Universidad Cenfotec

Maestría en Ciberseguridad

Documento final de Proyecto de Investigación Aplicada 2

Propuesta de Mejoras de Seguridad en el Desarrollo de Aplicaciones de Software: Un Caso Práctico

Paulo César Sequeira Gutiérrez

Agosto de 2022

Declaratoria de derecho de autor

Se declara que el presente proyecto de investigación fue realizado por el autor Paulo César Sequeira Gutiérrez, fundamentando los diferentes capítulos del trabajo en diferentes fuentes bibliográficas, literatura citada, las cuales tienen su respectiva referencia, respetando los derechos de autor de dichos trabajos.

Se autoriza la reproducción total o parcial de este trabajo, para ser usados como referencia de trabajos futuros de tipo académico y científico, en este caso, se solicita incorporar la referencia de este trabajo respetando los derechos de los autores.

Dedicatoria

El presente trabajo lo dedico enteramente a mi esposa: si en algo soy mejor hoy que cuando nos conocimos, es a ella a quien se lo debo.

> "¿Sabes? Siempre pensé que yo te había rescatado de la torre del dragón… [pero] no, fuiste tú quien me rescató a mí".
>
> Shrek, a Fiona

Agradecimientos

Quiero agradecer primeramente a mi familia: mi esposa, mis hijos, mis padres, hermanos, suegros y cuñado; a todos ellos por su paciencia y el apoyo que me brindaron todo este tiempo a fin de poder dedicarme de lleno a este trabajo hasta poder culminarlo.

Al máster Miguel Pérez Montero le estoy profundamente agradecido por su apoyo, su orientación y su paciencia; nunca hubiera llegado hasta acá sin ellas. Me siento muy afortunado y honrado de haberlo tenido como tutor.

De nuevo a mi esposa, Dunny Apuy Achío, por la elaboración a mano de los diagramas incluidos en varias secciones; esto me libró de batallar con las aplicaciones de diagramación en momentos que necesitaba enfocarme en avanzar con la redacción del documento.

A mis profesores a lo largo de todo el Programa de Maestría en Ciberseguridad de la Universidad Cenfotec, por su tiempo y esmero a fin de introducirme en este fascinante campo de la ciberseguridad y proveerme del fundamento necesario que me permitiera ambicionar con cumplir los objetivos fijados para este trabajo.

Finalmente, también quiero agradecer a los colaboradores en La Organización donde se llevó a cabo este estudio, quienes mostraron interés por la elaboración de este y me dieron la oportunidad de aplicar y poner a prueba los conocimientos adquiridos en este Programa.

Aprobación del Proyecto

Universidad Cenfotec
Carrera de Postgrado
Maestría Profesional en Ciberseguridad

TRIBUNAL EXAMINADOR

Este proyecto fue aprobado por el Tribunal Examinador de la carrera: **Maestría Profesional en Ciberseguridad,** requisito para optar por el título de grado de **Maestría,** para el estudiante: **Sequeira Gutiérrez Paulo César.**

MIGUEL PEREZ MONTERO (FIRMA)
Firmado digitalmente por MIGUEL PEREZ MONTERO (FIRMA)
Fecha: 2022.08.12 08:31:49 -06'00'

M. Sc. Miguel Pérez Montero
Tutor

ALVARO CORDERO PEÑA (FIRMA)
Firmado digitalmente por ALVARO CORDERO PEÑA (FIRMA)
Fecha: 2022.08.12 10:22:19 -06'00'

MAP. Álvaro Cordero Peña
Lector 1

IGNACIO TREJOS ZELAYA (FIRMA)
Firmado digitalmente por IGNACIO TREJOS ZELAYA (FIRMA)
Fecha: 2022.08.18 19:42:12 -06'00'

M. Sc. Ignacio Trejos Zelaya
Lector 2

San José, Costa Rica, 11 de agosto de 2022

Firmada digitalmente, de conformidad con la Ley de Certificados, Firmas Digitales y Documentos Electrónicos N° 8454, destacando el artículo 9°-

Tabla de Contenido

Resumen Ejecutivo

Antecedentes: Los procesos de desarrollo de software en muchas organizaciones todavía no logran incorporar las prácticas recomendadas para conseguir que el producto final sea no solo funcional sino también razonablemente seguro.

Objetivo: Analizar el caso de un producto de software en desarrollo para determinar qué métodos de ingeniería de software seguro le pueden aplicar y, a partir de estos, elaborar una propuesta de mejoras, tanto en la seguridad de su diseño como en su proceso de desarrollo.

Método: Revisión documental y sesiones de trabajo con involucrados en el desarrollo del producto; se aplican técnicas de análisis de amenazas y de superficie de ataque para identificar amenazas, y patrones de seguridad para evaluar posibles mejoras al diseño.

Resultados: Se identifica preferencia por técnicas de desarrollo seguro compatibles con procesos ágiles; la superficie de ataque es amplia ya que no se contiene el flujo de datos confidenciales a través del sistema; se proponen medidas de segregación de datos para reducir la superficie de ataque.

Conclusiones: La compatibilidad de las técnicas con procesos ágiles facilita su adopción; también, un programa de seguridad en la organización es un excelente fundamento para el desarrollo de software seguro, pero no es suficiente si no se acompaña con adecuada capacitación en las prácticas de seguridad que se pueden incorporar al proceso de desarrollo.

Palabras Clave: desarrollo seguro de software, procesos de desarrollo ágil, modelado de amenazas, seguridad multinivel, scrum seguro, HIPAA.

Keywords: *secure software development, agile processes, threat modeling, multilevel security, secure scrum, HIPAA.*

Abstract

Background: Software development processes in many organizations still fail to incorporate best practices to make the end product not only functional but also reasonably secure.

Objective: To analyze the case of a software product in development to determine what methods of secure software engineering can be applied and, from these, to develop a proposal for improvements, both in the security of its design and in its development process.

Method: Document review and working sessions with stakeholders in product development; apply threat and attack surface analysis techniques to identify threats, and security patterns to evaluate possible improvements to the design.

Results: Preference for secure development techniques compatible with agile processes is identified; the attack surface is wide as the flow of sensitive data through the system is not contained; data segregation measures are proposed to reduce the attack surface.

Conclusions: The compatibility of techniques with agile processes facilitates their adoption; Also, a security program in the organization is an excellent foundation for the development of secure software, but it is not enough if it is not accompanied by adequate training in security practices that can be incorporated into the development process.

Keywords: *secure software development, agile development processes, threat modeling, multi-level security, secure scrum, HIPAA.*

Introducción

Generalidades

El presente trabajo analiza el caso de un producto de software (en adelante, El Producto) que, si bien ya ha sido liberado con éxito al mercado, también continúa desarrollándose y mejorando. Este producto consiste de una aplicación web que se comercializa como un servicio (en inglés, *software as a service*), así como una serie de procesos de ingesta y enriquecimiento de información, la cual es luego servida a través de la aplicación web.

Antecedentes del Problema

Debido a que El Producto está dirigido al sector de la salud en los Estados Unidos de América, la cual se encuentra sujeta a importantes regulaciones para la protección de la privacidad y el manejo de información personal y de salud, la organización que lo desarrolla (en adelante, La Organización) naturalmente ha efectuado importantes esfuerzos para implementar un programa institucional de seguridad de la información: se han desarrollado diversas políticas de protección y manejo de la información, de uso de recursos tecnológicos, de procesos de respuesta a incidentes y de desarrollo y adquisición de sistemas de información, todas acompañadas de acciones concretas en la organización a fin de implementar dichas políticas.

Asimismo, en el proceso de desarrollo y operación de El Producto se han incorporado varias de las mejores prácticas de la industria para prevenir vulnerabilidades: capacitación de los desarrolladores en codificación segura para aplicaciones web, realización de pruebas de penetración, implementación de controles de acceso, análisis estático del código aplicado conforme se hacen cambios a este, entre otros.

Definición y Descripción del Problema

Si bien los resultados de todos estos esfuerzos para prevenir defectos de seguridad en El Producto, según evaluaciones internas, son positivos en términos del número y severidad de vulnerabilidades que las pruebas de seguridad revelan, también se han identificado posibilidades de mejora:

1. Se han detectado algunas decisiones de diseño que pueden generar vulnerabilidades y riesgos en el uso o la operación de El Producto, pero no hay claridad de si existen otras vulnerabilidades que no hayan sido identificadas, ni de cómo desarrollar este análisis de manera más sistemática y confiable.
2. Con frecuencia no hay confianza entre los desarrolladores, ni entre los encargados del proyecto, sobre cuáles pueden ser los requerimientos de seguridad que se deben considerar y honrar, ni de cómo estos, una vez identificados, van a traducirse en decisiones concretas de diseño.

Se desea, pues, efectuar una evaluación inicial de cuál es la situación actual de El Producto para determinar si existen, principalmente en su diseño, vulnerabilidades y riesgos que no han sido previamente identificados. A su vez, se espera que esta primera evaluación sirva de base para identificar maneras en que se puedan detectar y prevenir a futuro, idealmente desde las etapas tempranas del proceso de desarrollo, decisiones de diseño que puedan introducir nuevas vulnerabilidades y riesgos en El Producto.

Justificación

Para las organizaciones y profesionales encargados del desarrollo de software, la seguridad de este es una gran responsabilidad, especialmente cuando se involucra el manejo de información personal y de salud, pues es de muy alto valor, la reparación de los daños a raíz de

su filtración no autorizada es particularmente costosa, y su mal uso incluso puede poner en riesgo la vida de los pacientes (Seh et al., 2020).

Es también un reto mayúsculo ya que, contrario a lo que sucede con el manejo de requerimientos funcionales y algunos no funcionales como los de rendimiento o usabilidad, la capacitación en el manejo de requerimientos de seguridad ha sido mucho menos difundida y accesible entre los profesionales de desarrollo de software: (Zhu et al., 2013) recalca que relativamente pocos de los programas formativos en ciencias de la computación, ingeniería de software y tecnologías de la información educan sobre desarrollo seguro de software, y (Oyetoyan et al., 2016) reporta en los casos estudiados que, efectivamente, muchos de los desarrolladores dicen no contar con experiencia en relación con la seguridad del software, especialmente aquellos con menos años de experiencia profesional.

Incluso en los casos en que se provee entrenamiento sobre seguridad del software a los desarrolladores, este puede no ser suficiente o efectivo para lograr que el software desarrollado y desplegado tenga una seguridad adecuada. Por ejemplo, muchas organizaciones han visto *OWASP Top 10* como el estándar con el cual los desarrolladores deben ser entrenados; sin embargo, debe notarse que este es principalmente un documento para generar conciencia sobre riesgos de seguridad más comunes, y que se considera apenas el mínimo necesario y punto de partida para una verdadera capacitación de seguridad (OWASP Top 10 team, 2021).

Más aún, los principios del desarrollo de software seguro apuntan a que comenzar a considerar la seguridad al momento de codificar no es suficiente, sino que es necesario tener presente la seguridad desde las etapas más tempranas del ciclo de vida del desarrollo, incluyendo el diseño y la recolección y análisis de sus requisitos.

Este trabajo pretende explorar algunos de los métodos y técnicas asociados al desarrollo seguro de software, principalmente los aplicables durante las fases de análisis de requisitos y diseño, y mostrar cómo pueden ser aplicados a fin de identificar y prevenir vulnerabilidades en el diseño de software mediante el estudio del caso de El Producto.

Viabilidad

En esta sección se argumenta y justifica la viabilidad técnica, operativa y económica del trabajo que se plantea en el presente documento.

Punto de Vista Técnico

El autor del presente trabajo cuenta con 20 años de experiencia profesional en desarrollo e ingeniería de software, ha estado involucrado en casi todas las actividades relacionadas con el ciclo de desarrollo del software, tales como la recolección y análisis de requisitos, diseño, codificación, pruebas, despliegue, mantenimiento y soporte, donde casi la mitad de esta experiencia es con procesos de desarrollo ágil (principalmente Scrum).

El autor también ha cursado el programa de maestría en ciberseguridad, el presente trabajo es la culminación de este. A lo largo del programa, se han estudiado temas fundamentales para este trabajo, tales como los sistemas de gestión de la seguridad de la información, la gestión del riesgo de seguridad, la seguridad de las aplicaciones y bases de datos, y el análisis y detección de vulnerabilidades. El foco de este trabajo es el tema del desarrollo de software seguro, será necesario complementar la formación del programa con una revisión de literatura para asegurar una comprensión profunda y actualizada, la cual será llevada a cabo como parte integral del trabajo.

Punto de Vista Operativo

El autor también goza de una posición privilegiada para llevar a cabo el proyecto, pues ha estado involucrado de manera directa y activa con el desarrollo de El Producto por poco más de dos años, lo cual le ha permitido familiarizarse profundamente con la mayor parte de sus aspectos técnicos y operativos, los artefactos, el código fuente y la documentación disponible sobre este. Asimismo, se mantiene ya relación con la mayoría de los involucrados e interesados con El Producto de manera regular, por lo que no se espera que se dificulte el gestionar y llevar a cabo las entrevistas y sesiones que se requieran para el presente proyecto, especialmente porque estas también pueden formularse como parte del proceso de desarrollo y mantenimiento usual de El Producto; en todo caso, se espera que el componente que requerirá mayor tiempo y esfuerzo sea el trabajo de análisis y síntesis documental, que puede ser llevado a cabo por el autor por cuenta propia.

Punto de Vista Económico

El grueso del costo del presente proyecto consiste en el costo del tiempo dedicado por parte del investigador para llevarlo a cabo. Se estima que el proyecto requerirá de 12 horas de trabajo por semana, por un total de 15 semanas. La tarifa estimada para un profesional calificado, de un nivel al menos comparable al de un Líder de Equipo de Desarrollo de Software, es de $40/hora.

Además, se prevé incurrir en algunos otros gastos, tales como la adquisición de material bibliográfico y suscripción de servicios automatizados de traducción (ya que se requerirá traducir material desde y hacia el idioma inglés). Para los servicios de traducción, se puede tomar como referencia las tarifas publicadas en el sitio https://reverso.net. Todos estos gastos corren por cuenta del autor del trabajo; la Tabla 1 muestra el desgloce de estos.

Tabla 1

Estimación del Costo del Proyecto

Rubro	Costo
180 horas de trabajo del investigador, tarifa de $40/hora	$7,200
Servicios de traducción en línea, 50,000 palabras (aprox. 120 páginas)	$120
Compra de material bibliográfico complementario	$200
Total	$7,520

Nota: Confección propia; precios de servicio de traducción tomados de https://documents.reverso.net/Pricing.aspx?tab=onetime&lang=en

Objetivos

Se ha usado la taxonomía original de Bloom de 1956, para mostrar niveles jerárquicos en la producción del conocimiento y usar como punto de partida un estándar de amplia aceptación en la academia.

Objetivo General

Elaborar una propuesta de mejoras de seguridad en un caso de desarrollo de aplicaciones de software.

Objetivos Específicos

Describir técnicas, actividades, herramientas, métodos y mejores prácticas existentes tanto para el análisis de requerimientos de seguridad y cumplimiento, como para el diseño seguro de software.

1. Comprender cuáles de estas técnicas, actividades, herramientas, métodos y mejores prácticas pueden ser relevantes y pertinentes en el contexto del desarrollo de la aplicación bajo estudio.
2. Aplicar las técnicas de análisis de requerimientos y diseño seguro seleccionadas.
3. Comparar el diseño seguro resultante con el diseño actual de la aplicación bajo estudio.

Alcances y Limitaciones

En esta sección se presentan los alcances y limitaciones del proyecto.

Alcances

Como parte de este trabajo, se elabora lo siguiente:

1. Un reporte sumario de las técnicas, actividades, herramientas, métodos y mejores prácticas existentes, tanto para el análisis de requerimientos de seguridad y cumplimiento, como para el diseño seguro de software, para ser sometido a consideración por parte del líder del equipo a cargo de la aplicación y que se discuta la selección de aquellas que se desee emplear.
2. Un plan de trabajo para la organización de actividades de análisis de requerimientos de seguridad y cumplimiento, y de diseño seguro para El Producto.
3. Un reporte de requerimientos de seguridad y cumplimiento para El Producto.
4. Un reporte de propuesta de diseño de alto nivel para El Producto, el cual incorpore los controles y consideraciones necesarias que permitan satisfacer los requerimientos de seguridad y cumplimiento previamente identificados.

5. Un reporte comparativo del diseño actual de El Producto con el propuesto, que identifique brechas o vacíos entre estos a fin de que pueda evaluarse cubrirlos o subsanarlos.

Limitaciones

En contraparte con lo indicado en el segmento anterior, se aclaran las limitaciones que se han tenido con este trabajo:

1. Dado el alcance y funciones que comprende El Producto, no se han podido abarcar la totalidad de los requerimientos y/o consideraciones de diseño que atañen a esta. El subconjunto de requerimientos a cubrir se seleccionó con base en el nivel del riesgo asociado y su valor relativo estimado en conjunto con el líder del equipo a cargo de El Producto.
2. No se contempló llevar a cabo ningún esfuerzo por implementar alguna de las mejoras que se proponen.

Marco de Referencia Organizacional y Socioeconómico

La Organización que desarrolla El Producto y en la cual se lleva a cabo el presente trabajo es una empresa que sirve a empresas y organizaciones del sector de seguros de salud de los Estados Unidos de América; está basada en ese mismo país y cuenta con más de dos mil empleados.

Tipo de Negocio y Mercado Meta

El mercado meta para la aplicación de estas buenas prácticas son las empresas que desarrollan software y desean incorporar seguridad en sus productos desde el diseño mismo de las aplicaciones.

Para contextualizar el presente trabajo se describe a continuación el mercado al cual se orienta La Organización donde se ha desarrollado esta investigación.

Este mercado está compuesto por las empresas que administran planes de aseguramiento por gastos médicos en los EE. UU., en diversos programas de salud como el *Medicare* y *Medicaid*; estas organizaciones son las que pagan (los "pagadores") por los servicios de salud que los proveedores de estos servicios brindan a los asegurados.

Los servicios que La Organización brinda a sus clientes son variados, pero todos ellos van orientados a ayudarles a mejorar la calidad de la gestión y eficiencia de la operación de los planes de aseguramiento, pues este mercado meta enfrenta retos importantes como lo son un correcto y preciso manejo de la información que recibe de los proveedores de servicios de salud y el cumplimiento de los requerimientos de intercambio de información con las instituciones gubernamentales que las regulan, de los cuales depende no solo la rentabilidad de la operación del negocio, sino también su capacidad de seguir operando sin incurrir en faltas que pudieran representar sanciones contempladas en las leyes y regulaciones que les aplican.

Particularmente relevante para este trabajo es el servicio que presta La Organización a sus clientes de recibir y procesar la información que manejan de sus asegurados y que reciben también de los proveedores de servicios de salud, a fin de analizarla para: a) verificar su precisión y corrección, b) construir indicadores para medir la eficiencia con la que el programa se está desempeñando y c) identificar oportunidades para mejorar esta eficiencia y aprovechar los incentivos que ofrecen los programas estatales y federales para mejorar el acceso de la población a los servicios de salud y la reducción del costo de estos servicios mediante la atención preventiva.

Este servicio de analítica de los datos del cliente es el que se provee a través de El Producto y requiere la ingesta de un gran volumen de información sensible relacionada con la salud de las personas, formalmente protegida por regulaciones como HIPAA y, por consiguiente, La Organización debe mantener programas y controles robustos en el desarrollo y operación de los sistemas que procesan esta información.

Políticas Institucionales

La Organización ha desarrollado una serie de políticas y procedimientos de seguridad de la información a fin de asegurar el cumplimiento de las regulaciones que le son aplicables por su giro de negocio.

Entre las que pueden destacarse por su relevancia para el presente trabajo están:

- Política del Programa de Protección de la Información, y sus procedimientos
- Política de Gestión de Riesgos, y sus procedimientos
- Política de Protección de Datos y Privacidad, y sus procedimientos.

Estado de la Cuestión

A fin de comprender el estado de la cuestión en relación con el problema que se aborda en el presente trabajo, se ha optado por emplear el método de revisión sistemática de literatura para identificar estudios secundarios recientes que ya han tratado de establecer lo mismo con bastante amplitud y propiedad. Esta revisión se documenta y desarrolla en el Apéndice Revisión Sistemática de Estudios Secundarios en Desarrollo Seguro de Software y los estudios seleccionados a partir de esta se detallan a continuación, agrupados por el tema en el cual se enfocan.

Ingeniería de Software Seguro

Khan et al. (2021) exploran los diversos abordajes que existen para el desarrollo seguro de software y hacen análisis FODA de cada uno de estos; asimismo, identifican una serie de métodos de ingeniería de software seguro y proponen una taxonomía para clasificarlos.

También Khan et al. (2022) investigan en la literatura cuáles son los riesgos de seguridad y mejores prácticas que pueden ayudar a las organizaciones a gestionar la seguridad del software en cada etapa de su ciclo de desarrollo. A partir de los 121 estudios seleccionados, identifican en total 156 riesgos de seguridad y más de 400 prácticas, todos agrupados por fase del ciclo de desarrollo.

Métodos Ágiles y Seguridad del Software

En relación con Moneta (2018), este expone una revisión sistemática de la literatura para investigar sobre soluciones a los problemas que los métodos ágiles normalmente enfrentan al tratar de incorporar los marcos de trabajo de seguridad, los cuales en su vasta mayoría fueron desarrollados para procesos de desarrollo en cascada. El autor divide las diferentes soluciones identificadas en cuatro grupos: mejorar el factor humano (adecuado entrenamiento de los ingenieros y otros involucrados en el proyecto), hibridar técnicas de seguridad de procesos en cascada para adaptarlas a los procesos ágiles, emplear herramientas de medición para llevar el pulso del nivel de apercibimiento de la seguridad en el proyecto y, finalmente, adición de nuevos artefactos al proceso tales como un "listado de trabajo remanente de seguridad" e "historias de usuario malvado" (en inglés, "*security backlog*" y "*evil user stories*").

En un trabajo más reciente y con un alcance más amplio, Hron y Obwegeser (2022) exploran de qué maneras se ha estudiado en la literatura el hacer cambios y adaptaciones de Scrum y qué motivaciones ha habido detrás de ellos. Los autores identifican nueve motivos

comunes por los cuales se proponen o exploran modificaciones a Scrum, donde la seguridad es uno de ellos, y además agrupan en siete categorías las estrategias o formas en que se modifica la metodología. La lista de estudios seleccionados tiene mucho en común con la de Moneta (2018), aunque la de Hron y Obwegeser (2022) por supuesto incluye algunos estudios que son más recientes; sin embargo, dado que el enfoque principal este estudio no es la seguridad, los autores no se detienen con tanto detalle en analizar y contrastar las diferentes propuestas como lo hace Moneta (2018).

Por otro lado, Behutiye et al. (2020) elabora un mapeo sistemático para determinar el estado de la cuestión en relación con el manejo de los requerimientos de calidad (en inglés, "quality requirements") en las metodologías ágiles y de desarrollo rápido de software. Entre los resultados, encuentran que los requerimientos de seguridad y de rendimiento son los que más se reportan en la literatura; también identifican varias estrategias de gestión de requerimientos (prácticas, métodos, modelos, marcos de trabajo, herramientas, guías y recomendaciones). Adicionalmente, identifican y categorizan varios retos y dificultades que se enfrentan en esta área, destacando como los principales las limitaciones de los métodos de desarrollo ágil, la gestión de los requerimientos de calidad, restricciones debidas a los cortos ciclos de iteración, limitaciones en cuanto a las pruebas relacionadas a estos requerimientos, y el descuido de estos requerimientos.

Moyón et al. (2020) investigan sobre la situación del desarrollo ágil y su madurez para cumplir con requerimientos de seguridad y de regulaciones y estándares que puedan regir diversas industrias y actividades. Los autores encuentran que las publicaciones en este campo son escasas y las que hay son mayoritariamente discusiones teóricas sobre cómo se podrían integrar los requerimientos de normas y estándares al desarrollo ágil de software. El estudio

principalmente revela la necesidad de explorar no solo la integración de los requerimientos de seguridad de los estándares a los métodos ágiles, sino también cómo valorar el cumplimiento.

Ingeniería de Requerimientos de Seguridad

En el caso de Anwar Mohammad et al. (2019), identifican 20 abordajes para la Ingeniería de Requerimientos de Seguridad (*Security Requirements Engineering*, o SRE) y los comparan sobre la base de diferentes parámetros técnicos como "rendimiento en la fase de requerimientos", "facilidad de uso respecto del tamaño y complejidad del proyecto", "notación usada", entre otros. El análisis comparativo que desarrollan puede ser útil para evaluar cuáles pueden ser las más apropiadas dadas las necesidades de un proyecto dado.

