Universidad de Buenos Aires
Facultades de Ciencias Económicas,
Cs. Exactas y Naturales e Ingeniería

Maestría en Seguridad Informática.

Tesis.

Gestión de la Seguridad Informática en Procesos de Desarrollo de Software a Medida.

Autor: Karin Xiomara Marroquín Ortiz.

Director de Tesis: Ing. Jorge Luis Ceballos.

Año 2013

Cohorte 2010

A Dios que día a día me dio la sabiduría,

la fortaleza y la constancia necesarias

para poder culminar este proyecto.

A Gloria, Pedro, Harold, Fabricio y Fabián,

por su apoyo, confianza, paciencia,

oraciones, energía y cariño.

DECLARACIÓN JURADA

“Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Tesis vigente y que se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual”.

FIRMADO:

Karin Xiomara Marroquín Ortiz.

DNI. 94568240

RESUMEN

La presente investigación parte de la necesidad de establecer una estrategia que facilite la incorporación de controles de seguridad en el desarrollo de aplicaciones informáticas a medida, dadas ciertas falencias en los procesos de desarrollo aplicados, así como las diversas amenazas contra la información, activo principal de las empresas, en esta época de creciente globalización e interconectividad.

Por lo anterior y considerando la seguridad informática como una característica principal que deben atender las aplicaciones, se diseña y elabora una guía de implementación de controles de seguridad, a partir del análisis de estándares internacionales, de metodologías y marcos de trabajo propuestos por la industria, así como de la visión y experiencia de profesionales y analistas en desarrollo de software. Dicha guía cuenta con una base conceptual y una estructura de actividades que ayudan a identificar cuáles controles de seguridad informática incorporar en cada etapa del desarrollo, así como la forma de implementarlos.

En el desarrollo de este trabajo se han tenido en cuenta conceptos de metodología de investigación, caracterización de procesos y usabilidad.

Adicionalmente, para la evaluación de la guía propuesta, se realiza un proceso de validación con la participación de profesionales, consultoras de desarrollo de software y organizaciones públicas, cuyos resultados son analizados, para finalmente conformar una propuesta sistemática y presentar las conclusiones generales del proyecto.

Palabras Claves: Controles de seguridad informática, Desarrollo de software seguro, Gestión de riesgos, Gestión de seguridad informática, Modelos de capacidad y madurez.

TABLA DE CONTENIDO

1. AGRADECIMIENTOS

La realización del presente trabajo de investigación, requirió de bastante esfuerzo y dedicación, así como de la colaboración, proactiva y generosa de colegas, profesionales y amigos.

Agradezco inmensamente la colaboración, gestión y confianza del Ing. Esp. Jorge Luis Ceballos, director de la tesis, quién con su profesionalismo y esfuerzo, supo guiar y ayudar a llevar a buen término esta investigación.

Reconozco y agradezco especialmente por el tiempo invertido, los excelentes aportes y los consejos profesionales, a mi colega y amiga la Ing. Carolina López, quien amablemente contribuyo a esta investigación, analizando desde otra perspectiva temas referentes a calidad y procesos.

De igual manera quiero resaltar y agradecer profundamente la participación en el proyecto, y el apoyo de mis amigos y colegas, Pablo Folgar y Ana María Cortázar.

Muchas gracias a la Mg. Patricia Scalzone de VEMN, a Alejandro Páez de CENSYS, Nicolas Garrido del Gobierno de la Provincia de Neuquén, Claudia Ghisolfi de ANSES, Carlos Fontela de la UBA, y al personal de everis Argentina, por el tiempo, disposición e interés en participar en la entrevista y en el proceso de validación del presente trabajo.

Gracias a todas las personas que con su apoyo, confianza y palabras de aliento, ayudaron a la realización y culminación de este proyecto.

2. INTRODUCCIÓN.

Teniendo en cuenta que actualmente la información es uno de los principales activos de las organizaciones que más está expuesta a amenazas generadas por la globalización y la interconectividad, protegerla se ha convertido en objetivo fundamental de la tecnología.

La seguridad de la información se define como "la protección de la información de una amplia variedad de amenazas, con el objeto de asegurar la continuidad del negocio, minimizar los riesgos, maximizar el retorno de la inversión e incrementar las oportunidades de negocio".[1]

Aunque actualmente las consultoras de desarrollo de software a medida son más conscientes de la importancia de la seguridad en las aplicaciones que construyen, en muchos casos ésta termina siendo un requerimiento no funcional, que se aborda en etapas tardías del ciclo de vida, mediante la implementación de controles reactivos que tienen un impacto considerable en tiempo y costos.

Así mismo las estadísticas de ataques, vulnerabilidades explotadas y denegaciones de servicio tanto en aplicaciones de uso masivo como en aplicaciones de propósito específico y desarrollos a medida[2], demuestran que aún existen muchos riesgos que analizar y evaluar.

Por consiguiente, se plantea la necesidad de analizar cómo incorporar en el desarrollo de software a medida, en las diferentes etapas de su ciclo de vida, los controles de seguridad sugeridos por normas y estándares de la

[1] Concepto tomado de la definición de Seguridad de la Información propuesta en el estándar ISO/IEC 27002:2005. [7].
[2] Ver Boletín de estadísticas de seguridad: Kaspersky Security Bulletin 2012. The overall statistics for 2012. [16].

industria con el menor impacto posible en tiempo y costos, y con resultados que soporten los objetivos estratégicos de las organizaciones.

Esta investigación busca proponer estrategias y tácticas para facilitar la incorporación de controles de seguridad informática en procesos de desarrollo de software, agrupándolas en una guía de implementación, la cual de aquí en adelante se referencia como GICSI: Guía de incorporación de controles de seguridad informática.

Una guía de implementación que se ajuste a las características particulares de los proyectos de consultoría de software, que tenga en cuenta controles técnicos como de gestión, y que esté basada en estándares y modelos actuales de desarrollo seguro, ayuda a agilizar la adopción de conocimientos sobre la materia por parte del equipo de desarrollo de software, facilita la identificación de los controles necesarios para reducir los riesgos de seguridad en las aplicaciones y sirve como soporte para la evaluación del cumplimiento de normas y estándares. De esta manera actúa como elemento vinculante entre los conocimientos requeridos, las necesidades del negocio y las exigencias regulatorias, aspectos importantes a tener en cuenta en la incorporación de seguridad en las aplicaciones informáticas.

2.1 ESTRUCTURA DEL TRABAJO.

En el presente documento se relacionan brevemente las normas, metodologías y marcos de trabajo que sirven como base conceptual para el desarrollo de esta investigación; así mismo se referencian las herramientas de recopilación de información utilizadas.

Posteriormente se describe el proceso realizado para construir la GICSI, para después profundizar en cada una de las etapas que lo componen.

Es así como en las secciones posteriores se presenta el diseño y los resultados de la entrevista realizada.

Luego y atendiendo el proceso de elaboración mencionado, se documentan las etapas de diseño y construcción, para dar paso al contenido completo de la GICSI.

Finalmente se describe el proceso establecido para la validación de la guía, así como los resultados obtenidos.

2.2 METODOLOGÍA. [3]

Este trabajo se aborda como un estudio exploratorio[4] de las diferentes normas, marcos de trabajo y metodologías relacionadas con la seguridad informática y el desarrollo de software; así como de las opiniones que desarrolladores de diferentes perfiles tienen sobre la aplicación de estándares, políticas y controles de seguridad en su labor diaria.

A partir del análisis de esta información, se establece la estructura de la guía, para posteriormente relacionar los diferentes modelos estudiados, con los controles establecidos como mejores prácticas, y de esta manera proponer procesos orientados a la incorporación de dichas prácticas, en cada una de las partes de implementación correspondientes al ciclo de vida del software.

Para efectos de validación del trabajo desarrollado, se definen criterios y se realizan pruebas de campo en proyectos de desarrollo de software a medida, que permiten evaluar la pertinencia y aplicabilidad del mismo.

[3] En el capítulo 5 del presente documento, se describe el proceso de elaboración de la guía de implementación, siguiendo la metodología aquí establecida.
[4] Concepto tomado del libro Metodología de la Investigación: “Los estudios exploratorios se efectúan, normalmente, cuando el objetivo es examinar un tema o problema de investigación poco estudiado”. [5].

2.3 ALCANCES Y LIMITACIONES.

La GICSI está orientada a aplicarse en proyectos de desarrollo de aplicaciones web a medida, implementados por pequeñas y medianas empresas de consultoría de desarrollo de software, lo cual no impide que sea utilizada en el desarrollo de otro tipo de aplicaciones.

La metodología de entrevista utilizada se aplica a una muestra no probabilística de sujetos-tipo, y tiene como objetivo recopilar percepciones en la fase exploratoria de la investigación, por lo anterior no se pretende que de los resultados obtenidos se infieran conclusiones finales respecto a la forma en la que las pequeñas y medianas consultoras de desarrollo de software, implementan la seguridad informática en sus aplicaciones.

De los procesos del modelo del ciclo de vida del software propuesto por la norma ISO/IEC 12207[5], la GICSI tiene como alcance el proceso de Implementación del Software, por lo que no cubre procesos previos de gestión, o procesos posteriores de despliegue y mantenimiento.

[5] ISO/IEC-12207:2008- Modelo del ciclo de vida del software. [9].

3. OBJETIVOS.

3.1 OBJETIVO GENERAL.

Establecer una guía de implementación de controles de seguridad informática que facilite la aplicación de estándares y buenas prácticas de desarrollo seguro, en proyectos de construcción de aplicaciones web a medida.

3.2 OBJETIVOS ESPECÍFICOS.

- Identificar las principales causas posibles de la baja adopción de prácticas de desarrollo seguro de aplicaciones, en los procesos de construcción de software.

- Evaluar en qué medida las propuestas actuales de desarrollo seguro de aplicaciones informáticas, dictadas por las tendencias de la industria TIC[6], contemplan diferentes tipos de proyectos y el cumplimiento de normas y estándares.

- Identificar puntos claves de control de seguridad informática en los procesos de desarrollo de software a medida, a partir del análisis de metodologías, y estándares que contemplen aspectos de seguridad en esta área.

[6] TICS: Tecnologías de la información y las comunicaciones.

- Establecer lineamientos para la implementación de controles, que permitan mitigar los riesgos de seguridad informática que puedan presentarse durante las diferentes etapas de los procesos de desarrollo de software a medida, tomando en cuenta las restricciones analizadas en el primer ítem. (Posibles causas de la baja adopción de prácticas de desarrollo seguro de aplicaciones).

4. MARCO TEÓRICO.

El presente trabajo se realiza considerando conceptos, normas, metodologías y marcos de trabajo relacionados con la seguridad informática y el proceso de desarrollo de software, los cuales se describen a continuación.

4.1 SEGURIDAD INFORMÁTICA

La seguridad informática se define como las medidas y controles que garantizan la confidencialidad, integridad y disponibilidad de los activos del sistema de información, incluyendo hardware, software, firmware, así como de la información que se procesa, almacena y comunica. [7]

4.2 GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN.

4.2.1 ISO/IEC 27002:2005 – Código para la práctica de la gestión de la seguridad de la información

La fuente primaria para la identificación de los principales controles de seguridad de la información es la norma ISO/IEC 27002:2005 – Código para la práctica de la gestión de la seguridad de la información, la cual define las condiciones y características en las que debe implementarse, operarse y

[7] Concepto tomado del Glosario de términos claves de seguridad de la información – NIST [26]

mantenerse un sistema de gestión de la seguridad de la información en las empresas.

En el diseño de la entrevista como en la elaboración de la guía, se tienen en cuenta lineamientos generales propuestos por la norma, así como los siguientes objetivos de control y sus controles asociados:

- Gestión de activos – Clasificación de la información.
- Gestión de comunicaciones y operaciones – Supervisión.
- Control de acceso.
- Adquisición, desarrollo y mantenimiento de los sistemas.
- Cumplimiento.

4.2.2 Objetivos de control para la información y la tecnología relacionada. – COBIT.

Otra de las fuentes relevantes para la elaboración de la guía, es el marco de referencia COBIT, creado por el IT Governance Institute en 1996, con el fin de aportar herramientas que permitan a los dirigentes, auditores y usuarios, gestionar adecuadamente los recursos y riesgos, así como gobernar y controlar todos los aspectos relacionados a las tecnologías de la información, en los diferentes tipos de organizaciones.

COBIT sugiere determinar las actividades y los riesgos a administrar clasificándolos en dominios, en cada dominio se definen los procesos de TI correspondientes, así como los objetivos de control que van a permitir realizar seguimiento y control efectivo de cada uno de dichos procesos.[8]

Estos dominios corresponden a:

- Planear y organizar.

[8] La información de esta sección es recopilada del marco de trabajo COBIT 4.1. [15].

- Adquirir e implementar.
- Entregar y dar soporte.
- Monitorear y evaluar.

En la guía de implementación se hace referencia a los controles que deben implementarse en las aplicaciones que soportan los procesos de negocio, así como también a controles genéricos de los diferentes dominios.

4.2.3 ISO/IEC 27034:2011 – Tecnologías de la información – Técnicas de Seguridad – Seguridad en aplicaciones.[9]

El estándar ISO/IEC 27034, se encuentra actualmente en desarrollo, en noviembre de 2011 fue liberada la primera parte correspondiente a generalidades y conceptos, en ésta se introducen definiciones, principios y procesos relacionados con la seguridad en aplicaciones.

El objetivo de esta norma es proveer una guía para asistir a las organizaciones en la integración de seguridad en los procesos usados para la gestión de sus aplicaciones.

En la parte 1 de la norma se presentan los siguientes principios, que sirven como base para su desarrollo:

- La seguridad debe manejarse como un requerimiento, como lo son la calidad, la usabilidad y la funcionalidad. Los requerimientos de seguridad deben ser dirigidos, analizados y gestionados para cada etapa del ciclo de vida de la aplicación.
- La seguridad de una aplicación depende del contexto; el tipo y alcance de los requerimientos de seguridad están determinados por los riesgos a los cuales la aplicación está sujeta, lo cual depende de tres contextos: Contexto regulatorio, contexto de negocio y contexto tecnológico.

[9] La información presentada es extraída de la norma ISO/IEC 27034:2011 Part 1: Overview and concepts. [10]

- Inversión apropiada en la seguridad de las aplicaciones. Los costos de implementar controles de seguridad en las aplicaciones y de realizar las auditorías correspondientes debe estar acorde con el nivel de confianza esperado; dicha inversión reduce costos, responsabilidades del dueño de la aplicación y consecuencias legales de violaciones de seguridad.

- La seguridad de las aplicaciones debe ser demostrada, los procesos de auditoría deben hacer uso de la evidencia proporcionada por los controles de seguridad de la aplicación, para de esta manera determinar si la aplicación cumple con el nivel de confianza esperado.

A continuación se mencionan las otras partes de la norma ISO/IEC 27034, que actualmente se encuentran en elaboración.

- Parte 2: Marco normativo de la organización.

- Parte 3: Proceso de gestión de la seguridad de las aplicaciones.

- Parte 4: Validación de la seguridad de las aplicaciones.

- Parte 5: Protocolos y controles de seguridad de las estructuras de datos de las aplicaciones.

Los principios presentados por esta norma han sido tenidos en cuenta en el diseño de la guía de implementación, en tanto que tienen referencia con otros estándares que han servido como fuente primaria de información para el presente trabajo.

4.2.4 Análisis y evaluación de riesgos.

El grupo de normas ISO/IEC 27000 y COBIT, resaltan la importancia de la creación de controles derivados de un adecuado proceso de análisis y evaluación de riesgos.

“La gestión del riesgo es una herramienta fundamental para la identificación de los requerimientos de seguridad de una organización y para

la creación de un sistema eficaz de gestión de la seguridad de la información.” [10]

Acorde a lo anterior, en la etapa de especificación de requerimientos presentada en la guía de implementación, se hace referencia a normas y marcos de trabajo como los que se mencionan a continuación.

4.2.4.1 ISO/IEC 27005:2011 - Gestión de Riesgos de Seguridad de la Información.[11]

La norma ISO/IEC 27005:2011 es compatible con los conceptos generales especificados en la norma ISO/IEC 27001:2005, y está diseñada para apoyar la adecuada implementación de seguridad de la información, basada en la gestión de riesgos.

La norma define siete procesos para la gestión del riesgo:

- Establecimiento del contexto.
- Valoración del riesgo.
- Tratamiento del riego.
- Aceptación del riesgo.
- Comunicación del riesgo.
- Monitoreo y revisión del riesgo.

Así mismo propone métodos para la valoración de activos, listado de amenazas e identificación de vulnerabilidades.

[10] Concepto tomado de la norma ISO/IEC 27005:2011 Tecnología de la información – Técnicas de seguridad. Gestión de Riesgos de seguridad de la información. [13]
[11] La información presentada en esta sección es recopilada de la norma ISO/IEC 27005:2011. [13]

4.2.4.2 Guía para la gestión de riesgos en sistemas de tecnología de la información. NIST 800-30.[12]

El instituto nacional de estándares y tecnología, perteneciente al departamento de comercio de los Estados Unidos, creo en el año 2002, la guía NIST 800-30 para la gestión de riesgos en sistemas de tecnologías de la información.

Esta guía tiene por propósito aportar sugerencias a las organizaciones para que éstas puedan lograr sus objetivos, a través del aseguramiento de los sistemas que almacenan, procesan y transmiten su información. Adicionalmente provee herramientas para la comunicación adecuada de las decisiones de gestión de riesgos que permita la justificación de los gastos que hacen parte del presupuesto de TI. Finalmente plantea como la gestión de riesgos y la documentación resultante soporta los procesos de acreditación, y cumplimiento de regulaciones.

El NIST define la gestión de riesgos como el balance que deben hacer los responsables de los procesos, entre los costos operacionales y económicos de implementar medidas de protección que les permita ganar terreno en el cumplimiento de su misión de proteger los sistemas de información.

4.2.4.3 Metodología de análisis y gestión de riesgos de los sistemas de información. MAGERIT.[13]

La Metodología de Análisis y Gestión de Riesgos de sistemas de información fue creada por el Consejo Superior de Administración Electrónica

[12] La información que se presenta en esta sección es recopilada de la guía NIST 800-30. [27]

[13] La información presentada en esta sección es recopilada de la documentación de MAGERIT, correspondiente al método. [24]

del Gobierno de España. En este documento se toma como base la versión número dos, publicada en Mayo del 2006.

El análisis se enfoca en la determinación de los activos, definidos como elementos del sistema de información, en donde no solo se identifican los directamente relacionados con éste, sino todos aquellos que lo soportan, y que ayudan al cumplimiento de los objetivos estratégicos de la organización.

Para cada uno de los activos dependiendo de sus características, se propone la medición de las dimensiones asociadas con la seguridad de la información, autenticidad, confidencialidad, integridad, disponibilidad, y trazabilidad.

MAGERIT propone técnicas de valoración de los activos, controles y aspectos de seguridad, con el fin de cubrir todas las situaciones y los tipos de organización; aunque inicialmente fue diseñada para entidades del sector público.

4.3 CICLO DE VIDA DEL SOFTWARE – ISO/IEC 12207:2008

Según la norma ISO/IEC 12207:2008 el ciclo de vida del software es:

"Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso".[14]

La norma clasifica los procesos relacionados al ciclo de vida del software dentro de siete (7) categorías, las cuales se describen en función del propósito y los resultados que se desean obtener, presentando actividades y tareas para la consecución de dichos resultados.

[14] ISO/IEC-12207:2008- Modelo del ciclo de vida del software. [9].

4.3.1 Categorías de los procesos de ciclo de vida de software.

A continuación se describen las categorías propuestas por la norma:

1. Procesos de acuerdo.
2. Procesos organizativos de habilitación del proyecto.
3. Procesos del proyecto.
4. Procesos técnicos.
5. Procesos de implementación del software.
6. Procesos de soporte de software.
7. Procesos de reutilización del software.

El presente trabajo se enfoca y toma como referencia los subprocesos correspondientes al proceso de implementación del software, perteneciente a la categoría de procesos técnicos.

4.3.2 Análisis de requisitos del software.

En este proceso se deben establecer los requerimientos de los elementos de software del sistema, dentro de los cuales se encuentran las especificaciones de seguridad de acceso, incluyendo aquellas que comprometan la confidencialidad de la información.

4.3.3 Diseño de la arquitectura del software.

Este proceso tiene por propósito plantear un diseño para el software que implemente los requerimientos establecidos para el mismo, y que pueda ser verificado contra estos.

4.3.4 Diseño detallado del software.

Tiene por objeto proveer un diseño para el software, que implemente y pueda ser verificado contra los requisitos y la arquitectura del software, y sea lo suficientemente detallado como para permitir la codificación, y la ejecución de pruebas.

4.3.5 Construcción del software.

Su objetivo es generar unidades de software ejecutable, que reflejen apropiadamente el diseño del software.

4.3.6 Integración del software.

Tiene por propósito combinar las unidades de software y los componentes, para producir ítems de software integrados, consistentes con el diseño del software, que demuestren que los requerimientos funcionales y no funcionales son satisfechos sobre una plataforma operacional equivalente o completa.

4.3.7 Pruebas de calificación del software.

Su objetivo es confirmar que el producto integrado de software, cumple con los requerimientos definidos.

4.4 MODELOS DE MADUREZ Y CAPACIDAD, CRITERIOS DE EVALUACIÓN.

A continuación se mencionan otros estándares que se tuvieron en cuenta para la elaboración de la guía, entre los que se encuentran modelos de madurez y capacidad, y criterios de evaluación de productos de software.

4.4.1 CMMI – Modelo de madurez y capacidad integrado.

El modelo de madurez y capacidad integrada CMMI[15] es una colección de buenas prácticas para desarrollar productos y servicios de calidad, dicho modelo define 22 (Veintidós) áreas de proceso clasificadas en categorías, a continuación se mencionan las cinco áreas de la categoría de Ingeniería, que aplican al desarrollo del software:

- Desarrollo de requisitos.
- Solución técnica.
- Verificación.
- Integración del producto.
- Validación.

Estas áreas de proceso guardan relación con los procesos de implementación del software, definidos en la norma ISO/IEC 12207:2008.

[15] CMMI® para Desarrollo, Versión 1.3. Mejora de los procesos para el desarrollo de mejores productos y servicios, del Software Engineering Process Management Program. Carnegie Mellon University.

Modelo del ciclo de vida del software, los cuales se han seleccionado como base para el desarrollo del presente trabajo de investigación.

4.4.2 ISO/IEC 21827:2008 – Tecnología de la información – Ingeniería de Seguridad en Sistemas – Modelo de Madurez de Capacidad – (SSE- CMM).[16]

La norma ISO/IEC 21827:2008 es un modelo de referencia que describe las características esenciales que debe tener una organización para tener una buena ingeniería de seguridad en sus sistemas de información. Este estándar internacional no describe un proceso en particular, sino que captura mejores prácticas observadas por la industria, con el fin de que sean utilizadas en los procesos ya existentes para facilitar el incremento de madurez y capacidad de los mismos.

La Ingeniería de seguridad se define como una disciplina, que integra esfuerzos de otras áreas y especialidades para lograr la confiabilidad de un sistema, partiendo de la importancia que ha tomado la seguridad debido a la alta interoperabilidad e interconectividad de los sistemas de información actuales.

El modelo está dirigido a diversos tipos de organizaciones, consumidoras, proveedoras, evaluadoras de productos y servicios de tecnología informática, y ha clasificado las prácticas en dos dominios, a saber:

Dominio: Prácticas bases que de manera colectiva definen la ingeniería de seguridad.

Capacidad: Prácticas genéricas que indican gestión de procesos e institucionalización de la capacidad, deben realizarse como parte de las prácticas base.

[16] La información presentada en esta sección es recopilada de la documentación de la norma ISO/IEC 21827:2008. [8]

Para el desarrollo del presente trabajo de investigación se tuvieron en cuenta las prácticas base de las áreas de proceso que aplican a organizaciones desarrolladoras de aplicaciones, cómo son:

- PA01 Administración de controles de seguridad.
- PA02 Evaluación de impacto.
- PA03 Evaluación de riesgos de seguridad.
- PA04 Evaluación de amenazas.
- PA05 Evaluación de vulnerabilidades.
- PA06 Construir un argumento de garantía.
- PA07 Coordinar la seguridad.
- PA08 Monitorear el estado de la seguridad.
- PA09 Proveer entradas de seguridad.
- PA10 Especificar las necesidades de seguridad.
- PA11 Verificar y validar la seguridad.

Cada área de proceso tiene un listado de prácticas base las cuáles no obedecen a una etapa del ciclo de vida en particular sino que pueden aplicarse de manera transversal.

4.4.3 ISO/IEC 15408:2005. Tecnología de la información – Técnicas de Seguridad – Criterios de evaluación para la seguridad en TI. [17]

La norma ISO/IEC 15408:2005 establece un conjunto de criterios comunes que sirven como base para la evaluación de las propiedades de seguridad de los sistemas y productos de TI.

