funcionalidad está debidamente diseñada con respecto a la seguridad, incluida una rigurosa validación de todos los datos antes de su procesamiento o una implementación criptográficamente robusta de bibliotecas para los servicios de cifrado. El término *características de seguridad* describe la funcionalidad de un programa que tiene implicaciones para la seguridad, como la autenticación Kerberos o un firewall.

*Cualquier proceso de desarrollo de software debe incluir un método formal de aplazamiento de excepciones o errores. Un gran número de aplicaciones se basan en código y diseños heredados, por lo que puede ser necesario aplazar determinadas medidas de seguridad o privacidad como resultado de las restricciones técnicas.*

La actividad de los requisitos de diseño incluye varias acciones obligatorias. Algunos ejemplos son la creación de especificaciones de diseño en materia de seguridad y privacidad, la revisión de las especificaciones y la especificación de requisitos mínimos de diseño criptográfico. Las especificaciones de diseño deben describir las características de seguridad o de privacidad que van a estar directamente expuestas a los usuarios, como las que requieren la autenticación del usuario para obtener acceso a datos específicos o que necesitan el consentimiento del usuario para usar una característica con un alto riesgo para la privacidad. Además, todas las especificaciones de diseño deben describir cómo se implementa de manera segura toda la funcionalidad proporcionada por una característica o función determinada. Se recomienda validar las especificaciones de diseño con la especificación funcional de la aplicación. La especificación funcional debe:

* Describir de manera exacta y completa el uso previsto de una característica o función.
* Describir cómo se implementa de forma segura la característica o función.

#### Procedimiento 6 de SDL: reducción de la superficie de ataques

La reducción de la superficie de ataques está estrechamente relacionada con los modelos de riesgos, si bien aborda los problemas de seguridad desde una perspectiva ligeramente diferente. La reducción de la superficie de ataques es una forma de reducir el riesgo dando a los atacantes menos oportunidades para aprovechar un posible punto débil o una posible vulnerabilidad. Para reducir la superficie de ataques, se cierra o se restringe el acceso a los servicios del sistema, se aplica el principio de privilegios mínimos o se usan en la medida de lo posible defensas por capas.

#### Procedimiento 7 de SDL: modelos de riesgos

Los modelos de riesgos se usan en entornos donde existe un riesgo significativo para la seguridad. Este procedimiento permite a los equipos de desarrollo considerar, documentar y describir de manera estructurada las implicaciones para la seguridad de los diseños en el marco de su entorno operativo previsto. Los modelos de riesgos también permiten estudiar los problemas de seguridad en el nivel de los componentes   
o aplicaciones. La creación de un modelo de riesgos es un trabajo de equipo que implica a los administradores de programas o proyectos, desarrolladores y evaluadores; además, es la principal tarea de análisis   
de seguridad que se realiza en la fase de diseño del software.

*El método preferido para generar modelos de riesgos es la herramienta de creación de modelos de riesgos de SDL. Esta herramienta se basa en la taxonomía de la clasificación de amenazas según el método STRIDE.*

### Tercera fase: implementación

#### Procedimiento 8 de SDL: usar herramientas aprobadas

Todos los equipos de desarrollo deben definir y publicar una lista de las herramientas aprobadas y de las comprobaciones de seguridad asociadas, como las advertencias y las opciones del compilador o del vinculador. Esta lista la debe aprobar el asesor de seguridad del equipo de proyecto. En términos generales, los equipos de desarrollo deben procurar usar la versión más reciente de las herramientas aprobadas a fin de aprovechar las nuevas protecciones y funciones de análisis de seguridad.

#### Procedimiento 9 de SDL: prohibir funciones no seguras

Un gran número de las funciones y API de uso común no son seguras de cara al actual entorno de amenazas. Los equipos de proyecto deben analizar todas las funciones y API que se van a usar con un proyecto de desarrollo de software y prohibir las que consideran inseguras. Una vez determinada la lista de funciones prohibidas, los equipos de proyecto deben usar archivos de encabezado (por ejemplo, banned.h y strsafe.h), compiladores más recientes o herramientas de análisis de código para comprobar si hay funciones prohibidas en el código (incluido código heredado si procede) y reemplazarlas por alternativas más seguras.