Diseño de Software Seguro

(Bafandeh Mayvan et al., 2017) identifican tendencias recientes (a la fecha del estudio) en temas relacionados con patrones de diseño y proponen una clasificación para estos temas en el campo. En su análisis, encuentran que el tema de seguridad ha cobrado gran importancia en relación con los patrones de diseño.

Por un lado, términos como "Patrones de Seguridad" y otros asociados han resultado ser los que aparecen con mayor frecuencia en el conjunto de estudios seleccionados. Desde el punto de vista de desarrollo de patrones, este resultado sugiere que el tema de seguridad es uno de los más han ganado interés entre los investigadores para introducir un nuevo patrón o lenguaje de patrones; por otro lado, tanto la seguridad como la arquitectura son las dos áreas en las que mayormente se ha estudiado la aplicación de los patrones de diseño.

Los patrones de seguridad son medios para encapsular y comunicar soluciones probadas de seguridad. (Jafari y Rasoolzadegan, 2020) exploran los esfuerzos de investigación en este tema y presentan un análisis sobre aspectos como notaciones utilizadas, criterios de clasificación,

técnicas de evaluación y entornos en los que se usan. Encuentran que el tema es un campo de investigación activo y creciente, donde el desarrollo de nuevos patrones corresponde a nuevos paradigmas de ingeniería (tales como la computación en la nube) que requieren nuevas soluciones de seguridad, y que su mayor uso es un signo de madurez en el campo: los patrones de seguridad son cada vez más usados para desarrollar nuevas metodologías de desarrollo, o mejorar los abordajes existentes.

En cuanto a Washizaki et al. (2021), ellos desarrollan una taxonomía para clasificar la investigación sobre patrones de seguridad, junto con un sondeo cuyos resultados los autores esperan que puedan mejorar la comunicación entre los practicantes y los investigadores y mejorar la efectividad de los patrones de seguridad.

Entre los resultados del estudio, los autores destacan que la mayoría de los estudios se enfocan en patrones de seguridad, donde se identifican más de 230 de ellos, aunque esto no sucede en lo que respecta a los patrones de ataque. En cuanto a los estudios que reportan o consideran los métodos o procesos con los que se pueden aplicar los patrones, estos son en su mayoría métodos guiados por modelos (en inglés, "model-driven"), y la notación más utilizada es UML. En relación con cómo se impacta la calidad de la seguridad con la aplicación de los patrones, menos de un tercio de los estudios hacen alusión a vulnerabilidades o amenazas que abordan, y solo unos pocos adoptan medidas de seguridad para evaluar los patrones.

Seguridad Aplicable a Tipos Específicos de Software

Si se parte del supuesto de que la necesidad de proteger los APIs web de actores maliciosos requiere que estos se hagan seguros desde su diseño, y que para que este diseño se logre de manera efectiva es necesario conocer las vulnerabilidades y amenazas a las que comúnmente se exponen estos APIs, así como los mecanismos existentes para defenderse de

estos, Díaz-Rojas et al. (2021) llevan a cabo un estudio de mapeo sistemático para recolectar esta información.

Como resultado de esto, los autores encuentran 66 amenazas descritas en la literatura científica y observan que las más reportadas están relacionadas con la suplantación/falsificación (en inglés, "spoofing") y la adulteración (en inglés, "tampering"), principalmente relacionadas con el tráfico de la red con el que la API interactúa. En contraste, las amenazas menos reportadas están relacionadas con el repudio. Los autores también identificaron 21 técnicas, 11 patrones y 34 métodos que pueden ser empleados a nivel de diseño para detectar, resistir, reaccionar y recuperarse de tales amenazas.

Marco Conceptual

El Desarrollo de Software Seguro

La mayoría de las metodologías tradicionales de desarrollo de software no toman en consideración cuestiones del riesgo asociado con los activos de información implicados en los sistemas que se desarrollan, y típicamente la seguridad en estos se agrega como una ocurrencia tardía, lo cual con frecuencia resulta en una implementación inadecuada de controles de seguridad (Futcher y Solms, 2012).

A fin de combatir esta problemática, los ingenieros necesitan aprender a tener bien en mente la seguridad cuando se recolectan y documentan los requerimientos, en el momento que se concibe y documenta el diseño, cuando se desarrolla el software, en el tiempo que se prueba y cuando se despliega; es decir, a través de la totalidad de su ciclo de vida (Pothamsetty, 2005). A este proceso de desarrollo de software, en el cual las consideraciones de seguridad están presentes de manera integral en todas sus etapas de principio a fin, es lo que se le conoce como *Desarrollo de Software Seguro*.

Desarrollo de Software Ágil y Seguridad

Muchos de los métodos, procesos y estándares de desarrollo seguro mejor conocidos parten del proceso de desarrollo de software tradicional lineal o en cascada, lo cual plantea problemas y retos particulares para aplicarlos en contextos donde el desarrollo del software se hace mediante procesos ágiles (Maier et al., 2017, p. 1).

La literatura revisada sugiere que diversos desafíos en los procesos y métodos tradicionales de desarrollo ágil para abordar las preocupaciones de seguridad han impulsado la propuesta y el estudio de soluciones. Moneta (2018) las agrupa de acuerdo con el área y el alcance del problema que están abordando:

- aquellas que tratan de abordar el factor humano como la preocupación principal;
- aquellas que procuran adaptar los marcos y procesos convencionales a la metodología ágil, creando procesos híbridos;
- aquellas que más bien parten de los principios, ceremonias y artefactos propios de la metodología ágil para introducir nuevos artefactos y prácticas que procuran atender las consideraciones de seguridad.

Soluciones que Abordan el Factor Humano. Moneta (2018) reporta encontrar que hay consenso en las investigaciones de que la falta de capacitación y comprensión de las cuestiones de seguridad se consideró como la causa raíz de las vulnerabilidades de seguridad en el software que se está desarrollando, pero que hay cierto debate sobre cómo abordar este problema.

El principal enfoque discutido entre varios investigadores fue proporcionar una formación adecuada a los desarrolladores de software y otros interesados en el proyecto, pero se destacan varios puntos en relación con este:

1. Se debe prestar atención para que las *presiones de la organización* no impidan la aplicación de los conocimientos y prácticas de seguridad (por ejemplo, atención a la seguridad en conflicto con las limitaciones de tiempo y la presión para liberar el producto, falta de compromiso y comprensión adecuada del valor de la seguridad por parte de los gerentes).
2. Además de la capacitación en pruebas de seguridad y codificación segura, que son más frecuentes, otro tema en el que se considera que la capacitación es muy necesaria es el *diseño seguro*.

3. Un *entorno eficaz que fomente la difusión de los conocimientos* y la reproducción de los éxitos en materia de seguridad es importante para maximizar los beneficios de las actividades de capacitación y los beneficios de la experiencia.
4. Además, el entrenamiento tiende a tener el mayor impacto cuando se entrega en el momento en que se necesita, en lugar de tener una gran cantidad de información arrojada a las personas, pero que puede no tener un uso inmediato.

Un enfoque diferente de la problemática de la experiencia de los desarrolladores que se sugiere en los estudios es la adición de una o más figuras profesionales al equipo de desarrollo; por ejemplo:

1. introducir un *Security Master* (del inglés, jefe o maestro de seguridad), análogo al tradicional *Scrum Master*, que esté a cargo del *security backlog* (del inglés, listado de trabajo remanente de seguridad), el cual también es análogo al *backlog* convencional de *Scrum*;
2. o, alternativamente, un *ingeniero especialista* que debe considerar la seguridad en cada paso del desarrollo.
3. Una propuesta aún más extrema consiste en nombrar a varios profesionales, incluyendo un *Gerente de Seguridad* (en inglés, *Security Manager*, a cargo de las características tradicionales relacionadas con la seguridad, como la certificación ISO y los aspectos legales del desarrollo), un *Arquitecto de Seguridad* (responsable de transformar el documento que presenta los requisitos generales no funcionales en una descripción más técnica), un *Security Master* (responsable de las características de seguridad durante el desarrollo, así como el análisis de riesgo que estima las consecuencias negativas de un ataque y su probabilidad de

ocurrir) y un *Encargado de Pruebas de Penetración* (parte del equipo de control de calidad, que ejecutaría pruebas exploratorias y verificaría que el sistema podría resistir un uso malicioso inadecuado), todo para integrarse con el equipo de desarrollo ágil.

Si bien se percibió que la adición de expertos aumentó sustancialmente la seguridad general del producto, no estuvo exenta de preocupaciones:

1. Las funciones especializadas conllevan costes adicionales para el proyecto.
2. Tener *especialistas en un equipo va en contra del principio ágil de los equipos multifuncionales* (en inglés, *cross-functional teams*), que establece que "los equipos de desarrollo no contienen subequipos dedicados a dominios particulares como las pruebas o el análisis de negocios".

Soluciones que Abordan el Proceso Ágil Mediante la Hibridación. Una realidad de los procesos y marcos de seguridad es que, durante muchos años, se desarrollaron para el modelo de cascada más tradicional como base. Por lo tanto, con el fin de llevar las prácticas de seguridad a los procesos de desarrollo ágil, un enfoque muy común ha sido portar prácticas y modelos desde los marcos existentes y tratar de adaptarlos a los ágiles. La Tabla 2 muestra una lista de varios de estos esfuerzos de adaptación, con algunas breves notas sobre ventajas y desventajas para cada una.

Tabla 2

Soluciones de Seguridad Híbrida para Procesos Ágiles

Solución	Pros y contras
Aplicar actividades y guías del MS SDL para métodos ágiles	**Pros:** Chóliz et al. (2015) reportan resultados positivos al emplear varias de las actividades del proceso SDL de Microsoft en su caso de estudio. **Contras**: el verdadero proceso SDL para Agile no se siguió realmente, sino que se requirió tener un equipo de seguridad dedicado y separado para llevar a cabo actividades centradas en la seguridad (especificación de reqs de seguridad, evaluación de amenazas, pruebas de seguridad); por lo tanto, es una desviación significativa de los principios ágiles establecidos.
Aplicar actividades seleccionadas de múltiples marcos (MS SDL, *Touchpoints*, *Common Criteria*, CLASP)	**Pros:** Ayalew et al. (2013) y Baca y Carlsson (2011) analizan e identifican de forma independiente las actividades de varios marcos que tienen más posibilidades de ser compatibles con los procesos ágiles. En un estudio posterior, Baca et al. (2015) y Boldt et al. (2017) reportan resultados positivos al aplicar un "Proceso Ágil Mejorado de Seguridad" (*SEAP*, por sus siglas en inglés), un refinamiento de su trabajo anterior. Se descubrió que el *SEAP* reduce el riesgo de seguridad no abordado, o bien este se aborda antes; dicho proceso incorpora las actividades de seguridad en el proceso ágil. Actividades propuestas para ser añadidas al proceso ágil: 1. Identificar los requisitos de seguridad 2. Capturar una matriz de roles 3. Escribir casos de abuso 4. Desarrollar gráficos de contramedidas, llevar a cabo análisis de riesgo 5. Documentar supuestos y posibles ataques 6. Hacer inspecciones de requisitos **Contras**: requiere la adición de varias funciones de seguridad al equipo y, aunque se estima que es rentable para abordar las preocupaciones de seguridad, las funciones adicionales representan un costo complementario

	del proyecto (aunque este enfoque en el costo puede no ser apropiado).
Proceso Scrum mejorado con prácticas tomadas de ISO SSE-CMM	**Pros:** Maier et al. (2017) propusieron un proceso que integra las actividades de seguridad, al tiempo que se esfuerza por mantener la agilidad; incluye un componente de evaluación de riesgos. Actividades propuestas para ser añadidas al proceso ágil: 1. Iniciar la planificación de la seguridad 2. Método ágil de análisis de riesgos 3. Identificar los requisitos de seguridad, escribir historias relacionadas con la seguridad y criterios de aceptación 4. Documentar controles de seguridad 5. Revisiones de código (con perspectiva de seguridad) **Contras**: la evaluación de la propuesta aún no es empírica, pero se basa en la encuesta de los profesionales de la industria.
Proceso ágil con enfoque en la gestión de riesgos de seguridad	**Pros:** Franqueira et al. (2011) posicionan, y Salin y Lundgren (2022) desarrollan más tarde, un marco ágil con la evaluación de riesgos integrada en el proceso; estas prácticas puede ayudar a los *Product Owners* (dueños de producto) a tomar decisiones sobre las prioridades de las tareas en cada iteración (en inglés, *sprint*). Busca reducir el nivel de experiencia o pericia necesarios para gestionar el riesgo mediante la obtención de información de catálogos públicos como el *NVD* del *NIST*, permitiendo así potencialmente que el equipo permanezca lo más autónomo posible. **Contras**: se centra solo en los riesgos, no incluye orientación sobre los requisitos de ingeniería ni diseño seguro.
Proceso ágil con métricas de riesgo de seguridad	**Pros:** Savola et al. (2012) informan sobre un piloto de un proceso con la evaluación de riesgos integrada en el desarrollo, de manera ágil; da visibilidad de las métricas y estado de riesgo a todos los interesados en el proyecto. **Contras:** La orientación sobre los requisitos de seguridad y el diseño es limitado.

Nota: Adaptado de Moneta (2018); el análisis de pros y contras es de confección propia.

Soluciones que Abordan el Proceso Ágil Mediante la Adición de Artefactos o Prácticas Inspirados en Marcos Ágiles. Algunos autores proponen y estudian mejoras a los procesos ágiles mediante variaciones de sus prácticas, artefactos y ceremonias existentes.

Bartsch (2011) lleva a cabo entrevistas con varios profesionales ágiles para que compartan sus perspectivas sobre diversos temas planteados por formulaciones más teóricas de la literatura de seguridad, y deriva algunas recomendaciones que no se encontraban previamente.

Por su parte, Ghani et al. (2014) ejecutan un análisis de la idoneidad de las prácticas de llevar un *security backlog* y designar a un *Security Master* en un proceso de desarrollo ágil, y encuentran resultados alentadores.

Barbosa y Sampaio (2015) desarrollan una guía, basada en un estudio de la literatura, con diversas prácticas que podrían añadirse a un proceso ágil, luego lo evalúan varios profesionales (no todos son especialistas en seguridad). Encuentran que, si bien los especialistas en seguridad son en general receptivos a las prácticas incluidas en la guía, los otros entrevistados tienden a descartar varios de ellos, citan, por ejemplo, preocupaciones de costo y eficiencia de tiempo que prevalecen sobre el deseo de una mejor seguridad en el software producido.

En relación con lo mencionado, Binti Arbain et al. (2014) propusieron un nuevo enfoque que crea un "Modelo de Proceso de Trazabilidad" (*TPM*, por sus siglas en inglés) que toma en consideración los requisitos no funcionales.

En la Tabla 3 se presenta un resumen de los artefactos y actividades sugeridos en la literatura, junto con algunas ventajas y desventajas para cada uno.

Tabla 3

Artefactos y Actividades de Seguridad Ágil

Artefacto o Actividad	Pros y contras
Abuse Cases (casos de abuso), *Evil User Stories* (historias de usuarios malvados)	Historias de usuario escritas como si un hacker o un usuario malicioso estuviera planeando hacer un uso inapropiado del sistema. Descrito por Barbosa y Sampaio (2015 [17, 18, 37, 38]) **Pros**: Permite identificar posibles problemas de seguridad y resaltar la necesidad de contramedidas para historias maliciosas. **Contras**: Se requiere algo de experiencia para escribirlas, así como la participación de múltiples partes interesadas; no es una técnica sistemática en su enfoque para identificar áreas con potenciales vulnerabilidades.
Nombrar un *Security Master*	Análogo al *Scrum Master*, este papel garantizará que las preocupaciones de seguridad se consideren y aborden a lo largo de la ejecución del proyecto. Proporcionará orientación mientras otros miembros del equipo ejecutan las actividades relacionadas con la seguridad. Descrito por Barbosa y Sampaio (2015 [10, 16, 17, 29, 36]), Ghani et al. (2014). **Pros**: Permite la ejecución de actividades de seguridad y proporciona liderazgo técnico para ellas; es un rol integrado con el proyecto y el equipo; a medida que mejora la capacidad del equipo para abordar los problemas de seguridad a través de la formación y la experiencia, este recurso se puede compartir con varios proyectos. **Contras**: No muchas personas podrían estar calificadas para desempeñar el papel; añade costes al proyecto; algunos consideran que este papel es necesario solo provisionalmente mientras el equipo desarrolla la conciencia y la habilidad.
Security Backlog	Un subconjunto de los elementos en el *backlog* convencional del

	proyecto, y que están marcados como tales (p. ej., por el *Security Master*) ya que requieren atención para la seguridad. Descrito por Barbosa y Sampaio (2015 [10, 16]) y por Ghani et al. (2014). **Pros**: Las preocupaciones de seguridad se anticipan y se destacan; puede documentar las actividades de seguridad relacionadas y facilitar la trazabilidad de los requisitos de seguridad. **Contras**: Se requiere de experiencia para desarrollarlo y mantenerlo (p. ej., esto hecho por el *Security Master*); establece un precedente de que otros requisitos no funcionales (p. ej., usabilidad, rendimiento) pueden merecer *backlogs* especializados.
Protection Poker (Póker de Protección)	Complementa la actividad tradicional de *Planning Poker*, con la atención hacia los posibles problemas de seguridad que una historia puede implicar. Los puntos estimados dados a una historia indican riesgo asociado en lugar de esfuerzo o complejidad. Descrito por Barbosa y Sampaio (2015 [6, 22]) **Pros**: Incorpora una forma básica de evaluación de riesgos en la planificación y permite la priorización basada en el riesgo; trae múltiples partes interesadas y perspectivas al debate; aumenta la conciencia de los problemas de seguridad en el equipo. **Contras**: Requiere que los participantes tengan algunas habilidades para ser eficaces en la identificación y evaluación de riesgos; puede requerir materiales de apoyo, tal como la información disponible en los *evil user stories*, el *security backlog*, las listas de amenazas/activos/vulnerabilidad y listas de verificación de seguridad (en inglés, *security checklists*).
Gráficos de contramedidas	Un método de análisis de riesgos que se centra en identificar las características de seguridad y priorizarlas. Descrito por Baca y Carlsson (2011 [16]). **Pros:** Permite llevar a cabo el modelado de amenazas de una manera incremental e iterativa.

	Contras: No muy evaluado en estudios de casos reales.
Plan de incidentes	Los planes de incidentes se utilizan para identificar y planificar cómo responder a las amenazas potenciales que pueden ocurrir en las liberaciones. Descrito por Barbosa y Sampaio (2015 [17, 39]) **Pros:** La planificación hace que la organización esté mejor preparada para aplicar contramedidas, por lo que la recuperación puede ocurrir de manera rápida y eficiente con un impacto negativo mitigado. **Contras:** Requiere un poco de habilidad y puede ser laborioso de desarrollar.
Investigación de vulnerabilidades	Investigar posibles vulnerabilidades asociadas con el proyecto para implementar contramedidas. Implica buscar información actualizada sobre vulnerabilidades y ataques en boletines de seguridad y bases de datos. Descrito por Barbosa y Sampaio (2015 [17, 29]). **Pros:** Permite evaluar de forma proactiva las vulnerabilidades que pueden afectar el proyecto y desarrollar contramedidas. **Contras:** Puede ser laborioso y consume mucho tiempo; debe repetirse a lo largo del tiempo.
Formulación de los requisitos de seguridad y otros requisitos no funcionales como "*definitions of done*" (del inglés, "definiciones de hecho")	A veces, los requisitos de seguridad pueden incorporarse en las "*definitions of done*". Descrito por Bartsch (2011, p. 4). **Pros:** Muy ligero y eficaz para mantener la conciencia sobre los requisitos de seguridad que deben cumplirse a medida que se lleva a cabo el trabajo; una vez que se escribe la definición, se reduce la participación del especialista en seguridad. **Contras:** Depende de tener requisitos de seguridad correctamente identificados y sintetizados de antemano; no todos los requisitos de seguridad podrían abordarse de esta manera.
Código y configuración	Los casos de prueba, las políticas de autorización y otros artefactos pueden proporcionar documentación implícita de los aspectos de

como documentación de seguridad	seguridad del sistema. Descrito por Bartsch (2011, p. 5) **Pros:** Soluciona problemas con la documentación de seguridad que se vuelve obsoleta o que pierde sincronía con la realidad. **Contras:** Requiere habilidad y esfuerzo adicional por parte del desarrollador para construir abstracciones efectivas y sostenibles que capturen preocupaciones de seguridad; hacer que la información sea accesible y transparente para los no desarrolladores puede ser un desafío.
Verificaciones de Seguridad	Se recomiendan tres controles de seguridad: análisis estático, análisis dinámico y *fuzz testing*. Descrito por Barbosa y Sampaio (2015 [17, 28, 29, 42, 43]). **Pros:** Ejercita medidas de seguridad y controles; tiene como objetivo detectar vulnerabilidades y errores en la implementación antes de la liberación. **Contras:** No muchos especialistas de pruebas están familiarizados con las pruebas de seguridad a fondo, las herramientas de análisis estático pueden ser propensas a producir falsos positivos y generar ruido.

Nota: Adaptado de Moneta (2018); el análisis de pros y contras es de confección propia.

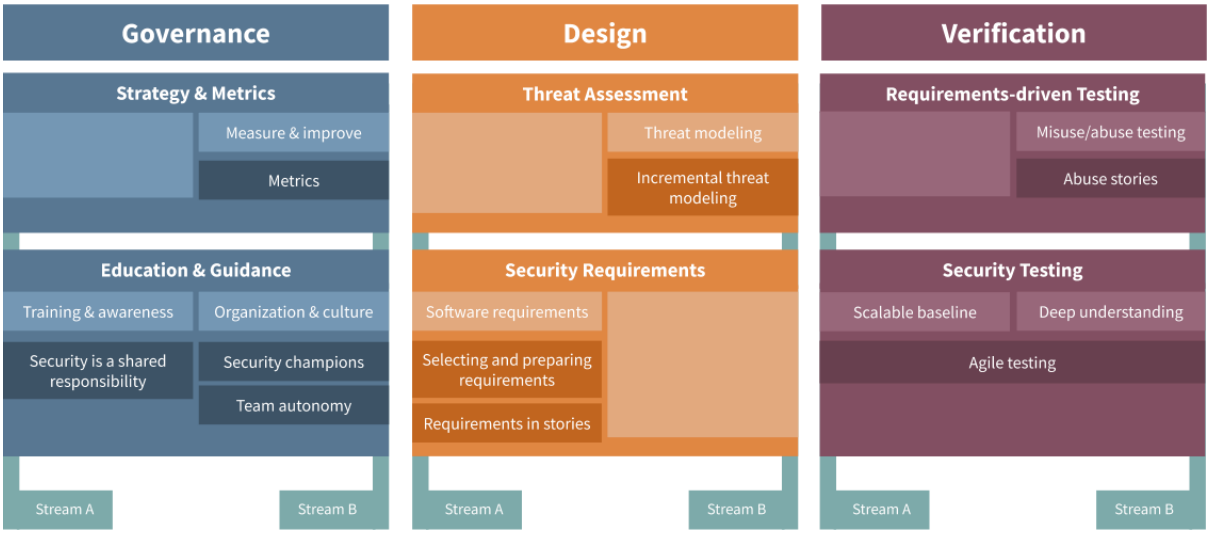
Modelos de Madurez de Desarrollo Seguro

Los modelos de madurez son marcos que ayudan a las organizaciones a evaluar, formular e implementar una estrategia de seguridad de software que se puede integrar en su ciclo de vida de desarrollo de software (SDLC) existente. Van más allá de simplemente prescribir una serie de actividades y prácticas a incorporar en el proceso de desarrollo, como lo hacen los marcos y procesos de referencia, y proponen una hoja de ruta para que las organizaciones puedan incorporar en sus procesos, progresivamente y según sus intereses y necesidades, las diferentes prácticas prescritos por los procesos y marcos de referencia.

OWASP SAMM. Adoptado por el proyecto OWASP, SAMM es un modelo de madurez que está disponible libremente para cualquier persona. Comprende 15 prácticas agrupadas en cinco funciones de negocio (Gobernanza, Diseño, Implementación, Verificación, Operaciones), y cada práctica tiene tres niveles de madurez (SAMM Community, n.d.-b). También tiene una guía específica para los procesos ágiles (SAMM Community, n.d.-a), el cual se muestra en la Figura 1.