La norma está compuesta por tres partes:

Parte 1: Introducción y modelo general: Define los conceptos y principios generales de la evaluación de la seguridad en TI y presenta un

[17] Contenido extraído de las partes 1, 2, y 3 de la norma ISO/IEC 15408:2005, Criterios de evaluación para seguridad en TI.

modelo general de evaluación. Ofrece lineamientos para seleccionar y definir requerimientos de seguridad.

Parte 2: Requerimientos funcionales de seguridad: Propone una forma estándar de expresar los requerimientos funcionales mediante un conjunto de componentes clasificados en familias y clases.

A continuación se presentan la clasificación de los requerimientos funcionales propuestos por la norma:

- Auditoria de seguridad.
- Comunicación.
- Soporte criptográfico.
- Protección de datos de usuario.
- Identificación y autenticación.
- Gestión de la seguridad.
- Privacidad.
- Protección de las funciones de seguridad del producto/sistema a evaluar.
- Utilización de recursos.
- Acceso al objetivo de evaluación.
- Rutas y canales seguros.

Parte 3: Requerimientos de garantía de seguridad: Establece un conjunto de componentes clasificados en familias y clases para expresar los requerimientos que determinan la confianza en que un producto o sistema cumpla con sus objetivos de seguridad, así mismo define los criterios de evaluación y presenta los niveles de confianza en la evaluación.

Como corolario, se destaca que la norma ISO/IEC 27002:2005[18] en el dominio A.12 “Adquisición, desarrollo y mantenimiento de los sistemas de información”, menciona el estándar ISO/IEC 15408-1:2005 como guía para la especificación de los requerimientos de seguridad.

[18] ISO/IEC 27002:2005 - Código para la práctica de la gestión de la seguridad de la información.

4.4.4 ISO/IEC 9126-1:2001. Ingeniería de software. Calidad de producto. Parte 1: Modelo de calidad. [19]

La norma ISO/IEC 9126 cuya primera versión fue publicada en 2001, definen los requisitos de calidad que debe cumplir un producto de software, así como los criterios para su evaluación.

Este modelo de calidad define seis características, cada una con sub-características, como se puede observar en la Figura No. 1.

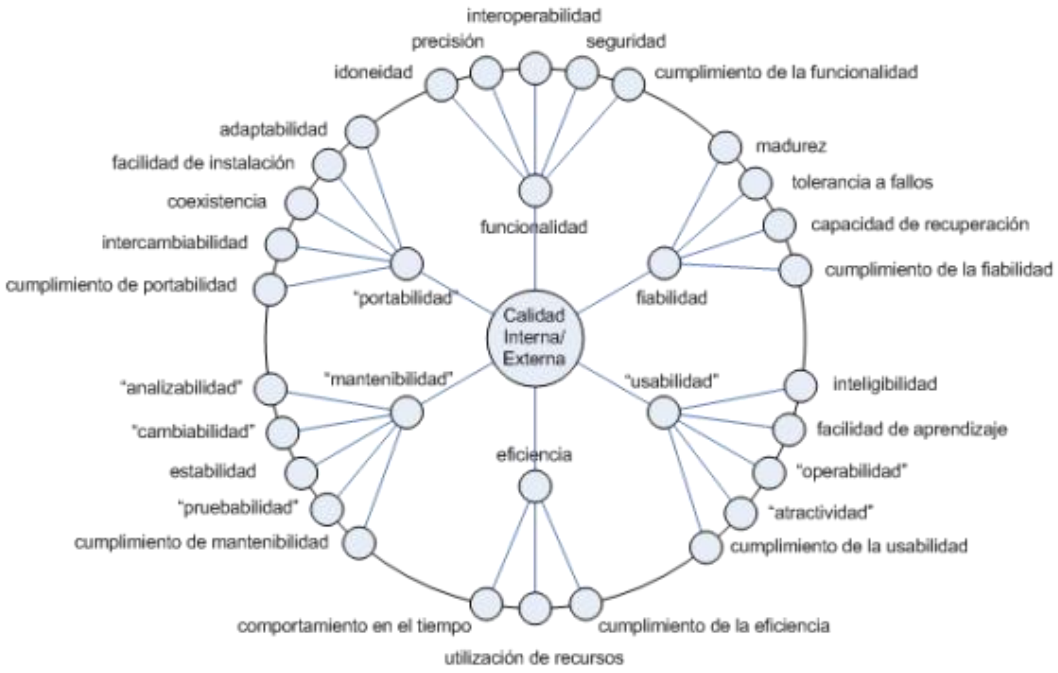


Figura No. 1: Características de la calidad según la ISO/IEC 9126-1:2001. [20]

La seguridad se establece como sub-característica de la funcionalidad, esta última corresponde al grado en que las necesidades asumidas o descritas se satisfacen.

[19] Las definiciones y conceptos mencionados en este apartado han sido extraídos de la norma ISO/IEC 9126-1:2001. [11]
[20] Gráfica tomada del sitio web www.iso25000.com. [14].

Así mismo la norma ISO/IEC 9126 define la seguridad como el grado en que un acceso no autorizado (accidental o deliberado) se prevenga y se permita un acceso autorizado.

Se establecen métricas acorde a las siguientes vistas:

- Interna: Aplicables durante el proceso de desarrollo.
- Externa: Aplicables al producto de software en ambiente productivo.
- Calidad en el uso: Productividad y efectividad que percibe el usuario al usar el software.

En las vistas interna y externa se tienen en cuenta las 6 características mencionadas anteriormente. En la vista de calidad en el uso se verifica la efectividad, productividad, satisfacción y seguridad.

En la vista de calidad en el uso, la seguridad se define como la capacidad del software para cumplir con los niveles de riesgo permitidos tanto para posibles daños físicos como para posibles riesgos de datos.

La ISO/IEC 9126 actualmente está siendo reemplazada por el grupo de normas ISO/IEC 25000, en donde la seguridad es definida como una característica principal, cuyas sub-características son la confidencialidad, integridad, no repudio, responsabilidad, y autenticidad.[21]

4.4.5 ISO/IEC 14598-3:2000. Evaluación del producto de software. Parte 3: Proceso para desarrolladores. [22]

La serie de normas ISO/IEC14598 provee métodos para la medición, valoración y evaluación de la calidad y conformidad de un producto de software.

[21] Descripción propuesta en la norma ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, tomada del libro Calidad del producto y proceso software. [2]

[22] Las definiciones y conceptos mencionados en este apartado han sido extraídos de la norma IRAM-ISO/IEC 14598-3, la cuál es una adopción idéntica de la norma ISO/IEC 14598-3:2000. [42]

Esta parte de la norma está orientada a brindar apoyo al desarrollador para medir las características del software durante el proceso de desarrollo, centrándose en aquellos indicadores que son útiles para predecir la calidad del producto final midiendo la calidad de los productos intermedios, de esta manera se puede lograr la conformidad del producto terminado

La norma es utilizada durante el desarrollo y es aplicable a todas las actividades de dicho proceso.

4.5 DESARROLLO SEGURO DE APLICACIONES.

Existen diversas propuestas de la industria, que sugieren y agrupan mejores prácticas para implementar controles de seguridad en todas las etapas del ciclo de vida, dichos controles pueden ir variando a medida que se descubren y explotan nuevas vulnerabilidades.

En el presente trabajo de investigación se tomaron como base fundamental las propuestas de OWASP (Open Web Application Security Project) y del MSDL (Microsoft Security Development Lifecycle).

4.5.1 OWASP – Open Web Application Security Project

OWASP es una comunidad libre, abierta y sin ánimo de lucro; que está continuamente trabajando en varios proyectos relacionados con la seguridad de aplicaciones, con el fin de dar soporte a las organizaciones, ayudando a concebir, desarrollar, adquirir, operar y mantener aplicaciones confiables.[23]

A continuación se mencionan varios de los proyectos OWASP, que son referenciados en las diferentes secciones del presente trabajo:

[23] La información presentada en esta sección ha sido recopilada del sitio web de OWASP: https://www.owasp.org.

- Modelo de madurez para el aseguramiento del software (SAMM) Versión 1.0, el cual consiste en un marco de trabajo que le permite a las organizaciones evaluar prácticas de seguridad implementadas, definir y medir actividades, demostrar mejoras y construir un programa de seguridad con actividades concretas.[24]

- OWASP Top Ten – 2013 rc1. Los diez riesgos más importantes en aplicaciones web.[25]

- Estándar de verificación de seguridad en aplicaciones (OWASP Application Security Verification Standard), el cual define requerimientos de verificación para aplicaciones web, que pueden ser usados como métricas, guías para la construcción de controles; y en la adquisición, proporciona bases para establecer la forma en que se deben verificar los requerimientos de seguridad.[26]

- Prácticas de código seguro.[27]

- Guías para autenticación, validación y criptografía.[28]

- Guía para pruebas.[29]

Es importante resaltar que la gran mayoría de controles de seguridad propuestos por las guías de desarrollo seguro de OWASP están relacionados con objetivos de control de COBIT, de igual manera ofrecen información sobre la implementación del estándar para manejo de datos de tarjetas de crédito PCI-DSS[30].

[24] *OWASP Software Assurance Maturity Model.* [30].
[25] *OWASP Top 10 Application Security Risks – 2013.* [38].
[26] *OWASP Application Security Security Verification Standard - Web Application Standard.* [29]
[27] *OWASP Secure Coding Practices Quick Reference Guide.* [31]
[28] Esta información se encuentra registrada en la Guía OWASP para construir aplicaciones y servicios web seguros.
[29] *OWASP Testing guide*. [36]
[30] PCI-DSS: *Payment Card Industry Data Security Standard*.

Organizaciones tanto del sector público como del sector privado a nivel mundial, han utilizado, referenciado y contribuido con los proyectos desarrollados por OWASP. [31]

4.5.2 Microsoft – Security Development Lifecycle.[32]

El SDL (Security Development LifeCycle) propuesto por Microsoft, es un proceso de desarrollo de software que ayuda a los desarrolladores a construir software más seguro y a cumplir con los requerimientos de seguridad mientras se reducen los costos del desarrollo.

El SDL está compuesto por una fase inicial de capacitación y concientización, y 17 (Diecisiete) prácticas distribuidas en 6 (seis) fases que a continuación se describen:

- Fase 1: Requerimientos: Tener en cuenta la seguridad y la privacidad como base fundamental en el inicio del proyecto y analizar si la inversión para mejorar estos aspectos es coherente con las necesidades del negocio.
- Fase 2: Diseño: Establecer las mejores prácticas a seguir a través de las especificaciones funcionales y de diseño y realizar el análisis de riesgos para identificar amenazas y vulnerabilidades.
- Fase 3: Implementación: Definir las mejores prácticas de desarrollo para detectar y remover problemas de seguridad y privacidad de forma más temprana en el ciclo de desarrollo.
- Fase 4: Verificación: Verificar si el código cumple con los requisitos establecidos en las fases previas, mediante la realización de pruebas de seguridad y privacidad, para lo cual se sugiere que la totalidad del equipo esté enfocado en la actualización del modelo de amenazas, la revisión del código, las pruebas, y la elaboración y revisión de toda la documentación.

[31] Ver referencias de la industria TIC. [37]
[32] La información presentada en esta sección ha sido recopilada del documento Microsoft Security Development LifeCycle. Version 5.0. [23].

- Fase 5: Liberación: Realizar una revisión final de seguridad y privacidad antes de liberar la versión, y diseñar un plan de acción para la atención a incidentes que puedan presentarse a causa del descubrimiento de vulnerabilidades en el software.
- Fase 6: Respuesta: Después de la liberación del software el equipo de desarrollo debe estar preparado para atender cualquier posible vulnerabilidad de seguridad o problema de privacidad que requiera pronta respuesta, por lo que deberán ejecutarse los planes de acción establecidos en la fase de liberación.

El MSDL ofrece tres propuestas dependiendo del tipo de aplicación:

- SDL para productos.
- SDL para aplicaciones de negocio y aplicaciones web.
- SDL para metodologías agiles de desarrollo, las cuales se enfocan en implementar requerimientos en ciclos cortos de desarrollo, que satisfagan necesidades directas del cliente.

4.6 ENTREVISTA.

Dentro del proceso de elaboración de la GICSI se ha considerado la necesidad de realizar entrevistas con un enfoque metodológico, que permitan recopilar información sobre la visión que tienen analistas y profesionales respecto de la seguridad informática en el proceso de desarrollo de software.

A continuación se describen los conceptos de diseño y aplicación de entrevistas que son utilizados en la realización de esta actividad durante el presente trabajo de investigación.

La entrevista se define como una situación interpersonal, una conversación entre dos personas acerca de un tema de interés mutuo. Es

una forma de interacción humana específica en la cual el conocimiento evoluciona a través del diálogo.[33]

4.6.1 Entrevista de Investigación

La entrevista de investigación tiene por objeto obtener información y comprensión de los asuntos relacionados con los objetivos generales y preguntas específicas de un proyecto de investigación.[34]

Para los efectos de la presente investigación se elabora una entrevista semi-estructurada[35], en la cual el investigador cuenta con una guía de preguntas o temas muy específicos a ser cubiertos; a pesar de lo anterior, el entrevistado cuenta con libertad a la hora de responder; pueden surgir e incluirse otras preguntas a medida que avance la conversación.

4.6.2 Etapas de una entrevista.

Según Kvale[36], la entrevista tiene siete etapas a saber:

1. Tematización: Definición del tema que se va a tratar en la entrevista, y del propósito de la misma.
2. Diseño: Cómo se va a obtener el conocimiento sobre el tema definido, establecer que tipos de preguntas van a permitir lograr el propósito de la entrevista.

[33] Concepto tomado del libro Social Research Methods: *Qualitative and Quantitative Aproaches*. [1]
[34] Concepto tomado del libro "The Research Interview". [4]
[35] Basado en el concepto de entrevista semi-estrucutara presentado en el libro *Social Research Methods: Qualitative and Quantitative Aproaches*. [1].
[36] Steinar Kvale es Catedrático de Psicología de la Educación y Director del Centro de Investigación Cualitativa de la Universidad de Aarthus, Dinamarca y profesor del Instituto Saybrook en San Francisco. Su libro InterViews: Learning the Craft of Qualitative Research Interviewing, ha sido la base para el diseño y elaboración de la entrevista realizada en la presente investigación.[18]

3. Entrevista: Conducción cuidadosa de la entrevista, teniendo a mano una guía que indique los temas, el orden y la manera en que éstos van a ser abordados.
4. Transcripción: Conversión de la entrevista a texto.
5. Análisis: Análisis basado en el tipo apropiado de investigación.
6. Verificación: Comprobación de la validez, confiabilidad y generalización de los hallazgos.
7. Reporte: Comunicación de los hallazgos de una manera científica y ética.

4.6.3 Tipos de muestra.[37]

La muestra es un subgrupo de la población, ésta última debe cumplir con un conjunto de características definidas que son pertinentes para el estudio que se requiere realizar. Las muestras se dividen en probabilísticas y no probabilísticas:

En las muestras probabilísticas todos los elementos de la población tienen la misma probabilidad de ser seleccionados.

En las muestras no probabilísticas la elección de los elementos no depende de la probabilidad, en este caso la selección depende de las características de la investigación y/o del investigador.

4.6.3.1 Muestras no probabilísticas.

Las muestras no probabilísticas o dirigidas son útiles para determinado diseño de estudio que requiere no tanto de representatividad de elementos de una población, sino de una cuidadosa y controlada elección de sujetos con ciertas características especificadas previamente en el planteamiento del problema.

[37] Información recopilada del capítulo 8: ¿Cómo seleccionar una muestra?, del libro Metodología de la Investigación.[5]

Dentro de las muestras no probabilísticas se encuentran las muestras de expertos, sujetos-tipos, voluntarios y por cuotas, en la presente investigación se hace uso de una muestra de sujetos-tipo.

4.6.4 Análisis y presentación de resultados.

Para el análisis de los resultados, se requiere previamente codificar y registrar las respuestas.

El proceso de codificación de las respuestas para preguntas cerradas puede realizarse antes de la realización de la entrevista, en el caso de las preguntas abiertas, una vez se conozcan todas las respuestas de los entrevistados, se identifican patrones generales correspondientes a respuestas similares o comunes, para posteriormente asignar un valor numérico o un símbolo a cada patrón.[38]

Después de la codificación, se aplica estadística descriptiva para cada variable, calculando la distribución de frecuencias, medidas de tendencia central y medidas de la variabilidad[39]; dichos métodos se aplican dependiendo del interés del investigador.

[38]Extraído de la sección 9.6.2 Cuestionarios del libro Metodología de la Investigación de Hernández Sampieri. [5]
[39] Tomado del capítulo 10. Análisis de datos del libro Metodología de la investigación de Hernández Sampieri. [5]

5. PROCESO DE ELABORACIÓN DE LA GICSI.

En el presente capítulo se describe el proceso aplicado para diseñar y construir la guía de implementación, la cual es el principal objetivo de este trabajo de investigación.

El enfoque utilizado para la definición del proceso de diseño y construcción, toma en consideración los conceptos del ciclo de vida del software[40]. A continuación se describen cada una de las etapas establecidas en dicho proceso:

- Etapa I – Recopilación y análisis de información:
  - Recopilación y análisis de información de normas, estándares, metodologías, marcos de trabajo y buenas prácticas, relacionadas con el desarrollo de software y con la gestión de seguridad de la información.
  - Recopilación y análisis de la visión y experiencia de profesionales en desarrollo de software, a partir del diseño y realización de una entrevista.[41]
- Etapa II - Diseño: Diseño de la estructura de la guía, a partir de los resultados de la entrevista, las normas analizadas, y las propuestas de la industria relacionadas con controles y buenas prácticas para desarrollar software seguro.
- Etapa III - Construcción: Elaboración de la guía, teniendo en cuenta la estructura definida y el diseño establecido en la etapa II.

[40] Ver marco teórico sección 5.3: Ciclo de vida del software – ISO/IEC 12207:2008.
[41] Ver capítulo de diseño y elaboración de la entrevista.

- Etapa IV - Validación: Definición y ejecución de un proceso de validación, con el fin de evaluar si la guía construida cumple con las necesidades y los objetivos establecidos.

En la Figura No. 2 puede observase un esquema del proceso descrito anteriormente.

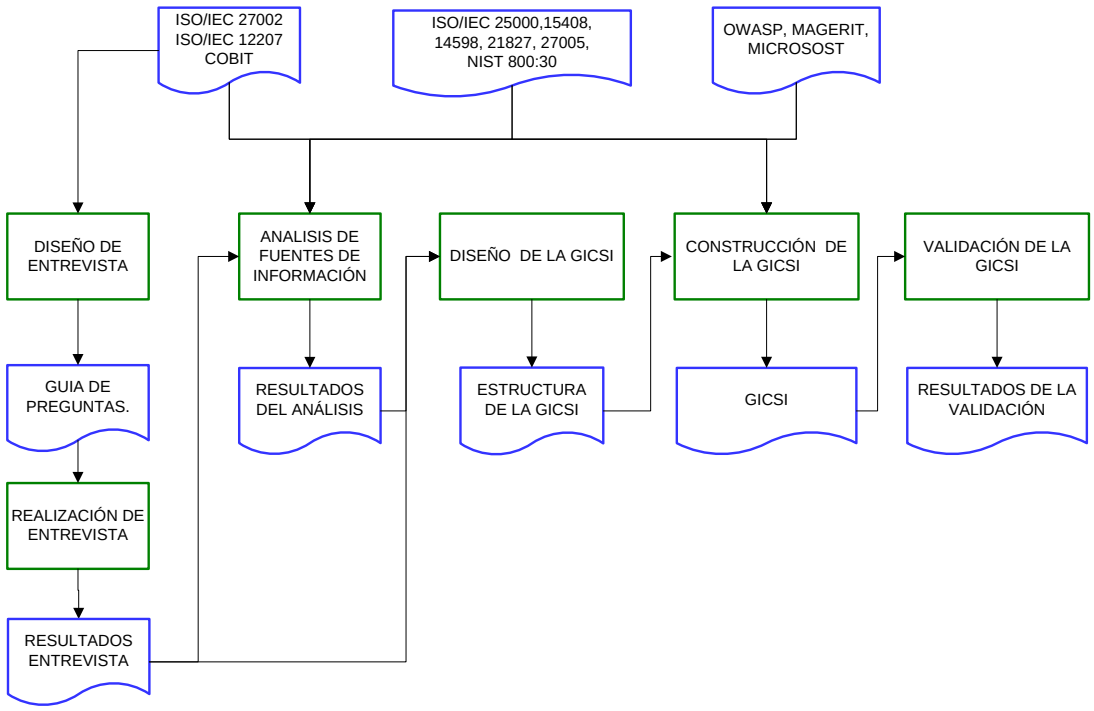


Figura No. 2: Proceso de elaboración de la GICSI.

En los siguientes capítulos, se detallan los resultados de cada una de las etapas definidas.

6. DISEÑO Y REALIZACIÓN DE LA ENTREVISTA.

Acorde al proceso establecido para la elaboración de la guía presentado en el capítulo anterior, la etapa I de recopilación y análisis de información, incluye la realización de una entrevista a diferentes integrantes de un proyecto de software a medida, con el fin de obtener datos sobre la experiencia, visión y opinión respecto de la incorporación de controles de seguridad al proceso de desarrollo de software que aplican en el día a día.

A continuación se describe el procedimiento utilizado, el cual tiene como base las etapas mencionadas por Kvale[42], para el diseño y realización de entrevistas.

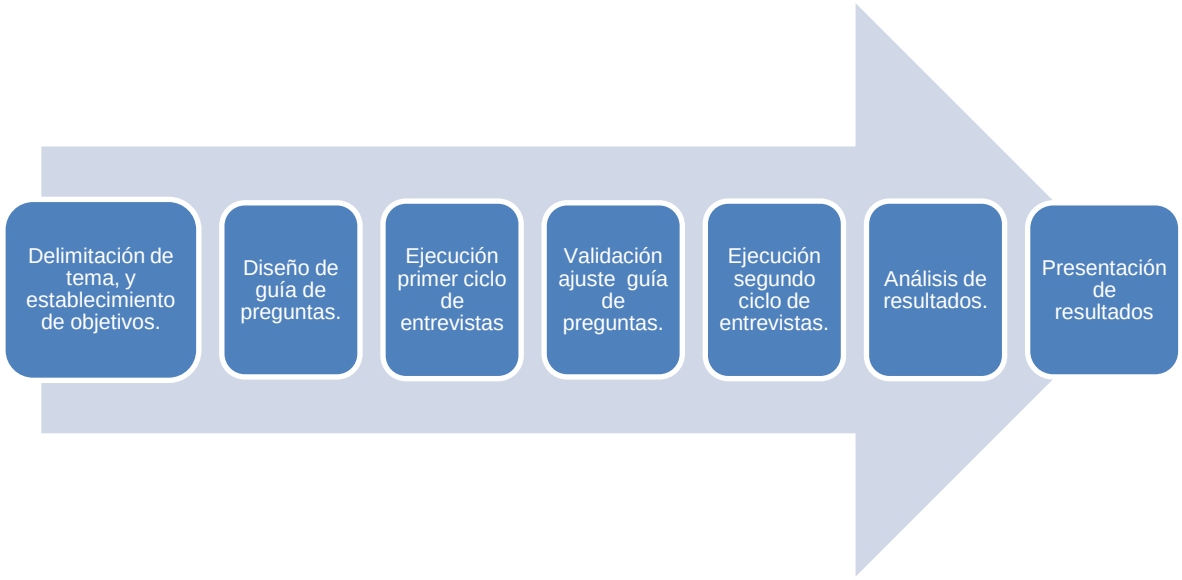


Figura No. 3: Metodología para el diseño y realización de la entrevista.

[42] Etapas de entrevista, descritas en el libro: InterViews: Learning the Craft of Qualitative Research Interviewing, pp 135. [17].

6.1 DELIMITACIÓN DEL TEMA Y ESTABLECIMIENTO DE OBJETIVOS.

Como se observa en la Figura No. 3, inicialmente se realiza la etapa de tematización en la cual se establece el tema, los objetivos de la entrevista y se definen los criterios de selección de los entrevistados.

6.1.1 Objetivos de la entrevista.

1. Establecer el grado de conciencia que se tiene sobre la importancia de la seguridad informática en los procesos de desarrollo de software a medida.
2. Identificar la aceptabilidad que tiene la aplicación de estándares y controles de seguridad informática en los procesos de desarrollo de software a medida.
3. Identificar posibles estrategias para facilitar la incorporación de controles de seguridad informática en procesos de desarrollo de software a medida.

6.1.2 Selección de las personas a entrevistar.

Se selecciona una muestra de sujetos-tipo[43] de un grupo de analistas de desarrollo de software, teniendo en cuenta su experiencia, así como las funciones de los cargos que han desempeñado en proyectos de desarrollo de software a medida.

Con el fin de cubrir tanto perfiles técnicos, como de gestión y funciones comerciales, se seleccionan 12 (doce) personas con los siguientes cargos:

[43] Ver sección 3.6.3.1 Muestras no probabilísticas, del presente documento.

- Analista programador.
- Analista funcional.
- Líder de proyecto.
- Gerente de proyecto.
- Gerente de empresa.