#### Procedimiento 10 de SDL: análisis estático

*En líneas generales, los equipos de desarrollo deben determinar cuál es la frecuencia óptima con la que deben realizarse análisis estáticos de modo que haya un equilibrio entre la productividad y una cobertura adecuada de la seguridad.*

Los equipos de proyecto deben realizar un análisis estático del código fuente. Este tipo de análisis permite revisar el código de seguridad de forma escalable y contribuye a asegurar que se observan las directivas de codificación segura. En general, por sí solo, el análisis de código estático no puede reemplazar una revisión de código manual. El equipo de seguridad   
y los asesores de seguridad deben ser conscientes de las ventajas   
y desventajas de las herramientas de análisis estático y deben estar preparados para ampliar dichas herramientas con otras o con la revisión humana, según procede.

### Cuarta fase: comprobación

#### Procedimiento 11 de SDL: análisis dinámico de los programas

Es necesario comprobar los programas de software en tiempo de ejecución para asegurar que su funcionalidad sea acorde con el diseño. Esta tarea de comprobación debe especificar las herramientas que supervisan el comportamiento de las aplicaciones para detectar si la memoria está dañada, si hay problemas con los privilegios de los usuarios u otro tipo de problemas de seguridad críticos. El proceso SDL usa herramientas en tiempo de ejecución como AppVerifier, junto con otras técnicas como pruebas de exploración de vulnerabilidades mediante datos aleatorios, para lograr los niveles de cobertura deseados de las pruebas de seguridad.

#### Procedimiento 12 de SDL: pruebas de exploración de vulnerabilidades mediante datos aleatorios

Este tipo de pruebas es una forma especializada de análisis dinámico que se usa para provocar errores en los programas introduciendo deliberadamente datos aleatorios o de formato incorrecto en una aplicación. Esta estrategia se deriva del uso previsto de la aplicación y de sus especificaciones funcionales y de diseño. Es posible que el asesor de seguridad requiera un número adicional de estas pruebas o amplíe su ámbito o duración.

#### Procedimiento 13 de SDL: revisión de los modelos de riesgos y de la superficie de ataques

En muchas ocasiones, una aplicación se desvía de manera significativa de las especificaciones funcionales y de diseño creadas durante las fases de requisitos y de diseño de un proyecto de desarrollo de software. Por ello, es muy importante que se revisen los modelos de riesgos y la medición de la superficie de ataques de una aplicación una vez completado su código. De este modo, se asegura que se toman en consideración los cambios de diseño o de implementación realizados en el sistema y que se revisan y se mitigan los nuevos vectores de ataque creados como resultado de los cambios.

### Quinta fase: lanzamiento

#### Procedimiento 14 de SDL: plan de respuesta a incidentes

Cada lanzamiento de software sujeto a los requisitos de SDL debe incluir un plan de respuesta a incidentes. Incluso los programas sin vulnerabilidades conocidas en el momento de su lanzamiento pueden estar expuestos a nuevas amenazas. El plan de respuesta a incidentes debe incluir:

* Un equipo de ingeniería sostenida identificado o, si el equipo es demasiado reducido para tener recursos de ingeniería sostenida, un plan de respuesta a emergencias en el que se identifica el personal de ingeniería, marketing, comunicaciones y administración que representa el primer punto de contacto en caso de una emergencia de seguridad.
* Contactos con capacidad para tomar decisiones y disponibilidad las 24 horas del día y los 7 días de la semana.
* Planes de servicios de seguridad para código heredado de otros grupos dentro de la organización.
* Planes de servicios de seguridad para código de terceros bajo licencia, incluidos nombres de archivo, versiones, código fuente, datos de contacto de terceros y permiso contractual para realizar cambios (si procede).