Figura 1

Actividades de SAMM para Métodos Ágiles

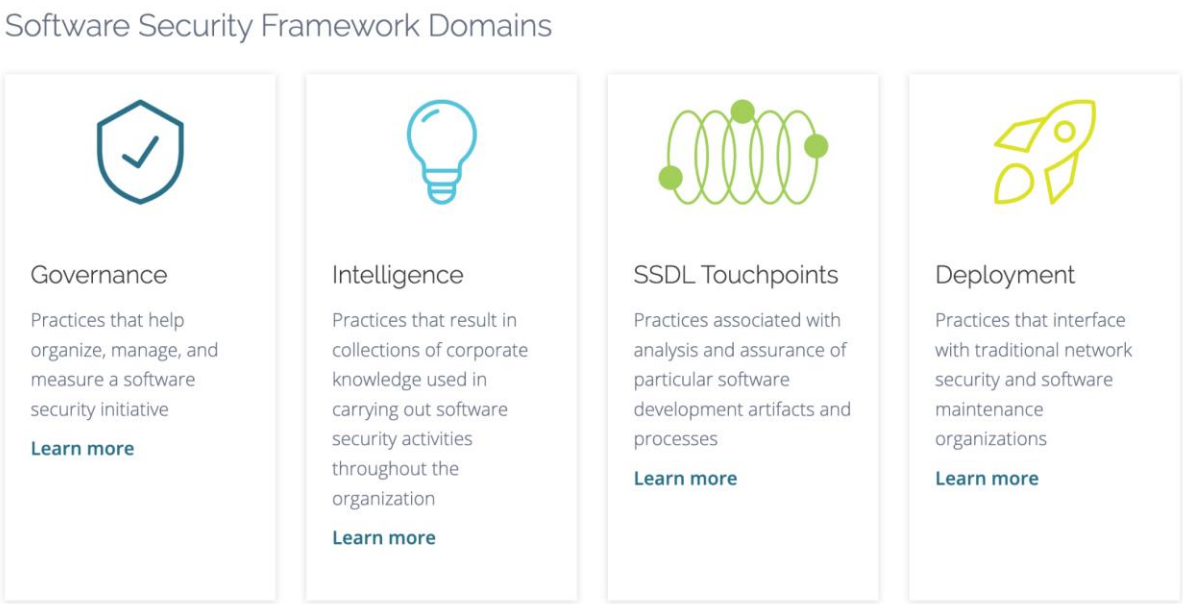


Nota: Tomado de SAMM Community (n.d.-a).

BSIMM. (BSIMM Community, n.d.) es otro modelo de madurez; este es bastante descriptivo (en contraste con el SAMM, que es prescriptivo), reportando las prácticas observadas a través de varias organizaciones que son miembros del programa. Las 12 prácticas en este caso están organizadas en cuatro dominios (Gobernanza, Inteligencia, Puntos de Contacto SSDL, Despliegue). La Figura 2 ilustra los dominios estipulados por este modelo.

Figura 2

Dominios del Marco de Seguridad de Software de BSIMM



Nota: Tomado de BSIMM Community (n.d.).

Principales riesgos para la ingeniería de requisitos y el diseño del software

Khan et al. (2022) argumentan que, a pesar de las diversas metodologías, estrategias y modelos que han sido propuestos y desarrollados para abordar la seguridad del software, sean estos ágiles o no, todavía hay retos que superar para integrar los procedimientos de seguridad en el Ciclo de Vida del Desarrollo de Software, por lo que estudiar y aprender sobre los riesgos de seguridad del software, y prácticas asociadas, permitirá diseñar mejor los procesos de desarrollo de software para que sean seguros.

Es natural que, cuando una organización intenta incorporar estándares, métodos y recomendaciones a sus procesos de desarrollo de software, muy probablemente haga ajustes y adaptaciones de los primeros, por lo que no necesariamente termina aplicando fielmente las recomendaciones originales. Sea cual fuere el caso, tener en cuenta los riesgos de seguridad que

han sido identificados como predominantes en la industria y la literatura, servirá de guía para que no se pierda de vista los objetivos más apremiantes en los esfuerzos que se ejecuten con miras a mejorar la seguridad de los resultados.

En la Fase de Ingeniería de Requisitos. Los riesgos más comúnmente asociados con esta fase se resumen en la Tabla 4.

Tabla 4

Principales Riesgos de Seguridad de Software en la Fase de Ingeniería de Requisitos

Número	Riesgos de seguridad en la fase de ingeniería de requisitos
i	Los requisitos de seguridad a menudo se descuidan o se consideran un requisito no funcional
ii	Falta de negociación y gestión de los requisitos de seguridad
iii	Falta de validación de los requisitos de seguridad
iv	Evaluación inadecuada del riesgo
v	Falta de análisis de riesgos de seguridad

Nota: Adaptado y traducido de Khan et al. (2022, Tabla 4).

En la Fase de Diseño del Software. Además, los riesgos más comúnmente asociados con la fase de diseño se enumeran en la Tabla 5.

Tabla 5

Principales Riesgos de Seguridad de Software en la Fase de Diseño

Número	Riesgos de seguridad en fase de diseño
i	Falta de desarrollo de modelos de amenazas durante la fase de diseño
ii	Falta de atención para seguir los principios de diseño de seguridad
iii	Falta de conocimientos, orientación y capacitación en materia de diseño de medidas de seguridad
iv	Documentación de seguridad de diseño inadecuada
v	Falta de creación y mantenimiento de modelos de casos de abuso y patrones de ataque
vi[a]	Inadecuada revisión del diseño de seguridad y su verificación

Nota: Adaptado y traducido de Khan et al. (2022, Tabla 5).

[a] Este riesgo se presenta con la misma frecuencia que los dos anteriores.

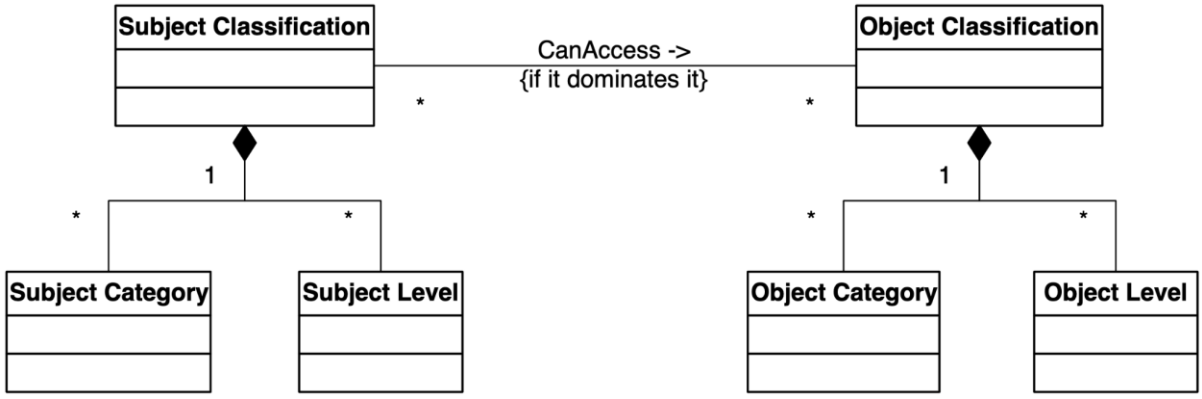
Seguridad Multinivel

Este es un patrón estructural de seguridad que trata de proveer un mecanismo para manejar el acceso en un sistema con varios niveles de seguridad Wassermann y Cheng (2003, p. 56).

Dada esta jerarquía de niveles de seguridad, a cada objeto por asegurar se asigna uno o más niveles, y lo mismo se hace con los sujetos que desean tener acceso a los objetos. Una representación de la información que se puede asociar con cada objeto y sujeto se muestra en la Figura 3.

Figura 3

Diagrama de Clases del Patrón de Seguridad Multinivel



Nota: Tomado de Wassermann y Cheng (2003, fig. 21).

Como se observa en la Figura 3, cada sujeto en el sistema tiene asociada una Clasificación de Sujeto; asimismo, cada objeto tiene asociada una Clasificación de Objeto. Estas clasificaciones consisten en uno o más niveles de seguridad, tanto del sujeto como del objeto, así como categorías (que definen particiones transversales a los niveles de seguridad). El acceso de un sujeto a un objeto dado solo se permite si la clasificación del primero "domina" a la clasificación del segundo.

Confidencialidad y el Modelo de Bell y La Padula. Para garantizar la confidencialidad de la información, se prescriben una serie de reglas que controlan y restringen el flujo de información de niveles de seguridad alto a niveles menores, y de una categoría a otra. Estas reglas fueron inicialmente desarrolladas por Bell y La Padula (1976).

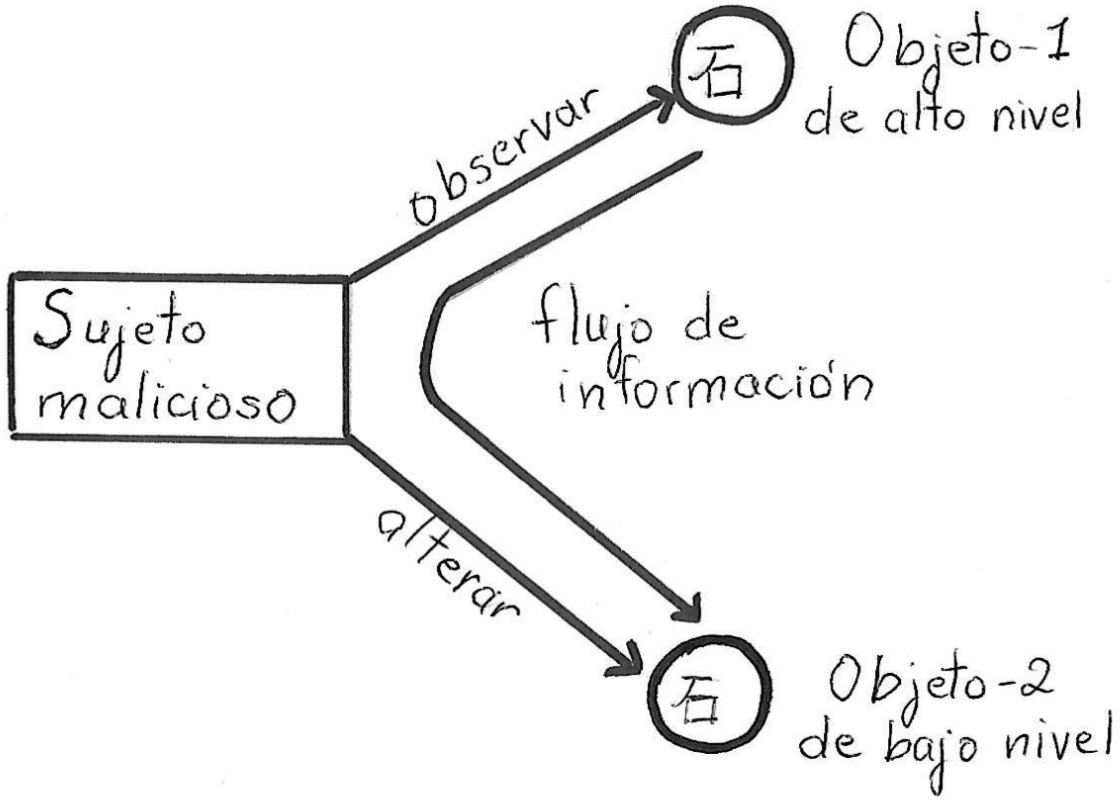
La primera regla, y la más intuitiva, es que una solicitud de acceso por parte de un sujeto a un objeto, se deben comparar los niveles de seguridad asignados a cada uno, y el acceso se concede solamente si el nivel del sujeto es igual o superior al del objeto; los sujetos con un nivel de seguridad bajo, no podrán obtener del todo acceso a ningún objeto con niveles de seguridad

superiores. Esta se conoce como la “propiedad de seguridad simple” (en inglés, *simple-security property* o también *ss-property*).

La “Propiedad-*”. Sin embargo, Bell y La Padula (1976) se dan cuenta de que, para garantizar la protección de la confidencialidad, es necesaria una regla adicional, menos obvia al principio, pero que se hace evidente al considerar la manera en que la información podría fluir de un nivel de seguridad alto hacia uno más bajo. A esta regla se le denominó “propiedad-*” o “propiedad-estrella” (en inglés, **-property*), y en la Figura 4 se muestra el flujo de información que esta propiedad procura prevenir.

Figura 4

*Flujo de Información Mostrando la Necesidad de la Propiedad-**



Nota: Tomado de Bell y La Padula (1976, fig. 4).

Nota. Un sujeto malicioso que tenga la posibilidad de observar un objeto de un nivel de seguridad alto, y que al mismo tiempo pueda actualizar la información en un nivel inferior, puede comprometer la confidencialidad del objeto escribiendo lo observado en el nivel inferior.

Esta regla exige que, si un sujeto tiene la posibilidad de actualizar la información a un nivel de seguridad determinado, no podrá observar objetos de un nivel de seguridad superior. Es una restricción fuerte, pero es necesaria cuando no es posible confiar en que el sujeto no llegará a

comprometer, de manera intencionada o no, la confidencialidad de la información a la que podría acceder.

Sujetos de Confianza. No obstante lo dicho en la sección La "Propiedad-*", el modelo de Bell y La Padula reconocen que hay razones válidas para permitir, en ciertas circunstancias, que un sujeto o un proceso no tenga que estar restringido por la propiedad-*.

Esto puede darse cuando se tiene por seguro que el sujeto o proceso en cuestión no va a revelar inapropiadamente la información del nivel de seguridad alto. También podría serlo cuando es apropiado que se actualice la información del nivel de seguridad inferior, ya sea porque se está rebajando la clasificación de la información, o porque se le está aplicando alguna redacción o transformación que sean aceptables para la nueva clasificación.

Con toda claridad, estos procesos y circunstancias deberán ser examinados y autorizados con sumo cuidado.

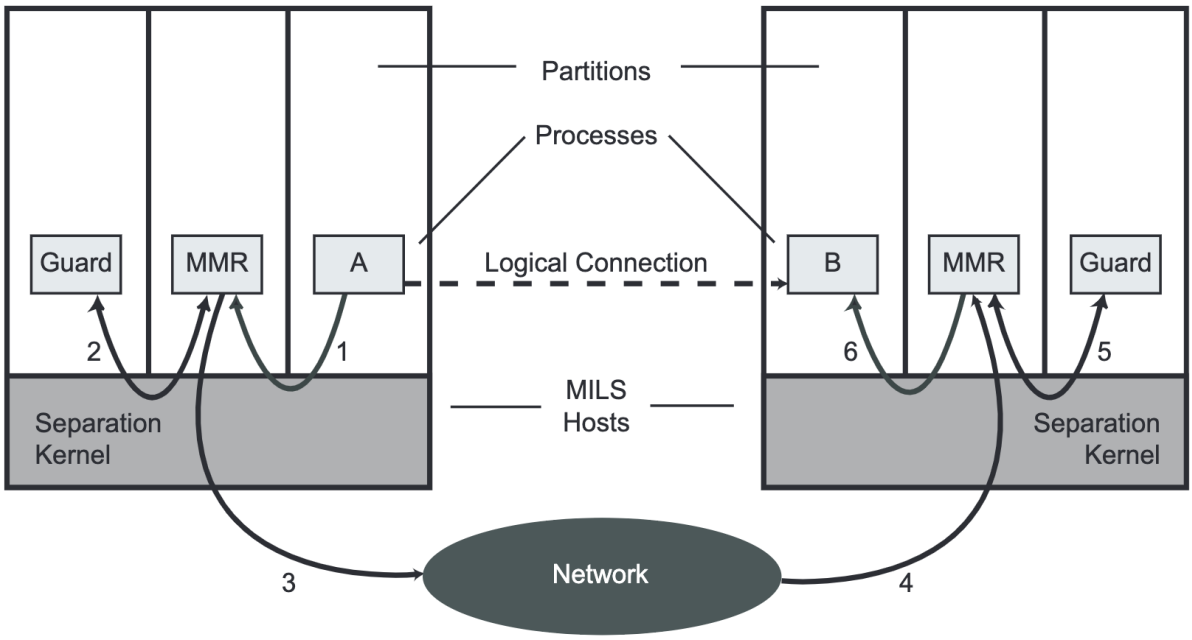
Seguridad Múltiple de Un Solo Nivel

Los sistemas de seguridad multinivel son sumamente rigurosos, y de hecho fueron concebidos para el contexto de sistemas de alta sensibilidad y confiabilidad, típicamente militares o gubernamentales, lo cual requiere incluso soporte especializado desde el sistema operativo de los equipos, este generalmente no se encuentra disponible en los ambientes convencionales.

Un enfoque alternativo en estos casos se conoce como "seguridad múltiple de un solo nivel" (en inglés, *multiple single-level security*) y una de las primeras propuestas de este enfoque fue la Arquitectura MILS (Harrison et al., 2005), y una representación de esta se muestra en la Figura 5.

Figura 5

La Arquitectura MILS



Nota: Tomado de Harrison et al. (2005, fig. 1).

Nota. Esta arquitectura se basa en un núcleo de separación, que garantiza el aislamiento de diferentes particiones en un sistema, y algunos componentes confiables de seguridad (*MMR*, *Guard*); en conjunto, permiten controlar la comunicación entre aplicaciones y procesos no confiables (A, B).

Ahora bien, incluso la arquitectura MILS, así como extensiones de esta a ambientes distribuidos (Jarmakiewicz y Podlasek, 2015; Luo y Kang, 2009), requieren de soporte especializado en el sistema operativo y/o capas intermedias (en inglés, *middleware*). Una variación menos ambiciosa y más al alcance de los ambientes convencionales consiste en manejar la información de diferentes clasificaciones en ambientes separados (Wolter, 2021).

También debe notarse que, dependiendo de la sensibilidad al riesgo y el análisis costo-beneficio, no necesariamente deben crearse ambientes completos dedicados a cada clasificación, sino que también es posible optar por el uso de mecanismos de segregación disponibles dentro de un ambiente. Por ejemplo, en diseños de bases de datos relacionales, se puede aplicar particionamiento vertical para almacenar los datos sensibles en particiones aparte, a las que se les puede aplicar controles de seguridad adicionales (Microsoft, 2022, sec. Vertical partitioning); estas particiones pueden ser esquemas dentro de una misma instancia de base de datos, o bien instancias de bases de datos separadas, pero siempre existiendo dentro del mismo ambiente.

Diversos Conceptos del Dominio

En esta sección se introducen varios conceptos clave del dominio de El Producto, tales como el de Ajuste de Riesgos, y algunos elementos de la normativa de privacidad aplicable al sector salud.

Ajuste de Riesgos en el Sector Salud, EUA

En los Estados Unidos de América, existen programas de salud en los que el Estado provee beneficios y coberturas de salud a las personas a través de aseguradores privados.

Un ejemplo de estos programas es *Medicare Advantage* (Centers for Medicare & Medicaid Services, 2021, p. 3). En este, la agencia conocida como los Centros de Servicios de Medicare y Medicaid (en inglés, *Centers for Medicare & Medicaid Services*, *CMS*), que es parte del Departamento de Salud y Servicios Humanos (en inglés, *Department of Health and Human Services*, *HHS*), paga un monto mensual a los aseguradores privados por cada uno de los beneficiarios inscritos en sus planes de aseguramiento.

En programas anteriores, el pago que se hacía por cada beneficiario se calculaba con base en un costo promedio según la zona de residencia del beneficiario. Sin embargo, se encontró que

este sistema creaba incentivos para que los aseguradores procuraran incluir en sus planes solamente personas que eran más saludables que el promedio, mientras que las personas con condiciones de salud deterioradas tendrían más problemas para inscribirse en un plan de aseguramiento. En respuesta a esta problemática, se crearon modelos de cálculo del monto a pagar por beneficiario que tomaran en cuenta características demográficas y de la condición de salud del beneficiario; a este sistema en el que lo que se paga al asegurador por cada beneficiario es un monto ajustado a las condiciones particulares de este último, se le denomina "ajuste de riesgos" (Centers for Medicare & Medicaid Services, 2021, p. 4).

La naturaleza de estos programas hace necesario que los aseguradores privados sean eficientes en el manejo de sus planes, lo cual requiere un fuerte apoyo de sus sistemas de información (o servicios dedicados para tal fin). Por ejemplo, Ciox Health (2021) argumenta que, con la información y tecnología correctas, se logra gestionar los datos de poblaciones específicas, seleccionar los miembros (beneficiarios) que deben ser monitoreados, identificar aquellos con alta probabilidad de tener serias condiciones de salud que no están documentadas y automatizar la extracción de los registros médicos (para cumplir con los procesos de documentación y reporte formales).

Debe destacarse que estos sistemas de analítica del ajuste de riesgo y de la gestión de los planes de aseguramiento, de los cuales El Producto es un ejemplo, se alimentan y procesan una gran cantidad de información sensible, protegida por regulaciones (ver La Normativa de Privacidad), por lo que resulta ineludible el deber de asegurarlos frente a las amenazas que naturalmente surgen en torno a ellos.

HIPAA

En los Estados Unidos de América se emitió, en 1996, la Ley de Responsabilidad y Portabilidad del Seguro de Salud (en inglés, *Health Insurance Portability and Accountability Act, HIPAA*). Su propósito es el de mejorar la portabilidad y continuidad de la cobertura de seguros de salud y simplificar la administración de los seguros de salud, entre otros más (Assistant Secretary for Planning and Evaluation, n.d.).

El propósito de la ley es facilitar la estandarización y el intercambio de información entre diversos actores del sector salud, y de hecho precisamente por eso, la ley exige que se recomienden y se establezcan regulaciones sobre estándares de privacidad sobre el uso e intercambio de esta información.

La Normativa de Privacidad

Cuando se emitió originalmente la ley HIPAA, esta no incluyó legislación específica en relación con regulaciones de privacidad sobre la información de salud que identificara a individuos (en inglés, *individually identifiable health information*), por lo que fue el Departamento de los Estados Unidos de Salud y Servicios Humanos (*HHS* por sus siglas en inglés) el que emitió estas regulaciones en el año 2000, y posteriormente algunas modificaciones en el 2002. A estas en conjunto, se les conoce como la "Normativa de Privacidad" (en inglés, *the Privacy Rule*) (Office for Civil Rights (OCR), 2013).

Esta normativa define el concepto de *información de salud protegida* (en inglés, *protected health information*, *PHI*), que es clave ya que las políticas de protección de datos y de privacidad la utilizan para guiar en la identificación de la información sensible que debe ser manejada con cautela.

Información de Salud Desidentificada

Otro concepto clave establecido en la Normativa de Privacidad es el de la *información de salud desidentificada,* así como los métodos formalmente reconocidos para obtenerla a partir de la información. La desidentificación es útil pues permite derivar información de salud para la cual la normativa no establece restricción alguna para su uso o divulgación (Office for Civil Rights (OCR), 2013, sec. "De–Identified Health Information").

La normativa reconoce dos métodos para derivar información desidentificada a partir de información protegida (Office for Civil Rights (OCR), 2012). El primero depende de la determinación, por parte de un experto debidamente calificado, de que con la aplicación de principios y métodos científicos y estadísticos, la información ya no permitirá identificar a ningún individuo. Naturalmente, la dificultad en este caso es que con frecuencia no se puede contar con la ayuda de este experto, y además el proceso será laborioso y costoso.

El Método del "Puerto Seguro" ("Safe Harbor"). Este segundo método está diseñado para que sea más sencillo y accesible para las organizaciones en general y consiste en requerir la remoción u ofuscamiento de un conjunto predeterminado de elementos que pudieran estar presentes en la información de salud. Una vez removidos estos elementos, si además la organización no tiene razones para creer que la información resultante puede ser usada para identificar al individuo, ya sea por sí sola o junto con alguna otra que esté disponible, la información se considera desidentificada.

Algunos de los elementos que deben ser removidos u ofuscados son los siguientes:

1. Nombres.
2. Todas las subdivisiones geográficas menores a las del estado (ciudad, condado, dirección de calle, código postal).

3. Todos los elementos de una fecha (exceptuando el año) para fechas directamente relacionadas con el individuo (fecha de nacimiento, de admisión, de salida o de deceso), así como aquellos que apunten a una edad del individuo mayor a 89 años.
4. Números de teléfono, de fax, dirección de correo electrónico, URLs.
5. Número del seguro social.
6. Números de registro médico, o de beneficiario en el plan de salud, números de cuenta.
7. Identificadores biométricos, fotografías del rostro, entre otros.