Las personas seleccionadas trabajan en las consultoras y empresas de desarrollo de software a medida que se presentan a continuación:

- everis – Argentina: Es una consultora de software multinacional que ofrece soluciones de negocio, estrategia y desarrollo, mantenimiento de aplicaciones tecnológicas y outsourcing. La consultora cubre los sectores de telecomunicaciones, entidades financieras, industria, utilities & energía, seguros, administraciones públicas, media y sanidad.[44]
- VEMN: Es un equipo de profesionales con más de 20 (veinte) años de trayectoria en el mercado local y regional, especializados en: Arquitectura e Ingeniería de Software, Metodologías Ágiles, SharePoint, Team Foundation Server y SQL Server.[45]
- CogisCAN: Empresa canadiense líder en desarrollo de hardware y software para rastreo, seguimiento y control de herramientas y materiales para la industria manufacturera. [46]

6.2 DISEÑO DE LA ENTREVISTA.

En el diseño de la entrevista se tienen en cuenta los tipos de preguntas propuestas por Kvale[47], así como las etapas del proceso de

[44] Tomado de la página oficial everis – Argentina. http://www.everis.com/argentina/es-AR/sobre-everis/compania/Paginas/compania.aspx
[45] Tomado de la página oficial de VEMN. http://www.vemn.com.ar/es/#/empresa-vemn-sistemas/.
[46] Tomado de la página oficial de CogisCan Inc. http://www.cogiscan.com/about-cogiscan/.
[47] Tomado del libro InterViews: Learning the Craft of Qualitative Research Interviewing. [17]

desarrollo de software definidas en la norma ISO/IEC 12207:2008[48] y los objetivos de control pertinentes, de la norma ISO/IEC 27002:2005[49].

Como resultado del proceso anterior se obtiene la guía de preguntas No. 1 (uno). [50]

6.3 REALIZACIÓN DE LA ENTREVISTA.

La entrevista se aplica en dos ciclos, el primer ciclo tiene por objeto medir la validez[51] del instrumento, en el mismo se aplica la guía de entrevista No. 1 (uno) a 4 (cuatro) personas de diferentes perfiles de la empresa everis.

Se analizan los resultados de este primer ciclo, con el fin de validar si los datos recopilados a partir de las preguntas realizadas, permiten cumplir con los objetivos establecidos para la entrevista y servir como fuente primaria para el desarrollo de la presente investigación.

Después de realizar dicha validación, se ajustan las preguntas y se diseña la segunda guía de entrevista.[52]

En el segundo ciclo, la guía de entrevista No. 2 (dos) es aplicada a 12 (doce) personas, 7 (siete) de la consultora everis - Argentina, 4 (cuatro) de la Pyme VEMN y 1 (uno) de la empresa CogisCAN; cuyos perfiles se presentan en la Tabla No.1.

---

[48] ISO/IEC 12207:2008 *Systems and software engineering - Software life cycle processes.* [9]

[49] ISO/IEC 27002:2005 Tecnología de la Información – Técnicas de Seguridad - Código para la práctica de la seguridad de la información. [7]

[50] Ver Anexo 2: Guía de preguntas No. 1.

[51] "La validez se refiere al grado en que un instrumento de medición mide realmente la(s) variable(s) que pretende medir". Sección 9.6.7. ¿Cómo se codifican las respuestas a un instrumento de medición?, del libro Metodología de la Investigación de Hernández Sampieri. [5]

[52] Ver Anexo 3. Guía de preguntas No. 2. Diseño Final.

Tabla No. 1: Perfil de personas entrevistadas.

Empresa	Perfiles entrevistados.	Ciclo
everis – Argentina.	• Analista programador junior: Un año de experiencia. • Analista Programador semi-senior: Dos años y medio de experiencia. • Analista Programador senior: Seis años de experiencia. • Analista funcional senior: Seis años de experiencia.	1
everis - Argentina	• Analista programador junior, con dos años de experiencia. • Analista Programador senior con siete años de experiencia en empresas de servicios petroleros, empresas de telecomunicaciones y del sector público. • Analista programador senior con siete años de experiencia en pymes, banca y sector público. • Analista funcional senior con siete años de experiencia. • Líder de proyecto con siete años de experiencia, en empresas de telecomunicaciones y sector público. • Gerente de proyecto con experiencia en empresas de telecomunicaciones y sector público. • Gerente de proyecto con experiencia en empresas del sector público y pymes.	2
VEMN	• Presidente y fundadora de la empresa. Magister en Informática, con amplia experiencia en la aplicación de metodologías ágiles en procesos de desarrollo software a medida. • Arquitecto de software. • Analista Funcional Senior. • Analista Funcional Junior.	2
CogisCAN	Analista QA (Quality Assurance) senior con 9 años de experiencia.	2

6.4 ANALISIS Y PRESENTACIÓN DE RESULTADOS.

Después de realizar la entrevista, se codifican las preguntas, y se registran las respuestas[53], con esta información para cada pregunta se calculan las frecuencias absolutas y relativas, se presenta la distribución de frecuencias mediante una gráfica y se realizan comentarios sobre los resultados obtenidos.

Los resultados y el análisis para cada pregunta de la guía de entrevista No. 2 (dos), se encuentran en el Anexo 4.

6.5 CONCLUSIONES DE LA ENTREVISTA

Teniendo en cuenta los objetivos establecidos y acorde al análisis de las respuestas recopiladas, a continuación se presentan las conclusiones más importantes de la entrevista realizada:

- Si bien se tiene cierta conciencia sobre la seguridad de la información, se puede inferir que la misma no es tenida en cuenta frecuentemente por las consultoras como característica principal de las aplicaciones dentro de las propuestas comerciales, a menos que el modelo de negocio tenga una necesidad puntual ya identificada o sea muy evidente, la seguridad es considerada costosa y poco valorada por el cliente.

- Los conceptos de disponibilidad e integridad de la información no son regularmente relacionados con la seguridad informática.

- El éxito de la incorporación de controles y de la concientización de los equipos de desarrollo, va de la mano con el seguimiento que se le pueda realizar a la aplicación de las políticas y lineamientos establecidos.

[53] En el apartado 5.6.4. Análisis y presentación de resultados, del presente documento, se detallan los procesos a tener en cuenta para la codificación de la información recopilada en la entrevista, así como para la presentación de los resultados.

- Según los analistas y líderes entrevistados, la implementación de controles de seguridad requiere de conocimientos específicos y profesionales calificados, perfiles con los que actualmente no cuentan la mayoría de las consultoras de desarrollo de software.

- Aunque los estándares son percibidos como herramientas que aportan a la calidad de las aplicaciones, algunos de los entrevistados consideran que su implementación completa exige bastante esfuerzo y altos costos que el cliente no asume, y por consiguiente puede impactar en la productividad del equipo de desarrollo, al menos con una mirada sesgada al impacto inmediato.

- Una guía de implementación de controles de seguridad en el proceso de desarrollo debe ser didáctica, independiente de la metodología de desarrollo, basada en un marco conceptual, que identifique los beneficios y relaciones con estándares y las mejores prácticas.

7. DISEÑO DE LA GICSI.

Como resultado de la ejecución de la etapa II del proceso de elaboración de la guía, se obtiene la estructura que se describe a continuación.

La GICSI está estructurada en dos secciones, una sección introductoria y la sección correspondiente al cuerpo principal.

7.1 INTRODUCCIÓN.

La sección inicial contiene los siguientes apartados:

- Objetivo general.
- Marco conceptual, en el cual se relacionan todas las fuentes primarias de información en las que está basada la guía.
- Alcances y limitaciones.
- Conceptos y definiciones: Términos y conceptos relacionados con seguridad informática, a los que se hace referencia en la guía.
- Beneficios: En este apartado se resaltan las ventajas que se pueden obtener al aplicar la guía, así como los beneficios de la incorporación de controles en cada una de las etapas del proceso de implementación del software.
- Recomendaciones generales: Se presentan recomendaciones generales que aplican de manera transversal a todas las partes de la guía.

7.2 ESTRUCTURA PRINCIPAL.

La sección correspondiente al cuerpo principal de la guía, está organizada en 6 (seis) partes, cada una de éstas presenta la siguiente estructura:

- **Título:** Relaciona cada parte con el proceso de implementación de software correspondiente, y establece en términos generales el alcance de la misma.
- **Objetivo:** Establece cuales son los resultados a obtener en cada parte, en cuanto a seguridad informática se refiere.
- **Cumplimiento:** Identifica los controles propuestos por la norma ISO/IEC 27002:2005 y los objetivos de control de COBIT 4, pertinentes a cada etapa.
- **Actividades (A)**: Acciones a realizar para lograr el objetivo de cada una de las partes.
- **Entradas:** Información y requisitos necesarios para desempeñar las actividades en cada parte.
- **Salidas:** Resultados visibles de la correcta ejecución de las actividades establecidas en cada parte.
- **Participantes**: Los roles que deben participar en cada parte y que tienen la capacidad de tomar decisiones, ejecutar las actividades y hacerles seguimiento.[54]

[54] De acuerdo a COBIT 4.1. "El entendimiento de los roles y responsabilidades para cada proceso es clave para un gobierno efectivo". [15]

Los elementos mencionados se presentan en diagramas similares al que se puede observar en la Figura No. 4.

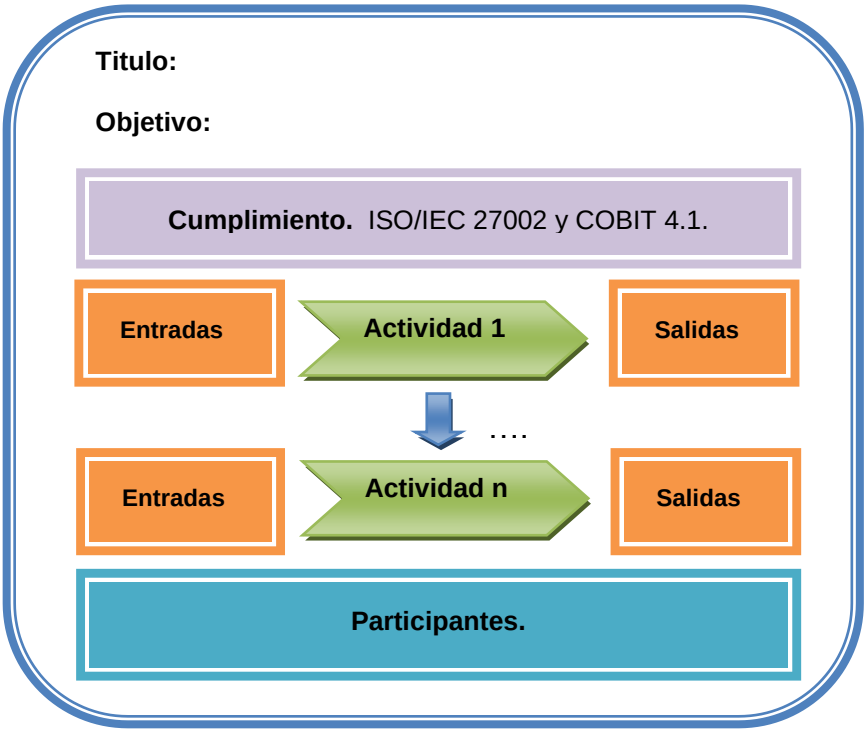


Figura No. 4: Estructura de la GICSI.

7.2.1 Detalle por actividad.

En cada una de las actividades se establecen los siguientes ítems, para orientar o facilitar la realización de las mismas:

- **Descripción:** Descripción general de la actividad a realizar. (Qué).
- **Tareas (T):** Acciones correspondientes a cada actividad.
- **Qué[55]:** Aspectos a tener en cuenta para asegurar que se están atendiendo razonablemente las necesidades y características que atañen a la seguridad informática en esta actividad; los mismos están

[55] Estos aspectos se basan en la información recolectada de normas y estándares, así como de guías y listas de verificación sugeridas por la industria.

expresados en forma de pregunta para facilitar su entendimiento. Con el fin de simplificar la aplicación de la guía para organizaciones más pequeñas, las preguntas se han clasificado en “necesarias” y “deseables”, estas últimas se presentan en letra cursiva para facilitar su identificación.

- **Cómo[56]:** Buenas prácticas sugeridas, recopiladas de normas, marcos de trabajo, y propuestas del mercado; que dan respuesta al “qué” y ayudan a cubrir los requerimientos de seguridad establecidos.

En la Figura No. 5 puede observarse un ejemplo de cómo se describen cada una de las actividades.

An Nombre Actividad

Descripción de la actividad

Tn.1 Descripción de la tarea 1

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
Preguntas…	Prácticas sugeridas, referencias a marcos de trabajo, normas y propuestas de la industria TIC.

Tn.x Descripción de la tarea x

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
Preguntas…	Prácticas sugeridas, referencias a marcos de trabajo, normas y propuestas de la industria TIC.

Figura No. 5: Detalle de cada actividad.

[56] Las buenas prácticas fueron extraídas de documentos de OWASP, Microsoft, MAGERIT, entre otros.

8. CONSTRUCCION DE LA GICSI.

A partir de la ejecución de la etapa III del proceso de elaboración, y siguiendo la estructura establecida en la fase de diseño, se construye la guía de incorporación de controles de seguridad informática en procesos de desarrollo de software a medida, la cual es presentada en el capítulo 9 de este documento.

El contenido completo de la GICSI se encuentra disponible para consulta en un sitio web[57] construido para facilitar la lectura y el acceso a la información registrada en ésta.

[57] https://sites.google.com/site/segappguide/, para ingresar se requiere habilitar previamente el acceso.

9. GUIA PARA LA INCORPORACIÓN DE CONTROLES DE SEGURIDAD INFORMÁTICA EN PROCESOS DE DESARROLLO DE SOFTWARE A MEDIDA - GICSI

La presente guía se propone como una herramienta para facilitar la incorporación de controles de seguridad informática en procesos de desarrollo de software a medida, que permitirá a gerentes y líderes de proyecto, mitigar riesgos, minimizar impactos e implementar controles, adoptando la seguridad como parte de dichos procesos.

En la guía se establecen actividades concretas relacionadas con las dimensiones de la seguridad (confidencialidad, disponibilidad, integridad, autenticación y trazabilidad) para cada etapa del ciclo de vida del software, y se recopilan mejores prácticas, lineamientos y controles, definidos en estándares y en diferentes metodologías propuestas por la industria TIC.

9.1.1 OBJETIVO

Facilitar la adopción de conceptos y la incorporación de controles de seguridad informática en proyectos de desarrollo de software a medida.

9.1.2 MARCO CONCEPTUAL

La guía es el resultado de un proceso de análisis de normas y estándares internacionales, así como de metodologías y marcos de trabajo propuestos por la industria TIC.

Adicionalmente se toma en cuenta la visión y experiencia de profesionales y analistas, las cuales han sido recopiladas a través de entrevistas realizadas a equipos de desarrollo de software a medida.

El detalle del marco conceptual mencionado se puede observar en la Figura No. 6.

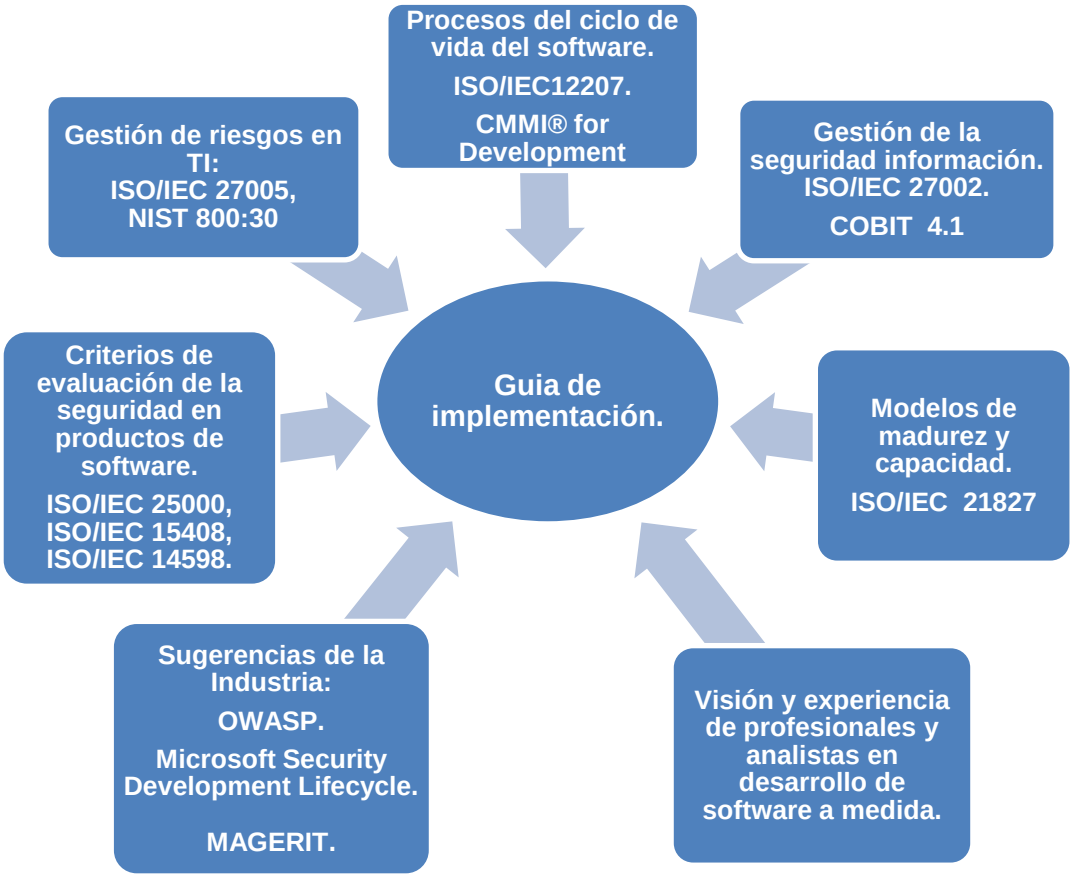


Figura No. 6: Marco conceptual en el que está basado la guía de implementación.

9.2 ALCANCE

La presente guía se focaliza en los procesos de implementación del software definidos por la norma ISO/IEC 12207:2008, los cuales se ilustran en la Figura No. 7 y se referencian a continuación[58]:

- Análisis de requisitos del software.
- Diseño de la arquitectura del software.
- Diseño detallado del software.
- Construcción del software.
- Integración del software.
- Pruebas de calificación del software.

Figura No. 7: Ciclo de vida del software. Proceso de implementación ISO/IEC 12207:2008.

[58] En la sección 1.1. Ciclo de vida del software del presente documento, se encuentran más detalles sobre la norma ISO/IEC 12207:2008.

9.3 CONCEPTOS Y DEFINICIONES:

A continuación se presentan las definiciones y conceptos a los que se hace referencia en la guía.

Activos de información: Recursos del sistema de información o relacionados con éste, necesarios para que la Organización funcione correctamente y alcance los objetivos propuestos por su dirección.[59]

Amenaza: Una causa potencial de un incidente no-deseado, el cual puede resultar en daño a un sistema u organización.[60]

Autenticación: El acto de verificar la identidad de un usuario y su elegibilidad para acceder a la información computarizada. La autenticación está diseñada para proteger contra conexiones de acceso fraudulentas.[61]

Confidencialidad: Propiedad de que la información no sea divulgada a entidades, personas o procesos no autorizados.[54]

Disponibilidad: Propiedad de que la información esté disponible y utilizable cuando lo requiera una entidad autorizada.[54]

Dueño de la información: Individuos, por lo general gerentes o directores, que tienen la responsabilidad de la integridad, el uso y el reporte preciso de los datos computarizados.[62]

Dueño del proceso: Individuos que tiene por responsabilidad la correcta ejecución de los procesos del negocio, soportados por las aplicaciones informáticas.

Información de negocio: Información sensible para los procesos de negocio soportados por el software a desarrollar, y que puede actuar como entrada, ser procesada, o ser generada por éste último.

---

[59] Tomado de MAGERIT – Método. [24]
[60] Tomado de la norma ISO/IEC 13335-1:2004 *Part 1: Concepts and models for information and communications technology security management.*[12]
[61] Tomado del Apéndice VII Glosario de COBIT 4.1. [15]
[62] Concepto tomado de COBIT 4.1. [15]

Integridad: La propiedad de salvaguardar la precisión y completitud de los recursos.[63]

No Repudio: Asegurar que el remitente no puede negar que envió y que el receptor no pueda negar que recibió.[64]

Patrones de diseño seguro: Tienen por objetivo eliminar la inserción accidental de vulnerabilidades dentro del código y mitigar las consecuencias de esas vulnerabilidades.[65]

Pruebas de calificación: Pruebas llevadas a cabo por el desarrollador y presenciadas por el adquirente (según corresponda), para demostrar que un producto de software cumple sus especificaciones y está listo para ser utilizado en el entorno seleccionado o para integrarse al sistema que lo contiene.[66]

Riesgo: Efecto de la incertidumbre sobre el logro de los objetivos. El riesgo se expresa frecuentemente en términos de una combinación de las consecuencias de un evento (incluyendo cambios en las circunstancias), y la probabilidad de ocurrencia asociada.[67]

Trazabilidad: Aseguramiento de que en todo momento se podrá determinar quién hizo qué y en qué momento.[68]

Usabilidad: La capacidad que tiene un producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo condiciones específicas.[69]

Validación: Es el conjunto de actividades que aseguran y generan confianza en que un sistema puede lograr su uso previsto, metas y objetivos.[60]

---

[63] Tomado de la norma ISO/IEC 13335-1:2004 *Part 1: Concepts and models for information and communications technology security management.*[12]
[64] Concepto tomado de *Risk Management Guide for Information Techonlogy Systems – NIST.* [27]
[65] Concepto tomado del documento *Secure Design Patterns.*[3]
[66] ISO/IEC-12207- Modelo del ciclo de vida del software. [9].
[67] GUIA ISO/IEC 73 Gestión de riesgos – Terminología.[41]
[68] Concepto tomado del Apéndice 1: Glosario de MAGERIT Versión 2. Método. [24]
[69] Definición tomada de la norma ISO/IEC 9126-1:2001 *Software engineering - Product quality - Part 1: Quality model.* [11]

Verificación: Es el conjunto de actividades que compara un producto del ciclo de vida con las características requeridas para dicho producto. Esto puede incluir, pero no se limita a, los requerimientos especificados, la descripción del diseño y el propio sistema.[70]

Vulnerabilidad: Debilidad que puede ser accidentalmente disparada o intencionalmente explotada.[71]

9.4 BENEFICIOS

9.4.1 Beneficios esperados de la aplicación de la GICSI.

La estructura que sigue la guía así como la forma en que se presenta la información, permite:

- Aclarar y reforzar conceptos asociados a la seguridad informática.
- Identificar errores, vulnerabilidades y mejoras de manera proactiva, ya que cada parte cuenta con actividades de revisión y verificación.
- Comunicar y concientizar sobre la incorporación de los controles de seguridad, sugiriendo tareas de documentación y comunicación en cada parte.
- Evaluar el estado de los controles de seguridad implementados en una aplicación en mantenimiento, así como estimar esfuerzos en correcciones y mejoras.
- Aportar a la mejora del proceso de desarrollo aplicado, mediante la incorporación de conceptos y prácticas relacionadas con aseguramiento de la calidad.

[70] ISO/IEC-12207- Modelo del ciclo de vida del software. [9].
[71] Concepto tomado de *Risk Management Guide for Information Techonlogy Systems – NIST.* [27]

- Identificar controles a incorporar, independiente de la metodología o marco de trabajo que se esté utilizando en el proceso de desarrollo (Incremental, Iterativo, Ágil, etc), puesto que se han clasificado según la característica de seguridad a la que atienden, de esta manera se les puede identificar para la aplicación en general, o para requerimientos en particular.

- Incrementar la proactividad del equipo, al permitir identificar responsabilidades de sus integrantes en cada parte.

9.4.2 Beneficios esperados de la incorporación de controles en cada una de las etapas del ciclo de vida.

A continuación se mencionan los beneficios de la incorporación de controles de seguridad informática en cada una de las etapas del proceso de implementación del software:[72]

Análisis de requerimientos.

- Focalización de esfuerzos en mitigar riesgos reales con impacto crítico para el negocio, optimizando la inversión de recursos; gracias a la ejecución de un proceso de análisis y evaluación de riesgos para el establecimiento de los requisitos de seguridad.

- Reducción de los costos asociados a cambios en arquitectura y diseño, para atender en etapas avanzadas, requisitos de seguridad que no fueron contemplados en etapas tempranas del desarrollo.

[72] En la identificación de los beneficios se tiene en cuenta entre otras fuentes, el modelo de madurez para el aseguramiento del software, propuesto por OWASP.[30]

Diseño de arquitectura y diseño detallado.

- Confiabilidad en la robustez de la aplicación dado que desde la definición de la arquitectura se contemplan técnicas y buenas prácticas orientadas a atender los requerimientos de seguridad.
- Creación de arquitecturas y plataformas de referencia, que enriquecen las bases de conocimiento de los proyectos y las organizaciones.

Construcción, integración y pruebas.