#### Procedimiento 15 de SDL: revisión de seguridad final

La revisión de seguridad final es una inspección deliberada de todas las actividades de seguridad realizadas con una aplicación de software antes de su lanzamiento. Esta revisión la lleva a cabo el asesor de seguridad con ayuda del personal de desarrollo habitual y de los responsables de los equipos de seguridad y de privacidad. La revisión de seguridad final no consiste en “penetrar y parchear”, ni tampoco en realizar las actividades de seguridad que se han omitido o se han olvidado. Suele consistir en un estudio de los modelos de riesgos, las solicitudes de excepciones, los resultados de las herramientas y el rendimiento teniendo en cuenta los umbrales de calidad y los límites de errores previamente determinados. La revisión de seguridad final da lugar a uno de estos tres resultados:

* **Revisión de seguridad final superada.** Se han corregido y mitigado todos los problemas de seguridad y de privacidad identificados en la revisión de seguridad final.
* **Revisión de seguridad final superada con excepciones.** Se han corregido y mitigado todos los problemas de seguridad y de privacidad identificados en la revisión de seguridad final y/o todas las excepciones se han resuelto de modo satisfactorio. Los problemas que no se pueden resolver (por ejemplo, vulnerabilidades debido a problemas heredados del “nivel de diseño”) se registran y se corrigen en la siguiente versión.
* **Revisión de seguridad final con remisión a una instancia superior.** Si un equipo no cumple todos los requisitos de SDL y el asesor de seguridad y el equipo de proyecto no alcanzan un acuerdo aceptable, el asesor de seguridad no podrá aprobar el proyecto, por lo que no se podrá realizar su lanzamiento. Los equipos deberán intentar cumplir en la medida de lo posible todos los requisitos de SDL antes de proceder al lanzamiento del proyecto o remitir el asunto a instancias superiores para que tomen una decisión al respecto.

#### Procedimiento 16 de SDL: lanzamiento o archivado

Las versiones RTM (Release To Manufacturing) y RTW (Release To Web) dependen de la finalización del proceso SDL. El asesor de seguridad asignado a la versión debe certificar (mediante la revisión de seguridad final y otros datos) que el equipo de proyecto ha cumplido los requisitos de seguridad. De manera similar, para todos los productos que tienen al menos un componente con el nivel de impacto en la privacidad P1, el asesor de privacidad del proyecto debe certificar que el equipo de proyecto ha cumplido los requisitos de privacidad para que se pueda enviar el software.

Además, se han de archivar todos los datos e información pertinentes a fin de permitir el mantenimiento del software después de su lanzamiento. Dichos datos incluyen todas las especificaciones, código fuente, archivos binarios, símbolos privados, modelos de riesgos, documentación, planes de respuesta a emergencias, términos de licencia y de mantenimiento de software de terceros así como otros datos necesarios para poder llevar a cabo las tareas de mantenimiento posteriores al lanzamiento.

## Actividades de seguridad opcionales

Las actividades de seguridad opcionales suelen llevarse a cabo cuando existe la posibilidad de que una aplicación de software vaya a usarse en entornos o escenarios críticos. En muchas ocasiones, las especifica el asesor de seguridad como parte de un conjunto negociado de requisitos adicionales con el fin de asegurar un mayor nivel de análisis de seguridad para determinados componentes de software. Los procedimientos descritos en esta sección son algunos ejemplos de tareas de seguridad opcionales pero no representan una lista exhaustiva.

#### Revisión de código manual

La revisión de código manual es una tarea opcional del proceso SDL que suelen llevar a cabo miembros altamente cualificados del equipo de seguridad y/o el asesor de seguridad. Si bien las herramientas de análisis se encargan en gran medida de detectar y marcar las vulnerabilidades, no son perfectas. Por consiguiente, la revisión de código manual suele centrarse en los componentes “críticos” de una aplicación. En la mayoría de los casos, se recurre a esta revisión cuando se procesan o se almacenan datos confidenciales, como datos de identificación personal (PII). También se utiliza para examinar otras funciones críticas, como las implementaciones criptográficas.

#### Pruebas de penetración

Las pruebas de penetración consisten en un análisis de seguridad de caja blanca de un sistema de software que llevan a cabo profesionales cualificados en el ámbito de la seguridad simulando las acciones de un hacker. Estas pruebas tienen como objetivo detectar posibles vulnerabilidades debidas   
a errores de codificación, errores en la configuración del sistema u otros puntos débiles de la implementación operativa. En muchas ocasiones,   
se llevan a cabo junto con revisiones de código automatizadas y manuales para alcanzar un nivel de análisis mayor de lo normal.