El Estándar del Mínimo Necesario y El Principio de "Necesitar Saber"

Moneypenny (2021) nota que los textos de HIPAA (la ley y directrices asociadas) no mencionan explícitamente este principio, pero que la conexión con este se encuentra en el Estándar del Mínimo Necesario; además, sostiene que posiblemente invocar el principio de "necesitar saber" resulte más efectivo en general para establecer una cultura para el manejo adecuado de la información sensible que HIPAA procura proteger.

Si bien es posible que las afirmaciones de Moneypenny (2021) no tengan más sustento que su experiencia particular y subjetiva, una y otra vez se observa un consenso general de que este principio está implicado por la regulación y sus directrices, o bien por la práctica estándar en el ramo (Edemekong et al., 2021; Kibbe, 2001; Kiel, 2012; Wysoker, 2003).

Un problema que persiste, sin embargo, es que todas estas alusiones al principio dejan por fuera una definición precisa de este. Wolter (2021) ofrece la siguiente definición: "Este principio dice que un usuario solo debe tener acceso a la información que su función requiere, indistintamente de su nivel de autorización" (párrafo 1).

Otra definición, más autoritativa aunque rigurosa, se encuentra en Office of the Under Secretary of Defense for Intelligence and Security (2017):

> Una determinación hecha por quien posee información clasificada de que un posible destinatario, en el interés de la seguridad nacional, tiene un requerimiento para acceder, conocer o poseer la información clasificada a fin de realizar tareas o servicios esenciales para cumplir con un programa oficial del Gobierno. La posesión, conocimiento o acceso a la información no será concedida a un individuo solamente en virtud de la oficina a la que éste pertenece, ni su posición o autorización de seguridad. (p. 82)

En ambos casos, debe llamar la atención el hecho de que el acceso a la información no se concede solamente por el rol o función que el destinatario ostente, sino por una necesidad real y legítima de acceder a la información para cumplir con una tarea que le corresponda ejecutar; es decir, un esquema de control de acceso basado en roles por sí solo no será suficiente para satisfacer este principio. En muchas organizaciones, se reconoce el rol de un "propietario de los datos", quien es el que autoriza a los individuos el acceso a la información cuando se necesita.

Marco Metodológico

Tipo de Investigación

El presente trabajo es una investigación de tipo aplicada, en la que se busca atender las inquietudes de los líderes a cargo del desarrollo de El Producto en relación con posibles vulnerabilidades en el diseño de este y recibir recomendaciones concretas sobre mejoras que se le pueden hacer al diseño para eliminar o mitigar las vulnerabilidades.

Alcance Investigativo

El alcance investigativo es exploratorio porque se desea comprender mejor el contexto y dominio del problema que El Producto en busca de técnicas de desarrollo seguro las cuales puedan aplicarse a su proceso de desarrollo, requerimientos de seguridad que pueden haber sido obviados, las amenazas más importantes que deben de considerarse, y las decisiones de diseño de alto nivel que pueden constituirse en debilidades de El Producto frente a estas amenazas.

Enfoque

Esta investigación utiliza un abordaje cualitativo a la hora de definir el enfoque, debido a que este documento tiene el fin de establecer mejoras en la seguridad del software que es desarrollado usando las buenas prácticas de la industria y las experiencias recolectadas de diferentes experiencias recolectadas durante esta investigación.

Diseño

El presente proyecto es una investigación para la acción que trata de resolver problemas de seguridad en el desarrollo de software. “Su propósito fundamental se centra en aportar información que guíe la toma de decisiones para programas, procesos y reformas estructurales” (Hernández et al., 2014, p. 510)

De acuerdo con Hernández et al. (2014) los tres pilares donde se fundamenta la investigación de este tipo son:

- Los participantes que están viviendo un problema son los que están mejor capacitados para abordarlo en un entorno naturalista.
- La conducta de estas personas está influida de manera importante por el entorno en que se encuentran.
- La metodología cualitativa es la mejor para el estudio de los entornos naturalistas.

Hernández et al. (2014) establece que estos estudios construyen el conocimiento por medio de la práctica y caracteriza los estudios de investigación para la acción de la siguiente forma:

- La investigación para la acción envuelve la transformación y mejora de una realidad.
- Parte de problemas prácticos y vinculados con un ambiente o entorno.
- Implica la total colaboración de los participantes en la detección de necesidades (ellos conocen mejor que nadie la problemática por resolver, la estructura por modificar, el proceso por mejorar y las prácticas que requieren transformación) y en la implementación de los resultados del estudio (p. 510).

Población y Muestreo

La población con la que se trabaja para recabar la información requerida para el presente estudio comprende buena parte de las figuras clave involucradas en el desarrollo de El Producto:

1. miembros del equipo de cumplimiento (en inglés, *compliance*), incluyendo el Oficial de Seguridad de la Información (en inglés, *Information Security Officer*)
2. miembros del equipo de infraestructura (de TI)

3. miembros de los equipos de desarrollo de software, incluyendo los jefes de equipo
4. dueños de producto (en inglés, *product owners*)

Instrumentos de Recolección de Datos

Para la recolección de datos, se usaron los siguientes instrumentos y técnicas:

1. Revisión de documentación (políticas, manuales, presentaciones del producto, documentos técnicos de diseño y operación del producto).
2. Revisión del código fuente del producto.
3. Sesiones de trabajo dirigidas.

Técnicas de Análisis de Información

Las técnicas para el análisis de la información recabada son las siguientes:

- Descripción de Arquitectura del Sistema [siguiendo específicamente lineamientos del enfoque de Puntos de Vista y Perspectivas de Rozanski y Woods (2012)]
- Modelado de Amenazas [en su variante "ágil" propuesta por Maier et al. (2017)]
- Análisis de Superficie de Ataque

Estrategia de Desarrollo de la Propuesta

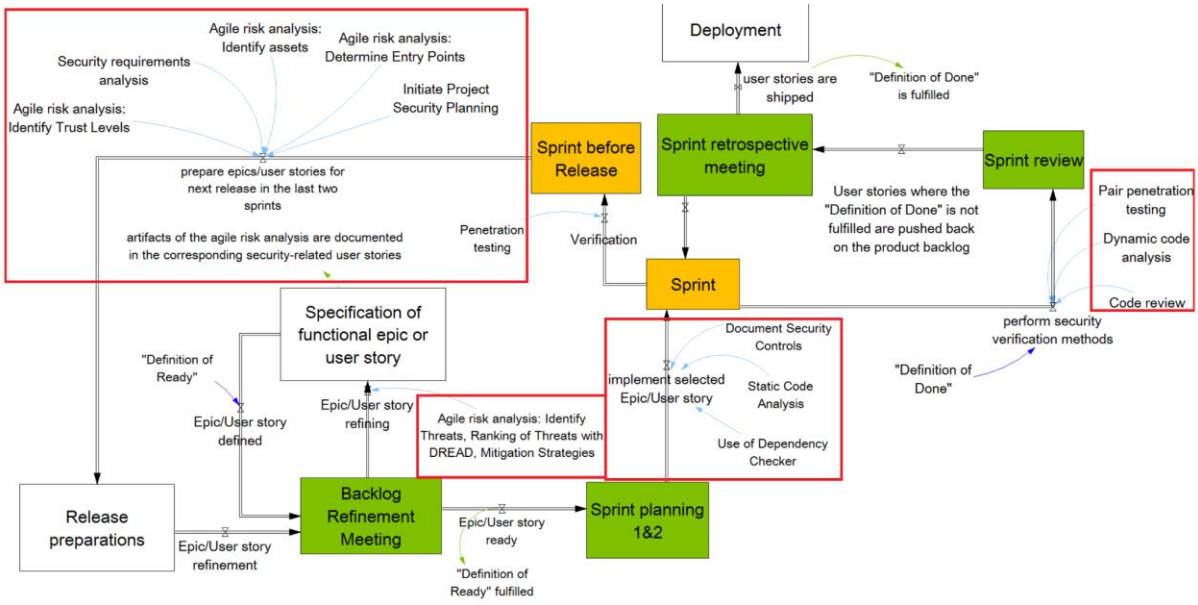
A partir de la revisión de literatura, se confeccionó un documento para resumir las diversas técnicas, actividades y métodos que se han propuesto para llevar a cabo procesos de desarrollo de software seguro. Los contenidos de este documento fueron la base para la elaboración de la sección El Desarrollo de Software Seguro, en el capítulo Marco Conceptual.

El documento de técnicas y métodos de desarrollo seguro fue revisado y discutido con los líderes a cargo del mantenimiento y desarrollo de El Producto, de lo cual surgieron los siguientes acuerdos:

1. Las técnicas de particular interés para el contexto de El Producto serán aquellas que sean compatibles con el proceso de desarrollo Scrum que ya se emplea en La Organización.
2. Considerando que en el proceso de desarrollo de El Producto ya se emplean herramientas de análisis estático para la detección de vulnerabilidades, y se efectúan pruebas de penetración con regularidad en los lanzamientos de nuevas versiones del producto, no se ve que las técnicas enfocadas en las fases de codificación y pruebas puedan aportar un valor adicional y no se tiene particular interés en estas en este caso.
3. La interrogante que persiste en el equipo y el liderazgo es si todavía es posible que no se tenga claridad sobre vulnerabilidades en el diseño que no se hayan detectado; es decir, que no se está seguro de lo que no se sabe (en inglés, *unknown unknowns*).
4. De las diferentes actividades que se muestran en la Figura 6, que es una representación gráfica de la adaptación hecha por Maier et al. (2017) como propuesta de un “Proceso Scrum Seguro”, llamaron la atención las relacionadas con la identificación de requisitos de seguridad y el modelado de amenazas, las cuales se ven como facilitadores para atender la interrogante anteriormente mencionada.

Figura 6

Proceso Scrum Seguro



Nota: Maier et al. (2017)

Así pues, a fin de identificar posibles mejoras a la seguridad del diseño de El Producto, se desea llevar a cabo un trabajo de recolección y análisis de requisitos de seguridad, y de modelado de amenazas para identificar aquellas que quizás no estén adecuadamente resueltas y para las que se deban implementar medidas de mitigación.

Para la recolección de requisitos de seguridad, se opta por comenzar con la inspección de la documentación de políticas y procedimientos de seguridad con las que ya cuenta La Organización, así como de documentación de El Producto, sus requisitos, complementados con sesiones de trabajo con desarrolladores y dueños de producto para esclarecer requisitos e inquietudes en relación con las consideraciones de seguridad y cumplimiento.

Análisis del Diagnóstico

En este capítulo se presentan y discuten los resultados del análisis de la información recolectada en relación con El Producto y su contexto.

Revisión de Documentación

La revisión de las políticas de seguridad de la información permite identificar los siguientes aspectos que se considera tienen un impacto directo en los requerimientos del software que se adquiere o desarrolla en la organización:

1. Se prescribe un sistema de clasificación de la información en el que se distinguen al menos dos clases o niveles de sensibilidad:
    a. “Confidencial”, en la que se incluye toda la información específicamente designada como tal por la regulación vigente, tal como la PHI y la PII.
    b. “De Uso Interno”, otra información generada y gestionada a lo interno de la organización y que no es de carácter público; esta es la clasificación que se asume cuando la información no cumple con las características de la clase confidencial.
2. Cada categoría de clasificación de datos, a su vez, conlleva una serie de requerimientos específicos sobre el manejo que se les debe dar y las medidas de protección que se consideran adecuadas para estos. Entre otros, se prescriben controles diferenciados para cada clasificación tales como el nivel de detalle de los registros de auditoría (en inglés, *audit trails*) que se deben llevar de los eventos de acceso, uso y modificación de la información, la necesidad o no de cifrar los archivos cuando se almacenan en disco (o incluso si no se permite del todo guardar la información en disco).

3. En relación con los controles de acceso y procesos asociados, cabe destacar que existen diferencias, en ocasiones significativas, en los requerimientos de control y manipulación de la información de diferentes niveles. Por ejemplo, mientras que para los datos de la categoría "De Uso Interno", se requiere del uso de Controles de Acceso Basados en Roles (en inglés, *Role-Based Access Control, RBAC*); estos no se consideran suficientes para la categoría de datos "Confidencial", sino que en este último caso se debe controlar el acceso con base en la "necesidad de saber" (en inglés, *need-to-know*) que se tenga para llevar a cabo el trabajo, y siguiendo el principio de acceder al "mínimo necesario" de información. Estas diferencias pueden tener un fuerte impacto en las decisiones de diseño y opciones de implementación de los sistemas de información que manipulan la información confidencial.
4. Se identifica el rol de Propietario de Datos (en inglés, *data owner*), el cual debe estar definido para cada activo de información. Este rol es responsable de determinar y asignar la clasificación correspondiente para el activo, así como autorizar al personal y funciones que van a tener acceso a este. En el caso de la información confidencial, es el Propietario de Datos quien personalmente autoriza a cada individuo para tener dicho acceso y vela porque la lista de personas con acceso se mantenga correcta y actualizada.

Sesiones de Trabajo Dirigidas

En sesiones de trabajo con los equipos de desarrollo, se identifican varias inquietudes en relación con las actividades de desarrollo y de operación de los sistemas:

1. Durante la operación del sistema, con frecuencia se requiere efectuar consultas a los datos que el sistema maneja o produce; el propósito de estas consultas es verificar la calidad y corrección de los datos y las salidas producidas, o bien ejecutar diagnósticos para identificar y corregir problemas en el sistema. Al respecto, cabe resaltar que, si bien estas consultas generalmente no están enfocadas en examinar información sensible, los desarrolladores y operadores deben ser vigilantes en todo momento para asegurarse de que estas consultas no devuelvan datos confidenciales que no son estrictamente necesarios para la tarea a la mano; de lo contrario, no estarían cumpliendo a cabalidad con los principios de "necesitar saber" y "mínimo necesario" que se imponen sobre la información en la clasificación confidencial.
2. Lo anterior se debe a que los almacenes de datos contienen una mezcla de datos confidenciales con otros que no lo son y, aunque esto resulta en que dichos almacenes se gestionan con los controles requeridos para información confidencial (en una mezcla de información de distintas clasificaciones, se imponen los requerimientos de la clasificación más sensible); esto se percibe como una exigencia que, si bien es necesaria, es también desgastante.
3. El aprovisionamiento de datos en los ambientes de pruebas y desarrollo es otra preocupación permanente. Con frecuencia resulta conveniente contar con conjuntos de datos con características muy semejantes a los que están presentes en los ambientes de producción (por ejemplo, para reproducir escenarios o eventos de este último, o para efectuar pruebas de rendimiento con cargas realistas antes de la liberación de cambios al sistema). Sin embargo, no es posible copiar

información del ambiente de producción al de pruebas debido a la información sensible que está presente en el primero que, por consideraciones de seguridad, no puede estar presente en los demás ambientes. Actualmente, existen retos con los procesos que derivan conjuntos de datos donde la información confidencial de producción ha sido removida u ofuscada para que esta sea apta para su uso en el entorno de pruebas, ya que se requiere la supervisión de los ingenieros experimentados para asegurarse de que estos procesos se mantengan al día con los cambios que se dan en el sistema principal de manera que se tenga correctamente identificada toda la información potencialmente sensible que debe ser removida para que esta no se filtre accidentalmente fuera del ambiente de producción. El riesgo de esta filtración accidental también ha llevado a que se tengan que implementar, en el ambiente de pruebas, controles comparables a los de producción, lo cual eleva el costo de operación de los ambientes secundarios.

4. En relación con los programas de seguridad y cumplimiento, algunos programadores expresan frustración por el hecho de que, aunque sí se recibe regularmente capacitación en temas de seguridad, esta todavía y en su mayoría es de concientización y genérica a las funciones del negocio; por otro lado, para las funciones de desarrollo de software esta es mucho más escasa, limitándose principalmente a un repaso de los contenidos del OWASP Top 10. También se carece de guías de seguridad específicas para aplicar en las actividades de desarrollo (p. ej. durante las discusiones de las historias de usuario, el diseño de los esquemas de datos, o bien las revisiones de código). Como consecuencia, los

programadores todavía no están muy claros sobre cómo aplicar consideraciones de seguridad en el día a día de las actividades de desarrollo.

Método de Evaluación del Riesgo

La documentación revisada incluyó la política y procedimientos de Gestión del Riesgo de Seguridad de la Información. Si bien estos están principalmente dirigidos a la gestión del riesgo organizacional y de procedimientos, proveen lineamientos útiles para el análisis de riesgos y amenazas de El Producto, como se verá más adelante.

En esta sección se repasan los aspectos del método de Evaluación del Riesgo de La Organización más relevantes para el presente trabajo.

Amenazas

El método reconoce las siguientes como potenciales amenazas comunes a considerar en evaluaciones de riesgo:

- **Internas Hostiles**: personal que trabaja en la organización que *intencionalmente* compromete la integridad, confidencialidad o disponibilidad de un activo de información.
- **Internas no Hostiles**: personal que trabaja en la organización que *accidentalmente* compromete la integridad, confidencialidad o disponibilidad de un activo de información.
- **Externas Hostiles**: personal externo a la organización que *intencionalmente* compromete la integridad, confidencialidad o disponibilidad de un activo de información.

- **Externas no Hostiles**: personal externo a la organización que *accidentalmente* compromete la integridad, confidencialidad o disponibilidad de un activo de información.
- **Problemas Técnicos**: defectos de software o hardware, fallas del sistema, código malicioso.

Evaluación del Impacto

Se identifican diferentes niveles de impacto para varios objetivos de negocio de La Organización. Para efectos de este trabajo, solamente se considerará el objetivo del cumplimiento con la normativa HIPAA y con las obligaciones contractuales a las que se está sujeta como custodio de la información de salud que fue provista por los clientes.

Se considera severa la filtración de registros de más de 500 personas; requerirá notificar de manera pública además de hacerlo a los afectados y por supuesto implica fuertes sanciones económicas.

Se considera mínimo el impacto de una filtración de registros de menos de 100 personas; requerirá notificar a los afectados pero no necesariamente hacerlo de manera pública; probablemente habrá sanciones económicas, aunque de menor cuantía.

Probabilidad de Ocurrencia del Riesgo

La estimación de la probabilidad de ocurrencia, según el método documentado por La Organización, se basa en una combinación de factores como la motivación de la amenaza (si es intencional o no), los medios disponibles (si se requiere o no de habilidad/experticia/tiempo para explotar la vulnerabilidad) y oportunidad (si las circunstancias en las que se puede explotar la vulnerabilidad están presentes en todo momento, o si más bien son limitadas en el tiempo y/o restringidas a ciertas condiciones).

Sin embargo, este esquema todavía deja a juicio de quien estima en qué configuración se conjugan estos factores para cada amenaza. Resultará útil entonces poder contar con alguna base objetiva para llevar a cabo estas estimaciones y, por fortuna, (Seh et al., 2020) desarrollaron recientemente un estudio que permite tener:

- Las filtraciones de datos son un fenómeno continuamente al acecho y van en aumento año con año.
- 62 % de las filtraciones de datos en los últimos 15 años han sido del Sector Salud; la proporción sube al 77 % si solo se consideran los últimos cinco años.
- Las filtraciones a raíz del hackeo son, por mucho, las que han comprometido más información. Los casos en que las filtraciones han sido facilitadas a lo interno de la organización son muy pocos, pero los ataques mediante phishing, malware e ingeniería social pueden vulnerar la información a la que tienen acceso el personal en las organizaciones.
- Si bien los incidentes de filtración de datos habían tenido lugar históricamente por el robo o pérdida de documentos físicos, en los años más recientes son las filtraciones de información de correos electrónicos y de servidores las que predominan.
- La mayoría de las filtraciones de datos tienen motivaciones económicas.

De lo anterior, pues, se puede asumir que las amenazas internas hostiles no serán comunes, pero sí se deberá tener más en cuenta la posibilidad de amenazas hostiles externas y problemas técnicos; la probabilidad de que una amenaza no hostil se convierta en un vector de ataque por parte de una hostil también deberá considerarse como significativa.

Arquitectura Base

A partir de la documentación y código fuente del producto inspeccionados, así como de las entrevistas organizadas con diversos involucrados, en esta sección se elabora una descripción de su arquitectura. Debe aclararse que esta no es una descripción completa de El Producto real, sino que solamente se incluyen los elementos que son relevantes para el desarrollo del presente estudio de caso.

La descripción se elabora basándose en los lineamientos del enfoque de Puntos de Vista y Perspectivas (en inglés, *Viewpoints and Perspectives*) de Rozanski y Woods (2012), escogiendo para este trabajo solamente el Punto de Vista de Contexto y el Punto de Vista Funcional.

Punto de Vista de Contexto

En esta sección se introduce a El Producto en su contexto de uso y operación, a fin de entender sus requisitos y características clave y los actores y entidades con los que el sistema interactúa, tanto internos a La Organización, como externos a esta.

Requisitos y Características Clave. En torno al Ajuste de Riesgos en el Sector Salud, EUA, una plataforma de analítica puede ser de gran apoyo para las instituciones involucradas en estos programas, por lo cual se ha desarrollado un mercado de soluciones comerciales que buscan atender sus necesidades. Whitehurst (2016) describe varios de los retos presentes de manera generalizada en las organizaciones que gestionan estos programas, los cuales sugieren cuáles pueden ser varias de las capacidades y características clave de un sistema que permita atender y lidiar con estos retos. Para efectos del presente estudio, se pueden resaltar las siguientes:

- *Gestión de los datos y el rendimiento*: el sistema debe permitir hacer análisis sofisticados con la información proveniente de varias fuentes a fin de comprender

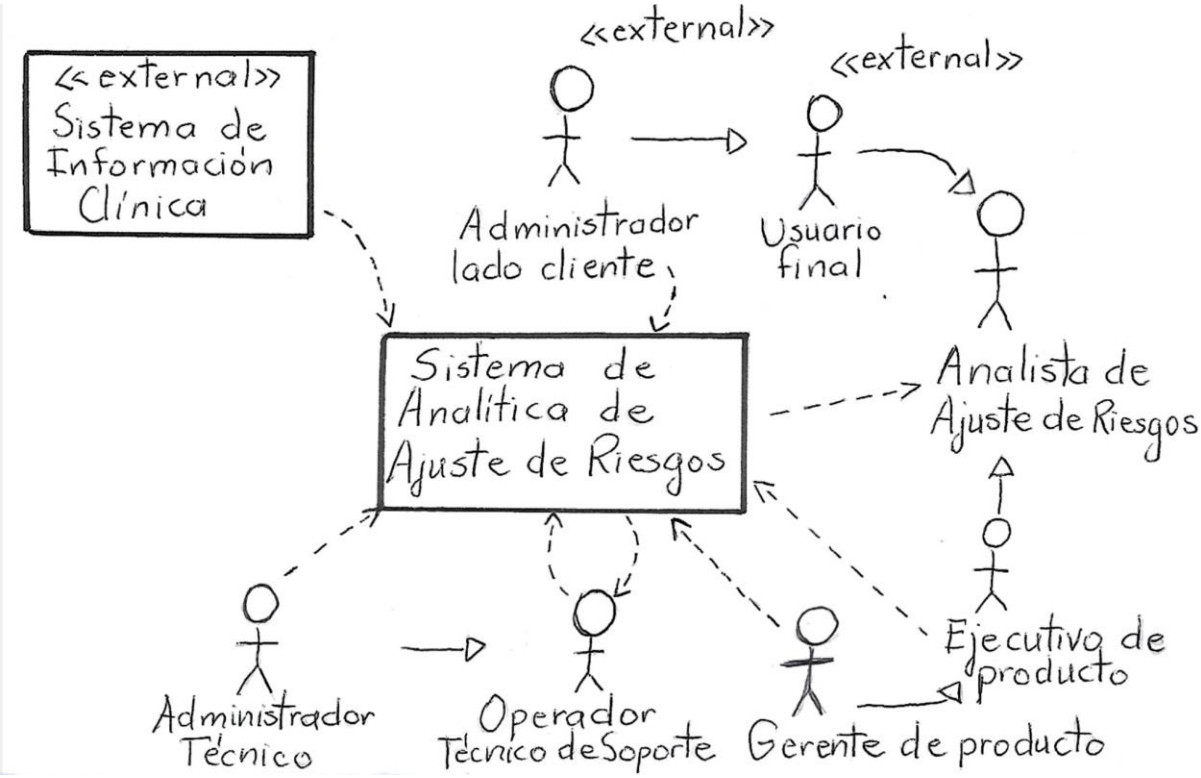
a cabalidad las condiciones de salud de los miembros del plan y aumentar la capacidad para hacer intervenciones focalizadas a fin de mejorar el retorno de la inversión de los planes. Mostrar métricas especializadas para medir el rendimiento de la gestión de los planes.