- Identificación temprana de vulnerabilidades gracias a revisiones de código acordes a parámetros y lineamientos de codificación segura.
- Verificación de que la implementación de nuevos requerimientos no afecta el comportamiento de los controles de seguridad implementados, mediante la ejecución del conjunto de pruebas unitarias de los controles en cada versión.
- La realización de pruebas de seguridad en las etapas de integración y validación permitirá identificar vulnerabilidades causadas por la integración de componentes.

9.5 RECOMENDACIONES GENERALES.

En esta sección se presentan algunas recomendaciones que pueden ser consideradas en las diferentes partes de la guía, para una aplicación eficaz y que permita el máximo provecho posible.

- **Aplicación progresiva:** Los controles sugeridos pueden incluirse de manera progresiva, y dentro de un proceso de gestión de cambios, a medida que se van implementando nuevas funcionalidades.

- **Validación continua:** Definir un responsable que valide que los requisitos establecidos se están atendiendo al finalizar la ejecución de las actividades correspondientes a cada parte de la GICSI.

- **Creación del perfil de riesgo:** Documentar el proceso de análisis y evaluación de riesgos, con el fin de crear un perfil de riesgo y establecer políticas de aceptación y transferencia de los mismos, para incluirlos en la gestión de riesgos de la organización o del proceso de desarrollo como tal.

- **Evaluación de sobre costos:** Es importante que a partir de la evaluación de riesgos se establezca el impacto económico que tiene la materialización de un riesgo tanto para el cliente como para el proveedor, frente al sobre costo inicial de incorporar los controles de seguridad en las aplicaciones; y así poder tomar decisiones de inversión con mayor certidumbre.

- **Registro de los resultados de las actividades:** En la medida de lo posible, los resultados de las actividades deben documentarse, con el fin hacer seguimiento y control de las mismas.

9.6 PARTES DE LA GICSI.

A continuación se presentan las 6 (seis) partes de la guía de asociadas a los 6 (seis) procesos de implementación de software mencionados anteriormente, siguiendo la estructura descrita en el sección 7.2 Estructura principal del capítulo 7: Diseño de la GICSI.

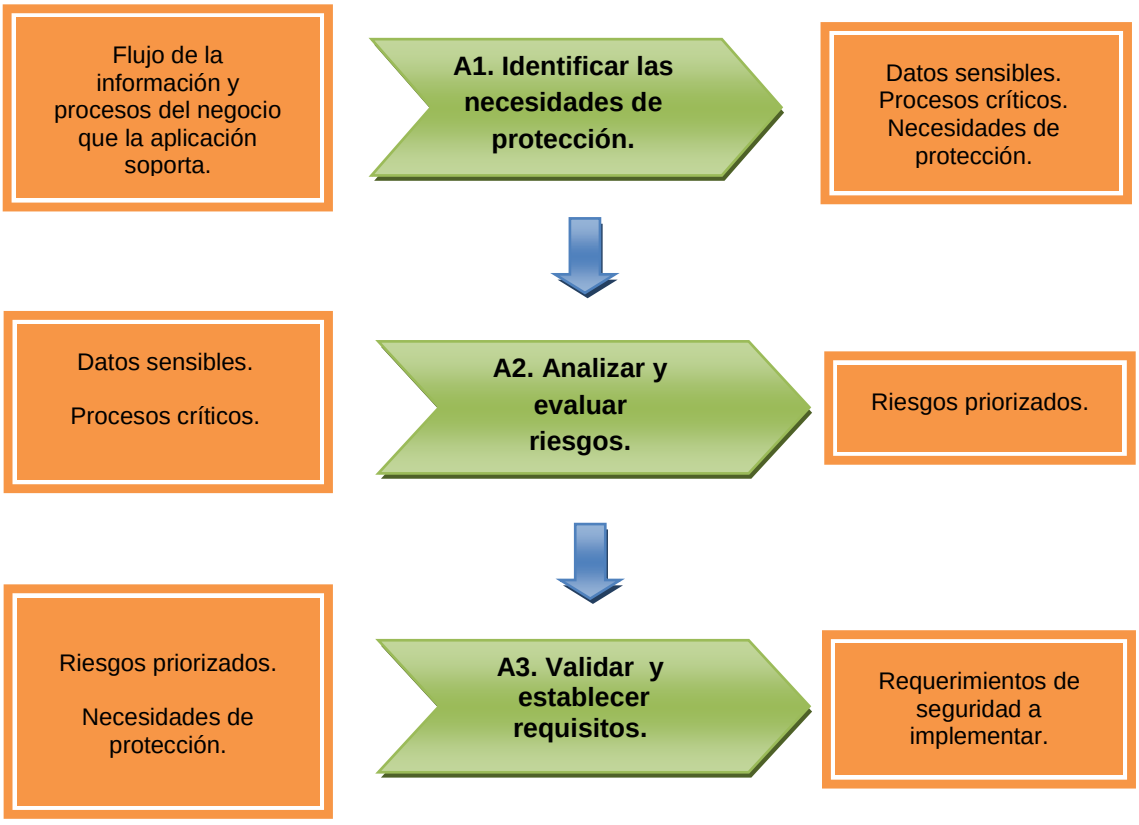
9.6.1 Parte 1: Análisis de requerimientos del software.[73]

Objetivo: Especificar claramente las necesidades de seguridad del software, asociadas a la disponibilidad, integridad y confidencialidad de la información que éste recibe, procesa y genera.

Cumplimiento.

ISO 27002: 4. Evaluación y tratamiento del riesgo, 7.2. Clasificación de la información, 10.10 Supervisión, 12.1. Requerimientos de seguridad de los sistemas.

COBIT: PO2.3 Esquema de clasificación de datos, PO9 Evaluar y administrar riesgos de TI, ME3 Garantizar el cumplimiento, DS5 Garantizar la seguridad de los sistemas.



Participantes: Líderes de proyecto, analistas funcionales, arquitectos de software

[73] Los objetivos de control de COBIT 4.1, y controles de la norma ISO/IEC 27002:2005, se mencionan como prácticas sugeridas puesto que ayudarán en gran porcentaje a definir los requerimientos de seguridad de la aplicación.

9.6.1.1 A1. Identificar necesidades de protección.[74]

Establecer las necesidades de protección que requieren los procesos críticos y la información sensible, en cuanto a confidencialidad, disponibilidad, integridad, trazabilidad y demás características de seguridad.[75]

T1.1 Identificar los procesos soportados por la aplicación y la información relacionada.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
1. ¿Cuáles son los principales procesos y funcionalidades del negocio que soporta la aplicación? 2. ¿Cuál es la información de entrada de los procesos de negocio? 3. ¿Qué información generan los procesos?	Conjuntamente con los dueños de los procesos y los dueños de la información, identificar: • Los procesos que van a ser soportados por la aplicación. • Información de entrada a los procesos. • Información procesada y generada por los procesos. • Información de entrada a los procesos que es proveída por terceros.

T1.2 Establecer la criticidad de los procesos identificados y la sensibilidad de la información relacionada.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
4. ¿Cuál es el nivel de disponibilidad que requiere cada uno de los procesos a los que da soporte la aplicación? 5. ¿En qué condiciones la información debe estar disponible para que los usuarios puedan acceder a ésta?	• Determinar conjuntamente con los dueños de los procesos y los dueños de la información, el nivel de disponibilidad requerido, mediante la identificación de consecuencias económicas, impacto en buena imagen de la empresa, entre otras; que puede traer la no disponibilidad del proceso o de la información.[76] • Identificar si existen condiciones

[74] Los objetivos de control de COBIT 4.1, y controles de la norma ISO/IEC 27002:2005, se mencionan como prácticas sugeridas puesto que ayudarán en gran porcentaje a definir los requerimientos de seguridad de la aplicación.
[75] Ver área de proceso PA10: Especificar necesidades de seguridad, de la norma ISO/IEC 21827:2008. [8].
[76] Un método para evaluar la importancia de la información y los procesos como activos de la organización, puede encontrarse en MAGERIT Versión 2. I-Método, sección 2.1.1 Paso 1: Activos. [24].

	contractuales de niveles de servicio, relacionadas con la disponibilidad de ejecución de los procesos y de acceso oportuno a la información.
6. ¿Qué información es de libre acceso? 7. ¿Qué información es confidencial?	Identificar que información: • Puede ser consultada por todos los usuarios. • Solo puede ser consultada por usuarios autorizados. • Que por exigencias legales o regulaciones gubernamentales debe ser de acceso público.
8. ¿Cuáles usuarios requieren autenticación? 9. ¿Cuáles son los niveles de autorización requeridos respecto de la ejecución de los procesos? 10. ¿Cuáles son los niveles de autorización requeridos respecto del acceso y modificación de la información?	Establecer que usuarios o grupos de usuarios pueden ejecutar cada proceso, tener en cuenta: • OWASP. Requerimientos de seguridad -2.A.Generar una matriz de control de acceso a los recursos y capacidades.[77] • COBIT, DS5.3 Administración de Identidad.[78] • ISO/IEC 27002:2005, 11.1 Requisitos del negocio para el control de acceso.[79]
11. ¿Cuáles son los requisitos de integridad de la información?	• Controles de aplicación de COBIT[51.] ▪ AC3 Chequeos de Exactitud, Integridad y Autenticidad. ▪ AC4 Integridad y Validez del Procesamiento ▪ AC6 Autenticación e Integridad de Transacciones. • Identificar reglas de negocio que estén relacionadas con la precisión y completitud de la información.
12. ¿Qué información va a registrarse para poder realizar trazabilidad de los procesos? 13. ¿Qué información va a registrarse para poder realizar la trazabilidad del acceso y modificación de los datos?	• Determinar conjuntamente con los dueños de los procesos y los dueños de la información, el nivel de trazabilidad requerido para los procesos y la información. • Identificar que información se va a registrar para efectos de auditorías, controles correctivos, controles detectivos, no repudio del origen. Tener en cuenta: ISO/IEC 27002:2005, 10.10.1 Registro de auditorías.[80]

[77] Ver OWASP Software Assurance Madurity Model. Versión 1.0. [30].
[78] Ver COBIT 4.1. [15].
[79] Ver ISO/IEC 27002:2005. Tecnología de la Información – Técnicas de Seguridad - Código para la práctica de la seguridad de la información. [7]
[80] Ver ISO/IEC 27002:2005. Tecnología de la Información – Técnicas de Seguridad - Código para la práctica de la seguridad de la información. [7]

T1.3 Determinar las necesidades de protección de la información y los procesos.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
14. ¿Cuáles reglamentaciones y normas internas de la organización, leyes y estándares relacionados con la seguridad informática, deben cumplir los procesos de negocio que el software soportará?[81]	• Identificar políticas de seguridad de la información establecidas por la organización. • Identificar si la información de negocio está sujeta a cumplimiento de leyes de de protección de datos personales, o reglamentaciones de entidades de control estatales. • Identificar si la información de negocio debe cumplir con normas y estándares internacionales como SOX[82], ISO/IEC 27002:2005, PCI-DSS[83], entre otros.
15. ¿Cuáles son los datos sensibles? 16. ¿Cuáles son los procesos críticos, a los cuales se les debe dar prioridad? 17. ¿Cuáles son las necesidades de protección de los datos sensibles y los procesos críticos?	A partir de la información recopilada en las tareas T1.1, T1.2 y T1.3, registrar: • Información sensible para la aplicación en las diferentes dimensiones de seguridad. • Procesos críticos en cuanto a autenticación, disponibilidad y trazabilidad. • Necesidades de protección de la información sensible y los procesos críticos.

9.6.1.2 A2. Analizar y evaluar riesgos.

Identificar los riesgos que van a ser mitigados, a partir de la información recopilada anteriormente y de la aplicación de un proceso de análisis y evaluación de riesgos.

[81] Ver práctica base BP.10.02. Identificar leyes aplicables, políticas y restricciones, de la norma ISO/IEC 21827:2008. [8]
[82] SOX: Ley Sarbanes – Oxley de Estados Unidos, creada para proteger al inversionista en empresas que cotizan en la bolsa. http://www.sec.gov/about/laws.shtml#sox2002.
[83] PCI-DSS: *Payment Card Industry Data Security Standard*.

T2.1 Identificar amenazas y vulnerabilidades

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
18. ¿Cuáles son las posibles amenazas a las que se encuentra expuesta la información que maneja la aplicación? 19. ¿Cuáles son las vulnerabilidades qué podrían ser explotadas, si las amenazas se materializan?	• MAGERIT Versión 2: Catálogo de Elementos. Sección 5 Amenazas.[84] • OWASP Top ten de riesgos en aplicaciones web.[85] • ISO/IEC 27005:2011, Gestión de Riesgos de Seguridad de la Información. Anexo C: Ejemplos de amenazas típicas, y Anexo D: Vulnerabilidades y métodos para evaluación de vulnerabilidades.[86] • STRIDE. Categorización de amenazas sugerida por Microsoft.[87]

T2.2 Identificar y evaluar los riesgos.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
20. ¿Cuál es la probabilidad de que las amenazas se materialicen y/o las vulnerabilidades sean explotadas? 21. ¿Cuál es el impacto técnico o daño sobre la información, si las amenazas se materializan o las vulnerabilidades se explotan? 22. ¿Cuáles son los riesgos y los niveles de riesgo asociados?	Si la empresa consultora o el cliente no tienen establecida una metodología de análisis y evaluación de riesgos, se puede hacer uso de una de las siguientes alternativas: • Metodología de valoración de riesgos de OWASP[88] • Modelado de Amenazas de Microsoft.[89] • ISO/IEC 27005:2011 Gestión de Riesgos de Seguridad de la Información.[90] • NIST 800:30: Guía para gestión de riesgos para los sistemas de tecnología de la información.[91]

[84] MAGERIT – Catálogo de elementos. [25]
[85] *OWASP Top 10 Application Security Risks* – 2013. [38]
[86] ISO/IEC 27005:2011 Tecnología de la información – Técnicas de seguridad. Gestión de Riesgos de seguridad de la información. [13]
[87] *The STRIDE Threat Model*. [18.
[88] *OWASP Risk Rating Methodology*. [32]
[89] *Improving Web Application Security*, Chapter 3 Threat Modeling.[22]
[90] ISO/IEC 27005:2011 Tecnología de la información – Técnicas de seguridad. Gestión de Riesgos de seguridad de la información. [13]
[91] *National Institute of Standards and Technology. US.- Risk Management Guide for Information Techonlogy Systems.* [27].

9.6.1.3 A3. Validar y establecer requisitos.

Establecer el listado de requisitos a implementar, a partir de las necesidades identificadas y de los riesgos a evitar y/o reducir.

T3.1 Definir los riesgos que van a evitarse y/o reducirse.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
23. ¿Cuáles son las opciones para el tratamiento de los riesgos identificados?	Conjuntamente con los dueños de proceso y los dueños de la información, y teniendo en cuenta la lista priorizada de riesgos, determinar[92]: • Riesgos a reducir. • Riesgos a evitar. • Riesgos a transferir. • Riesgos a aceptar.

T3.2 Identificar los requerimientos de seguridad.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
24. ¿Cuáles son los requisitos que se van a implementar?	Establecer el listado de requerimientos de seguridad que la aplicación debe atender, a partir del listado de riesgos a reducir y/o evitar, y de las necesidades de protección.
25. ¿Alguno de los requisitos establecidos presenta conflicto con requerimientos funcionales y no funcionales determinados previamente?	Identificar si los requisitos de seguridad anteriormente establecidos contradicen o impactan en otros requerimientos, como aquellos relacionados con rendimiento y/o usabilidad del software.
26. ¿Cada uno de los requerimientos establecidos, es técnicamente factible, alcanzable y verificable?	Evaluar si se requiere modificar o excluir algún requerimiento, por no ser técnicamente factible o difícil de verificar.

[92] Mayor información en la norma ISO/IEC 27005:2011 Gestión de Riesgos de Seguridad de la Información. Capítulo 9; Tratamiento de Riesgos de Seguridad de la Información. [13]

T3.3 Comunicar los requisitos a implementar.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
27. *¿Los requerimientos establecidos son entendidos claramente por todos los interesados?*[93]	• *Registrar en un documento los requisitos que se van a implementar y verificar su entendimiento por parte de los interesados.* • *Validar con los interesados, si es necesario incorporar dicho documento en un acuerdo de nivel de servicios.*

[93] Ver prácticas base BP.09.01 y BP.10.07 del estándar ISO/IEC 21827:2008 CMMI-SEC [8].

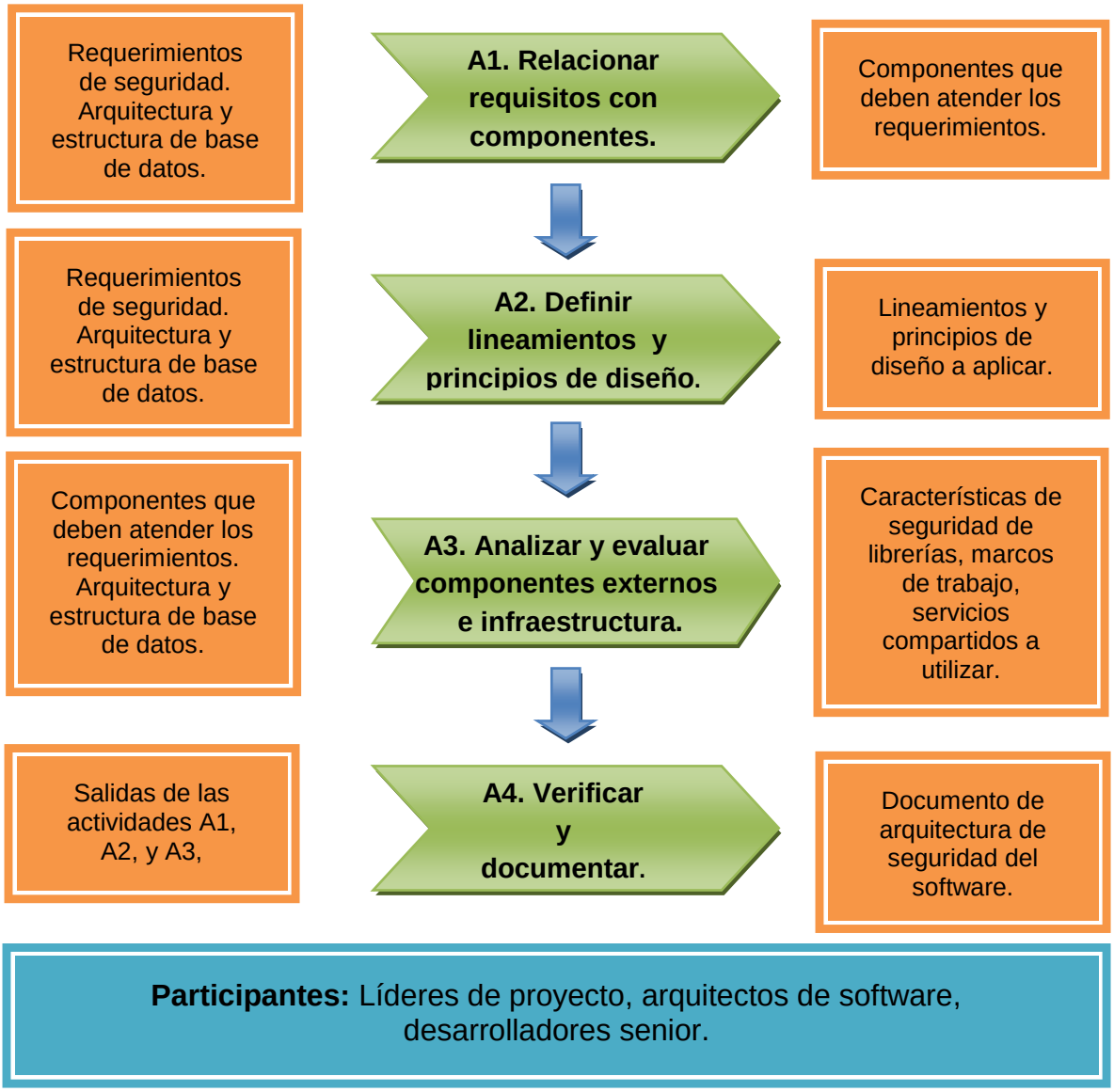
9.6.2 Parte 2: Diseño de la arquitectura del software.

Objetivo: Establecer cómo la arquitectura del software va a satisfacer los requerimientos de seguridad.

Cumplimiento.

ISO 27002: 10.10 Supervisión, 11.6 Controles de acceso a las aplicaciones y a la información, 12.2 Procesamiento correcto de las aplicaciones, 12.3 Controles criptográficos.

COBIT: AI2.1 Diseño de Alto Nivel, AI2.4 Seguridad y Disponibilidad de las Aplicaciones, DS5 Garantizar la seguridad de los sistemas, PO2.4 Administración de Integridad, PO8.3



Participantes: Líderes de proyecto, arquitectos de software, desarrolladores senior.

9.6.2.1 A1. Relacionar requisitos con componentes.

Identificar que componentes de alto nivel deben atender los requerimientos de seguridad.

T1.1 Establecer los componentes en los que se incorporan los controles de seguridad que están asociados a los requerimientos establecidos.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
28. ¿Qué componentes son los encargados de cubrir cada uno de los requerimientos establecidos?	• A partir del listado de requerimientos a implementar y de los componentes de la aplicación identificados, establecer en que componentes se incorporan los controles necesarios para atender los requerimientos de seguridad definidos. • Esta relación puede establecerse teniendo en cuenta que los requerimientos establecidos en la Parte 1, están clasificados en las diferentes dimensiones de seguridad, a saber: ▪ Confidencialidad. ▪ Autenticación. ▪ Integridad. ▪ Disponibilidad. ▪ Trazabilidad. • Registrar en el documento de arquitectura las nuevas responsabilidades para cada componente involucrado.

9.6.2.2 A2. Definir lineamientos y principios de diseño.

Establecer lineamientos, principios y patrones de diseño seguro a tener en cuenta.

T2.1 Establecer los principios de diseño a aplicar.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
29. ¿Cuáles son los principios de diseño seguro que se van a adoptar?	Entre los principios de diseño seguro a tener en cuenta se encuentran: [94] • Mantener el diseño de los controles de seguridad lo más simple posible. • Diseño abierto, evitar seguridad por oscuridad. • Fallar de manera segura. • Valores predeterminados seguros. • Ejecución con mínimos privilegios. • Controles por oposición. • Equilibrio entre los controles de seguridad, la usabilidad y el rendimiento de la aplicación. • Defensa en profundidad. • Mantener los datos, los ejecutables y la configuración separados. • Reducir la superficie de ataque: Minimizar el número de entradas y salidas.

T2.2 Identificar los patrones de diseño a aplicar.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
30. ¿Cuáles son los patrones de diseño orientados a la seguridad, que se van a aplicar?	A continuación se presentan algunos patrones a contemplar, a nivel de arquitectura: • Separación y reducción de privilegios[95]. • Inicio único de sesión para el manejo de la autenticación y la identificación de los usuarios. • Modelo de autorización para separación de funciones. Ver modelos descritos por Microsoft[96]. • Modelo centralizado de manejo de registro de errores y trazabilidad. • Implementar arquitecturas multi-capa.

[94] Principios de diseño extraídos del libro Writing Secure Code. Chapter 3: Security Principles to Live By. [6], y de la publicación *"The Ten Best Practices for Secure Software Development"* del *"International Information Systems Security Certification Consortium"*. [43]

[95] Existen diferentes patrones a nivel de arquitectura asociados al manejo de los privilegios otorgados a cada componente del software, como lo son: La descomposición desconfiada, la separación de privilegios y la delegación en el kernel, una descripción detallada de los mismos puede encontrarse en el reporte técnico Secure Design Patterns de la Universidad de Carnegie Mellon. [3].

[96] Design Guidelines for Secure Web Applications. Microsoft. [21].

T2.3 Verificar y registrar que principios y patrones van a implementarse.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
31. ¿Los lineamientos de diseño establecidos afectan los requerimientos de seguridad?	• Evaluar si los lineamientos de diseño pre-establecidos no impactan en el cubrimiento de los requerimientos de seguridad definidos. • Documentar los resultados de esta actividad.

9.6.2.3 A3. Analizar y evaluar componentes externos e infraestructura.

Identificar, analizar y evaluar de qué manera componentes externos, tales como librerías, marcos de trabajo, servicios compartidos, y características de infraestructura contribuyen al cumplimiento de los requerimientos de seguridad.