*Todos los problemas que se identifican mediante las pruebas de penetración se deben abordar y resolver   
antes de la aprobación del lanzamiento del proyecto.*

#### Análisis de vulnerabilidades de aplicaciones similares

En Internet, hay un gran número de fuentes de información acreditadas sobre las vulnerabilidades de software. En algunos casos, los análisis de vulnerabilidades de aplicaciones de software similares pueden arrojar luz sobre los problemas de diseño o implementación con los que se puede encontrar el software en fase de desarrollo.

# Otros requisitos del proceso

## Análisis de causa raíz

Aunque normalmente no forma parte del proceso de desarrollo de software, el análisis de causa raíz desempeña un papel importante a la hora de garantizar la seguridad del software. Cuando se detecta una vulnerabilidad desconocida, se ha de llevar a cabo una investigación para determinar dónde han fallado los procesos de seguridad. Dichas vulnerabilidades pueden tener diversas causas, como un error humano, un error en una herramienta o una directiva. El análisis de causa raíz tiene como objetivo conocer exactamente la naturaleza del error. Esta información ayuda a asegurar que se tendrán en consideración errores del mismo tipo en las futuras revisiones de SDL.

## Actualizaciones periódicas del proceso

Los riesgos a los que está sujeto el software no son estáticos. Por consiguiente, el proceso que se usa para proteger el software no puede ser estático. Las organizaciones deben aprovechar los conocimientos que les aportan procedimientos como el análisis de causa raíz, los cambios en las directivas así como las mejoras tecnológicas y mejoras en el ámbito de la automatización, y deben aplicarlos al proceso SDL según un calendario previsible. En general, una actualización anual será suficiente. Hay una excepción a esta regla: cuando se identifican vulnerabilidades desconocidas, es preciso revisar inmediatamente el proceso SDL para asegurar que se apliquen las mitigaciones apropiadas.

# Proceso de comprobación de la seguridad de las aplicaciones

Lógicamente, las organizaciones que desarrollan software seguro desean disponer de un medio para comprobar que se han seguido los procesos descritos en el ciclo de vida SDL de Microsoft. El acceso a datos centralizados referentes al desarrollo y las pruebas contribuye a la toma de decisiones en diversos escenarios importantes, como la revisión de seguridad final, el control de excepciones de los requisitos de SDL y las auditorías de seguridad. La comprobación de la seguridad de una aplicación implica diferentes procesos y personas:

* Se debe usar una aplicación especialmente designada para realizar un seguimiento de la conformidad con SDL. Esta aplicación sirve como repositorio central de todos los artefactos del proceso SDL, como notas de diseño e implementación, modelos de riesgos, cargas de registros de herramientas y otras atestaciones del proceso. Al igual que sucede con otras aplicaciones críticas, se han de usar controles de acceso para asegurar que:

Solo personal autorizado pueda usar la aplicación.

Haya una separación firme de los roles. Por ejemplo, un desarrollador puede usar la aplicación y cargar datos, pero tiene prohibido el acceso a la funcionalidad reservada para los asesores de seguridad y de privacidad, los responsables de los equipos de seguridad y los evaluadores.

* + - * Los responsables de los equipos de seguridad y de privacidad deben asegurar que se introducen en la aplicación de seguimiento y se clasifican correctamente los datos necesarios para emitir un dictamen objetivo.
      * La información que se introduce en la aplicación de seguimiento la usan los asesores de seguridad y de privacidad como marco de análisis para la [revisión de seguridad final](#_Must-Complete_Practice_15:).
      * Los asesores de seguridad y de privacidad se encargan de revisar los datos que se introducen en la aplicación de seguimiento (incluidos los resultados de la revisión de seguridad final y otras tareas de seguridad adicionales asignadas por los asesores); además, son los responsables de certificar el cumplimiento de todos los requisitos y/o la resolución de todas las excepciones.