- *Análisis prospectivo*: este se hace con el fin de identificar el cuidado adecuado que requerirán los miembros del plan, en el momento adecuado; lograr esto de manera efectiva permite mejorar tanto el desempeño clínico como financiero. El sistema deberá facilitar el identificar y alcanzar a los miembros que requerirían atención, coordinar con proveedores las visitas, e iluminar con los datos disponibles la priorización en el tiempo de los esfuerzos, todo lo cual es clave para optimizar los resultados.
- *Adquisición de datos*: las organizaciones deben habilitar el acceso completo a la información clínica, al día, que se puede obtener de los registros médicos (en inglés, *charts*); sin embargo, la recuperación y extracción de esta información a partir de los registros originales puede ser costosa. Por consiguiente, el sistema deberá facilitar el acceso de la información, por parte de múltiples departamentos y actores, sin tener que duplicar y repetir las extracciones.
- *Cumplimiento*: las organizaciones deben adoptar abordajes modernos para asegurar al máximo el apego a las regulaciones y protegerse ante posibles penalizaciones. Entre otras consideraciones, el diseño y operación del sistema deberá brindar protección adecuada a la confidencialidad e integridad de la información sensible que se le confía.

Modelo de Contexto. El contexto de El Producto se ilustra en la Figura 7.

Figura 7

Diagrama de Contexto de El Producto



En el diagrama se pueden apreciar las siguientes entidades alrededor de El Producto:

- *Sistema de Información Clínica*: sistema externo que resguarda la información clínica de los miembros suscritos a los planes operados por los clientes de La Organización. Esta es una designación genérica que puede representar diversos sistemas que gestionan esta información, puede variar de cliente en cliente. Estos sistemas proporcionan la información clínica que es el insumo esencial que El Producto requiere, el cual comprende una cantidad muy importante de material altamente sensible y objeto de fuertes regulaciones y controles que buscan garantizar su privacidad y seguridad.

- *Analista de Ajuste de Riesgos*: principal generalización de los diversos roles del usuario final. Es un actor principalmente interesado en ejecutar consultas y elaborar reportes a fin de analizar diversas características de la población de miembros del plan, tanto agregados como a nivel de miembros específicos, a fin de tomar decisiones sobre acciones que se pueden tomar para mejorar la atención de los miembros y el rendimiento de la gestión de los planes de aseguramiento. Si bien las consultas agregadas generalmente no exponen información sensible, las consultas y reportes a nivel individual de los miembros con frecuencia sí la contienen.
- *Usuario Final*: especialización del Analista de Riesgos que se refiere a los usuarios del cliente que hacen uso del producto; son externos a La Organización.
- *Administrador por Parte del Cliente*: usuario final, también externo a La Organización, que además gestiona la designación del personal del cliente que hacen uso del producto y que autorizan también cuáles de esos miembros pueden tener acceso a las diversas funciones de esta, como los distintos reportes y consultas a nivel agregado y de miembros individuales.
- *Ejecutivo del Producto*: actor interno de La Organización, cuya función es mantener el contacto con el cliente y asistirlo en el uso del producto. Con frecuencia requerirá hacer uso de las mismas funciones de El Producto que los usuarios finales externos, pero además podrá llevar a cabo otras tales como gestión de la configuración del producto para habilitar su uso por parte del cliente, e inspección de métricas y reportes que permiten monitorear el uso que el cliente hace del producto. Adicionalmente, este actor recibirá, por parte del cliente,

consultas y reportes de problemas, algunos de los cuales podrá resolver por sí mismo, y otros quizás requieran asistencia por parte del Operador Técnico de Soporte. Puede tener acceso a la misma información sensible de los miembros del cliente que los usuarios finales, aunque este acceso debe estar justificado por las necesidades de brindar la asistencia que el cliente requiere.

- *Gerente del Producto*: personal del negocio a cargo del producto y de los ejecutivos. Además de poder llevar a cabo las mismas funciones del Ejecutivo del Producto, gestiona la designación de los ejecutivos y los autoriza para acceder a la información de sus clientes asignados.
- *Operador Técnico de Soporte*: personal interno de La Organización encargado de velar porque El Producto opere con normalidad. Tiene acceso a aspectos técnicos internos de El Producto tales como sus bitácoras técnicas, los registros de entrada, intermedios y finales producto de la operación del sistema; esta información es útil para diagnosticar y resolver problemas, pero el operador debe tener la precaución de no acceder a los elementos de información sensibles a menos que sea estrictamente necesario, lo cual usualmente solo se da a raíz de la solicitud expresa por parte del Ejecutivo del Producto para atender un problema específico que se haya identificado al usar El Producto.
- *Administrador Técnico del Sistema*: personal técnico interno de La Organización a cargo de los Operadores Técnicos de Soporte, y es quien gestiona la designación de estos y los autoriza para tener el acceso a los elementos técnicos internos de El Producto.

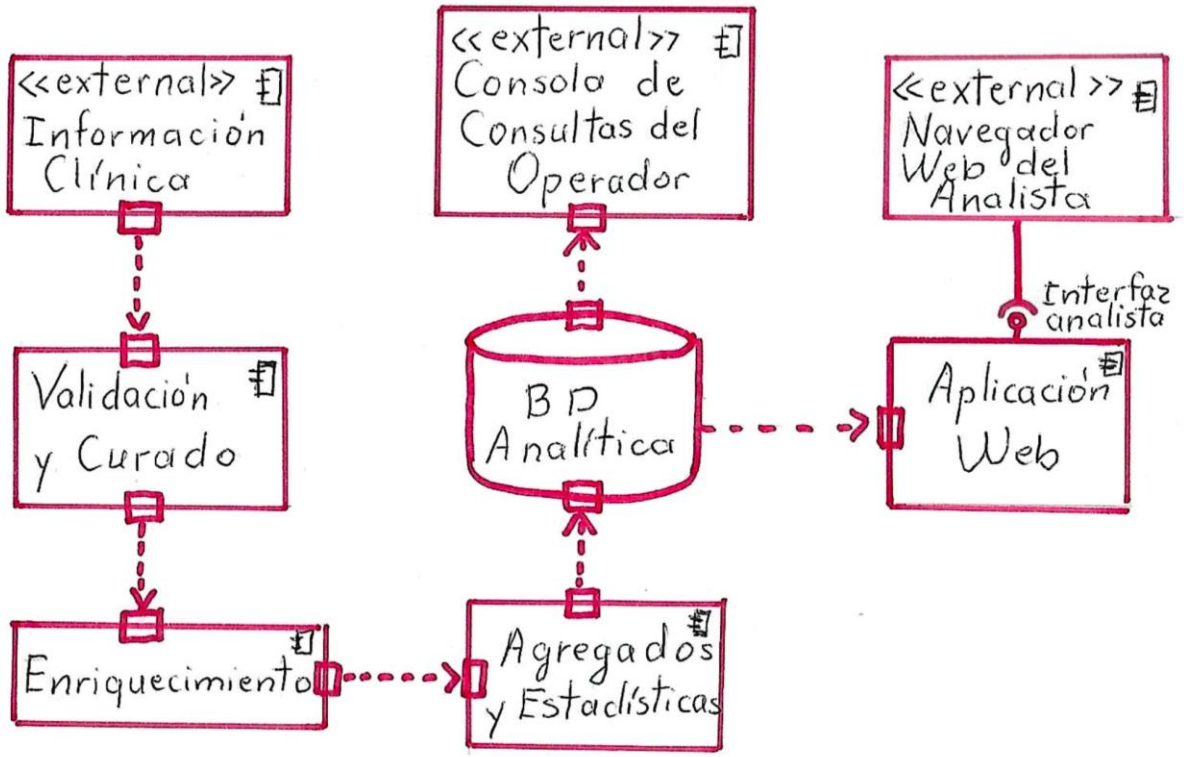
Punto de Vista Funcional

En esta sección se describen los diferentes elementos del sistema que llevan a cabo sus funciones.

Modelo de Estructura Funcional. En la Figura 8 se muestra el diagrama para el modelo de estructura funcional del sistema.

Figura 8

Diagrama Funcional de El Producto



Nota: En este modelo funcional de El Producto, se aprecia cómo la información fluye a través de los diferentes componentes. Dado que la información que proviene de la fuente contiene una mezcla de información confidencial y no confidencial, y en la arquitectura base no se toman medidas específicas para segregar ambos tipos de información, todos los componentes y flujos de información contienen información confidencial y deben ser sujetos a controles correspondientes a este nivel de seguridad; por esta razón, todos los componentes y flujos se resaltan en color rojo.

En este modelo se identifican los siguientes elementos:

- *Sistema de Información Clínica*: Este representa el sistema externo que suple la información clínica de los miembros del plan. La información que se provee combina tanto información confidencial como no-confidencial.
- *Validación y Curado*: Este componente se encarga de la ingesta de datos que provienen del Sistema de Información Clínica, y tiene el propósito de validar los registros de entrada y alertar si se detecta algún problema con este. En caso de que la información pase la validación inicial, se transforma para facilitar su procesamiento en las etapas posteriores.
- *Enriquecimiento*: Este componente procesa la información clínica curada para enriquecerla con los puntajes e índices de ajuste de riesgos. Es aquí donde se ejecutan cálculos complejos basados en modelos oficiales y/o estandarizados (ver Ajuste de Riesgos en el Sector Salud, EUA).
- *Agregados y Estadísticas*: Este componente parte de la información clínica enriquecida y produce agregados y estadísticas que son útiles para analizar el comportamiento del plan. El resultado de este proceso se convertirá en el insumo para desarrollar los análisis de rendimiento y de inteligencia de negocios para la toma de decisiones (el fin último de El Producto).
- *BD de Analítica*: Esta es la base de datos que almacena los resultados provistos por el componente de Agregados y Estadísticas para servirlas a la Aplicación Web.
- *Aplicación Web*: Este es el componente que representa el proceso servidor que hospeda la aplicación web con la que actúan los usuarios finales.

- *Navegador Web del Analista*: Este es el componente que representa el proceso cliente que el usuario final utiliza para interactuar con la aplicación web.
- *Consola de Consultas del Operador*: Este componente corresponde al proceso cliente de las herramientas de consulta que el operador técnico utiliza para interactuar directamente con la BD de Analítica, lo cual es frecuentemente requerido para brindar el soporte técnico solicitado por parte de los Ejecutivos y Gerentes de Producto.

Un aspecto a resaltar en este modelo funcional es que, a lo largo de todo el flujo de información del sistema, la información confidencial viaja junto con la información no-confidencial, por lo que todos los flujos de datos y componentes deben de ser sujetos de los controles requeridos para la protección de la información confidencial.

Modelo de Amenazas

En esta sección se documenta el análisis de amenazas para El Producto y se usa como guía los artículos de CheatSheets Series Team (n.d.-c) y Conklin y Drake (n.d.).

Activos de Información

El principal activo de información en El Producto es la información clínica de los miembros de planes de aseguramiento, especialmente la confidencial, que es fundamentalmente PHI, por lo cual cualquier impacto a esta afectaría el objetivo de cumplimiento de HIPAA y obligaciones contractuales con los clientes que suplen esta información.

Dependencias Externas

Las siguientes son las dependencias externas con las que El Producto interactúa:

- Navegadores usados para acceder a la aplicación web de El Producto
- Consola de Consultas del Operador

- Sistema de Información Clínica
- Base de datos de analítica

Puntos de Entrada

- Navegadores usados para acceder a la aplicación web de El Producto
- Interfaz del Analista de la aplicación web de El Producto
- Sistema de Información Clínica
- Consola de Consultas del Operador

Puntos de Salida

- Navegadores usados para acceder a la aplicación web de El Producto
- Estación de trabajo de usuarios de la aplicación web de El Producto
- Módulos del proceso servidor de la aplicación web de El Producto (con acceso a Internet)
- Consola de Consultas del Operador
- Estación de trabajo del Operador
- Módulo de validación y curado (con acceso a Internet)

Amenazas

En la Tabla 6 se enumeran algunos de los riesgos que pueden identificarse para el producto, para ello se emplea el método de evaluación de riesgos de La Organización para asignarles también una valoración de su riesgo.

Tabla 6

Amenazas de El Producto

Amenaza	Puntos de Entrada / Salida	Impacto	Probabilidad de Ocurrencia	Riesgo
Hostil Externo: induce a usuario a revelar reporte con información confidencial	Navegador web del usuario	Significativo (reportes pueden incluir decenas o cientos de registros)	Significativo: ataques de “spear” “phishing” o ingeniería social han ocurrido en el pasado	Significativo a severo
Problema técnico: secuestro de credenciales o sesión de usuario por malware	Navegador web de usuario	Significativo (reportes / consultas pueden incluir cientos de registros)	Significativo: ataques por malware han ocurrido en el pasado	Significativo a severo
Hostil Externo: induce a Operador a revelar información confidencial	Consola de Consultas del Operador	Severo (resultados de consultas pueden incluir miles de registros)	Posible: ataques de spear phishing o ingeniería social se han dado, pero requieren mayor esfuerzo	Severo
Problema técnico: secuestro de credenciales o sesión de operador por malware	Consola de consultas del operador	Severo (resultados de consultas pueden incluir miles de registros)	Posible: ataques de spear phishing o ingeniería social se han dado, pero requieren mayor esfuerzo	Severo
No-Hostil interno: operador crea copias no controladas de información confidencial	Consola de consultas del operador	Medio (salvo el caso de negligencia extrema, no se espera que el personal copie o transfiera un gran número de registros sin intención)	Posible: ha sucedido en el pasado que las consultas elaboradas devuelven información confidencial a pesar de que no era estrictamente necesaria	Medio

Nota: Se emplea el método de evaluación de riesgos de La Organización. No se trata de elaborar de manera exhaustiva todas las amenazas, sino que se limita a aquellas que se consideran representar el mayor riesgo.

Análisis de Superficie de Ataque

Woody y Ellison (2020) recomiendan complementar el modelado de amenazas con un análisis de la superficie de ataque. Esto permite identificar de manera más sistemática las posibles amenazas y además amplía la perspectiva de la valoración de posibles mitigaciones a las amenazas mediante la reducción del área de ataque, de manera que se reducen también las formas en que se puede atacar al objetivo.

Al observar la Figura 8, salta a la vista el hecho de que la totalidad de los componentes y flujos de datos en el sistema procesan o almacenan información confidencial, la cual sería el objetivo de alto valor para el atacante. Es decir, el área de ataque es sumamente amplia, por lo que requiere una gran cantidad de controles en cada uno de los activos y puntos de entrada y salida del sistema para tratar de mitigar las amenazas.

El enfoque de reducción del área de ataque sugiere se deberían hacer esfuerzos por encontrar maneras en que esta área se pueda reducir, tratando de contener y limitar el flujo de información confidencial de alto valor a un número reducido de componentes e interacciones. Este enfoque será el que guíe en gran medida las diversas propuestas de mejora del diseño de seguridad para El Producto en el siguiente capítulo.

Propuesta de Solución

En este capítulo se presenta una serie de recomendaciones que buscan atacar debilidades identificadas en el análisis sobre El Producto, llevado a cabo en el capítulo anterior, de sus requisitos de seguridad, de las amenazas que representan mayor riesgo, y de su diseño.

Recomendaciones Generales sobre la Arquitectura

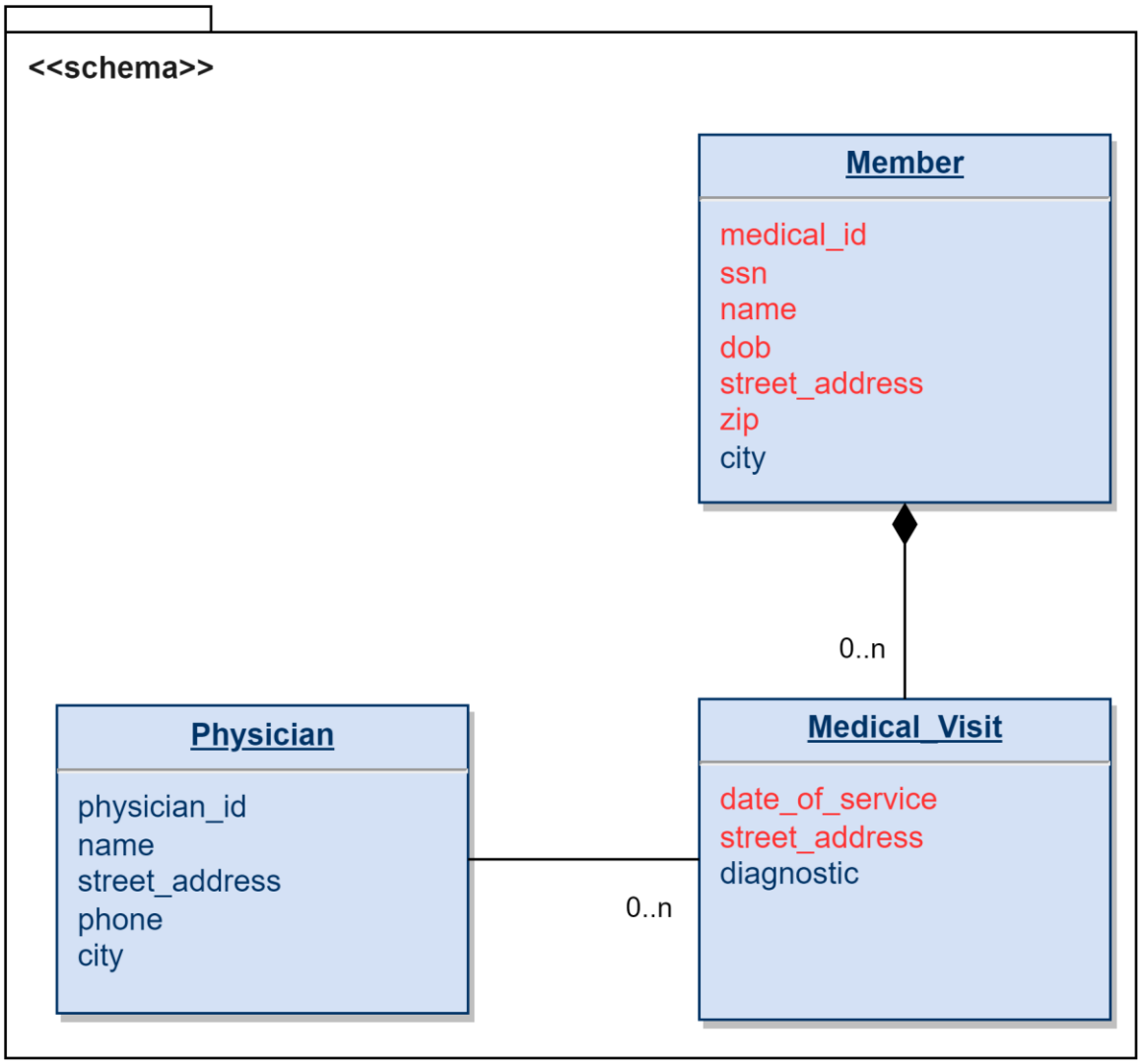
En esta sección se presentan algunas recomendaciones que son aplicables a nivel general de la arquitectura del sistema, caracterizadas por impactar a la vez el diseño de varios de los componentes del sistema.

Segregar los Datos según su Clasificación

Un diseño común del esquema (en inglés, *schema*) de base de datos se muestra en la Figura 9. En este, se puede observar que las tablas contienen una mezcla de atributos cuya clasificación es confidencial (resaltados en rojo), con otros que no lo son.

Figura 9

Diseño de Tablas Sin Segregar por Clasificación



Nota. Los campos confidenciales, resaltados en color rojo, se encuentran mezclados con campos no confidenciales. Las consultas efectuadas contra este esquema pueden retornar con facilidad información confidencial a menos que la consulta se elabore con cuidado.

Este diseño, aunque común, plantea varios retos:

1. al personal y aplicaciones que no requieren acceso a los datos confidenciales, generalmente se les concede acceso a estos, pues el mecanismo de control de acceso del motor de base de datos puede no ser lo suficientemente granulado como para limitar el acceso solo a ciertos campos;
2. la información confidencial debe extraerse solamente en la cantidad mínima necesaria; sin embargo, cuando se hacen consultas para extraer información en este diseño, estas deben prepararse de manera cuidadosa para evitar extraer información confidencial que no sea estrictamente necesaria, lo cual es probable que no sea una consideración que se tenga al hacerlas;
3. dado que este esquema contiene información confidencial, se le deben aplicar todos los controles correspondientes a esta clasificación (p. ej. control y monitoreo del ciclo de vida de los datos, auditoría de accesos, etc.), pero ellos normalmente no se justifican para la información no confidencial, por lo que pueden conllevar un sobrecosto de la gestión de los datos.

Por consiguiente, una primera y fundamental recomendación es la de tomar medidas para segregar en la medida de lo posible la información confidencial de la que no lo es. Esto generalmente implica introducir repositorios para información confidencial que están física o lógicamente separados de los repositorios de información convencional.

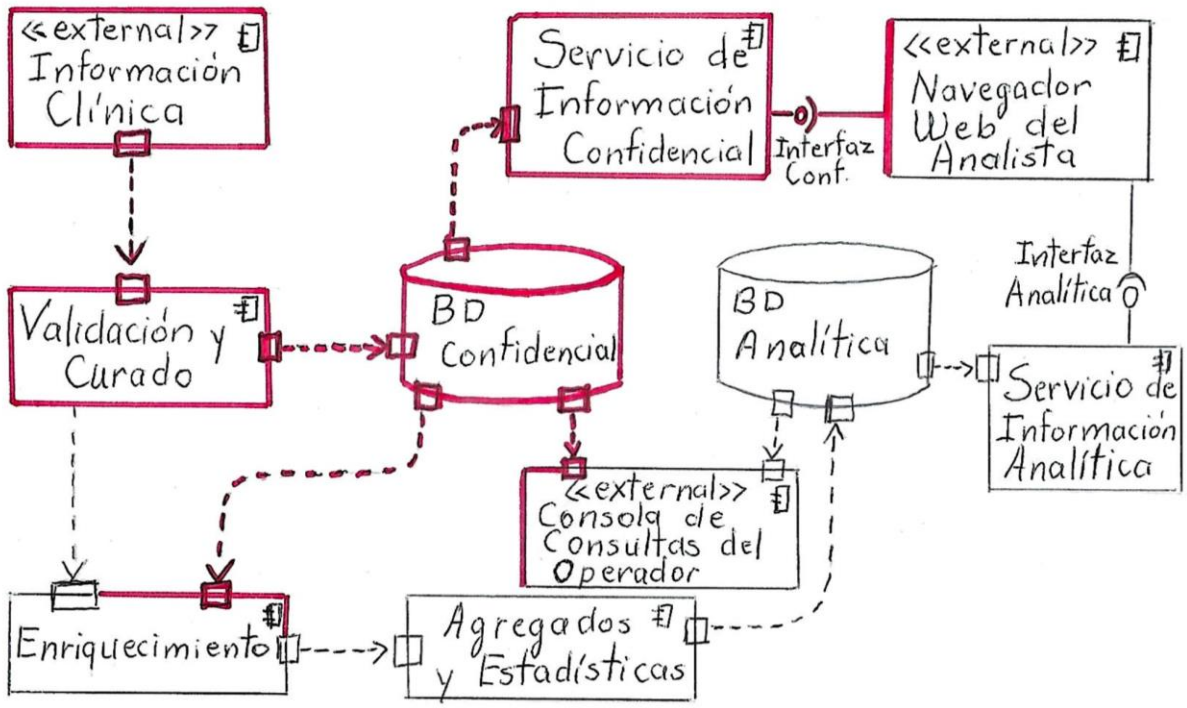
Mantener Segregación de los Procesos a lo Largo del Flujo de Datos

Al partir de la recomendación de mantener repositorios de información confidencial separados de los repositorios convencionales, se procede a considerar cómo esta segregación se debe mantener a lo largo del flujo completo de los datos en el sistema bajo estudio. El resultado

de la aplicación de estas recomendaciones dará como resultado una nueva estructura funcional como la que se muestra en la Figura 10.