T3.1 Identificar las características de seguridad de la plataforma en donde se ejecuta la aplicación.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
32. ¿Cuáles son las características de seguridad presentes en la plataforma e infraestructura, en la cual se ejecuta la aplicación?	• *Identificar las características de seguridad de los siguientes componentes, que se consideran relevantes para el cumplimiento de los requerimientos de seguridad.* ▪ *Servidores de aplicaciones* ▪ *Motores de bases de datos.* ▪ *Topología de red.* ▪ *Sistemas operativos.* • *Evaluar restricciones impuestas por el entorno de ejecución y/o por aplicaciones de terceros con las que se va a interactuar.*

T3.2 Identificar los servicios compartidos con los que se integra la aplicación.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
33. ¿Cuáles son los servicios compartidos relacionados con la seguridad, con los que se va a integrar la aplicación?	*Existen servicios compartidos que cubren varios de los principios y patrones mencionados en la actividad anterior, entre ellos se encuentran:* • *Servicios de inicio de sesión único y centralización de la autenticación como lo son: CAS, Kerberos, Microsoft Single Sign-On Service.* • *Implementaciones de servicios de directorios para la centralización de la autorización, entre los que se encuentran: Microsoft Active Directory, OpenLDAP, ApacheLDAP.*
34. ¿Cuáles son las características de los servicios compartidos que soportan el cumplimiento de los requerimientos?	*Identificar las características de seguridad y las posibles vulnerabilidades, de los servicios compartidos.*

T3.3 Identificar las características de seguridad de las librerías y/o marcos de trabajo que van a ser utilizados en la construcción de la aplicación.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
35. ¿Cuáles son las librerías y marcos de trabajo que se van a integrar a la aplicación?	*Identificar los componentes de terceros que van a cubrir funcionalidades como:* • *Conexiones a base de datos.* • *Manejo de transacciones.* • *Autenticación.* • *Autorización.* • *Encripción.* • *Registro de trazabilidad.* • *Validación de entradas, entre otros.*
36. ¿Cuáles son las características de seguridad de las librerías y marcos de trabajo de terceros, que se integran con la aplicación?	• *Identificar y evaluar las características de seguridad con las que cuentan los componentes.* • *Establecer listado de vulnerabilidades conocidas.*
37. ¿Cuáles son los componentes que se utilizan para integrar la aplicación con los servicios compartidos identificados?	*En la integración con servicios compartidos, tener en cuenta:* • *Java Authentication and Authorization Service - JAAS.* • *Spring (LDAP, security, remoting, etc), OWASP Enterprise Security API, entre otros.*

9.6.2.4 A4. Verificar y documentar.

Verificar los resultados de los procesos ejecutados y documentarlos.

T4.1 Evaluar las características de seguridad de los componentes externos.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
38. ¿Las características identificadas permiten cubrir los requerimientos establecidos?	• Identificar si las características que proporcionan los componentes externos son suficientes para cubrir los requerimientos. • Seleccionar los componentes adecuados, acorde al análisis de las características y vulnerabilidades identificadas.

T4.2 Registrar y comunicar las decisiones.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
39. ¿La integración de los componentes externos es conocida y entendida por el equipo de desarrollo?	• *Registrar en el documento de arquitectura de la aplicación, los componentes externos seleccionados, identificando los requerimientos a los que responde.* • *Elaborar guías de apoyo, que describan la forma adecuada de integrar dichas librerías y/o componentes a la aplicación.*

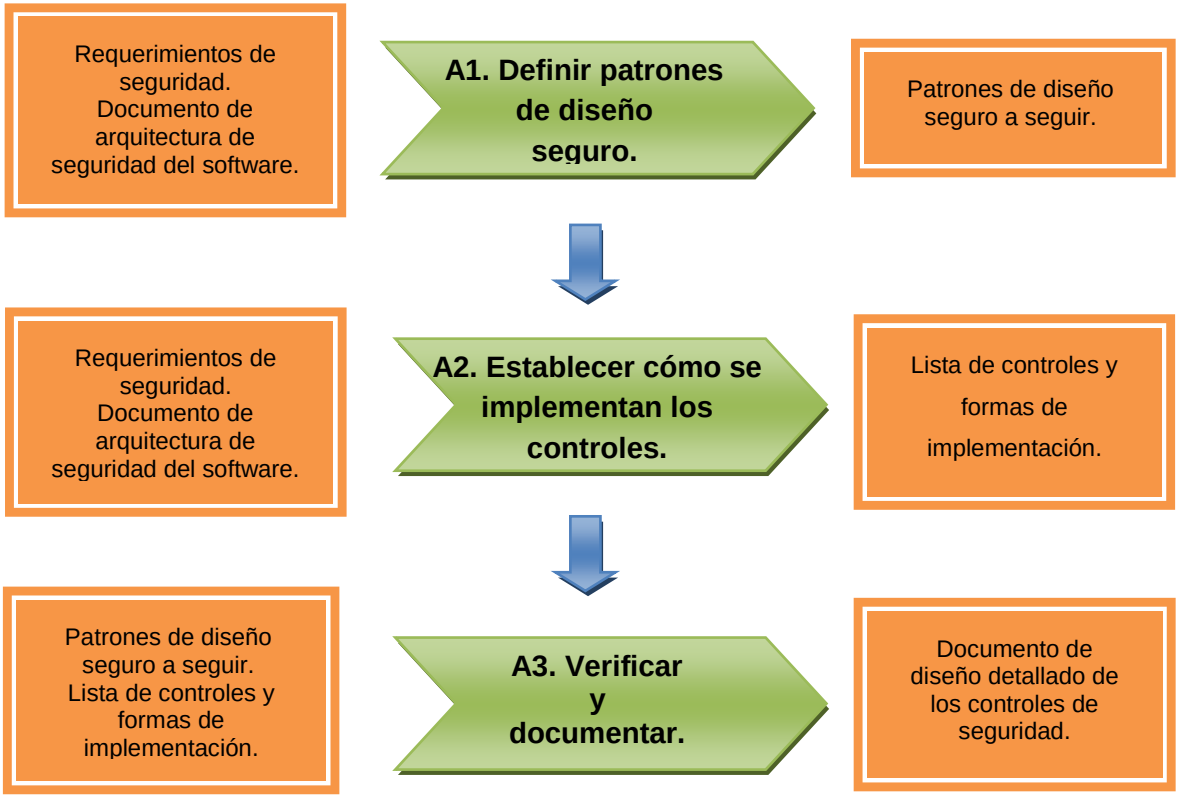
9.6.3 Parte 3: Diseño detallado del software.

Objetivo: Establecer como el software va a satisfacer los requerimientos de seguridad, determinando los controles a implementar para reducir y/o evitar los riesgos identificados, acorde a los lineamientos definidos en la arquitectura.

Cumplimiento.

ISO 27002: 10.10 Supervisión, 11.6 Controles de acceso a las aplicaciones y a la información., 12.2 Procesamiento correcto de las aplicaciones, 12.3 Controles criptográficos.

COBIT: AC1 Preparación y autorización de información fuente, AC2 Recolección y entrada de información fuente, AC3 Chequeos de exactitud, integridad y autenticidad, AC4 integridad y validez del procesamiento, AC5 Revisión de salidas, reconciliación y manejo de errores, AC6 Autenticación e integridad de transacciones, DS5.3 Administración de identidad, DS5.18 Administración de llaves criptográficas, DS5.11 Intercambio de datos sensitivos, DS11.6 Requerimientos de seguridad para la administración de datos.



Participantes: Arquitectos de software, desarrolladores senior.

9.6.3.1 A1. Definir patrones de diseño seguro.

Establecer los patrones de diseño que se van a aplicar en la construcción de la aplicación.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
40. ¿Qué patrones de diseño se aplican en la construcción de la aplicación?	Entre los patrones de diseño seguro se pueden encontrar[97]: • Secure factory. • Secure strategy factory. • Secure builder factory. • Secure chain of responsability. • Secure state machine. • Secure visitor.

9.6.3.2 A2. Establecer cómo se implementan los controles.

Elaborar el diseño de la implementación de los controles de seguridad que se van a incorporar en la aplicación.[98]

T2.1 Definir el diseño de los controles de autenticación[99]

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
41. ¿Cuál es el mecanismo de autenticación? 42. ¿Cuáles recursos u operaciones requieren autenticación, cuales no?	• Tener en cuenta mecanismos como la autenticación fuerte que mezcla dos de tres factores de autenticación a saber: ▪ Algo que se tiene (tokens). ▪ Algo que se conoce (nombre de usuario, contraseña). ▪ Algo que se es (controles biométricos). • Definir un único mecanismo de autenticación para la protección de los recursos y servicios. • Solicitar la re-autenticación del

[97] Se pueden encontrar más detalles en el reporte técnico Secure Design Patterns de la Universidad de Carnegie Mellon. [4].
[98] Ver lista de controles, sugerida en OWASP *Secure Coding Practices Quick Reference Guide*. [31].
[99] Ver *OWASP Top 10 Application Security Risks – 2013*. A2 - Broken Authentication and Session Management.[38] y la guía para autenticación OWASP. [33]

	usuario para la ejecución de procesos críticos. • Tener en cuenta la guía para autenticación OWASP.[100]
43. ¿En donde se almacenan y cómo se protegen las contraseñas o credenciales del usuario? 44. ¿Cómo es la política para el manejo de las contraseñas? *45. ¿Cómo se mantiene la identidad del usuario a través de las diferentes interacciones entre los componentes?*	• Utilizar cifrado de contraseñas no reversible, en cuyo caso se establece el uso de algoritmos de hash como el SHA-256. • Incluir en la política de contraseñas: Expiración, bloqueos administrativos, contraseñas fuertes, alertas. • Transmitir credenciales sobre canales seguros. Ver mejores prácticas para el manejo de contraseñas, en la guía para autenticación OWASP.[79]
46. ¿Cómo se maneja la autenticación de los servicios compartidos, como bases de datos, servicios de directorios, repositorios, etc? 47. ¿Cómo se almacenan las cadenas de conexión a la base de datos?	• Definir cuentas para cada uno de los servicios siguiendo el principio de mínimo privilegio. • Encriptar las cadenas de conexión y restringir el acceso a esta información. • Entre los mecanismos de autenticación a base de datos se encuentran[101]: ▪ Autenticación directa con la base de datos para identificar y autenticar a los usuarios. ▪ Autenticación externa a través del sistema operativo o servicios de red. • Proxy de autenticación, en donde un servidor intermedio autentica y asume la identidad del usuario.
48. ¿Cómo se realiza el manejo de sesiones de la aplicación?	Tener en cuenta las siguientes sugerencias: [102] • Limitar el tiempo de vida de la sesión. • Proteger el estado de la sesión de accesos no autorizados. • Hacer uso de los manejadores de sesión de los marcos de trabajo utilizados, mantener versiones actualizadas, evitar implementaciones propietarias.

[100] *OWASP Guide to authentication.* [33]
[101] Ver mecanismos de autenticación descritos por Microsoft en "*How do you authenticate with the database?*" [22] y por Oracle en "*Administering Authentication*". [39]
[102] Para mayor información ver manejo de sesiones propuesto por OWASP. [34]

T2.2 Definir el diseño de los controles de autorización[103].

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
49. ¿Cómo se controla el acceso de los usuarios finales, a las funcionalidades de negocio, servicios e información? 50. ¿Cómo es el control de acceso desde las aplicaciones a la base de datos? 51. ¿Qué privilegios tienen las aplicaciones sobre recursos del sistema cómo archivos, servicios de red, entre otros?	• Establecer roles, usuarios y privilegios, a partir de la matriz de control de acceso definida en la Tarea 1.2 de la Parte 1 de la presente guía, siguiendo principios de diseño como la separación de funciones y mínimos privilegios. • Denegar cualquier acceso de forma predeterminada, y validar privilegios explícitos para acceder a cada funcionalidad. • Aplicar los controles de acceso con la misma rigurosidad en las diferentes capas de la aplicación (negocio, vista, controladores), siguiendo el principio de defensa en profundidad. • Tener en cuenta productos y servicios compartidos que pueden encargarse de esta funcionalidad.

T2.3 Definir el diseño de los controles para la confidencialidad de la información.[104]

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
52. ¿Qué tipo de algoritmos se usan para encriptar la información que se definió como confidencial? 53. ¿En dónde se almacenan y cómo se protegen las llaves criptográficas?	• Identificar los algoritmos de encripción que van a ser utilizados: ▪ 3DES, AES con tamaños de clave entre 128 y 256 bits, para criptografía simétrica. ▪ RSA, Diffie-Hellman, con tamaños de clave entre 1024 bits y 2048 bits, para criptografía asimétrica. • Encriptar la información, tan cerca como sea posible de la fuente que la genera. • Hacer uso de librerías de encripción de terceros conocidas, y probadas, evitando las implementaciones propias; tener en cuenta las siguientes opciones:

[103] Ver *OWASP Top 10 Application Security Risks – 2013*. A7 - Missing Function Level Access Control, y A5 – Security Misconfiguration**.** [38]

[104] Ver *OWASP Top 10 Application Security Risks – 2013*. A6 – Sensitive Data Exposure, A1 – Injection y A3 – Cross-Site Scripting (XSS). [38] y guía para criptografía de OWASP. [35]

	▪ Bouncy Castle Crypto APIs, para java y C#. ▪ Java Cryptography Extension (JCE). ▪ Data Protection API de Microsoft. • Restringir el acceso al almacén de llaves o a la ubicación de las llaves en el sistema de archivos, mediante disminución de privilegios; si la aplicación lo requiere, puede evaluarse la utilización de hardware criptográfico. • Minimizar la cantidad de componentes que deben acceder directamente a las llaves. • Evitar almacenar las llaves en el código fuente.
54. ¿Cómo se protege la información sensible que se transmite a través de redes?	• Encriptar la información sensible que se va a transmitir, o encriptar el canal de comunicaciones.[105] • La encripción del canal de comunicaciones puede realizarse haciendo uso del protocolo TLS - Transport Layer Security (Actualización del SSL – Secure sockets layer.). • Evitar enviar información sensible a través del método GET de HTTP. • Evitar almacenar información sensible en cookies persistentes.

T2.4. Definir el diseño de los controles para proteger la integridad de la información.[106]

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
55. ¿En cuáles componentes se implementan los controles de validación de entradas de información? 56. ¿Cómo se realiza la validación de las entradas de información en los diferentes componentes?	• Identificar las fronteras de confianza de su aplicación, determinando interfaces con terceros e interacciones con redes públicas. • Identificar los puntos de entrada desde la vista, servicios web, base de datos, y otros componentes. • Determinar la estrategia de validación a aplicar en todos los componentes, se sugiere el patrón de validación positiva.

[105] Ver sección "*Sensitive Per User Data*", del documento "*Design Guidelines for Secure Web Applications*" de Microsoft. [21].
[106] Para mayor información ver la guía de validación de datos – OWASP. [28]

	• Aplicar las validaciones requeridas dependiendo de las responsabilidades de cada capa de la aplicación, evitando enviar datos inválidos a niveles subsiguientes, innecesariamente. • Definir las validaciones por tipos de datos, e implementarlas de manera consistente en todos los componentes que lo requieran, siguiendo el principio de defensa en profundidad. • Definir controles para contrarrestar los principales ataques de inyección de SQL, LDAP, XML y comandos del sistema operativo, los cuales explotan vulnerabilidades en la validación de las entradas.[107]
57. ¿En cuales componentes se incorporan los controles de integridad de la información? 58. ¿En cuales componentes se incorporan las validaciones de reglas de negocio? 59. ¿Cómo se implementan los controles de integridad de la información?	• Implementar los controles de integridad en aquellos puntos donde la información se transfiera desde una zona más confiable a una menos confiable. • Determinar el control de integridad a implementar, entre los que se pueden encontrar: HMAC – SHA256[108]), firma digital, encripción PGP[109], entre otras. • Tener en cuenta mecanismos de integridad en base de datos como son la integridad de entidad, la integridad referencial y el manejo de transacciones. • Implementar las reglas de negocio necesarias para evitar que información manipulada se tome como válida.

T2.5 Definir lineamientos para evitar posibles denegaciones de servicios.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
60. ¿Cómo maneja la aplicación los	• Tener en cuenta el manejo de:

[107] Ver *OWASP Top 10 Application Security Risks* – 2013. A6 – Sensitive Data Exposure, A1 – Injection y A3 – Cross-Site Scripting (XSS). [38]

[108] HMAC - Hash-based Message Authentication Code: Utilizar hashes con tamaño entre 128 y 256 bits.

[109] PGP - Pretty Good Privacy es un grupo de aplicaciones desarrolladas por Philip R. Zimmermann, que hace uso de técnicas de hashing, criptografía simétrica y asimétrica; para la encripción, desencripción y firma digital de datos.

recursos disponibles?	▪ Condiciones de concurrencia. ▪ Consumo de recursos de memoria y procesamiento. ▪ Pool de conexiones para la gestión de sesiones de la base de datos. ▪ Manejo de archivos y espacio en disco. • Asegurar que las funcionalidades no autenticadas o desprotegidas, utilicen la menor cantidad de recursos para prevenir un ataque de denegación de servicio.

T2.6 Definir el manejo de la auditoría de eventos y la trazabilidad de las operaciones.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
61. ¿Cuáles transacciones se auditan? 62. ¿Qué información de seguimiento se registra?	• Registrar quién, cuándo y que operación se ejecutó. • Identificar el usuario responsable de la ejecución de las operaciones a través de las diferentes capas de la aplicación. • Preservar y proteger los logs de auditoría de modificaciones y acceso no autorizado.

T2.7 Definir el manejo de errores y excepciones.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
63. ¿Cómo se manejan las condiciones de error y excepciones generadas por la aplicación? 64. ¿Cómo se manejan las transacciones activas cuando se presenta un error fatal?	• Capturar y manejar excepciones, haciendo uso de los manejadores que proveen los lenguajes utilizados. • Establecer lineamientos para el manejo de errores fatales, con el fin de que la aplicación falle de manera segura.
65. ¿Cómo se registran y presentan los mensajes de error?	• *Evitar registrar en los logs datos sensibles e información confidencial.* • *Registrar los errores en el log, con la información suficiente que permita hacer el seguimiento adecuado.* • *Evitar revelar información detallada de la aplicación en los mensajes de error que se le presentan al usuario, utilizar mensajes de error genéricos.* • *Registrar intentos fallidos de autenticación y fallas de control de*

	acceso. • *Hacer uso de marcos de trabajo de logging que permitan definir de manera centralizada el formato con el que se registrarán los eventos, tipos de eventos (Error, depuración, información, advertencia, etc.), el tamaño y cantidad de archivos de log que se generarán.*

9.6.3.3 A3. Verificar y documentar.

Verificar los resultados de los procesos ejecutados y documentarlos.

T3.1 Verificar el diseño.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
66. ¿El diseño propuesto atiende los requerimientos de seguridad y se ajusta a la arquitectura de diseño definida?	• Verificar mediante una matriz de trazabilidad si el diseño propuesto cubre cada uno de los requerimientos de seguridad establecidos. • Verificar que el diseño propuesto no va en contra de los principios y lineamientos definidos en la arquitectura de seguridad de la aplicación. • Validar que las estrategias propuestas no van en contra de requerimientos funcionales, o requerimientos no funcionales relacionados con rendimiento y usabilidad.

T3.2 Registrar y comunicar las decisiones.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
67. ¿Las decisiones de diseño, herramientas y estrategias son conocidas y entendidas por el equipo de desarrollo?	• *Registrar en el documento de diseño de la aplicación, los controles, estrategias y herramientas a utilizar, junto con la matriz de trazabilidad.* • *Verificar que los desarrolladores que construirán la aplicación comprenden las decisiones y lineamientos establecidos.*

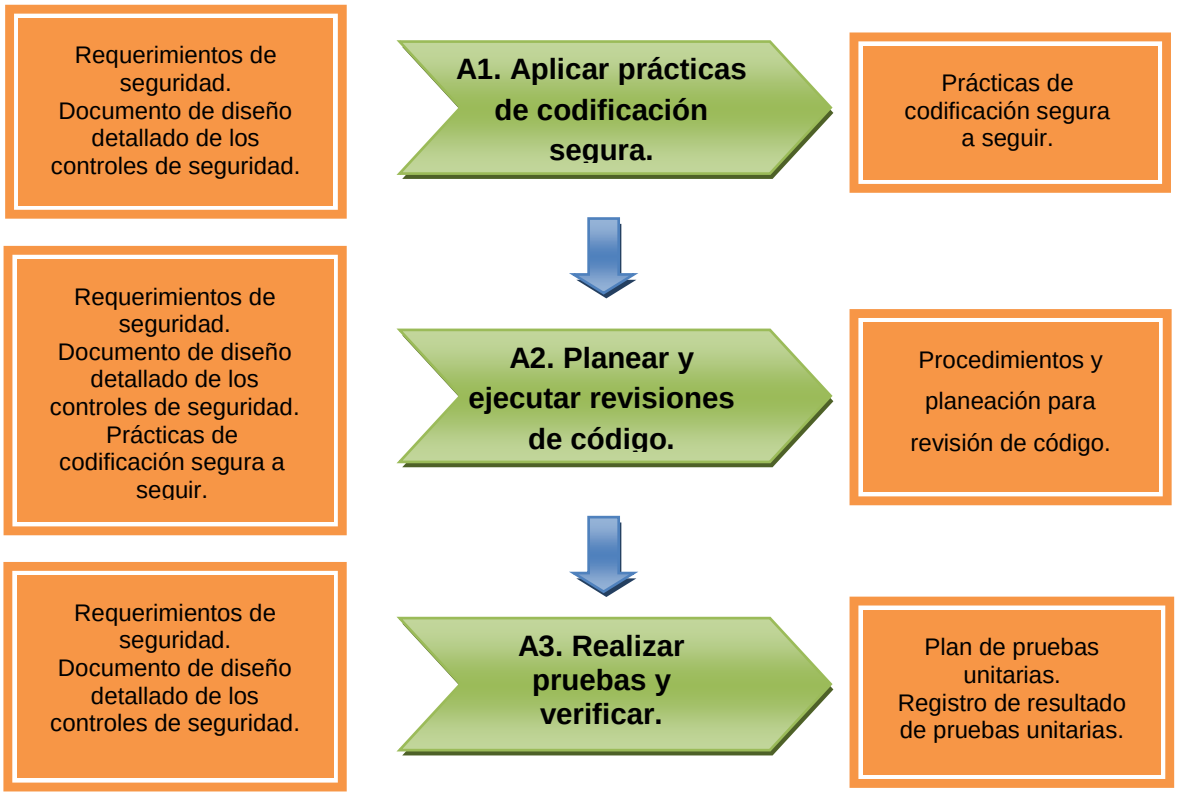
9.6.4 Parte 4: Construcción del software.

Objetivo: Construir los componentes de software que implementan los requerimientos de seguridad y verificar que se está aplicando el diseño definido.

Cumplimiento*.

ISO 27002: 12.4.2 Protección de los datos de prueba del sistema, 15.2.2 Chequeo del cumplimiento técnico.

COBIT: DS5.5 Pruebas, Vigilancia y Monitoreo de la Seguridad.



Participantes: Arquitectos de software, desarrolladores.

*Aplican los puntos de cumplimiento relacionados en la Parte 3.

9.6.4.1 A1. Aplicar prácticas de codificación segura.

Seleccionar y aplicar pautas de codificación segura acorde a los lenguajes de programación utilizados.

T1.1 Seleccionar y aplicar practicas de codificación segura.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
68. ¿Cuáles lineamientos y estándares de codificación deben seguirse?	Seleccionar prácticas de codificación acorde al lenguaje de programación a utilizar, tener en cuenta los siguientes recursos: • Pautas para codificación segura en Java.[110] • Estándares de codificación segura propuestos por el Instituto de Ingeniería del Software de la Universidad de Carnegie Mellon.[111] • Pautas de codificación segura de Microsoft.[112]

9.6.4.2 A2. Planear y ejecutar revisiones de código.[113]

Diseñar, planear y ejecutar revisiones de código.

T2.1 Identificar puntos críticos y definir criterios de revisión.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
69. ¿Cuáles son las funcionalidades críticas que requieren una revisión detallada de la implementación?	*Definir cuales requerimientos o funcionalidades críticas de seguridad requieren de una revisión de código detallada.*

T2.2 Definir procedimientos para la realización de la revisión de código.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
70. ¿Qué criterios de verificación se van a tener en cuenta? *71. ¿De qué manera se realizan las revisiones de código?*	• *Elaborar listas de chequeo, acorde a los puntos críticos, prácticas de codificación segura, y otras listas de chequeo pre-establecidas. Se pueden tener en cuenta las*

[110] *Secure Coding Guidelines for the Java Programming Language.*[40]
[111] *CERT Secure Coding Standards.* [45]
[112] *Secure Coding Guidelines*, Microsoft. [20]
[113] Actividad deseada pero no necesaria.

	siguientes referencias: ▪ *Revisión de código Microsoft.*[114] ▪ *OWASP Secure Coding Practices Quick Reference Guide.*[115] • *Seleccionar y aplicar herramientas análisis de código*[116]*, capacitar a los desarrolladores sobre el uso que deberían darle a la misma. Se pueden tener en cuenta los siguientes recursos*[117]*:* ▪ *Owasp Code Crowler.* ▪ *IBM Security AppScan Source.* ▪ *Sonar.* ▪ *Static Source Code Analysis Tools, sugeridas por el Instituto de Ingeniería del Software de la Universidad de Carnegie Mellon.*[118] ▪ *Revisar el código por pares.*

T2.3 Programar las revisiones de código.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
72. ¿En qué momento se realizan las revisiones de código? *73. ¿Quiénes son los responsables de analizar el código acorde a las funcionalidades críticas definidas?*	• *Definir el momento del proceso de construcción para aplicar los procedimientos de revisión seleccionados, previo a la liberación de los componentes.* • *Establecer quienes realizarán el análisis de código. La revisión por pares y la aplicación de las listas de chequeo obedecen al esquema de control por oposición.*

T2.4 Realizar las revisiones de código y documentar los resultados.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
74. ¿La implementación se está realizando siguiendo los principios de codificación segura establecidos? *75. ¿La implementación de los controles*	• *Aplicar los procedimientos establecidos, analizar los resultados, identificar no conformidades y registrar resultados.* • *Analizar los resultados con el fin de*

[114] *Code Review* Microsoft [19].
[115] *OWASP Secure Coding Practices Quick Reference Guide*. [31]
[116] En esta parte se sugieren herramientas de análisis estático de código, existen también herramientas de análisis dinámico, que se aplican en tiempo de ejecución y que son muy útiles para recopilar información durante la realización de pruebas de rendimiento y estrés.
[117] Se puede encontrar mayor información en la sección de recursos relacionados.
[118] *Static Source Code Analysis Tools.*[44]

de seguridad se ajusta al diseño definido?	*mejorar tanto la codificación como los procedimientos de revisión.*

9.6.4.3 A3. Realizar pruebas unitarias y verificar.

Implementar y ejecutar pruebas unitarias para verificar el cumplimiento de los requerimientos de seguridad.