El presente documento se centra en el nivel avanzado del modelo de optimización de SDL, donde (en la mayoría de los casos) los procesos de seguimiento rudimentarios resultan insuficientes. Sin embargo, las organizaciones con procesos menos sofisticados o grupos de recursos más reducidos, es decir, las que se ajustan al nivel básico o estandarizado del modelo de optimización de SDL, probablemente se las puedan arreglar con un proceso de seguimiento más sencillo.

Es muy importante que los procesos de seguimiento y de comprobación capturen con exactitud:

* Los requisitos de seguridad y de privacidad de la organización (por ejemplo, que no haya vulnerabilidades críticas conocidas en el momento del lanzamiento).
* Los requisitos funcionales y técnicos de la aplicación en fase de desarrollo.
* El contexto operativo de la aplicación.

Por ejemplo, si un equipo de desarrollo crea una aplicación de control de procesos que se va a ejecutar en un entorno crítico, se han de asignar los recursos y el tiempo adecuados a la creación y al mantenimiento del proceso de seguimiento para que los responsables de seguridad y de privacidad así como los directivos de la organización y otras partes implicadas, como evaluadores o auditores de conformidad, puedan realizar análisis objetivos. Dicho de otro modo, si se escatima en medios para el proceso de seguimiento, inevitablemente surgirán problemas más adelante, normalmente cuando se produzca una emergencia. Asegúrese de disponer de sistemas confiables para poder responder a situaciones críticas en momentos críticos.

# Conclusión

El objetivo del presente documento es proporcionar un marco *sencillo* para la incorporación pragmática de los procedimientos de seguridad en el proceso de desarrollo de software. En él, se describen diversas actividades de desarrollo de seguridad discretas y de dominio público que, si se combinan con una automatización eficaz de los procesos y una sólida orientación en materia de directivas, representan los pasos necesarios para que una organización pueda afirmar de modo objetivo su conformidad con el proceso SDL de Microsoft, tal y como se define en el nivel avanzado del modelo de optimización de SDL.

Si bien el proceso anteriormente descrito establece un umbral *mínimo* para la conformidad con SDL, no se trata de un proceso estándar. Los equipos de desarrollo deben usar este documento como *guía* para implementar SDL de acuerdo con el tiempo, los recursos y los procedimientos empresariales de la organización.

Los conceptos de ingeniería anteriormente descritos se consideran generalmente como procedimientos de seguridad apropiados y no son específicos de Microsoft ni de la plataforma Windows. Se pueden aplicar a diversos sistemas operativos, plataformas de hardware y metodologías de desarrollo. Incluso un enfoque poco sistemático en el que se usan algunas de estas técnicas probablemente tenga un efecto beneficioso sobre la seguridad y la privacidad de una aplicación en fase de desarrollo.

Se podría decir que incluso una simple orquestación de los procesos de seguridad puede contribuir a mejoras holísticas en la seguridad y privacidad que sobrepasan el valor de las tareas realizadas cada una por separado. Sin embargo, las amenazas que están al acecho en los entornos informáticos de hoy en día han evolucionado desde los primeros “script kiddies” con afán de alimentar su ego hasta el crimen financiero organizado a gran escala y, más recientemente, los nuevos campos de batalla en los conflictos de los estados nación. Esta evolución de las amenazas aboga firmemente por algo mucho más sustancial que el típico enfoque poco sistemático de la seguridad de software.

El proceso SDL de Microsoft es un proceso de libre disposición que permite mejorar la seguridad y la privacidad de un software. Se ha aplicado a cientos de programas de software y a cientos de millones de líneas de código de producción. SDL consta de acciones obligatorias que siguen el proceso de desarrollo de software tradicional, si bien es lo suficientemente flexible para que se puedan agregar otras directivas y técnicas de modo que se cree una metodología de desarrollo de software única para una organización. La combinación de procesos, formación y herramientas aporta diversos beneficios, como una mayor previsibilidad, capacidad técnica y un software más seguro, lo que se traduce en un menor riesgo para la organización y el usuario del software.

# Apéndice A: ilustración del proceso SDL