Figura 10

Diagrama Funcional con Flujos Segregados de Datos Confidenciales



Nota: En este modelo funcional de El Producto, se aprecia cómo la información fluye a través de los diferentes componentes, se resalta en rojo aquellos por los que fluye información confidencial. En esta estructura, se toman medidas para segregar la información confidencial de la no-confidencial, y se procura limitar el flujo de la primera solamente a los componentes donde es estrictamente necesario.

Adquisición de datos. En esta primera etapa, se recibe la información por parte de las fuentes iniciales; la entrega se hace mediante una combinación de archivos en diferentes formatos, en los que la información confidencial y no confidencial se encuentran entremezcladas. El repositorio donde se retiene esta información debe ser clasificado como confidencial, y todos los controles correspondientes a esta clasificación le son aplicados.

Transformación, limpieza y enriquecimiento. Esta es la primera etapa de procesamiento por parte del sistema bajo estudio; esta etapa se compone a su vez de varios procesos en secuencia, pero desde los primeros es posible aplicar una transformación con la que se separe la información confidencial de la que no lo es, y luego de este punto es posible de mantener separados tanto los repositorios como la mayor parte de los procesos que operan sobre cada uno. Aquellos procesos que requieran operar sobre los repositorios de datos confidenciales serán catalogados como sensibles y serán objeto de controles de seguridad diferenciados: inspecciones de seguridad del código, pruebas de seguridad, priorización en el parcheo de seguridad, monitoreo y auditoría diferenciados y prioritarios, etc.

Carga de información en aplicación final de usuario. La etapa anterior habrá dado como resultado dos conjuntos de datos, uno confidencial y el otro no. Ambos podrán ser cargados en las bases de datos y motores de búsqueda de la aplicación final, pero siempre manteniendo separados los procesos que acceden al conjunto de datos confidenciales, idealmente también dedicando repositorios de datos separados.

Interacción del usuario final con la aplicación. Finalmente, la interacción del usuario será con una única aplicación web, la cual deberá manejar tanto información confidencial como no-confidencial, por lo que esta será clasificada como sensible. Sin embargo, todavía será factible mantener segregación de diversos componentes: los procesos del lado del servidor pueden segregarse, proveyendo una API separada para las operaciones sobre información confidencial (interfaz de acceso a información confidencial), de manera que el área que debe ser sujeta de controles adicionales de seguridad aún es reducida.

Una ventaja importante de mantener una segregación fuerte de los procesos y repositorios en la arquitectura de El Producto es que resulta sencillo identificar cuáles son los procesos y

conjuntos de datos que tratan con información sensible y gestionarlos con el cuidado acorde a su sensibilidad. Por ejemplo, será evidente cuando una revisión de código se haga a un módulo sensible y por consiguiente habrá mayor consciencia de que esta deberá hacerse teniendo particular cuidado a los cambios que se introducen para no incluir vulnerabilidades que pongan en riesgo la seguridad de la información que el módulo gestione.

En las secciones siguientes se examinará con más detalle las implicaciones y desarrollo de estas recomendaciones generales sobre la arquitectura del sistema.

Introducir Códigos Identificadores Internos de Miembro

En diversos escenarios de uso normal del sistema, así como de su operación y mantenimiento, se hace necesario capturar y comunicar el identificador de uno o más miembros específicos. Por ejemplo, cuando un Ejecutivo de Producto llega a identificar uno o más registros de miembros que presentan una incorrección en su información y estos deben ser comunicados al Operador Técnico de Soporte, se le debe facilitar a este último los identificadores de dichos registros para que pueda llevar a cabo el análisis que permita diagnosticar el origen de la información incorrecta.

Un problema con la comunicación de estos identificadores es que estos pueden ser ellos mismos, información confidencial, y por ende toda la comunicación que los contenga debe ser protegida acordemente.

Además, dependiendo del identificador utilizado y de la organización de la información a lo interno del sistema, es posible que el uso de este identificador obligue al operador a tener que iniciar su análisis y ejecutar consultas a los elementos de información sensible, para luego descubrir con frecuencia que el análisis continúa enfocándose en elementos que ya no se consideran sensibles.

Aunque el acceso inicial a la información sensible se podría considerar justificado por la necesidad de atender una solicitud de resolución de un problema específico, lo cierto es que conlleva un mayor costo o complejidad para llevar a cabo las comunicaciones y riesgos adicionales derivados del acceso y circulación de la información sensible.

Una manera en la que podrían evitarse estos problemas es la introducción y empleo de códigos identificadores internos. Estos son identificadores que son generados para referirse a miembros sin revelar información sensible acerca de estos, y que pueden utilizarse en lugar de los identificadores convencionales como el número de beneficiario del plan, el número de registro médico o el número de seguro social, los cuales son explícitamente designados como información protegida en la regulación.

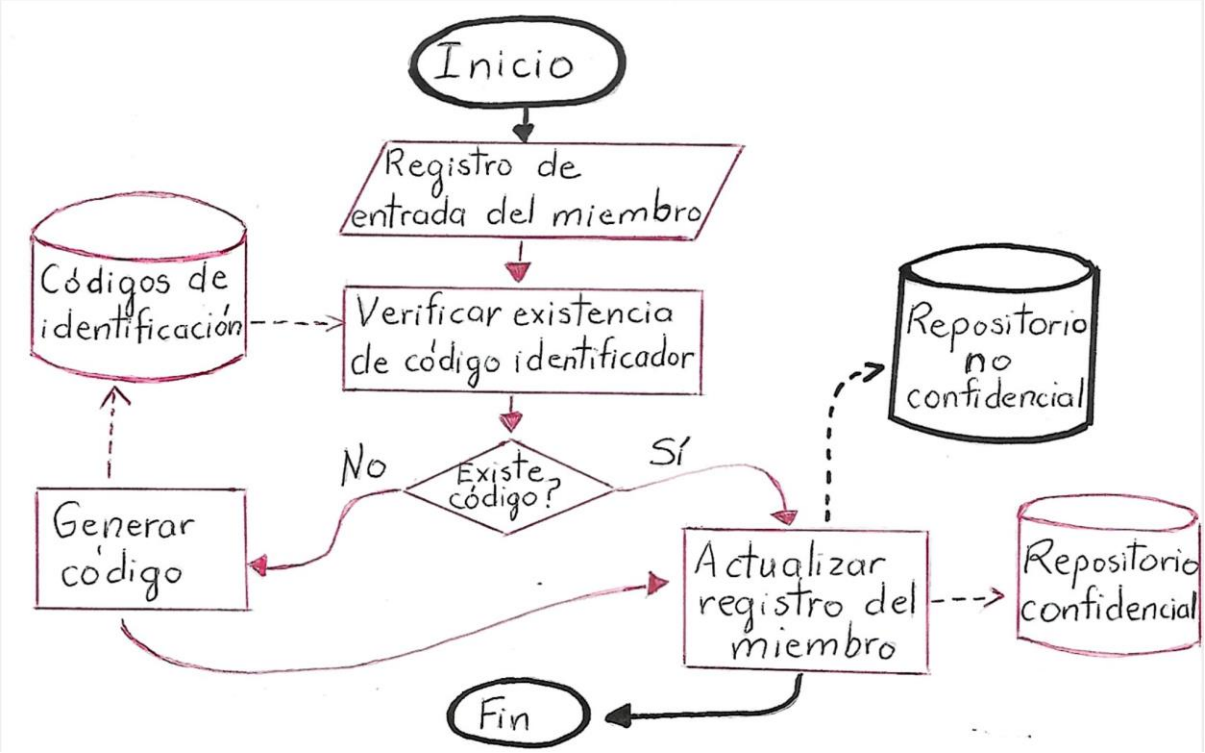
El empleo de códigos identificadores internos debe cumplir con ciertos requisitos técnicos y procedimentales, explicados en (Office for Civil Rights (OCR), 2012), sección "*Re-identification*", tales como los siguientes:

- Los códigos no pueden derivarse o estar relacionados con la información que identifica al individuo.
- No se debe divulgar ni facilitar el mecanismo para obtener de nuevo la identidad original del individuo a quien el código se refiere, si no es a alguien autorizado para acceder a dicha información confidencial.

En la Figura 11 se muestra el proceso mediante el cual se asignarían códigos de identificación al tiempo que se ingresan los registros clínicos en El Producto.

Figura 11

Diagrama de Flujo de Asignación de Códigos de Identificación



Nota. La asignación de códigos de identificación tiene lugar durante la ingesta de registros clínicos de los miembros. Este es un proceso sensible por estar manipulando y operando con datos confidenciales (tanto la información médica protegida del miembro, como la tabla de asignación de códigos de identificación). Al final del proceso, la información confidencial ha sido segregada de la no confidencial al ser almacenadas en los respectivos repositorios.

Estos códigos se emplearían de la siguiente manera:

1. Al cargar el primer lote de información de miembros en el sistema, para cada uno de ellos se genera un código único. Esta generación no se hace de manera totalmente independiente de la información sobre el miembro (por ejemplo, usando un número de secuencia de inserción).

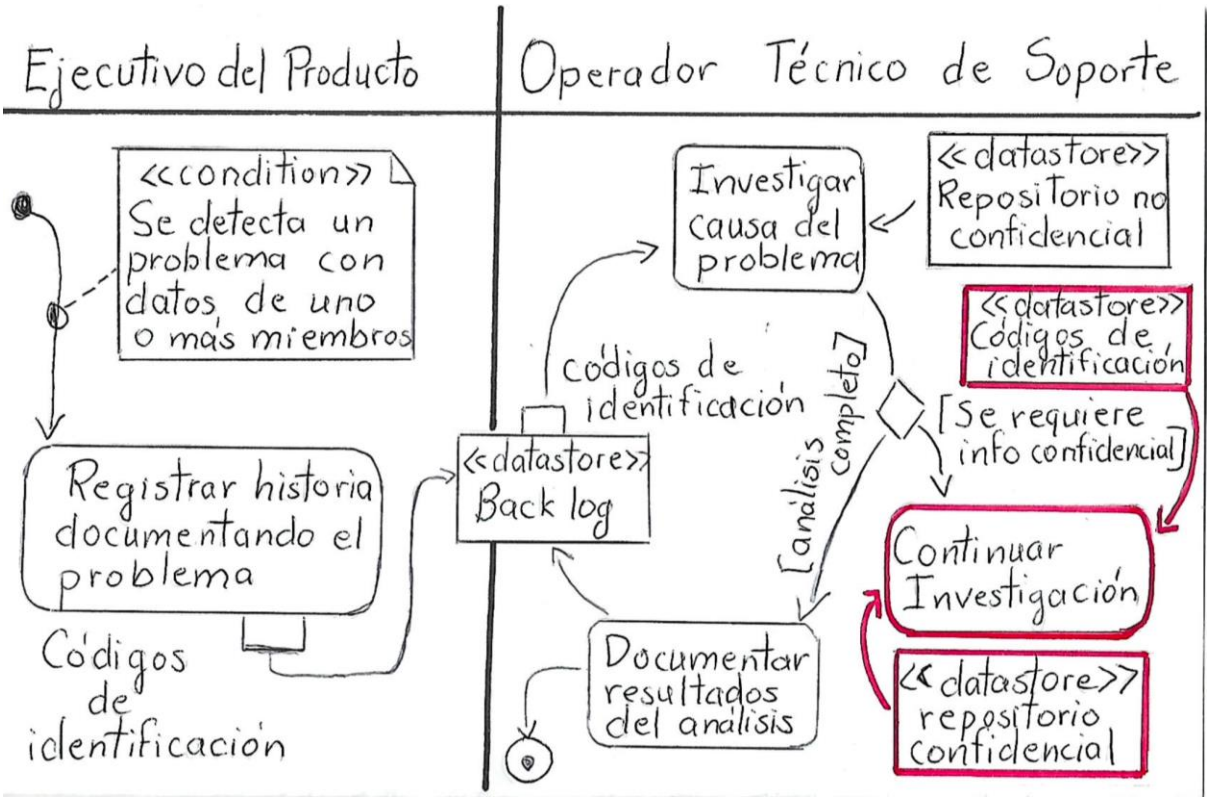
2. Se deberá mantener una tabla de asignación de identificadores que permita determinar el identificador original del miembro a partir del código interno en caso de ser requerido, y que garantice que en actualizaciones futuras al miembro se le siga asignando consistentemente el mismo código identificador. Esta tabla de asociación de identificadores se deberá clasificar como sensible y confidencial, y protegerse acordemente. McCallister et al. (2010, pp. 4-5) señalan que esta tabla, por ser el mecanismo de re-identificación, se deberá mantener en un sistema aparte.
3. Una vez asignado este código de identificación, esta se convertirá en la llave primaria con que se identificará al miembro, y toda la información asociada se referirá a este usando este código (por ejemplo, las llaves foráneas de los registros clínicos se basarán en el código de identificación interno en vez de utilizar el identificador original del miembro.
4. Este código de identificación se propagará junto con la información del miembro a lo largo de todo el proceso hasta que la información se cargue en la aplicación de usuario final.
5. Cuando un usuario final reporte un incidente relacionado con uno o varios miembros en particular, el Ejecutivo del Producto tomará nota de los identificadores internos asociados a estos, en vez de anotar algún otro identificador sensible. Es posible que los identificadores internos no sean visibles para el usuario final, pero sí deberán al menos ser visibles para el Ejecutivo del Producto.

6. Cuando el Ejecutivo del Producto refiera el caso al Operador Técnico de Soporte, solamente consignará los códigos identificadores internos sin mencionar ninguna información sensible (todo lo cual ahora podrá comunicarse de manera ordinaria sin tener que recurrir a protecciones adicionales que son requeridas cuando los mensajes contienen alguna información confidencial).
7. El Operador Técnico de Soporte ahora podrá efectuar su análisis sin tener necesariamente que recurrir a la consulta de alguna información confidencial; solamente cuando su análisis del caso le lleve a tener que examinar algún elemento confidencial, es entonces que el acceso a este se ejecutará, pues estará debidamente justificado.

En la Figura 12 se ilustra el proceso con el cual se investigaría un problema con los datos en El Producto.

Figura 12

Diagrama de Actividades de Análisis de Problema con Datos de Miembros



Nota. En esta colaboración entre el Ejecutivo del Producto y el Operador Técnico de Soporte, en la que se identifica un problema con los datos de uno o más miembros de un plan de salud y luego se le investiga para diagnosticarlo y documentarlo, se aprecia cómo el empleo de códigos internos de identificación para referirse a los miembros afectados permite prescindir del acceso a la información sensible de estos en la mayor parte del proceso: solamente en caso de que el operador determine que la información confidencial es necesaria para continuar y completar la investigación, se accede a esta.

Alternativa: Servicios de Generación de Símbolos Identificadores. Una alternativa a la generación interna de los códigos de identificación es el empleo de un servicio generación de

símbolos de identificación (en inglés, *tokenization*), tal como se amplía en la recomendación de Emplear Servicios de una Plataforma de Gestión de la Privacidad. Esta alternativa, si bien puede representar un costo mayor, tiene la ventaja de consistir generalmente en una solución ya probada y madura que además, por su naturaleza, ya cumple el requerimiento de gestionar el mecanismo de re-identificación en un sistema aparte, el cual puede controlarse y asegurarse de manera independiente.

Emplear Servicios de una Plataforma de Gestión de la Privacidad

Un servicio de tokenización, mediante el cual se reemplaza los datos sensibles almacenados por tokens (valores generados aleatoriamente), es un mecanismo alternativo que permite proteger la confidencialidad de la información. Ejemplos de estos servicios son los de Privitar Ltd (2021) y los de *ALTR* (n.d.).

Estos servicios tienen la ventaja de que no se necesita modificar el diseño de los esquemas para implementar un control de acceso diferenciado para los datos sensibles, y permiten todavía un preciso control de auditoría y monitoreo sobre cada solicitud de acceso que se hace a la información sensible. También quitan peso de encima al no tener que implementar los controles desde cero. Finalmente, algunos de estos servicios ofrecen variaciones como la "perturbación de los datos" de manera que se preservan algunas propiedades de los datos sensibles que son útiles para hacer ciertos análisis sobre estos, al tiempo que la información sensible se protege.

La principal desventaja es que tienen un costo adicional e introducen una dependencia externa que media cada acceso a la información sensible, lo cual puede representar un riesgo de disponibilidad de la información que no está presente en el caso de la segregación de los datos sensibles en un esquema aparte. Tampoco permiten establecer controles diferenciados para el

acceso en modo de escritura, por lo que no proveen esta protección a la integridad de la información sensible que sí puede lograrse con la recomendación Segregar los Datos Según su Clasificación.

Recomendaciones para las Bases de Datos

En esta sección se presentan recomendaciones específicamente aplicables al diseño de los esquemas y tablas en las bases de datos y repositorios de información.

Introducir Esquema Confidencial Segregado

Una estrategia con la que se pueden tratar los retos mencionados en la recomendación de Segregar los Datos según su Clasificación, se inspira en ideas tomadas de los patrones y sistemas de Seguridad Multinivel y Seguridad Múltiple de un solo Nivel.

Como se apunta en Seguridad Múltiple de un solo Nivel, en un ambiente de base de datos relacional típicamente se puede recurrir al uso de esquemas de bases de datos para emular múltiples particiones de seguridad pues es esperable que las funciones de control de acceso permitan establecer políticas diferenciadas de acceso para cada esquema. Así, el primer paso consistirá en introducir un nuevo esquema cuyo propósito es funcionar como contenedor de seguridad para los datos clasificados como confidenciales, mientras que el esquema existente se entenderá como uno al que corresponde la información clasificada como no-confidencial.

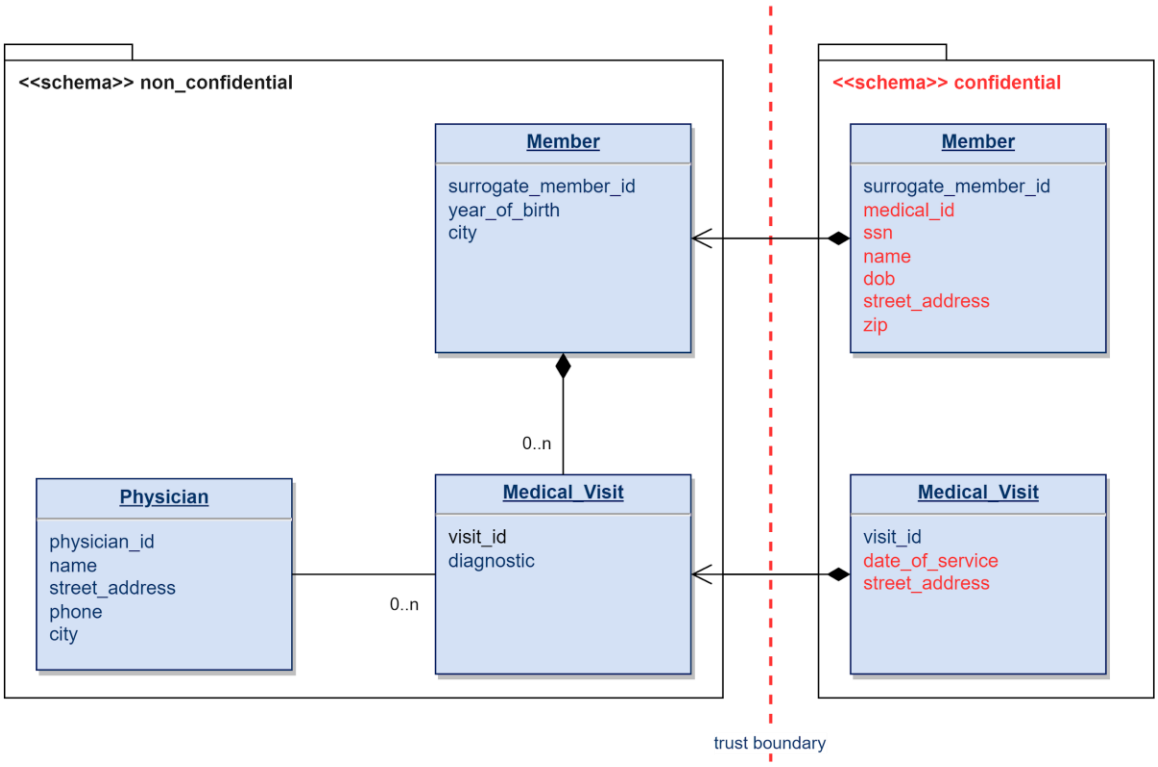
Seguidamente, se aplica la técnica de particionamiento vertical a las columnas identificadas como confidenciales en el esquema original, y se trasladan a una nueva tabla ubicada en el esquema confidencial, utilizando el mismo conjunto de llaves primarias en ambas tablas, de manera que se puede entender que estas se encuentran ligadas entre sí por la llave. Al terminar el proceso con todos los campos confidenciales, no quedará ninguno en el esquema

original. La Figura 13 ilustra cómo queda el diseño de las tablas en cada esquema luego de aplicar las transformaciones.

Posteriormente a esto, se deben introducir varios roles con privilegios específicos de manera que permitan aproximar las reglas de acceso de un sistema de seguridad multinivel.

Figura 13

Diseño de Tablas con Esquema Confidencial Segregado



Nota. Los campos confidenciales, resaltados en color rojo, se ubican únicamente dentro de un esquema específicamente dedicado a información de esta categoría. La línea punteada roja resalta un límite de confianza, en el que diferentes políticas de control de acceso pueden aplicarse a cada lado de manera independiente; el cruce de este límite en una u otra dirección implica un punto de control y verificación. El esquema no-confidencial no tiene información sensible y quienes tengan acceso a este último pueden realizar consultas sin preocuparse de poder extraer información sensible.

Diseño de Roles

Las reglas de control de acceso de Seguridad Multinivel, basadas en el modelo de Bell y La Padula, pueden emularse sobre la base del Esquema Confidencial Segregado explicado

anteriormente. Para el caso en el que solamente se manejan dos niveles de seguridad (p. ej. confidencial y no-confidencial), lo que procede es definir cinco roles de seguridad, como se detalla en la Tabla 7.

Tabla 7

Roles que Emulan el Modelo de Privacidad de Bell y La Padula

Rol	Acceso en Esquema No-Confidencial	Acceso en Esquema Confidencial
NO_CONFIDENCIAL_RO	Solo lectura	Ninguno
NO_CONFIDENCIAL_RW	Lectura y escritura	Ninguno
CONFIDENCIAL_RO	Solo lectura	Solo lectura
CONFIDENCIAL_RW	Solo lectura	Lectura y escritura
CONFIDENCIAL_TRUSTED	Lectura y escritura	Lectura y escritura

Nota. Este conjunto de roles busca emular el modelo de privacidad de Bell y La Padula para un sistema con esquema segregado de 2 niveles (confidencial y no-confidencial). Los roles *NO_CONFIDENCIAL_RO*, *NO_CONFIDENCIAL_RW* y *CONFIDENCIAL_RO* emulan la "propiedad de seguridad simple"; el rol *CONFIDENCIAL_RW* emula la "propiedad-*"; y el rol *CONFIDENCIAL_TRUSTED* emula el acceso que se concede a sujetos y procesos de confianza.

Ejemplos de Aplicación de Roles

En la Tabla 8 se muestran diversos procesos y escenarios para los cuales cada uno de los roles es aplicable como idóneo.

Tabla 8

Ejemplos de Aplicación de Roles

Rol	Proceso o Función
NO_CONFIDENCIAL_RO	Operador Técnico de Soporte
NO_CONFIDENCIAL_RW	Transformación/enriquecimiento de datos, desidentificación de información de salud (con enmascaramiento de campos confidenciales)
CONFIDENCIAL_RO	Usuario procurando contactar a un miembro del plan para coordinar una visita (“necesidad de saber”)
CONFIDENCIAL_RW	Unificación/deduplicación de Identidades
CONFIDENCIAL_TRUSTED	Asignación de Códigos de Identificación, Desidentificación de información de salud

Enmascarar Elementos Confidenciales en la Base de Datos

Muchos motores de bases de datos en la actualidad ofrecen una función generalmente denominada como “enmascaramiento dinámico de datos” (en inglés, “*dynamic data masking*”) (Dalibo SARL SCOP, n.d.; Microsoft Docs Community, 2022; Snowflake Inc, n.d.), mediante la cual se ofusca la información de columnas confidenciales o sensibles cuando es consultada, y solamente se muestra como es a usuarios con ciertos roles.

Por ejemplo, si un usuario con el rol NO_CONFIDENCIAL_RO hace una consulta para extraer los contenidos de la tabla de miembros, los campos confidenciales como el nombre, o los números de identificación personal, no aparecerán en los resultados los valores reales de los campos confidenciales, sino un valor sustituto, por ejemplo “***”. En la Figura 14, se muestra un ejemplo de cómo se define y aplica una política de enmascaramiento en la plataforma de base de datos en la nube *Snowflake*.