T3.1 Construir y ejecutar las pruebas unitarias.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
76. ¿Qué pruebas unitarias van a realizarse? 77. ¿Qué datos de prueba se requieren?	• Diseñar casos de prueba que permitan verificar el comportamiento de las unidades de software respecto a los requerimientos y lineamientos de diseño establecidos. • Hacer validación tanto positiva como negativa del comportamiento de los controles de seguridad. • Utilizar marcos de trabajo de pruebas unitarias, que permitan crear un conjunto general de pruebas que verifiquen el comportamiento de la aplicación respecto a [119]: ▪ Autenticación y autorización. ▪ Encripción. ▪ Validación de entradas. ▪ Auditoria. • Manejo de errores y excepciones, entre otros.

T3.2 Analizar y registrar el resultado de las pruebas unitarias.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
78. ¿La implementación realizada cumple con los requerimientos de seguridad? 79. ¿Los resultados de las pruebas unitarias son satisfactorios y suficientes?	• Verificar contra la matriz de trazabilidad que la implementación se ajusta a los requerimientos de seguridad establecidos. • Evaluar los resultados de las pruebas y registrarlos. • Evaluar si las pruebas unitarias cubren los requerimientos establecidos e identificar mejoras. • Registrar los resultados.

[119] Ver capítulo Developers' Security Tests de Owasp Testing Guide. [36].

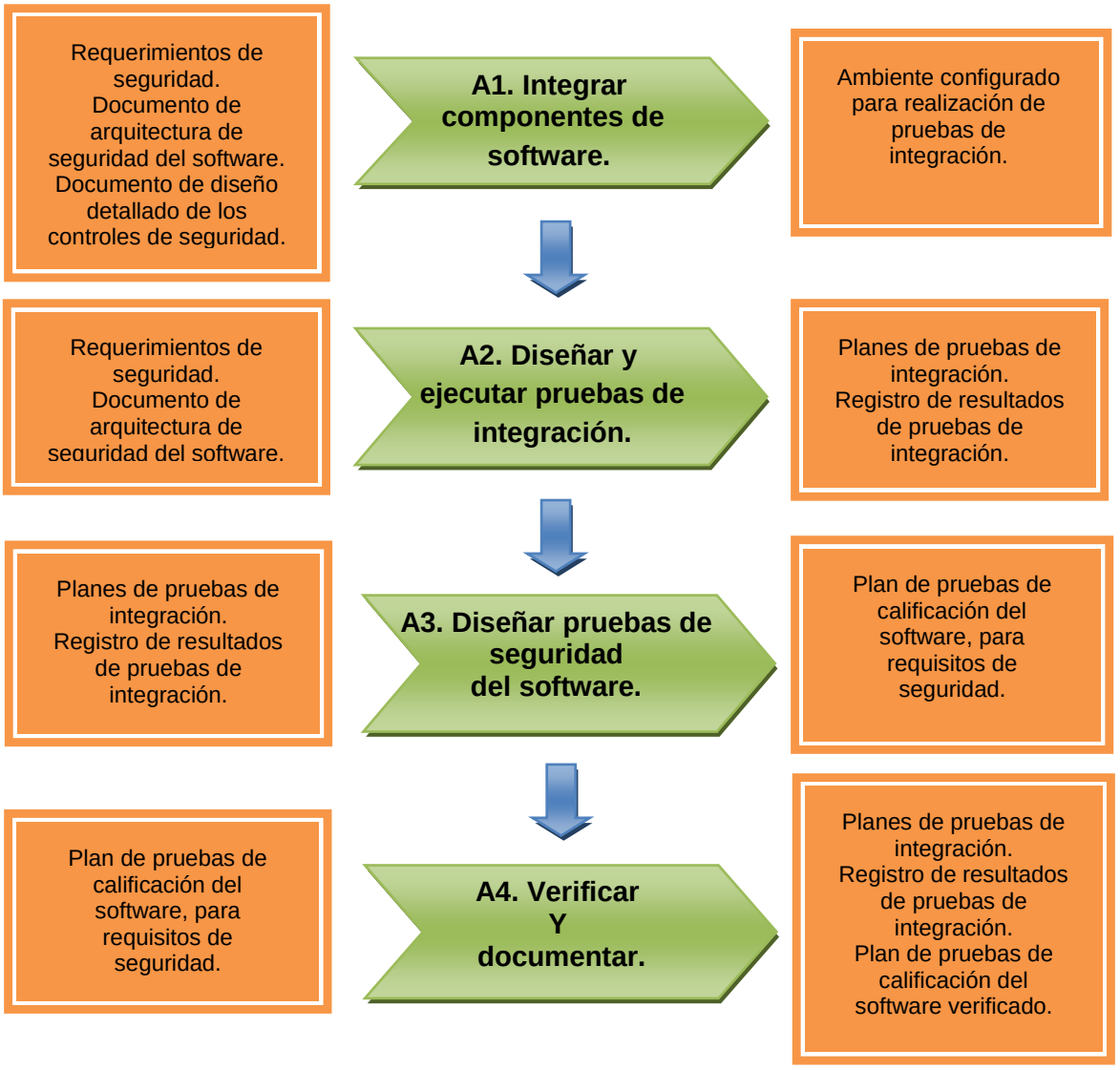
9.6.5 Parte 5: Integración del software.

Objetivo: Integrar los componentes de software acorde al diseño definido, en una plataforma operativa equivalente, en donde se puedan verificar que se están cumpliendo los requerimientos de seguridad establecidos.

Cumplimiento.

ISO 27002: 10.3.2 Aceptación del sistema.

COBIT: DS5.5 Pruebas, Vigilancia y Monitoreo de la Seguridad, AI3.4 Ambiente de Prueba de Factibilidad. AI7.2 Plan de Prueba



Participantes: Arquitectos, desarrolladores, analistas funcionales.

9.6.5.1 A1. Integrar componentes de software.

Realizar la integración de cada uno de los componentes y desplegar la aplicación en un ambiente operativo equivalente.

T1.1 Realizar la integración de los componentes.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
80. ¿En el proceso de integración se están respetando los lineamientos y principios establecidos en el documento de arquitectura de seguridad del software?	• Realizar la integración de los componentes de software entre sí, y con otros sistemas o servicios compartidos, teniendo en cuenta los lineamientos establecidos en las partes dos (2) y tres (3) de la presente guía.

T1.2 Preparar ambiente de pruebas.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
81. ¿En cuál ambiente se realizan las pruebas de integración? 82. ¿Cuáles servicios compartidos, componentes y librerías deben instalarse en el ambiente de pruebas? 83. ¿La configuración del ambiente de pruebas es igual o lo más parecida posible al ambiente operativo?	• Mantener separado el ambiente de desarrollo del ambiente para pruebas de integración. • Preparar el ambiente de pruebas de integración con los recursos necesarios para lograr simular de manera fiel el ambiente operativo en el que se desplegará la aplicación. • Tener en cuenta servicios compartidos, librerías, servidores versiones y configuraciones requeridas.

9.6.5.2 A2. Diseñar y ejecutar pruebas de integración.

Establecer y ejecutar planes de prueba que permitan verificar el comportamiento de la aplicación después de la integración, respecto a los requerimientos de seguridad establecidos.

T2.1 Diseñar un plan de pruebas de integración.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
84. ¿Qué tipo de pruebas se van a realizar? 85. ¿Qué datos de prueba se requieren? 86. ¿Cuáles son los resultados esperados?	• Identificar casos y escenarios de prueba basados en casos de uso y uso inadecuado de la aplicación. Acorde a los requerimientos que van a ser verificados, tener en cuenta[120]: ▪ Patrones de ataque. ▪ Comportamientos esperados. ▪ Resultados del proceso de análisis de riesgo. • Seleccionar datos de prueba con significado para el negocio y datos aleatorios sin significado. • Identificar los criterios de aceptación y falla de las pruebas. • Seleccionar los tipos de pruebas a realizar; entre las técnicas de pruebas enfocadas en verificar los requerimientos de seguridad se pueden encontrar[121]: ▪ Pruebas de datos aleatorios – "Fuzz testing". ▪ Pruebas de intrusión.[122] ▪ Inyección de fallas en código fuente y binarios. • Exploración de vulnerabilidades.
87. ¿Qué herramientas se utilizan para apoyar la ejecución de las pruebas?	*Evaluar y seleccionar herramientas que ayuden a automatizar la ejecución de las pruebas, acorde a las técnicas a aplicar, tener en cuenta:* • *Herramientas para realización de pruebas de caja negra genéricas como:* ▪ *OWASP WebScarab.* ▪ *Paros.* ▪ *BackTrack.* • *Escáneres de vulnerabilidades como:* ▪ *IBM Security AppScan Source.* ▪ *Nessus.* ▪ *Kyplex Security Scanner.* ▪ *OpenVAS.*

[120] Ver escenarios de ataque en *OWASP Top 10 Application Security Risks – 2013*. [38]
[121] Ver descripción de las diferentes técnicas de pruebas en Software *Security Testing. Software Assurance (SwA) Pocket Guide Resources*. [46]
[122] Ver técnicas de comprobación y metodología para la realización de pruebas de intrusión para aplicaciones web, propuesta por OWASP. [36]

T2.2 Ejecutar pruebas y documentar resultados.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
88. ¿Cuáles son los resultados de las pruebas? 89. ¿Se está llevando registro tanto del plan de pruebas como de los resultados de las mismas?	• Documentar el plan de pruebas y las herramientas utilizadas para ir construyendo un conjunto de pruebas de seguridad que se apliquen antes de la liberación de cada versión. • Registrar no conformidades y vulnerabilidades detectadas, en lo posible crear métricas que permitan evaluar la madurez de la aplicación respecto al manejo de las características de seguridad. • Estimar costos de refactorización y/o corrección, acorde a las debilidades y no conformidades encontradas.

9.6.5.3 A3. Diseñar pruebas de seguridad del software.

Establecer el plan de pruebas de calificación del software, que permitan validar los requerimientos de seguridad, en el ambiente de aceptación del cliente.

T3.1 Diseñar el plan de pruebas de calificación del software.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
90. ¿Cuáles pruebas de seguridad van a ejecutarse en el ambiente de aceptación del cliente?	• Seleccionar cuales pruebas del plan de pruebas de integración, van a ejecutarse en el ambiente de aceptación del cliente, con el fin de que éste pueda validar el cubrimiento de los requerimientos de seguridad establecidos.

9.6.5.4 A4. Verificar y documentar.

Verificar el plan de pruebas propuesto y documentarlo.

T4.1 Verificar el plan de pruebas diseñado.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
91. ¿Las pruebas de calificación del software para la validación de los	• Verificar que el plan de pruebas de calificación propuesto, cubre todos

requisitos de seguridad, son suficientes? 92. ¿Es viable realizar todas las pruebas propuestas, en el ambiente de aceptación del cliente? 93. ¿La aplicación se encuentra lista para la realización de las pruebas de calificación?	los requerimientos de seguridad establecidos. • Validar con los dueños de proceso si es posible realizar las pruebas propuestas en el ambiente de aceptación del cliente. • Registrar el plan de pruebas como la validación con el cliente.

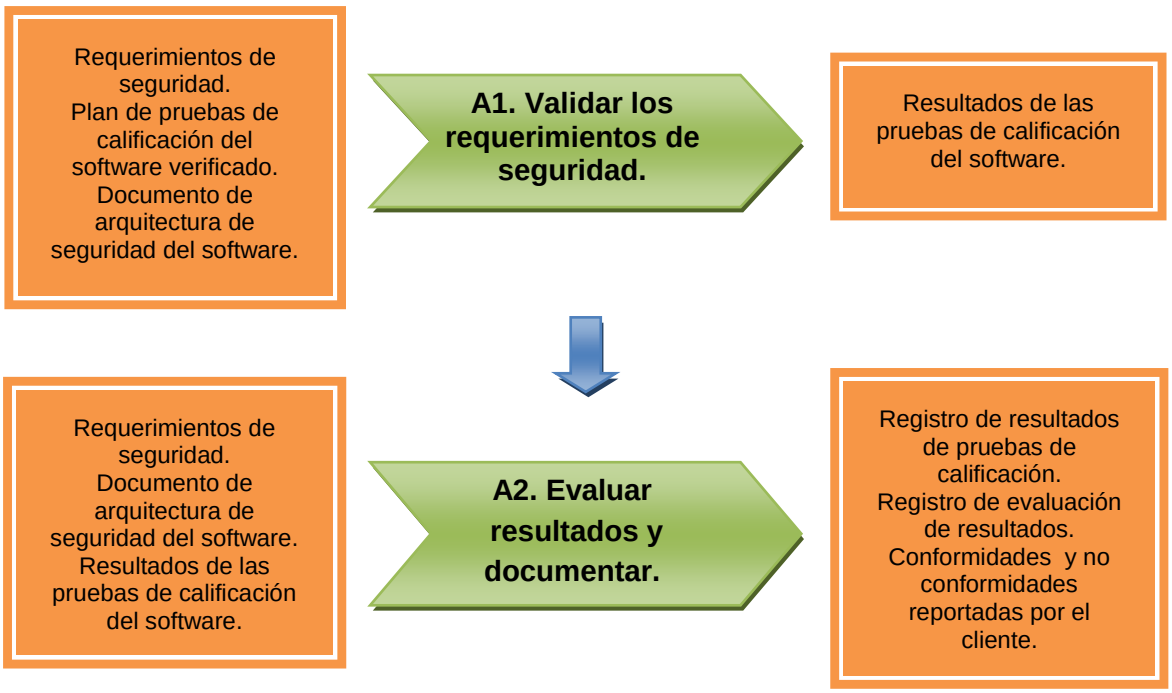
9.6.6 Parte 6: Pruebas de calificación del software.

Objetivo: Confirmar que la aplicación integrada cumple con los requerimientos de seguridad establecidos.

Cumplimiento.

ISO 27002: 10.3.2 Aceptación del sistema.

COBIT: DS5.5 Pruebas, Vigilancia y Monitoreo de la Seguridad, AI7.2 Plan de Prueba. AI7.4 Ambiente de Prueba.



Participantes: Líderes de proyecto, analistas funcionales.

9.6.6.1 A1. Validar los requerimientos de seguridad.

Ejecutar las pruebas de calificación del software, que le permiten al cliente validar los requerimientos de seguridad de la aplicación.

T1.1 Ejecutar plan de pruebas de calificación en ambiente de aceptación del cliente, en presencia de los dueños de procesos y los dueños de información.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
94. ¿Los dueños de procesos y de información conocen y están de acuerdo con el plan de pruebas a ejecutar? 95. ¿Qué datos de prueba se requieren? 96. ¿La configuración del ambiente de aceptación del cliente, es igual o lo más parecida posible al ambiente operativo?	• Asegurar que los dueños de información y los dueños de proceso conocen y aprueban el plan de pruebas a seguir. • Evitar la utilización de datos reales productivos como datos de prueba. • Verificar que las condiciones del ambiente de aceptación son iguales o lo más parecidas posibles al ambiente operativo, teniendo en cuenta calidad de los datos y privacidad, cargas de trabajo, prácticas operativas, y demás condiciones relevantes.

9.6.6.2 A2. Evaluar resultados y documentar.

Evaluar y registrar los resultados de la validación de los requerimientos de seguridad, por parte del cliente.

T2.1 Evaluar los resultados de las pruebas.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
97. ¿Los dueños de los procesos y los dueños de información están conformes con los resultados de las pruebas? 98. ¿Se han validado todos los requerimientos de seguridad establecidos?	Evaluar los resultados de las pruebas en conjunto con los dueños de proceso y dueños de información, teniendo en cuenta. • Cobertura de las pruebas respecto de los requerimientos de seguridad establecidos. • Impacto en otros requerimientos

99. ¿El comportamiento de los controles de seguridad implementados no afecta otros requerimientos funcionales y no funcionales?	funcionales y no funcionales de la aplicación. • Conformidad con los lineamientos de seguridad establecidos en el diseño. • Pertinencia de las pruebas realizadas.

T2.2 Registrar resultado de las pruebas.

Preguntas clave (QUE)	Prácticas sugeridas. (CÓMO).
100. ¿Se están registrando adecuadamente los resultados de las pruebas realizadas? 101. ¿Se está registrando adecuadamente la evaluación de los resultados de las pruebas?	• Registrar los resultados de las pruebas. • Registrar la evaluación de los resultados previo acuerdo con el cliente. • Registrar las conformidades y no conformidades reportadas por los dueños de proceso y dueños de información.

10. VALIDACIÓN DE LA GICSI.

Luego de finalizado el proceso[123] planteado para la construcción de la guía, es en la etapa IV en donde se pretende validar si la misma cumple con el objetivo de facilitar la adopción de conceptos y la incorporación de controles de seguridad informática en proyectos de desarrollo de software a medida.

A continuación se describe el proceso de validación establecido así como los resultados obtenidos a partir de su ejecución.

10.1 PROCESO DE VALIDACIÓN.

El contenido de la GICSI es presentado a referentes de diferentes organizaciones de desarrollo de software, para que posteriormente mediante la realización y/o análisis de las tareas indicadas en dicha guía, validen y expresen su opinión respecto a su utilización en el proceso de desarrollo que aplican diariamente.

Acorde a lo anteriormente mencionado, se les solicita a los participantes en el proceso, que después de navegar y explorar el sitio web en donde se encuentra publicado el contenido de la guía, realicen los pasos descritos en la Tabla No. 2.

[123] El proceso de elaboración de la GICSI, se describe en el capítulo 5 de este documento.

Tabla No. 2: Tareas del proceso de validación de la GICSI.

TAREA	DESCRIPCIÓN
Paso 1: Defina sus expectativas sobre los resultados que espera obtener a partir de la aplicación de la guía.	Definir las expectativas sobre los resultados que se esperan obtener a partir de la aplicación de la guía.
Paso 2: Defina la aplicación o funcionalidad sobre la que se aplicará la guía.	Seleccionar la aplicación o funcionalidad sobre la que se aplicará la guía.
Paso 3: Identifique las partes de la guía que se aplicarán acorde a los objetivos establecidos.	Seleccionar las partes de la guía que se aplicarán acorde a los objetivos establecidos.
Paso 4: Responda las siguientes preguntas, con el fin de evaluar si la guía cumple con su objetivo: 1. ¿Es simple en estructura? 2. ¿Es clara y se facilita su aplicación? 3. ¿Presta apoyo para el cumplimiento de normas, estándares y regulaciones? 4. ¿Permite priorizar las medidas que se van a implementar para contrarrestar amenazas de seguridad informática? 5. ¿Ayuda a identificar los posibles riesgos de seguridad informática? 6. ¿Permite establecer el costo-beneficio de la implementación de controles de seguridad informática? 7. ¿Cubre satisfactoriamente el proceso de desarrollo y las buenas	Responder el grupo de preguntas establecidas para validar la guía, según las posibles respuestas mencionadas a continuación: ▪ Totalmente de acuerdo. ▪ De acuerdo. ▪ En desacuerdo. ▪ Totalmente en desacuerdo. ▪ Sin información.

prácticas conocidas? 8. ¿Permite detectar el nivel de concientización y las necesidades de capacitación de los analistas de desarrollo, respecto de la seguridad informática en las aplicaciones?	

La lista de preguntas mencionadas en el paso 4 de la Tabla No.2, se establece teniendo en cuenta los objetivos definidos para el presente trabajo de investigación y para la guía de incorporación de controles.

10.1.1 Criterio de evaluación.

Para la validación de la guía, se define el siguiente criterio de evaluación:

Se considera que la guía tiene un nivel de validación aceptable, si en la muestra definida de evaluadores, el promedio de respuesta favorable es mayor al 70% y en ninguna pregunta el promedio individual es totalmente negativo. Se considera como respuestas favorables “Totalmente de acuerdo” y “De acuerdo”, la respuesta correspondiente a “Sin información” no se tiene en cuenta para el cálculo del promedio.

10.2 EJECUCIÓN DEL PROCESO DE VALIDACIÓN.

En la sección “Validación de la guía” del sitio web[124] en donde se encuentra publicada la GICSI, se presenta el detalle de las tareas a realizar

[124] https://sites.google.com/site/segappguide/proceso-de-validacion

por parte de las empresas consultadas, el dictamen de su experiencia y sus opiniones se registran en un formulario expuesto para tal fin.

A continuación se presenta la información de los referentes de las organizaciones desarrolladoras de software que participan en el proceso de validación:

- Alejandro Páez: Líder del área de organización y métodos de la empresa CENSYS.
- Pablo Folgar: Líder de equipo de desarrollo en everis – Argentina.
- Nicolás Garrido: Analista de desarrollo en el Gobierno de la provincia de Neuquén.
- Claudia Ghisolfi: Coordinador seguridad de aplicaciones y bases de datos en ANSES.- Administración Nacional de Seguridad Social.
- Carlos Fontela: Docente Universidad de Buenos Aires, Facultad de Ingeniería.

10.3 RESULTADOS DEL PROCESO DE VALIDACIÓN

En el Anexo 5 se registra el detalle de las respuestas y observaciones de los profesionales que participan en el proceso.

Acorde a los resultados registrados se obtiene un promedio de respuesta favorable[125] del 87,5%, y ninguna pregunta con promedio de respuesta no favorable, con lo cual se concluye que el nivel de validación de la guía es aceptable.

Por otro lado analizando las observaciones de los participantes, a continuación se presentan las respuestas obtenidas, agrupándolas en proactivas y reactivas.

[125] En la tabla No. 8 del Anexo 5, se encuentra el cálculo del porcentaje de respuesta favorable acorde al criterio de evaluación definido en la sección 10.1.1. del presente documento.

Respuestas proactivas.

- La guía de implementación ayuda a identificar el nivel de concientización y las necesidades de capacitación de los analistas de desarrollo, respecto a la seguridad informática en aplicaciones.

- La estructura de la guía facilita su aplicación y es claramente identificable la relación con las normas y estándares.

- La guía debería ser tenida en cuenta en toda su extensión para tipos de aplicaciones que manipulen datos sensibles en sectores como bancos, servicios públicos, administraciones públicas, entre otras.

Respuestas reactivas.

- Aún sigue siendo complejo establecer el costo-beneficio de implementar los controles de seguridad, lo cual depende de la magnitud del proyecto, pues para ciertos requerimientos seguir la guía puede implicar perder competitividad en tareas de desarrollo menores.

- Aplicar los conceptos de seguridad se hace difícil para las empresas con niveles de madurez incipientes así como para los analistas de desarrollo con poca experiencia.

Finalmente se destaca que como resultado del análisis de las conclusiones aportadas por los participantes y a los efectos de que la guía sea accesible a organizaciones de desarrollo más pequeñas, se decide agregar una sub-categorización a las preguntas propuestas, clasificándolas en: “Preguntas necesarias” y “Preguntas deseables”.

11. CONCLUSIONES

A continuación se sintetizan las actividades que han sido realizadas durante el trabajo de investigación y construcción de la GICSI.

- Desarrollo de una metodología para la elaboración de la guía de incorporación de controles de seguridad informática en procesos de desarrollo de software a medida.

- Diseño y realización de una entrevista para recopilar la percepción que tienen analistas, líderes y gerentes de proyecto, respecto a la incorporación de controles de seguridad informática en el proceso de desarrollo de software que aplican diariamente.

- Definición y aplicación de un proceso de validación de la guía, que ha permitido interactuar con referentes de diferentes organizaciones desarrolladoras de software, y cuyos resultados se han registrado y analizado para identificar puntos de mejora.

En función de lo anterior, y acorde a los resultados del proceso de validación realizado, y al análisis de la bibliografía referenciada, se entiende que se ha cumplido el objetivo de desarrollar una guía de incorporación de controles de seguridad informática que facilite la aplicación de estándares y buenas prácticas de desarrollo seguro, en proyectos de construcción de aplicaciones web a medida.

Adicionalmente y dada la experiencia en desarrollo de software que tiene la autora del presente documento, se considera que la guía puede ser una herramienta valiosa en lo referente a concientización y capacitación en seguridad informática, tanto para analistas como para líderes de desarrollo,

ya que concentra aspectos regulatorios, conceptuales y de procedimiento, que les permite tener una fuente primaria de consulta, cuando el negocio demande que las aplicaciones que lo soportan implementen determinados requerimientos de seguridad.

Por otra parte, se ha podido relevar adecuadamente la percepción de los grupos de interés (como ser analistas, líderes y gerentes de proyectos que participaron en la entrevista), sobre la incorporación de controles de seguridad informática en el proceso de desarrollo de software. Cabe resaltar que los participantes en la entrevista, argumentan que normalmente no hay una implementación adecuada y sistemática de los controles de seguridad informática, a menos que haya un requerimiento explícito del cliente; lo anterior lo justifican con una interpretación de sobrecostos asociados sin ponderar el impacto de los riesgos, y por no contar con profesionales calificados para responder a dichos requerimientos.

11.1 FACTORES RELEVANTES DEL TRABAJO DE INVESTIGACIÓN.

La información recopilada a través de la entrevista realizada, es muy útil para la etapa de diseño de la GICSI, en tanto que permite identificar el nivel de concientización que tienen los entrevistados respecto a la importancia de la seguridad informática y la aplicación de estándares, así como sugerencias para una mejor comprensión y aplicación de los controles.

Se considera que las propuestas de desarrollo seguro de OWASP[126] y MSDL[127] que han servido como base para el diseño y la construcción de la GICSI, son muy completas; OWASP está enfocado en el desarrollo de aplicaciones web, independiente de la metodología utilizada; mientras que Microsoft ha ajustado su modelo para aplicarlo no solo a productos, sino a aplicaciones web a medida, y aplicaciones construidas utilizando metodologías ágiles de desarrollo.

[126] OWASP – Open Web Application Security Project.
[127] Microsoft Security Development LiveCycle.

Se resalta que la referencia a estándares y normas es más explícita en las guías de desarrollo de OWASP, que en el MSDL, sin embargo para este último se puede encontrar información de apoyo que permite alinearse con algunos estándares.

11.2 ASPECTOS DESTACABLES DE LA GICSI

La estructura propuesta por la guía respeta los lineamientos y el alcance planteados, en tanto que identifica los controles de seguridad y la forma de incorporarlos, establece la relación con normas y regulaciones, propone un marco conceptual, en toda su extensión se resalta la importancia de estar alineados con el negocio y se proponen tareas de verificación y comunicación, que apoyan la capacitación y concientización de los involucrados en el proceso de desarrollo respecto a la seguridad informática.

Se considera que la GICSI es flexible dado que es posible aplicarla a cualquier modelo de proceso de desarrollo e incluso a metodologías ágiles, que contemplen las áreas de proceso propuestas por la norma ISO/IEC 12207:2008[128].

11.3 RECOMENDACIONES DE APLICACIÓN DE LA GICSI

Para lograr un cubrimiento amplio de las características de seguridad de una aplicación, se recomienda la implementación de toda las partes de la guía; sin embargo la misma es escalable en tanto que puede servir de consulta para cuando se requiere abordar requerimientos particulares, ya que la estructura propuesta facilita la identificación de los controles para cada una de las dimensiones de la seguridad (Autenticación, confidencialidad, integridad, disponibilidad y trazabilidad).

[128] ISO/IEC-12207:2008 - Modelo del ciclo de vida del software. [9].

11.4 FUTURAS LINEAS DE INVESTIGACIÓN.

Teniendo en cuenta el estado del arte en el momento de la finalización del presente trabajo, y la importancia que tiene para las organizaciones que las aplicaciones que soportan su negocio, cuenten con los controles de seguridad informática requeridos, se considera pertinente profundizar y ampliar los siguientes aspectos:

- Análisis e identificación de controles de seguridad informática en otros procesos del modelo de ciclo de vida del software, que no fueron cubiertos en esta versión inicial de la guía de incorporación, como pueden ser los procesos de mantenimiento y soporte del software, entre otros.

- Aplicación de un proceso de validación más extenso y detallado, teniendo en cuenta que algunas de las fases propuestas en dicho proceso no pudieron ser realizadas a completitud por parte de las empresas consultadas, durante el desarrollo de la presente investigación.

- Análisis e incorporación de los lineamientos propuestos por las partes 2, 3 y 4 de la norma ISO/IEC 27034:2011 Tecnologías de la información – Técnicas de seguridad – Seguridad en aplicaciones, que se encuentran actualmente en elaboración, las cuales se mencionan a continuación:

  - Parte 2: Marco normativo de la organización.

  - Parte 3: Proceso de gestión de la seguridad de las aplicaciones.

  - Parte 4: Validación de la seguridad de las aplicaciones.

  - Parte 5: Protocolos y controles de seguridad de las estructuras de datos de las aplicaciones.

- Identificación de conceptos que permitan reforzar y/o complementar los controles de seguridad informática referenciados en la guía, a partir del estudio más detallado del estándar ISO/IEC 15408: Criterios de evaluación para la seguridad en TI.

12. GLOSARIO[129]

Activos de información: Recursos del sistema de información o relacionados con éste, necesarios para que la Organización funcione correctamente y alcance los objetivos propuestos por su dirección.[130]

Amenaza: Una causa potencial de un incidente no-deseado, el cual puede resultar en daño a un sistema u organización.[131]

Autenticación: El acto de verificar la identidad de un usuario y su elegibilidad para acceder a la información computarizada. La autenticación está diseñada para proteger contra conexiones de acceso fraudulentas.[132]

Confidencialidad: Propiedad de que la información no sea divulgada a entidades, personas o procesos no autorizados.[117]

Disponibilidad: Propiedad de que la información esté disponible y utilizable cuando lo requiera una entidad autorizada.[117]

Dueño de la información: Individuos, por lo general gerentes o directores, que tienen la responsabilidad de la integridad, el uso y el reporte preciso de los datos computarizados.[133]

Dueño del proceso: Individuos que tiene por responsabilidad la correcta ejecución de los procesos del negocio, soportados por las aplicaciones informáticas.

---

[129] La mayoría de las definiciones presentadas en este capítulo también son relacionadas en la guía de implementación, en la sección 9.3 Conceptos y definiciones.
[130] Tomado de MAGERIT – Método. [24]
[131] Tomado de la norma ISO/IEC 13335-1:2004 *Part 1: Concepts and models for information and communications technology security management.*[12]
[132] Tomado del Apéndice VII Glosario de COBIT 4.1. [15]
[133] Concepto tomado de COBIT 4.1. [15]

Información de negocio: Información sensible para los procesos de negocio soportados por el software a desarrollar, y que puede actuar como entrada, ser procesada, o ser generada por éste último.

Integridad: La propiedad de salvaguardar la precisión y completitud de los recursos.[134]

NIST: National Institute Standard and Technology.

No Repudio: Asegurar que el remitente no puede negar que envió y que el receptor no pueda negar que recibió.[135]

Patrones de diseño seguro: Tienen por objetivo eliminar la inserción accidental de vulnerabilidades dentro del código y mitigar las consecuencias de esas vulnerabilidades.[136]

Pruebas de calificación: Pruebas llevadas a cabo por el desarrollador y presenciadas por el adquirente (según corresponda), para demostrar que un producto de software cumple sus especificaciones y está listo para ser utilizado en el entorno seleccionado o para integrarse al sistema que lo contiene.[137]

Riesgo: Combinación de la probabilidad de un suceso y sus consecuencias.[138]

Seguridad de la información: "Protección de la información de una amplia variedad de amenazas, con el objeto de asegurar la continuidad del negocio, minimizar los riesgos, maximizar el retorno de la inversión e incrementar las oportunidades de negocio".[139]

---

[134] Tomado de la norma ISO/IEC 13335-1:2004 *Part 1: Concepts and models for information and communications technology security management.*[12]
[135] Concepto tomado de *Risk Management Guide for Information Techonlogy Systems – NIST.* [27]
[136] Concepto tomado del documento *Secure Design Patterns.*[3]
[137] ISO/IEC-12207- Modelo del ciclo de vida del software. [9].
[138] GUIA ISO/IEC 73 Gestión de riesgos – Terminología.[41]
[139] Definición de Seguridad de la Información propuesta en el estándar ISO/IEC 27002:2005.[7].

Trazabilidad: Aseguramiento de que en todo momento se podrá determinar quién hizo qué y en qué momento.[140]

Usabilidad: La capacidad que tiene un producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo condiciones específicas.[141]

Validación: Es el conjunto de actividades que aseguran y generan confianza en que un sistema puede lograr su uso previsto, metas y objetivos.[142]

Verificación: Es el conjunto de actividades que compara un producto del ciclo de vida con las características requeridas para dicho producto. Esto puede incluir, pero no se limita a, los requerimientos especificados, la descripción del diseño y el propio sistema.[143]

Vulnerabilidad: Debilidad que puede ser accidentalmente disparada o intencionalmente explotada.[144]

[140] Concepto tomado del Apéndice 1: Glosario de MAGERIT Versión 2. Método. [24]
[141] Definición tomada de la norma ISO/IEC 9126-1:2001 *Software engineering - Product quality - Part 1: Quality model.* [11]
[142] ISO/IEC-12207- Modelo del ciclo de vida del software. [9].
[143] ISO/IEC-12207- Modelo del ciclo de vida del software. [9].
[144] Concepto tomado de *Risk Management Guide for Information Techonlogy Systems – NIST.* [27]

ANEXO 1: TIPOS DE PREGUNTAS DE UNA ENTREVISTA.

Según KVale[145] la guía de entrevista puede elaborarse teniendo en cuenta los siguientes tipos de preguntas.

- Preguntas introductorias: Pueden ser preguntas que arrojen descripciones ricas y espontáneas donde los sujetos ofrecen lo que consideran según su experiencia las principales dimensiones del fenómeno investigado.
- Preguntas de seguimiento: Pueden surgir a partir de las respuestas a las preguntas iniciales, el entrevistador debe estar atento a interpretar las luces rojas –"Red Lights" en las respuestas, tales como entonación fuerte, términos inusuales.
- Preguntas de sondeo: Permiten al investigador perseguir respuestas sondeando su contenido pero sin establecer que dimensiones deben ser tenidas en cuenta.
- Preguntas específicas: Ayudan a obtener descripciones más precisas sobre las percepciones del entrevistado.
- Preguntas directas: En este caso el entrevistador introduce directamente temas y dimensiones, se sugiere que están preguntas sean realizadas en la última parte de la entrevista después de que el entrevistado haya dado su descripción propia y espontánea y haya expuesto que aspectos del fenómeno son importantes para él. Sin embargo con el fin de delimitar el tema de la entrevistas algunas preguntas directas podrían realizarse en la introducción.

[145] Steinar Kvale es Catedrático de Psicología de la Educación y Director del Centro de Investigación Cualitativa de la Universidad de Aarthus, Dinamarca y profesor del Instituto Saybrook en San Francisco. Su libro InterViews: Learning the Craft of Qualitative Research Interviewing, ha sido la base para el diseño y elaboración de la entrevista realizada en la presente investigación. [17].

- Preguntas indirectas: Son preguntas proyectivas que pueden implicar referencias a las percepciones de otras personas, se debe prestar atención especial en la interpretación de las respuestas.
- Preguntas estructuradas: El entrevistador es responsable por el rumbo que tome la entrevista, y debería indicar cuando un tema ya se ha agotado, por lo que éste puede interrumpir amablemente una respuesta larga que es irrelevante para el tema de investigación, a partir de una pregunta que permita introducir a un nuevo aspecto.
- Preguntas de interpretación: Permiten reformular y/o complementar una respuesta.
- Silencios: Con el fin de no hacer de la entrevista un interrogatorio, el entrevistador deber darle tiempo al entrevistado, para que organice sus ideas y rompa el silencio con respuestas más elaboradas e información más significativa.

ANEXO 2: GUÍA DE ENTREVISTA No. 1.

1. ¿Qué entiende usted por seguridad Informática en aplicaciones?.
2. ¿Qué tan importante considera usted que es la seguridad como característica de las aplicaciones informáticas?.
   - Como usuario.
   - Como proveedor.
3. De acuerdo a su experiencia ¿Considera que la seguridad informática es tenida en cuenta de manera adecuada en los procesos de desarrollo en los que ha participado?
   - Si: ¿Cómo se ha incorporado?
   - No: ¿Cuáles considera usted que son las causas por las que no se tiene en cuenta la seguridad informática a lo largo de las diferentes etapas del proceso de desarrollo de software?
4. Como cree usted que la empresa/proyecto toma en cuenta la disponibilidad, integridad, confidencialidad y trazabilidad de la información?, en las diferentes etapas del proceso de desarrollo de software.
5. Cuando el usuario o el equipo de desarrollo identifican posibles incidentes de seguridad, ¿cómo considera usted que son atendidos, con que prioridad y lineamientos?
6. ¿Ha escuchado el concepto de desarrollo seguro, si es así podría definirlo?
7. Acorde a la respuesta anterior, ¿Ha aplicado o conoce algún estándar, buena práctica o marco de trabajo de desarrollo seguro que considere útil?, ¿cuáles y como fue la experiencia?
8. Tiene conocimiento de la existencia y/o aplicación de algún lineamiento (procedimientos, buenas prácticas, estándares, políticas) respecto de la

seguridad informática, que haya sido establecido en la empresa y/o proyecto, y que impacte en su trabajo diario.

9. ¿Cuán importante considera que es el establecimiento de políticas de seguridad dentro del marco del proceso de desarrollo que actualmente aplica?
    - En la labor del equipo de desarrollo.
    - En la interacción con el cliente.
    - En la interacción con proveedores.
10. En las reuniones con el cliente, cuando se ofrece un servicio o solución, ¿qué grado de importancia se le da a la seguridad informática como característica del mismo/a?
11. ¿Qué grado de conciencia considera usted que tienen sus clientes, respecto a los riesgos de seguridad informática a los que se encuentran expuestos?
12. En el momento de establecer los requerimientos de una aplicación y/o sistema, ¿en qué medida considera beneficioso identificar la legislación vigente relacionada con la seguridad informática?, dado que esto puede tener impacto en el manejo que las aplicaciones le dan a la información.
13. Según su experiencia ¿bajo qué condiciones o contexto tener en cuenta la seguridad informática en el proceso de desarrollo, puede ser valor agregado o ventaja competitiva de las soluciones que se ofrecen?
14. ¿Qué importancia considera usted que las empresas del sector le están dando a la seguridad informática; la están incluyendo dentro de su portafolio de servicios?
15. ¿Cuál es su percepción de la importancia que le están dando actualmente a la seguridad Informática, sus compañeros de trabajo y colegas que trabajan en otras consultoras?
16. ¿Bajo qué condiciones considera que una guía de implementación para la incorporación de controles de seguridad a lo largo del proceso de desarrollo podría minimizar costos, facilitar implementación y maximizar beneficios?

ANEXO 3: GUÍA DE ENTREVISTA No. 2.

Fecha:

Esta entrevista forma parte de una investigación académica sobre Gestión de la Seguridad Informática en procesos de desarrollo de software a medida, para la Maestría en Seguridad Informática de la Universidad de Buenos Aires. Argentina.

Su participación, así como sus opiniones, percepciones y experiencias son muy importantes como fuentes de información primaria para dicha investigación, y serán de carácter confidencial.

Objetivos

1. Establecer el grado de conciencia que se tiene sobre la importancia de la seguridad informática en los procesos de desarrollo de software a medida.
2. Identificar la aceptabilidad que tiene la aplicación de estándares y controles de seguridad informática en los procesos de desarrollo de software a medida.
3. Identificar posibles estrategias para facilitar la incorporación de controles de seguridad informática en procesos de desarrollo de software a medida.

Consideraciones

Autorización por parte de la empresa: La empresa por intermedio del gerente o responsable del proyecto en el que usted trabaja, ha dado la autorización para la realización de esta entrevista.

Confidencialidad: La empresa está al tanto del objetivo de la entrevista, así como de la información que se tratará en la misma, con el fin de garantizar que no haya compromiso de la confidencialidad de la información tanto de la empresa, sus clientes o del entrevistado.

Responsabilidad: Las percepciones, opiniones e información proporcionada por los entrevistados serán utilizadas única y exclusivamente para los propósitos de la investigación de la cual hace parte esta entrevista.

Datos profesionales del entrevistado.

Nombre: *Opcional.*

Descripción del perfil profesional:

Preguntas

1. ¿Qué entiende usted por Seguridad Informática en aplicaciones?
2. ¿Qué nivel de importancia considera usted que se le está dando actualmente a la seguridad informática en el desarrollo de software a medida?

   a) Alto
   b) Medio
   c) Bajo.

3. ¿En qué porcentaje considera usted que las consultoras de desarrollo de software están ofreciendo seguridad informática en sus aplicaciones como parte de su portafolio de servicios?

   a) Alto
   b) Medio
   c) Bajo.

4. ¿Qué nivel de conciencia considera que tienen tanto sus compañeros de trabajo como colegas de otras empresas, respecto a controles y mejores prácticas de seguridad informática en los procesos de

desarrollo de software? Si considera que es medio o bajo, explique sus razones.

a) Alto
b) Medio
c) Bajo.

5. ¿Considera que sus compañeros aplican lineamientos relacionados con la seguridad informática en las diferentes etapas del proceso de desarrollo de software? Si considera que excepcionalmente o nunca, explique sus razones.

a) Frecuentemente.
b) Excepcionalmente.
c) Nunca

6. ¿Acorde a su experiencia en qué nivel considera usted que el seguimiento y aplicación de estándares aporta a la calidad de las aplicaciones que desarrolla? Si considera que es medio o bajo, explique sus razones.

a) Alto
b) Medio
c) Bajo.

7. ¿Qué grado de conciencia considera usted que tienen sus clientes respecto de los riesgos de seguridad a los que se encuentra expuesta la información sensible de su negocio?

a) Alto
b) Medio
c) Bajo.

8. En las reuniones de especificación de requerimientos con el cliente, ¿en qué grado cree usted que se muestra interés por conocer las leyes, regulaciones o políticas que en materia de seguridad informática deben cumplir?.

a) Frecuentemente
b) Excepcionalmente
c) Nunca.

9. De acuerdo a su experiencia, ¿qué criterios se utilizan para priorizar la atención a incidentes de seguridad informática que se presentan en las diferentes etapas del proceso de desarrollo de software que aplica diariamente?

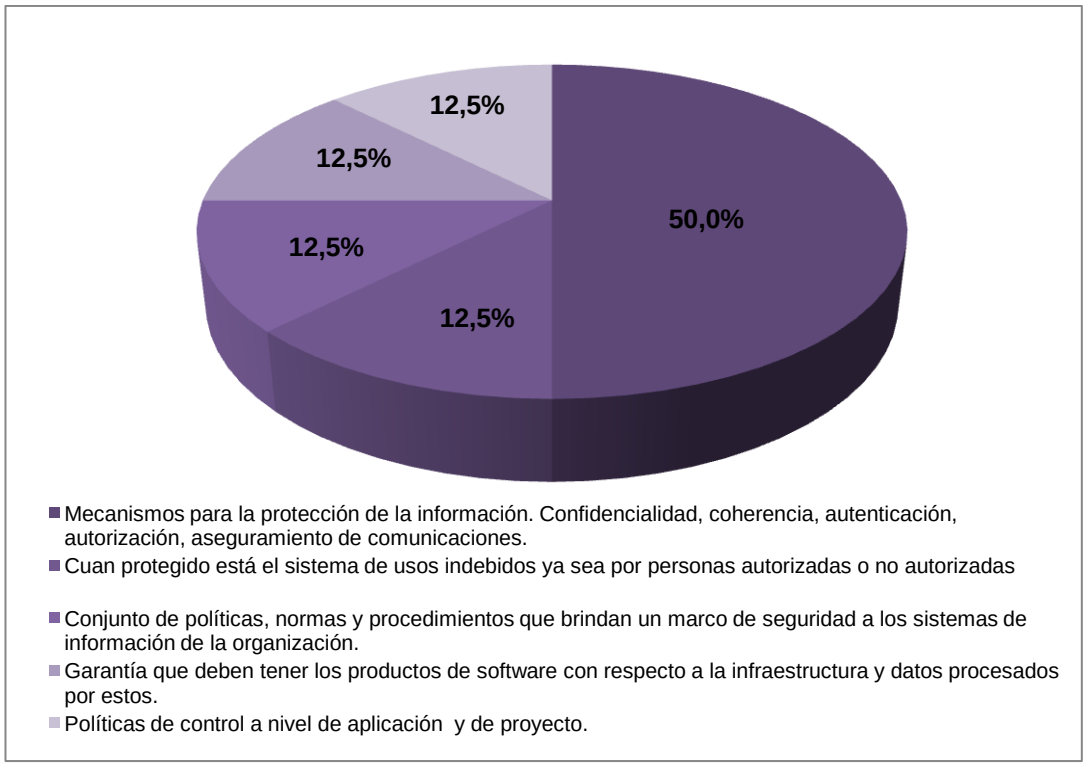
10. ¿En qué condiciones considera útil el cumplimiento de políticas, directrices o lineamientos de seguridad informática dentro del marco del proceso de desarrollo de software que usted aplica?

11. ¿Qué ventajas considera que le traería a la estrategia de negocio de su empresa, la incorporación de manera formal de controles de seguridad informática, en el proceso de desarrollo que aplica diariamente?

12. ¿Qué condiciones considera que debería cumplir una estrategia de incorporación de controles de seguridad informática, para que usted se sienta motivado a adoptarla y aplicarla en el proceso de desarrollo que ejecuta diariamente?

ANEXO 4: RESULTADOS DE LA ENTREVISTA.

Pregunta No. 1 ¿Qué entiende usted por Seguridad Informática en aplicaciones?

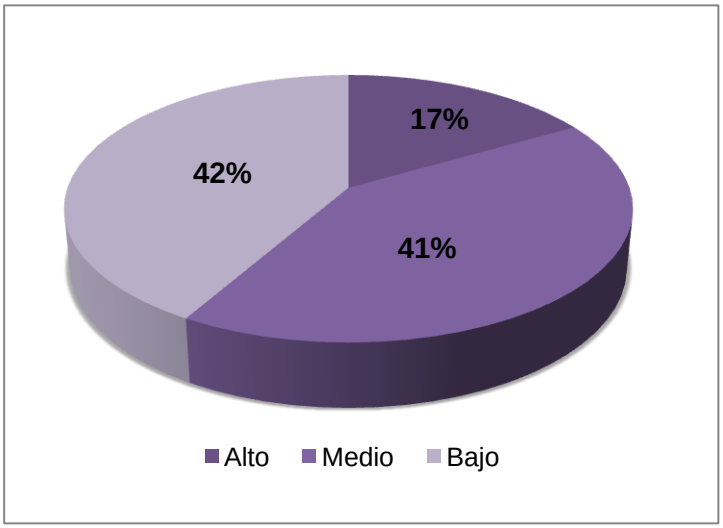


Observaciones:

Acorde a los resultados se puede encontrar que el 50% de los entrevistados relacionan los conceptos de confidencialidad, autenticación y autorización con seguridad informática.

- El otro 50% maneja un concepto más genérico, mencionando seguridad a nivel de proyecto y de aplicación.
- Ninguno de los entrevistados relacionó los conceptos de disponibilidad e integridad con seguridad.

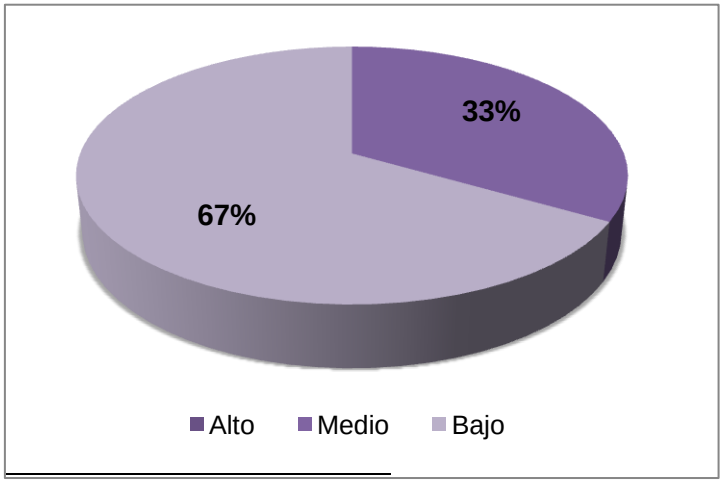
Pregunta No.2 ¿Qué nivel de importancia considera usted que se le está dando actualmente a la seguridad informática en el desarrollo de software a medida?[146]



Observaciones:

Un porcentaje reducido (16%) de los entrevistados considera que se le da un nivel de importancia alta, mientras que el 42% opina que su nivel de importancia es bajo; estos últimos justificaron su respuesta argumentando que: "La seguridad es el primer requerimiento no funcional que se recorta en un proyecto de desarrollo de software, ante la falta de recursos y/o la exigencia del cumplimiento de nuevos requerimientos funcionales".