Figura 14

Ejemplo de Definición de Política de Enmascaramiento en Snowflake

```
create or replace masking policy name_mask as (val string)
returns string ->
 case
  when is_granted_to_invoker_role('CONFIDENCIAL_RO')
  then val
  else '***'
 end;

alter table if exists confidential.member modify column name
set masking policy name_mask;
```

Nota: Tomado de Snowflake Inc (n.d.).

El enmascaramiento de columnas confidenciales puede considerarse tanto una alternativa como un complemento a la recomendación de Introducir Esquema Confidencial Segregado. Es una alternativa por cuanto permite implementar un control de acceso diferenciado para los datos sensibles, aunque puede que sea menos factible mantener el control de auditoría y monitorización como sí puede hacerse con la segregación; otro control que no puede lograrse con el enmascaramiento es el de la protección de la integridad (si se conceden permisos de escritura sobre los campos no-confidenciales, se estaría concediendo igual sobre los confidenciales, por lo que el usuario o proceso podría modificar el campo confidencial aunque no pueda leerlo).

Aun si se opta por la segregación de datos en su propio esquema aparte, el enmascaramiento puede resultar útil para compensar una debilidad importante de la primera. Considere el caso en el que se descubre que un campo confidencial fue incorrectamente clasificado y agregado al esquema no-confidencial; cuando el error se descubre, debe corregirse cuanto antes, pero de seguro no será posible modificar inmediatamente los esquemas moviendo el campo de uno al otro porque esto estropearía el funcionamiento de las rutinas y consultas que

fueron escritas accediendo al campo en el esquema no-confidencial. En estas circunstancias, probablemente será mucho más factible aplicar una política de enmascaramiento al campo en el esquema no-confidencial, de manera que la privacidad de la información puede protegerse de inmediato, dando tiempo a que se efectúe una migración del campo al esquema confidencial, la cual podría extenderse por semanas o meses, pues dependerá de la velocidad con la que todas las rutinas dependientes del campo puedan actualizarse.

Implementar Proceso para Desidentificar Datos Confidenciales

Como se vio en la sección de Información de Salud Desidentificada, el método del Puerto Seguro puede ser utilizado para derivar, a partir de información de salud protegida, información que ya no se considera protegida por la regulación. Esto quiere decir que la información resultante de aplicar el procedimiento ya no calificaría como información confidencial, por lo cual puede trasladarse con más libertad a otros contextos, tales como un ambiente de pruebas y/o desarrollo, lo cual puede ser sumamente beneficioso para las actividades que se desempeñan en esos ambientes.

Una manera en que se puede implementar este proceso es aplicando primero la recomendación de Enmascarar Elementos Confidenciales en la Base de Datos, se ha de tener cuidado de que las reglas de enmascaramiento estén apegadas a los requisitos del método del Puerto Seguro. De esta manera, se puede correr de manera segura un proceso con el rol NO_CONFIDENCIAL_RW que copie los contenidos del ambiente de producción al ambiente de pruebas; dado que las reglas de enmascaramiento no permitirán al proceso tener acceso a la información real de los campos confidenciales, la copia de los datos resultante en el ambiente de pruebas consistirá en información desidentificada.

Si no se opta por el enmascaramiento de los campos confidenciales, el proceso de desidentificación deberá implementarse como un script que aplique las transformaciones requeridas por el método y almacene los resultados en el ambiente de pruebas. En este caso, el proceso deberá ejecutarse con el rol CONFIDENCIAL_TRUSTED para que le sea posible tanto leer la información del esquema confidencial como escribir en los esquemas del ambiente de pruebas, cuya clasificación es no-confidencial.

Alternativa: Servicios de Generación de Símbolos Identificadores. Como alternativa al empleo del enmascaramiento, que podría no estar soportado de manera nativa por el manejador de la base de datos, se puede emplear un servicio generación de símbolos de identificación (en inglés, *tokenization*), tal como se amplía en la recomendación de Emplear Servicios de una Plataforma de Gestión de la Privacidad. Con esta, las columnas confidenciales se ofuscarían y reemplazarían sus valores reales con los símbolos de identificación generados por el servicio, de manera que nuevamente sería posible correr un proceso no privilegiado que copie los datos de producción al ambiente de pruebas.

Recomendaciones para la Aplicación de Usuario Final

En esta sección se presentan recomendaciones específicas para los componentes de la aplicación web y la interfaz de usuario final.

Enmascarar Elementos Confidenciales en la Interfaz de Usuario

A fin de alinearse con el requerimiento de evitar acceder a la información confidencial a menos que sea necesario, se puede recurrir al enmascaramiento de los campos confidenciales para que, por defecto, sus contenidos no se muestren en pantalla. Si el usuario tiene permisos de acceso a información confidencial, se podrá habilitar un control en la interfaz de usuario para que este solicite desenmascarar los datos, y solamente entonces es que esta información se extraería

para mostrarla en pantalla. Existe la posibilidad de desenmascarar solamente un campo a la vez, o bien desenmascararlos todos, donde la primera es preferible desde el punto de vista de la seguridad; pero para decidir sobre la aplicación de una u otra experiencia de usuario, idealmente deberá hacerse una evaluación de usabilidad.

Si se sigue la recomendación de Enmascarar Elementos Confidenciales en la Base de Datos, la información ya vendría enmascarada desde la base de datos siempre y cuando la conexión a esta última se haga empleando el rol NO_CONFIDENCIAL_RO. Cuando se solicite desenmascarar la información confidencial, será necesario utilizar una conexión diferente a la base de datos para activar el rol CONFIDENCIAL_RO y que esta vez las consultas retornen la información desenmascarada.

Si se sigue la recomendación de Introducir Esquema Confidencial Segregado, no sería posible siquiera incluir los campos confidenciales en la consulta utilizando el rol NO_CONFIDENCIAL_RO, y será la lógica de la interfaz de usuario la responsable de suplir el enmascarado o bien indicar de alguna otra forma que esa información no está inicialmente disponible. Nuevamente, una conexión diferente será necesaria para poder ejecutar la consulta que permita extraer la información confidencial si el usuario la solicita y tiene los permisos necesarios.

Implementar Registro de Auditoría Sobre Acceso a Datos Confidenciales

Dado que las políticas prescriben que se debe llevar un registro de auditoría detallado de cada acceso a la información confidencial, se deben registrar en las bitácoras de las aplicaciones y servicios los eventos en que se accedió a esta información. CheatSheets Series Team (n.d.-a) contiene lineamientos y consideraciones de seguridad sobre cómo registrar adecuadamente estos eventos, tales como los siguientes:

- Dónde registrar la información (considérese enviar el registro a un sistema centralizado de ser posible).
- Qué información registrar (fecha y hora del evento y del registro, identidad del usuario, dirección IP de origen de la interacción, identificador de la interacción, elementos confidenciales accedidos y un identificador interno que permita correlacionar luego quién es el sujeto cuya información se accedió, pantalla/ventana/módulo donde se solicitó y/o mostró la información, entre otros).
- Cómo proteger los registros de la bitácora.

Si se sigue la recomendación de Mantener Segregación a lo Largo del Flujo de Datos, un componente idóneo desde donde se puede ejecutar este registro sería el encargado de atender las solicitudes de información confidencial (que expone la interfaz de acceso a datos confidenciales). Adicionalmente, si se sigue la recomendación de Introducir Esquema Confidencial Segregado, es esperable que pueda llevarse una bitácora de acceso a los datos confidenciales a nivel de la base de datos ya que todos estos, y solo estos, estarán contenidos en un esquema dedicado para tal efecto.

Conclusiones y Recomendaciones

En este capítulo se presentan las conclusiones y recomendaciones derivadas del presente trabajo.

Conclusiones

1. En la literatura académica se reporta una gran cantidad de enfoques, marcos, métodos y técnicas propuestos para la realización de actividades, como parte del ciclo de desarrollo de software, con el fin de mejorar la capacidad del software resultante de resistir ataques y que pueda funcionar correctamente aún en condiciones adversas; es decir, hacerlo seguro. Sin embargo, la posibilidad real de su aplicación efectiva en un contexto dado está condicionada por diversidad de factores tales como el nivel de conciencia del liderazgo técnico y del negocio sobre el valor de atender las consideraciones de seguridad, la disponibilidad de recurso humano debidamente capacitado y la compatibilidad del enfoque con los procesos y dinámicas organizacionales. En contextos donde prevalece el agilismo surgen retos adicionales porque muchos de los enfoques y marcos de desarrollo seguro predominantes tuvieron su origen en el contexto de procesos de desarrollo en cascada con los que los métodos ágiles buscaron precisamente romper, y a la tensión inherente entre los principios de análisis y construcción incrementales, y la necesidad que muchas consideraciones de seguridad tienen de un abordaje holístico y coherente para que sean efectivas.
2. En el contexto del caso bajo estudio, se encuentran condiciones favorables de una organización cuyo liderazgo comprende el valor de la seguridad como parte del giro del negocio (en gran medida debido a la prevalencia de las regulaciones en el

campo) y con programas de seguridad de la información y cumplimiento consolidados, principalmente en las áreas de TI y procesos de negocio, pero con debilidades en el área y procesos de ingeniería del software debido a la carencia de personal capacitado, y brechas de comunicación entre los programas de seguridad y cumplimiento y algunos de los procesos y facetas del desarrollo de software. Las actividades y técnicas por las que La Organización muestra preferencia son primordialmente las compatibles con métodos ágiles; entre estas, se priorizan las de identificación y análisis de requisitos de seguridad (a partir de las políticas de seguridad y cumplimiento), y los patrones de diseño de seguridad.

3. El análisis de las políticas de seguridad permite identificar la necesidad de aplicar controles diferenciados a los activos de información (control de acceso, auditoría, integridad, cifrado, disponibilidad), según su nivel de sensibilidad en el esquema de clasificación de datos. El análisis de la arquitectura base frente a las posibles amenazas que enfrenta señalan que la primera falla en contener la propagación de información confidencial a través de los distintos puntos del flujo de datos de El Producto, lo cual conduce a un área de ataque muy extendido a lo largo del sistema y dificulta o encarece la aplicación efectiva de los controles de seguridad acordes con la clasificación de los datos requeridas por las políticas de seguridad. La revisión de patrones de diseño de seguridad permite identificar la segregación de datos y las reglas de acceso de la seguridad multinivel como soluciones que pueden ayudar a mitigar los riesgos de seguridad identificados.
4. A pesar de la existencia de un programa de seguridad de la información y cumplimiento establecidos, persisten las dificultades para comunicar y difundir

estos requisitos de seguridad a los equipos de desarrollo y los dueños de producto, por lo que algunas de las decisiones de diseño de El Producto presentan vulnerabilidades o dificultades para cumplir con estos requisitos.

5. Si bien las políticas de seguridad son una fuente rica de información para identificar requisitos de seguridad, estas no son idóneas para comunicar los requisitos a los equipos de desarrollo de software ya que contienen también información que no es relevante para este propósito, o bien su redacción y estructura no son familiares para los desarrolladores, lo cual dificulta su lectura y análisis por parte de estos.
6. La segregación de los flujos de datos de El Producto, según su clasificación, permite reducir la exposición de información confidencial, mejorar la aplicación diferenciada de los controles de seguridad (auditoría, integridad, control de acceso), potencialmente reduciendo también el costo de la aplicación de dichos controles.

Recomendaciones

1. Recurrir a las políticas de la organización, particularmente las que emite el programa de seguridad de la información, en búsqueda de requerimientos de seguridad para las aplicaciones.
2. Crear un documento específicamente dirigido a los equipos de desarrollo de software, en el cual se condense los requisitos de seguridad que se derivan de las políticas de seguridad y cumplimiento de La Organización y que hagan referencia a estas.

3. Crear una Comunidad de Práctica (Knaster, 2015) en torno al desarrollo seguro de software para el cultivo y desarrollo continuos de las habilidades de los *Maestros de Seguridad* en los diversos equipos de desarrollo de la organización.
4. Facilitar el entrenamiento y aprendizaje en temas de diseño seguro y codificación segura. Estos temas pueden ser considerados como muy necesarios y apetecibles por parte de los equipos de desarrollo (Oyetoyan et al., 2016). Algunos temas que se pueden recomendar para el entrenamiento de los miembros del equipo de desarrollo, así como recursos asociados, se enumeran en Tabla 9.
5. Otro aspecto que vale la pena rescatar por su efecto potenciador de los procesos de difusión del conocimiento y de generación de ideas que los equipos pueden aplicar, es el facilitar dinámicas como las charlas técnicas, demostraciones y actividades similares de "construcción de comunidad" (Oyetoyan et al., 2016, p. 553).

Tabla 9

Temas Sugeridos para Entrenamiento de Seguridad

Tema	Objetivos	Recursos
Hacking de aplicaciones, Ataques y defensas	Enseñar a los desarrolladores sobre patrones de ataque prácticos y defensas recomendadas en código y configuración. Idealmente, deberían desarrollar un "pensamiento de adversario" (en inglés, *adversarial thinking*) y distinguir las defensas comunes, pero ingenuas o defectuosas, de aquellas que realmente funcionan.[a]	Prácticas en laboratorios y cursos como (PortSwigger, n.d.) (Clarified Security, n.d.), (SANS Institute, n.d.); aplicaciones web vulnerables como (Kimminich, n.d.); otros recursos de aprendizaje en línea como (CheatSheets Series Team, n.d.-b),
Modelado de amenazas y análisis de riesgos de arquitectura	Enseñar a los desarrolladores sobre el análisis de una arquitectura de sistema para buscar riesgos y vulnerabilidades en su diseño y planificar para remediarlos	Cursos como (Synopsys, n.d.), (Security Innovation, 2019), (Hill, n.d.), (Toreon, 2021), (Secura, n.d.); otros recursos de aprendizaje como (Microsoft, n.d.), (OWASP Threat Model Community, n.d.), (Drake, n.d.), (CheatSheets Series Team, n.d.-c), (OWASP Application Security Verification Standard Community, 2019), (OWASP Proactive Controls Community, n.d.).

Nota: los recursos mencionados en esta tabla fueron identificados y seleccionados mediante una búsqueda no-sistemática, y tampoco fueron evaluados para determinar su calidad. Estos recursos se muestran como ejemplos para ilustrar el material que podría ser útil para alcanzar los

objetivos de aprendizaje sugeridos, pero no necesariamente se recomiendan de manera autoritativa.

[a] Tenga en cuenta que "OWASP Top 10" se considera entrenamiento de "nivel de básico o elemental" (OWASP Top 10 team, 2021) y por ende se recomienda procurar abarcar estándares más amplios, como el (OWASP Application Security Verification Standard Community, 2019).

Trabajos a Futuro

El trabajo elaborado hasta ahora consistió en una revisión puntual de algunos de los requisitos de seguridad y parte del diseño de El Producto, pero queda pendiente la cuestión de cómo integrar la realización de dichas actividades al proceso de desarrollo de manera sostenida en el tiempo, especialmente a la luz de reportes como los de Poller et al. (2017), que advierten del hecho de que la preocupación por cuestiones de seguridad puede debilitarse y desvanecerse luego de haber tenido el impulso inicial que incluso contaba con el apoyo del liderazgo organizacional. Aspectos como el que las mismas prácticas y principios de desarrollo ágil pueden desincentivar un adecuado tratamiento de los riesgos de seguridad podrían ser abordados con los expertos en desarrollo ágil de La Organización para crear conciencia sobre los peligros de considerar la seguridad como "una clase más de requisitos cualitativos".

Otra línea de investigación a futuro es la de ampliar el análisis de requisitos de seguridad y del diseño de la arquitectura base a fin de derivar propuestas de mejora adicionales de El Producto. Por ejemplo, dado que el foco del análisis en este trabajo se ha centrado mayoritariamente en la confidencialidad de la información, nuevos análisis en relación con la integridad y su disponibilidad podrían considerarse. Otra forma de ampliar el análisis de seguridad es recurrir a listados adicionales de requisitos de seguridad, tales como (OWASP Application Security Verification Standard Community, 2019), particularmente centrado en requisitos para los componentes de la aplicación web.

Referencias

ALTR. (n.d.). ALTR. Retrieved July 2, 2022, from https://www.altr.com/use-cases/privileged-access

Anwar Mohammad, M. N., Nazir, M. y Mustafa, K. (2019). A Systematic Review and Analytical Evaluation of Security Requirements Engineering Approaches. *Arabian Journal for Science and Engineering*, *44*(11), 8963–8987.

Assistant Secretary for Planning and Evaluation. (n.d.). *Health Insurance Portability and Accountability Act of 1996*. ASPE. Retrieved June 2, 2022, from https://web.archive.org/web/20220518164948/https://aspe.hhs.gov/reports/health-insurance-portability-accountability-act-1996

Ayalew, T., Kidane, T. y Carlsson, B. (2013). Identification and Evaluation of Security Activities in Agile Projects. *Secure IT Systems*, 139–153.

Baca, D., Boldt, M., Carlsson, B. y Jacobsson, A. (2015). A Novel Security-Enhanced Agile Software Development Process Applied in an Industrial Setting. *2015 10th International Conference on Availability, Reliability and Security*, 11–19.

Baca, D. y Carlsson, B. (2011). Agile development with security engineering activities. *Proceedings of the 2011 International Conference on Software and Systems Process*, 149–158.

Bafandeh Mayvan, B., Rasoolzadegan, A. y Ghavidel Yazdi, Z. (2017). The state of the art on design patterns: A systematic mapping of the literature. *The Journal of Systems and Software*, *125*, 93–118.

Barbosa, D. A. y Sampaio, S. (2015). Guide to the Support for the Enhancement of Security Measures in Agile Projects. *2015 6th Brazilian Workshop on Agile Methods (WBMA)*, 25–31.

Bartsch, S. (2011). Practitioners' Perspectives on Security in Agile Development. *2011 Sixth International Conference on Availability, Reliability and Security*, 479–484.

Behutiye, W., Karhapää, P., López, L., Burgués, X., Martínez-Fernández, S., Vollmer, A. M., Rodríguez, P., Franch, X. y Oivo, M. (2020). Management of quality requirements in agile and rapid software development: A systematic mapping study. *Information and Software Technology*, *123*, 106225.

Bell, D. E. y La Padula, L. J. (1976). *Secure computer system: Unified exposition and multics interpretation*. MITRE CORP BEDFORD MA. https://apps.dtic.mil/sti/citations/ADA023588

Bennasar, H., Essaaidi, M., Bendahmane, A. y Ben-Othman, J. (2021). A Systematic Literature Review of Cloud Computing Cybersecurity. *Advances in Dynamical Systems and Applications*, *16*(2), 1883–1919.

Binti Arbain, A. F., Ghani, I. y Wan Kadir, W. M. N. (2014). Agile non functional requirements (NFR) traceability metamodel. *2014 8th. Malaysian Software Engineering Conference (MySEC)*, 228–233.

Boldt, M., Jacobsson, A., Baca, D. y Carlsson, B. (2017). Introducing a Novel Security-Enhanced Agile Software Development Process. *International Journal of Secure Software Engineering* , *8*(2), 26–52.

BSIMM Community. (n.d.). *Software Security Framework - BSIMM*. BSIMM. Retrieved April 18, 2022, from https://web.archive.org/web/20210815222749/https://www.bsimm.com/framework.html

Centers for Medicare & Medicaid Services. (2021). *REPORT TO CONGRESS: RISK ADJUSTMENT IN MEDICARE ADVANTAGE*. Centers for Medicare & Medicaid

Services. https://www.cms.gov/files/document/report-congress-risk-adjustment-medicare-advantage-december-2021.pdf

CheatSheets Series Team. (n.d.-a). *Logging - OWASP Cheat Sheet Series*. OWASP Cheat Sheet Series. Retrieved June 5, 2022, from https://web.archive.org/web/20220424195254/https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html

CheatSheets Series Team. (n.d.-b). *OWASP Cheat Sheet Series*. OWASP Cheat Sheet Series. Retrieved April 18, 2022, from https://web.archive.org/web/20220619020719/https://cheatsheetseries.owasp.org/

CheatSheets Series Team. (n.d.-c). *Threat Modeling Cheat Sheet*. OWASP Cheat Sheet Series. Retrieved April 18, 2022, from https://web.archive.org/web/20220613200142/https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html

Chóliz, J., Vilas, J. y Moreira, J. (2015). Independent Security Testing on Agile Software Development: A Case Study in a Software Company. *2015 10th International Conference on Availability, Reliability and Security*, 522–531.

Ciox Health. (2021, February 12). *What is Risk Adjustment?* Ciox. https://web.archive.org/web/20210612204817/https://www.cioxhealth.com/blog/payers/risk-adjustment-faqs/

Clarified Security. (n.d.). *Web Application Security*. Clarified Security. Retrieved April 18, 2022, from https://web.archive.org/web/20210225024527/https://clarifiedsecurity.com/web-application-security-training/

Conklin, L. y Drake, V. (n.d.). *Threat Modeling Process*. OWASP. Retrieved May 28, 2022, from https://web.archive.org/web/20220502030217/https://owasp.org/www-community/Threat_Modeling_Process

Dalibo SARL SCOP. (n.d.). *PostgreSQL Anonymizer*. PostgreSQL Anonymizer. Retrieved June 16, 2022, from https://web.archive.org/web/20220417135606/https://postgresql-anonymizer.readthedocs.io/en/stable/

Díaz-Rojas, J. A., Ocharán-Hernández, J. O., Pérez-Arriaga, J. C. y Limón, X. (2021). Web API Security Vulnerabilities and Mitigation Mechanisms: A Systematic Mapping Study. *2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 207–218.

Drake, V. (n.d.). *Threat Modeling*. OWASP. Retrieved April 18, 2022, from https://web.archive.org/web/20220428115521/https://owasp.org/www-community/Threat_Modeling

Edemekong, P. F., Annamaraju, P. y Haydel, M. J. (2021). *Health Insurance Portability and Accountability Act*. StatPearls Publishing, Treasure Island (FL).

Franqueira, V. N. L., Bakalova, Z., Tun, T. T. y Daneva, M. (2011). Towards agile security risk management in RE and beyond. *Workshop on Empirical Requirements Engineering (EmpiRE 2011)*, 33–36.

Futcher y Solms. (2012). SecSDM: A usable tool to support IT undergraduate students in secure software development. *HAISA*. https://www.researchgate.net/profile/Lynn-Futcher/publication/258859723_SecSDM_A_usable_tool_to_support_IT_undergraduate_students_in_secure_software_development/links/53d8df530cf2a19eee838c54/SecSDM-A-usable-tool-to-support-IT-undergraduate-students-in-secure-software-development

Ghani, I., Azham, Z. y Jeong, S. R. (2014). Integrating Software Security into Agile-Scrum Method. *KSII Transactions on Internet and Information Systems (TIIS)*, *8*(2), 646–663.

Harrison, W. S., Hanebutte, N. y Oman, P. (2005). The mils architecture for a secure global information grid. *Crosstalk: The Journal of*. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.587.3139&rep=rep1&type=pdf

Hernández, R., Fernández, C. y Baptista, P. (2014). *Metodología de la investigación*. McGraw-Hill Education.

Hill, G. (n.d.). *Threat Modeling*. Infosec. Retrieved April 18, 2022, from https://web.archive.org/web/20220131191634/https://www.infosecinstitute.com/skills/learning-paths/threat-modeling/

Hron, M. y Obwegeser, N. (2022). Why and how is Scrum being adapted in practice: A systematic review. *The Journal of Systems and Software*, *183*, 111110.

Jafari, A. J. y Rasoolzadegan, A. (2020). Security patterns: A systematic mapping study. *Journal of Computer Languages*, *56*, 100938.

Jarmakiewicz, J. y Podlasek, T. (2015). Design and implementation of multilevel security subsystem based on XACML and WEB services. *2015 International Conference on Military Communications and Information Systems (ICMCIS)*, 1–8.

Khan, R. A., Khan, S. U., Ilyas, M. y Idris, M. Y. (2020). The State of the Art on Secure Software Engineering: A Systematic Mapping Study. *Proceedings of the Evaluation and Assessment in Software Engineering*, 487–492.

Khan, R. A., Khan, S. U., Khan, H. U. y Ilyas, M. (2021). Systematic mapping study on security approaches in secure software engineering. *IEEE Access*, *9*, 19139–19160.

Khan, R. A., Khan, S. U., Khan, H. U. y Ilyas, M. (2022). Systematic Literature Review on Security Risks and its Practices in Secure Software Development. *IEEE Access*, *10*, 5456–5481.

Kibbe, D. C. (2001). A problem-oriented approach to the HIPAA security standards. *Family Practice Management*, *8*(7), 37–43.

Kiel, J. M. (2012). HIPAA and its effect on informatics. *Computers, Informatics, Nursing: CIN*, *30*(1), 1–5.

Kimminich, B. (n.d.). *Juice Shop - Insecure Web Application for Training*. OWASP. Retrieved April 18, 2022, from https://web.archive.org/web/20220525120541/https://owasp.org/www-project-juice-shop/

Kitchenham, B. y Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (No. EBSE-2007-01). University of Durham. https://www.researchgate.net/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering

Knaster, R. (2015, June 29). *Communities of Practice*. Scaled Agile Framework. https://web.archive.org/web/20211215221622/https://www.scaledagileframework.com/communities-of-practice/

Luo, J. y Kang, M. (2009). *An Infrastructure for Multi-Level Secure Service-Oriented Architecture (MLS-SOA) Using the Multiple Single-Level Approach*. NAVAL RESEARCH LAB WASHINGTON DC CENTER FOR HIGH ASSURANCE COMPUTING SYSTEMS …. https://apps.dtic.mil/sti/citations/ADA514453

Maier, P., Ma, Z. y Bloem, R. (2017). Towards a Secure SCRUM Process for Agile Web Application Development. *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 1–8.

McCallister, E., Grance, T. y Scarfone, K. A. (2010). *Guide to protecting the confidentiality of Personally Identifiable Information (PII)*. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-122

Microsoft. (n.d.). *Threat Modeling Security Fundamentals Learning Path*. Microsoft Docs. Retrieved April 18, 2022, from https://web.archive.org/web/20220506105602/https://docs.microsoft.com/en-us/learn/paths/tm-threat-modeling-fundamentals/

Microsoft. (2022, February 7). *Data Partitioning Guidance*. Microsoft Docs. https://web.archive.org/web/20220308150745/https://docs.microsoft.com/en-us/azure/architecture/best-practices/data-partitioning

Microsoft Docs Community. (2022). *Dynamic Data Masking* (Version SQL Server 2019). Microsoft. https://web.archive.org/web/20220604210055/https://docs.microsoft.com/en-us/sql/relational-databases/security/dynamic-data-masking?view=sql-server-ver15

Mishra, S. y Khanum, M. A. (2020). A SYSTEMATIC LITERATURE REVIEW ON SECURE SOFTWARE DEVELOPMENT: AGILE PERSPECTIVE. *IJARET*. https://doi.org/10.34218/IJARET.11.12.2020.278

Mohamed, M. A., Challenger, M. y Kardas, G. (2020). Applications of model-driven engineering in cyber-physical systems: A systematic mapping study. *Journal of Computer Languages, 59*, 100972.

Mohammed, N. M., Niazi, M., Alshayeb, M. y Mahmood, S. (2017). Exploring software security approaches in software development lifecycle: A systematic mapping study. *Computer Standards & Interfaces*, *50*, 107–115.

Moneta, F. M. (2018). *State of the art techniques for creating secure software within the Agile process: a systematic literature review* [University OF Bologna]. https://amslaurea.unibo.it/16803/1/main.pdf

Moneypenny, M. (2021, March 30). *HIPAA Need to Know Basis: What, When, How — Etactics*. Etactics | Revenue Cycle Software. https://web.archive.org/web/20220626152311/https://etactics.com/blog/hipaa-need-to-know-basis

Moyón, F., Almeida, P., Riofrío, D., Mendez, D. y Kalinowski, M. (2020). Security Compliance in Agile Software Development: A Systematic Mapping Study. *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 413–420.

Nina, H., Pow-Sang, J. A. y Villavicencio, M. (2021). Systematic Mapping of the Literature on Secure Software Development. *IEEE Access*, *9*, 36852–36867.

Office for Civil Rights (OCR). (2012, September 7). *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule*. HHS.gov; US Department of Health and Human Services. https://web.archive.org/web/20220528215535/https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html

Office for Civil Rights (OCR). (2013, July 26). *Summary of the HIPAA Privacy Rule*. U.S. Department of Health & Human Services.

https://web.archive.org/web/20220524212536/https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html

Office of the Under Secretary of Defense for Intelligence and Security. (2017). *PROCEDURES FOR THE DOD PERSONNEL SECURITY PROGRAM (PSP)* (DODM 5200.02). Department of Defense of the. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodm/520002m.pdf

Onduto, B. (2021). *Gamification of Cyber Security Awareness--A Systematic Review of Games* [University of Turku]. https://www.utupub.fi/bitstream/handle/10024/152929/Onduto_Barack_Thesis_Final.pdf?sequence=1

OWASP Application Security Verification Standard Community. (2019). *OWASP Application Security Verification Standard*. OWASP. https://web.archive.org/web/20220613200137/https://owasp.org/www-project-application-security-verification-standard/

OWASP Proactive Controls Community. (n.d.). *OWASP Proactive Controls*. OWASP. Retrieved June 25, 2022, from https://web.archive.org/web/20220610003045/https://owasp.org/www-project-proactive-controls/

OWASP Threat Model Community. (n.d.). *OWASP Threat Modeling Project*. OWASP. Retrieved April 18, 2022, from https://web.archive.org/web/20220525111942/https://owasp.org/www-project-threat-model/

OWASP Top 10 team. (2021). *How to use the OWASP Top 10 as a standard*. OWASP. https://web.archive.org/web/20220504084607/https://owasp.org/Top10/A00_2021_How_to_use_the_OWASP_Top_10_as_a_standard/

Oyetoyan, T. D., Cruzes, D. S. y Jaatun, M. G. (2016). An Empirical Study on the Relationship between Software Security Skills, Usage and Training Needs in Agile Settings. *2016 11th International Conference on Availability, Reliability and Security (ARES)*, 548–555.

Pereira-Vale, A., Márquez, G., Astudillo, H. y Fernandez, E. B. (2019). Security Mechanisms Used in Microservices-Based Systems: A Systematic Mapping. *2019 XLV Latin American Computing Conference (CLEI)*, 01–10.

Petersen, K., Vakkalanka, S. y Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, *64*, 1–18.

Poller, A., Kocksch, L., Türpe, S., Epp, F. A. y Kinder-Kurlanda, K. (2017). Can Security Become a Routine? A Study of Organizational Change in an Agile Software Development Group. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2489–2503.

PortSwigger. (n.d.). *Web Security Academy: Free Online Training from PortSwigger*. PortSwigger. Retrieved April 18, 2022, from https://web.archive.org/web/20220619161423/https://portswigger.net/web-security

Pothamsetty, V. (2005). Where security education is lacking. *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development*, 54–58.

Privitar Ltd. (2021, April 30). *Privacy Techniques*. Privitar. https://web.archive.org/web/20210616084059/https://www.privitar.com/data-privacy/privacy-techniques/

Riihelä, L. (2019). *Teaching information security: A systematic mapping study* [LUT University]. https://lutpub.lut.fi/bitstream/handle/10024/159739/diplomityo_riihela_lassi.pdf?sequence=1

Rozanski, N. y Woods, E. (2012). *Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives* (2nd ed.). Addison-Wesley.

Saldanha, L. R. y Zorzo, A. (2019). Security requirements in agile software development: a systematic mapping study. *Pontifical Catholic University of Rio Grande Do Sul*. https://www.pucrs.br/politecnica/wp-content/uploads/sites/166/2019/07/Technical_Report_087-Leandro_Ripoll_Saldanha.pdf

Salin, H. y Lundgren, M. (2022). Towards Agile Cybersecurity Risk Management for Autonomous Software Engineering Teams. *Journal of Cybersecurity and Privacy*, *2*(2), 276–291.

SAMM Community. (n.d.-a). *SAMM Agile Guidance*. SAMM. Retrieved April 18, 2022, from https://web.archive.org/web/20220425013751/https://owaspsamm.org/guidance/agile/

SAMM Community. (n.d.-b). *The Model - SAMM*. SAMM. Retrieved April 18, 2022, from https://web.archive.org/web/20220503123629/https://owaspsamm.org/model/

SANS Institute. (n.d.). *Application Security: Securing Web Applications, APIs, and Microservices*. SANS. Retrieved April 18, 2022, from

https://web.archive.org/web/20220626154811/https://www.sans.org/cyber-security-courses/application-security-securing-web-apps-api-microservices/

Savola, R. M., Frühwirth, C. y Pietikäinen, A. (2012). Risk-Driven Security Metrics in Agile Software Development - An Industrial Pilot Study. *JUCS - Journal of Universal Computer Science*, *18*(12), 1679–1702.

Secura. (n.d.). *Threat Modeling Training*. Secura. Retrieved April 18, 2022, from https://web.archive.org/web/20210419161058/https://www.secura.com/services/people/training-courses/threat-modeling-training

Security Innovation. (2019, January 29). *DES 212 - Architecture Risk Analysis & Remediation*. Security Innovation. https://web.archive.org/web/20210419130600/https://www.securityinnovation.com/course-catalog/architecture-risk-analysis-remediation/

Seh, A. H., Zarour, M., Alenezi, M., Sarkar, A. K., Agrawal, A., Kumar, R. y Khan, R. A. (2020). Healthcare Data Breaches: Insights and Implications. *Healthcare (Basel, Switzerland)*, *8*(2). https://doi.org/10.3390/healthcare8020133

Shukla, A., Katt, B., Nweke, L. O., Yeng, P. K. y Weldehawaryat, G. K. (2021). System Security Assurance: A Systematic Literature Review. *arXiv Preprint arXiv:2110.01904*.

Snowflake Inc. (n.d.). *Using Dynamic Data Masking — Snowflake Documentation*. {Snowflake Inc}. Retrieved June 5, 2022, from https://web.archive.org/web/20210724022933/https://docs.snowflake.com/en/user-guide/security-column-ddm-use.html

Synopsys. (n.d.). *Software Architecture Risk Analysis - Application Security Course*. Synopsys. Retrieved April 18, 2022, from

https://web.archive.org/web/20210625043620/https://www.synopsys.com/software-integrity/training/software-security-courses/software-architecture-risk-analysis.html

Toreon. (2021, October 11). *Threat Modeling training or Whiteboard Hacking training*. Toreon. https://web.archive.org/web/20220626153125/https://www.toreon.com/threat-modeling-training/

Tuma, K., Calikli, G. y Scandariato, R. (2018). Threat analysis of software systems: A systematic literature review. *The Journal of Systems and Software*, *144*, 275–294.

Van Den Berghe, A., Scandariato, R., Yskout, K. y Joosen, W. (2017). Design notations for secure software: a systematic literature review. *Software & Systems Modeling*, *16*(3), 809–831.

Venson, E., Guo, X., Yan, Z. y Boehm, B. (2019). Costing Secure Software Development: A Systematic Mapping Study. *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 1–11.

Villamizar, H., Kalinowski, M., Viana, M. y Fernández, D. M. (2018). A Systematic Mapping Study on Security in Agile Requirements Engineering. *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 454–461.

Washizaki, H., Xia, T., Kamata, N., Fukazawa, Y., Kanuka, H., Kato, T., Yoshino, M., Okubo, T., Ogata, S., Kaiya, H., Hazeyama, A., Tanaka, T., Yoshioka, N. y Priyalakshmi, G. (2021). Systematic Literature Review of Security Pattern Research. *Information. An International Interdisciplinary Journal*, *12*(1), 36.

Wassermann, R. y Cheng, B. H. C. (2003). Security patterns. *Michigan State University, PLoP Conf. Citeseer*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.7818&rep=rep1&type=pdf

Whitehurst, S. (2016, December 14). *Risk Adjustment Optimization From All Angles*. Health Fidelity, an Edifecs Company. https://web.archive.org/web/20201125221834/https://healthfidelity.com/blog/360-degree-risk-adjustment-framework-optimization/

Wolter, A. (2021, February 3). *Security: The Need-to-know principle*. Microsoft Tech Community. https://web.archive.org/web/20211205180922/https://techcommunity.microsoft.com/t5/azure-sql-blog/security-the-need-to-know-principle/ba-p/2112393

Woody, C. y Ellison, R. (2020). *ATTACK SURFACE ANALYSIS: Reduce System and Organizational Risk.* CARNEGIE-MELLON UNIV PITTSBURGH PA. https://apps.dtic.mil/sti/citations/AD1110322

Wysoker, A. (2003). HIPAA and Psychiatric Nurses. *Journal of the American Psychiatric Nurses Association*, *9*(5), 173–175.

Zhu, J., Lipford, H. R. y Chu, B. (2013). Interactive support for secure programming education. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 687–692.

Apéndice 1

Revisión Sistemática de Estudios Secundarios en Desarrollo Seguro de Software

El primer paso preliminar para acercarse a la comprensión del estado de la cuestión en el campo de interés es llevar a cabo una exploración de mapeos y revisiones sistemáticos existentes que arrojen luz sobre: a) las preguntas de investigación en el campo de interés que ya se han abordado, b) si se ha intentado establecer con anterioridad el estado del arte en relación con los enfoques que garanticen o mejoren la seguridad del software desde su etapa de requerimientos y diseño, c) cuáles son los métodos, técnicas y herramientas identificados por esos estudios, e) en qué espacios se han publicado los estudios más importantes en el campo de interés y, f) cuáles son los métodos y parámetros con los que se han realizado con éxito estos estudios secundarios. Todo lo anterior servirá como fundamento para determinar si las revisiones sistemáticas existentes permitirán establecer adecuadamente el estado de la cuestión de interés particular para el presente trabajo de investigación, o en su defecto proceder a desarrollar una para tal efecto.

Para realizar esta exploración, se seguirán los lineamientos ya establecidos para estudios de mapeo sistemático y revisión sistemática recogidos en (Kitchenham y Charters, 2007) y (Petersen et al., 2015), siguiendo como ejemplos de su aplicación los trabajos de (Mohammed et al., 2017) y (Khan et al., 2021). Si bien esta exploración a realizar no es estrictamente un mapeo sistemático, dichos lineamientos se consideran útiles también en este caso por ayudar a establecer un protocolo para la realización de la exploración de manera más sistemática y objetiva.

Preguntas de Investigación

En la Tabla 10 se presentan las preguntas de investigación y sus motivaciones. Los resultados de estas preguntas serán usados para guiar una investigación posterior sobre la

intersección que pueda haber entre los campos de la seguridad del software y la ingeniería guiada por modelos.

Tabla 10

Preguntas de Investigación

Identificador	Pregunta	Motivación
RQ1	¿Qué mapeos y revisiones sistemáticas se han realizado en los campos de la ingeniería de la seguridad del software?	El propósito de esta pregunta es tratar de identificar los trabajos de investigación que han analizado de manera amplia cuál es el estado del arte en el campo de la ingeniería de seguridad del software. A fin de contestar esta pregunta, se analizará la literatura con base en las siguientes preguntas más específicas:
RQ1.1	¿Cuáles son los enfoques, métodos, técnicas y herramientas que se han estudiado, aplicables a las fases de establecimiento de requerimientos y de diseño, y que se consideran más efectivos para garantizar o mejorar la seguridad del software?	Para explorar los métodos existentes en la literatura relacionados con la seguridad en el desarrollo del software.
RQ1.2	¿Cuáles son los enfoques y métodos que se han estudiado para la medición y verificación de la seguridad del software y del cumplimiento de sus requerimientos de seguridad?	Para explorar los métodos existentes en la literatura relacionados con la medición y verificación de los atributos de seguridad del software.

Estrategia de Búsqueda

La búsqueda se lleva a cabo mediante una frase de búsqueda compuesta de términos clave, y ejecutada contra varias bibliotecas digitales y motores de búsqueda.

En este caso, de hecho se derivan dos frases de búsqueda separadas, una enfocada en el área de la seguridad en el desarrollo de software, y la otra en el área de la ingeniería guiada por modelos. En ambos casos, frases de búsqueda derivadas, así como las bases de datos a utilizar

para las búsquedas, fueron validados con aquellos utilizados en otros estudios tales como (Khan et al., 2021) y (Mohamed et al., 2020).

Frase de Búsqueda. Siguiendo los lineamientos indicados, se determina primeramente la estructura PICOC (acrónimo en inglés para "población, intervención, comparación, resultado y contexto") de la frase de búsqueda es la siguiente:

Población. Aplicaciones web y microservicios.

Intervenciones. El desarrollo seguro de software, particularmente en lo concerniente a las actividades de análisis de requerimientos y de diseño.

Comparación. Este aspecto no es relevante para la búsqueda.

Resultado. Métodos y enfoques para el desarrollo seguro de software.

Contexto. Revisiones sistemáticas de literatura o mapeos sistemáticos publicados en los últimos 5 años (2017 a principios de 2022).

A partir de los términos en la estructura *PICOC* y una validación preliminar con varias búsquedas en bases de datos académicas, se identifican los siguientes términos y sinónimos para utilizar en la frase de búsqueda:

Revisiones de Literatura o Mapeos Sistemáticos. (intitle:"systematic review" OR intitle:"systematic literature review" OR intitle:"mapping study" OR intitle:"systematic mapping")

Aplicaciones Web y Microservicios. ("web application*" OR "web system*" OR "microservice*" OR "micro-service*")

Desarrollo Seguro de Software, Requerimientos y Diseño. ("secure development" OR "secure software development)" AND (design OR requirements)

Frase de Búsqueda Resultante. (intitle:"systematic review" OR intitle:"systematic literature review" OR intitle:"mapping study" OR intitle:"systematic mapping") AND ("web application*" OR "web system*" OR "microservice*” OR "micro-service*") AND ("secure development" OR "secure software development)" AND (design OR requirements)

Selección de Fuentes. Para la identificación de las fuentes sobre las cuales se realizará la búsqueda, se optó por utilizar aquellas que se han observado en otros estudios como los de (Khan et al., 2021) y (Mohammed et al., 2017), así como otros sugeridos por los supervisores de este trabajo:

- *Google Scholar*: https://scholar.google.com/
- *IEEE Xplore*: https://ieeexplore.ieee.org/Xplore/home.jsp
- *ACM Digital Library*: https://dl.acm.org/
- *Science Direct*: https://www.sciencedirect.com/
- *SpringerLink*: https://link.springer.com/
- *Wiley Online Library*: https://onlinelibrary.wiley.com/

La Tabla 11 muestra las cadenas de búsqueda utilizadas y los resultados encontrados.

Tabla 11

Cadenas de Búsqueda por Biblioteca Digital

Biblioteca Digital	Cadena de Búsqueda	Resultados
Seguridad en el Desarrollo de Software		
Google Scholar	(intitle:"systematic review" OR intitle:"systematic literature review" OR intitle:"mapping study" OR intitle:"systematic mapping") ("secure development" OR "secure software development") (design OR requirements) ("web application" OR "web applications" OR "web system" OR "web systems" OR "microservice" OR "microservices" OR "micro-service" OR "micro-services")	38
IEEE Xplore	("Document Title":"systematic review" OR "Document Title":"systematic literature review" OR "Document Title":"mapping study" OR "Document Title":"systematic mapping") AND ("secure development" OR "secure software development") AND (design OR requirements)[a]	5
ACM Digital Library	[[Title: "systematic review"] OR [Title: "systematic literature review"] OR [Title: "mapping study"] OR [Title: "systematic mapping"]] AND [[All: "secure development"] OR [All: "secure software development"]] AND [[All: "web application*"] OR [All: "web system*"] OR [All: "microservice*"] OR [All: "micro-service*"]] AND [Publication Date: (01/01/2017 TO 02/28/2022)]	4
Science Direct	("secure development" OR "secure software development") (design OR requirements) ("web application" OR "web system" OR microservice OR "micro-service"); Title: "systematic review" OR "systematic literature review" OR "mapping study" OR "systematic mapping"; Year: 2017-2022	5
SpringerLink	("secure development" OR "secure software development") AND ("web application" OR "web system" OR microservice OR "micro-service"); title: systematic; date: 2017-2022[b]	4

Wiley Online Library	"(micro-service* OR microservice* OR "web application*" OR "web system*") (design OR requirements) ("secur* development" OR "secur* software development")" anywhere and ""systematic review" OR "systematic literature review" OR "mapping study" OR "systematic mapping"" in Title	0
Total		56

[a] Se elimina el requerimiento de que los resultados contengan los términos “web application” OR “microservice” a fin de no excluir algunos resultados que se consideran relevantes; de lo contrario, solamente 1 publicación estaría incluida en los resultados.

[b] No fue posible expresar en el formulario de búsqueda avanzada la restricción de que el título incluyera también las diversas frases para identificar revisiones y mapeos sistemáticos, por lo que se optó por usar un término general que podría devolver resultados que no son de interés, pero que luego se descartaron aplicando los criterios de exclusión establecidos.

Selección de Estudios

En esta etapa, se toman los resultados de las búsquedas de la etapa anterior y se inspeccionan para determinar si cumplen con criterios de inclusión y exclusión que permitan establecer si son relevantes o no para ayudar a responder las preguntas de investigación planteadas; así mismo, se identifican y eliminan estudios duplicados.

Criterios de Inclusión. Se plantea los siguientes criterios de inclusión:

IC1. La publicación explora el tema del desarrollo seguro de software, particularmente de aplicaciones web y/o microservicios, bajo un esquema convencional propietario (es decir, no es desarrollo de código abierto).

IC2. La publicación es un estudio terciario de mapeo sistemático.

IC3. La publicación logra identificar enfoques, técnicas y herramientas que son o podrían ser aplicables en las etapas de análisis de requerimientos y de diseño; o bien enfoques, técnicas y herramientas para verificar, medir y/o evaluar su estado.

Criterios de Exclusión. Se plantea los siguientes criterios de exclusión:

EC1. Publicaciones que no provienen de revistas académicas (*journals*), conferencias, talleres o simposios especializados.

EC2. Investigaciones que no pertenecen al campo de la ingeniería del software.

EC3. Publicaciones que consisten meramente de resúmenes o abstracts de talleres.

EC4. Libros, páginas web y artículos de revistas no académicas (*magazines*).

EC5. Estudios duplicados (esto es, aquellos que aparecen múltiples veces y que reportan resultados del mismo estudio original; solamente será seleccionada la versión más completa de la publicación, o la primera publicación del estudio original).

Resultados de la Selección. Teniendo en cuenta los criterios de inclusión y exclusión citados anteriormente, se hizo la lectura del título, el abstract, las conclusiones y, cuando fuese necesario también, los resultados de cada una de las 56 publicaciones resultantes del proceso de búsqueda. Como resultado de este proceso, se seleccionaron 25 publicaciones.

Resultados

Las 25 publicaciones seleccionadas fueron agrupadas en 6 grupos temáticos: estudios sobre ingeniería de software seguro en general, la seguridad en el contexto de los métodos de desarrollo ágiles, la ingeniería de requerimientos de seguridad, el diseño seguro de software, conceptos de seguridad aplicables a tipos específicos de software (microservicios, servicios web, computación en la nube) y otros temas. La Tabla 12 muestra cómo se asignan las publicaciones a cada grupo. Estos estudios servirán de punto de partida para identificar, utilizando la técnica de

bola de nieve (en inglés, “snowballing”), estudios primarios de relevancia para la presente investigación.

Tabla 12

Publicaciones Seleccionadas por Grupo Temático

Identificador	Estudios
Ingeniería de software seguro (*secure software engineering*)	(Khan et al., 2022), (Shukla et al., 2021), (Khan et al., 2021), (Nina et al., 2021), (Khan et al., 2020), (Mohammed et al., 2017)
Métodos ágiles y seguridad del software	(Hron y Obwegeser, 2022), (Mishra y Khanum, 2020), (Moyón et al., 2020), (Behutiye et al., 2020), (Moneta, 2018)
Ingeniería de requerimientos de seguridad (*security requirements engineering*)	(Anwar Mohammad et al., 2019), (Saldanha y Zorzo, 2019), (Villamizar et al., 2018)
Diseño de software seguro	(Washizaki et al., 2021), (Jafari y Rasoolzadegan, 2020), (Tuma et al., 2018), (Van Den Berghe et al., 2017), (Bafandeh Mayvan et al., 2017)
Seguridad aplicable a tipos específicos de software	(Díaz-Rojas et al., 2021), (Bennasar et al., 2021), (Pereira-Vale et al., 2019)
Otros	(Onduto, 2021), (Riihelä, 2019), (Venson et al., 2019)