Pregunta No. 3. ¿En qué porcentaje considera usted que las consultoras de desarrollo de software están ofreciendo seguridad informática en sus aplicaciones como parte de su portafolio de servicios?



Observaciones:

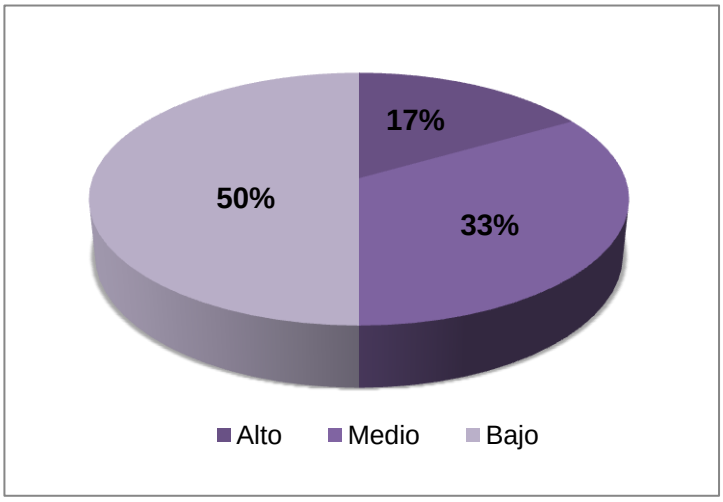
El 67% considero un porcentaje bajo de inclusión de la seguridad informática como característica de sus aplicaciones,

[146] Pregunta relacionada con el objetivo No. 1 de la entrevista. Ver apartado 5.1.1. Objetivos de la entrevista.

argumentan que en muchas ocasiones el cliente asume que ya hay un manejo implícito de la misma y confían en que la consultora realice un trabajo adecuado pues no cuentan con herramientas para evaluarlo, por lo anterior puede que los controles ofrecidos no sean coherentes con lo realmente implementado.

Así mismo quienes consideran que el porcentaje es medio, mencionan que la seguridad es una inversión muy alta, que no es fácil de justificar, y que es altamente dependiente del tipo de negocio del cliente.

Pregunta No. 4. ¿Qué nivel de conciencia considera que tienen tanto sus compañeros de trabajo como colegas de otras empresas, respecto a controles y mejores prácticas de seguridad informática en los procesos de desarrollo de software?[147] Explique sus razones.



Observaciones:

El 50% de los entrevistados considera que el nivel de conciencia de sus colegas respecto a la seguridad informática, es bajo, ellos coinciden en que en el ámbito de la seguridad informática deben tomarse decisiones difíciles, y no es común que los desarrolladores estén capacitados para esto, se requieren amplíos conocimientos técnicos y del modelo del negocio del cliente; por otro lado argumentan que las decisiones que se toman al respecto no son comunicadas adecuadamente al resto del equipo.

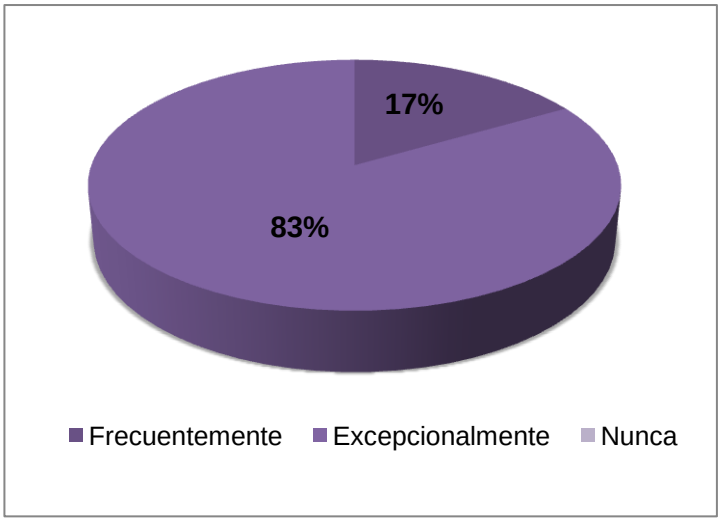
El 33% considera que ese nivel de conciencia se ha incrementado en los últimos tiempos, ya que muchas herramientas y frameworks incluyen

[147] Pregunta relacionada con el objetivo No. 1 de la entrevista. Ver apartado 5.1.1. Objetivos de la entrevista.

aspectos de seguridad en cuestiones básicas, y en la mayoría de las aplicaciones se ofrecen módulos de control de acceso a las aplicaciones.

El 17% que lo catálogo en un nivel alto, mencionó que conocen colegas cuyos trabajos están relacionados directamente con conceptos de seguridad, por el tipo de negocio del cliente; en tales casos, dichas personas han estado obligadas ha enfocarse en temas puntuales de seguridad para cubrir los requerimientos.

Pregunta No. 5. ¿Considera que sus compañeros aplican lineamientos relacionados con la seguridad informática en las diferentes etapas del proceso de desarrollo de software?. Si considera que excepcionalmente o nunca, explique sus razones.

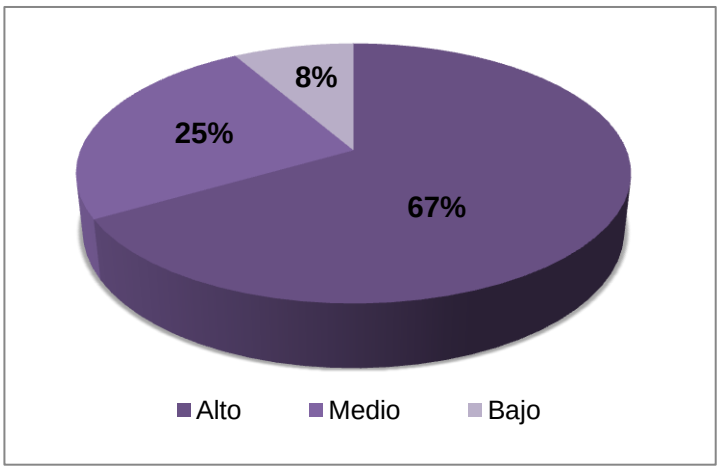


Observaciones:

El 83% de los entrevistados considera que de manera excepcional sus compañeros aplican lineamientos de seguridad durante el proceso de desarrollo, argumentan que es un área que requiere de perfiles especializados que generalmente son costosos; por otro lado se evita tomar decisiones respecto a temas de seguridad por el impacto que puedan llegar a tener, se prefiere ceder esa responsabilidad.

Otro porcentaje considera que es frecuente que se apliquen lineamientos dependiendo de la proactividad de los desarrolladores y del tipo de aplicación que se está desarrollando.

Pregunta No. 6. ¿Acorde a su experiencia en qué nivel considera usted que el seguimiento y aplicación de estándares aporta a la calidad de las aplicaciones que desarrolla? Si considera que es medio o bajo, explique sus razones.



Observaciones:

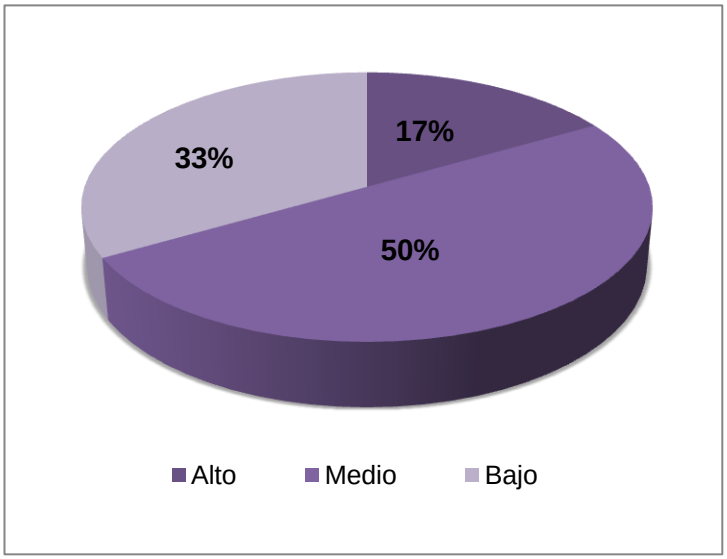
Un gran porcentaje considera que la aplicación de estándares aporta calidad a las aplicaciones, sin embargo condicionan dicha utilización a la madurez de los procesos del cliente, y consideran que aplicarlos en su totalidad es costoso y no se percibe el retorno de la inversión.

Un 13% argumenta que el nivel de aporte es bajo, en tanto que la implementación completa de un estándar solo puede realizarse en un mundo perfecto con recursos ilimitados, y que en su mayoría se implementan más con objetivos de cumplimiento que de mejoramiento.

Pregunta No. 7. ¿Qué grado de conciencia considera usted que tienen sus clientes respecto de los riesgos de seguridad a los que se encuentra expuesta la información sensible de su negocio?

Observaciones:

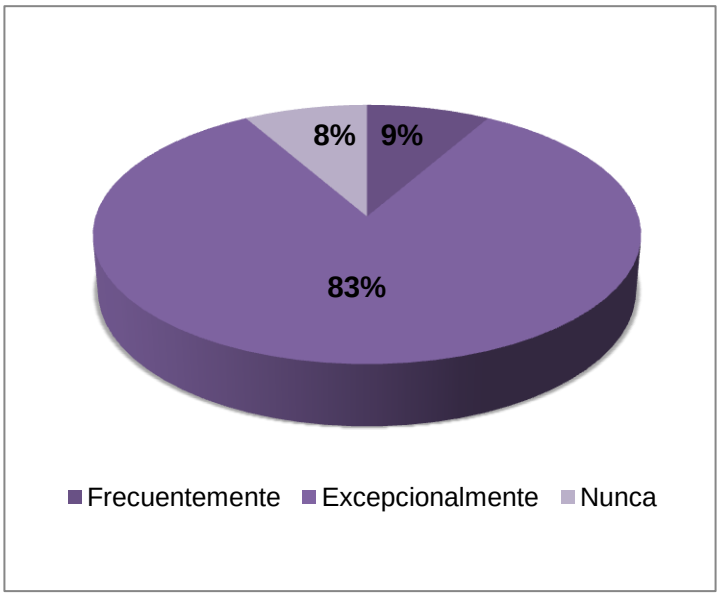
El 50% de los entrevistados considera que el cliente tiene un nivel medio de consciencia respecto a los riesgos de seguridad, algunos argumentan que los clientes son conscientes de las vulnerabilidades pero no tienen la capacidad de discernir si las soluciones propuestas por las consultoras son las adecuadas.



Un 17% considera que el cliente tiene un nivel de consciencia alto, siempre que su negocio esté basado en datos.

El 33% restante argumenta que el nivel normalmente es bajo pues se le da mayor prioridad a los requerimientos funcionales.

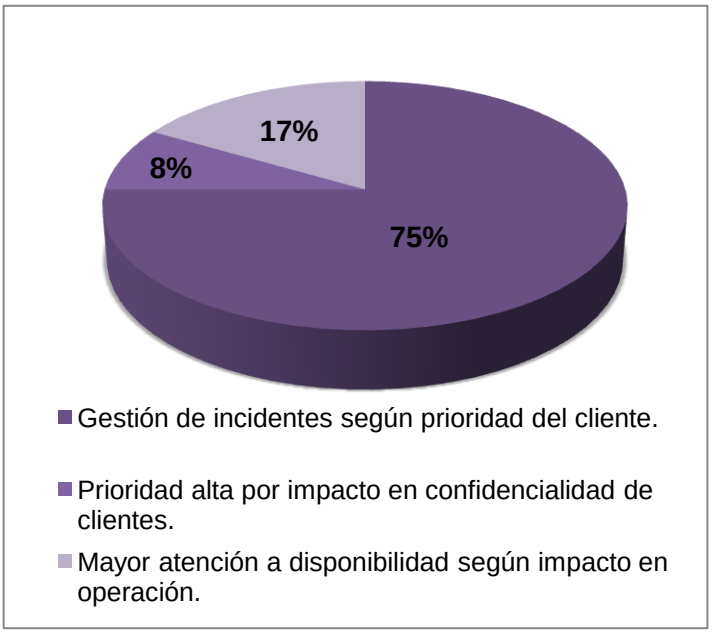
Pregunta No. 8: En las reuniones de especificación de requerimientos con el cliente, ¿en qué grado cree usted que se muestra interés por conocer las leyes, regulaciones o políticas que en materia de seguridad informática deben cumplir?



Observaciones:

El 84% de los entrevistados consideran que los clientes no son muy conscientes de las regulaciones que deben cumplir a menos que éstas estén estrechamente relacionadas con los requerimientos funcionales.

Pregunta No. 9: De acuerdo a su experiencia, ¿qué criterios se utilizan para priorizar la atención a incidentes de seguridad informática que se presentan en las diferentes etapas del proceso de desarrollo de software que aplica diariamente?

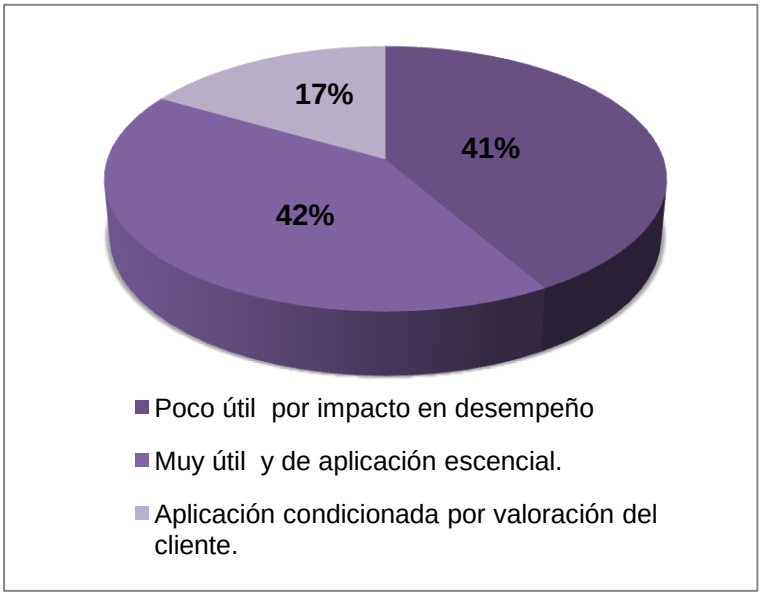


Observaciones:

El 75% argumenta que los incidentes se atienden acorde a lo pactado con el cliente, no hay criterios particulares para seguridad. Los incidentes de seguridad que surjan pueden no atenderse si están fuera del alcance establecido.

El 17% considera que los incidentes de seguridad, en particular los relacionados con la disponibilidad, se atiende acorde al impacto que tengan en la operación.

Pregunta No. 10. ¿En qué condiciones considera útil el cumplimiento de políticas, directrices o lineamientos de seguridad informática dentro del marco del proceso de desarrollo de software que usted aplica?



Observaciones:

Un 42% considera que la aplicación de políticas es poco útil en tanto que su seguimiento normalmente tiene impacto en el

desempeño de las personas frente a la realización de ciertas operaciones, adicionalmente mientras no se tenga un adecuado seguimiento se va a tender a no cumplirlas.

Un porcentaje similar argumenta que la aplicación de políticas es escencial en el desarrollo de aplicaciones, dado que estás se han convertido en un blanco fácil para los atacantes.

Un 16% reconoce la importancia de la implementación de políticas, pero la condiciona a la valoración que el cliente le da a los resultados de aplicarlas, en muchos casos se desestiman por los costos que implican y que el cliente no asume.

Pregunta No. 11. ¿Qué ventajas considera que le traería a la estrategia de negocio de su empresa, la incorporación de manera formal de controles de seguridad informática, en el proceso de desarrollo que aplica diariamente?

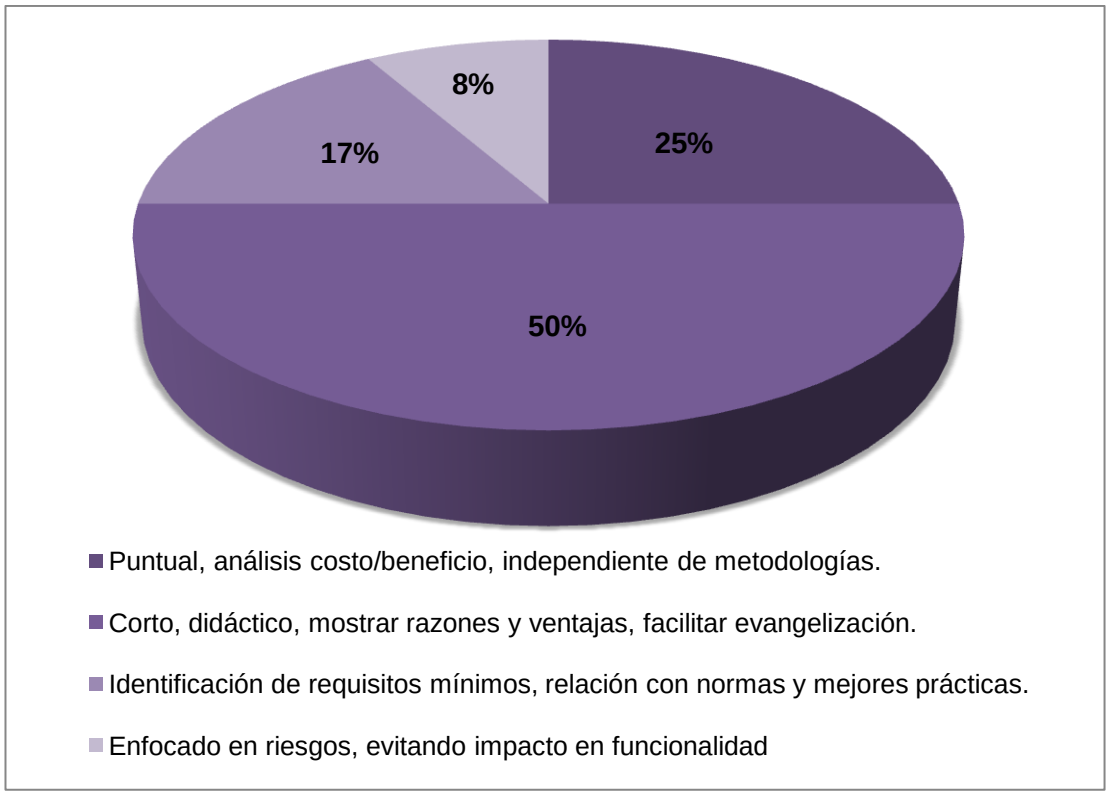
Observaciones:

El 58% de los entrevistados opinan que incorporar controles de seguridad al proceso de desarrollo aportaría valor agregado y sería una ventaja competitiva, ya que a su juicio son pocas las consultoras que lo tienen en cuenta, sin embargo esto dependerá de la consciencia e importancia que el cliente le preste a la seguridad de su información.

El 25% considera que los controles de seguridad son valorados por el cliente, y tienen un buen impacto en la imagen de la empresa, sin embargo hay que evangelizar al cliente.

El 17% generalmente no incluye aspectos de seguridad en sus propuestas comerciales a menos que la aplicación tenga algún requerimiento de seguridad específico, y consideran que si se incluyera no sería un criterio importante para la selección de las propuestas, ya que el cliente no asume los costos que implica la incorporación y mantenimiento de estos controles, como lo son la capacitación y contratación de personal calificado en desarrollo seguro de aplicaciones.

Pregunta No 12. ¿Qué condiciones considera que debería cumplir una estrategia de incorporación de controles de seguridad informática, para que usted se sienta motivado a adoptarla y aplicarla en el proceso de desarrollo que ejecuta diariamente?[148]



[148] Esta pregunta está relacionada con el objetivo No. 3 de la entrevista (Ver apartado 5.1.1. Objetivos de la entrevista), y sus resultados fueron tenidos en cuenta en el diseño de la guía de implementación.

Observaciones:

La mayoría de los entrevistados coincidieron en que una buena estrategia para la incorporación de controles de seguridad al proceso de desarrollo, debe ser didáctica, resaltando las ventajas y razones, estableciendo una base conceptual que permite la fácil evangelización de los desarrolladores.

Por otro lado el 15% considero que es importante que sea independiente de la metodología de desarrollo implementada por la empresa, para que sea aplicable a diferentes proyectos, y que tenga en cuenta el análisis costo/beneficio de la incorporación de ciertos controles.

El 17% hizo énfasis en la relación con normas y mejores prácticas, y en la identificación de recursos críticos que deben ser asegurados.

Un porcentaje menor (8%) argumento que debe establecerse un conjunto mínimo de requisitos basado en un análisis de riesgos, preservando ante todo la funcionalidad de la aplicación.

ANEXO 5: RESULTADOS DEL PROCESO DE VALIDACIÓN.

Los resultados del proceso de validación se capturaron a través de un formulario publicado en el sitio web[149] de la GICSI, a continuación se presentan los resultados registrados.

Tabla No. 3: Proceso de validación. Paso 1: Defina sus expectativas sobre los resultados que espera obtener a partir de la aplicación de la guía.

Alejandro Páez CENSYS S.A.	Contar con una "herramienta/metodología" que permita considerar los aspectos de seguridad a tener en cuenta en el desarrollo de sistemas, tanto para la parte funcional como tecnológica.
Pablo Folgar everis.	Espero que los lineamientos de esta guía sirva para dirigir de forma correcta un proceso de desarrollo de software que cumpla con las normas de seguridad requeridas.
Nicolás Garrido Gobierno Provincia de Neuquén.	Considero que la seguridad debe, así como los casos de uso o requisitos o historias de usuario, guiar el desarrollo del producto software, pero no solo eso sino que en muchos casos indicar el enfoque, cómo pensar, plantear, diseñar e implementar los requisitos, casos de uso, historias de usuario, una especie de marco o contexto además del contexto del negocio pero del cual también es parte.
Claudia Ghisolfi ANSES.	Evitar los costos resultantes de la detección de vulnerabilidades una vez implementada una aplicación. El punto de partida para lograrlo sería cumpliendo las tres primeras partes de la guía.
Carlos Fontela UBA.	No se realizó este paso.

[149] https://sites.google.com/site/segappguide/proceso-de-validacion.

Tabla No. 4: Proceso de validación. Paso 2: Defina la aplicación o funcionalidad sobre la que se aplicará la guía.

Alejandro Páez CENSYS S.A.	La guía se aplico para el análisis funcional de cada uno de los requerimientos mayores a una determinada cantidad de horas solicitados por nuestros clientes.
Pablo Folgar everis.	Por la manera sencilla en que la guía es explicada considero que en cualquier aplicación de mediana y/o gran escala pueda ser implementada. Por otra parte considero esta guía como mandataria para tipos de aplicaciones que manipulen información sensible, como ser bancos, servicios de salud, administraciones públicas, etc.
Nicolás Garrido Gobierno Provincia de Neuquén.	No se realizó este paso.
Claudia Ghisolfi ANSES.	Aplicaciones críticas de acceso público.
Carlos Fontela UBA.	La puedo aplicar sobre productos de software presentados en tesinas de grado.

Tabla No. 5: Proceso de validación. Paso 3: Identifique las partes de la guía que se aplicarán acorde a los objetivos establecidos.

Alejandro Páez CENSYS S.A.	Parte 1 a 3.
Pablo Folgar everis.	Para poder llegar a cumplir los objetivos establecidos considero que es necesario respetar todo el proceso definido en esta guía.
Nicolás Garrido Gobierno Provincia de Neuquén.	No se realizó este paso.
Claudia Ghisolfi ANSES.	Partes 1, 2 y 3.
Carlos Fontela UBA.	Todas, ya que se trata de proyectos de software completos

Paso 4: Responda las siguientes preguntas, con el fin de evaluar si la guía cumple con su objetivo

1. ¿Es simple en estructura?

2. ¿Es clara y se facilita su aplicación?

3. ¿Presta apoyo para el cumplimiento de normas, estándares y regulaciones?

4. ¿Permite priorizar las medidas que se van a implementar para contrarrestar amenazas de seguridad informática?

5. ¿Ayuda a identificar los posibles riesgos de seguridad informática?

6. ¿Permite establecer el costo-beneficio de la implementación de controles de seguridad informática?

7. ¿Cubre satisfactoriamente el proceso de desarrollo y las buenas prácticas conocidas?

8. ¿Permite detectar el nivel de concientización y las necesidades de capacitación de los analistas de desarrollo, respecto de la seguridad informática en las aplicaciones?.

Tabla No. 6: Proceso de validación. Respuestas a preguntas del Paso 4.

Participante	Pregunta No. 1	Pregunta No. 2	Pregunta No. 3	Pregunta No. 4	Pregunta No. 5	Pregunta No. 6	Pregunta No. 7	Pregunta No. 8
Alejandro Páez CENSYS S.A.	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	En desacuerdo	En desacuerdo	De acuerdo
Pablo Folgar everis.	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
Nicolás Garrido Gobierno Provincia de Neuquén.	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo	En desacuerdo	De acuerdo	En desacuerdo
Claudia Ghisolfi ANSES.	De acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo	De acuerdo
Carlos Fontela UBA.	Totalmente de acuerdo	Totalmente de acuerdo	Sin información	De acuerdo	De acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo

Tabla No. 7: Proceso de validación. Observaciones.

Alejandro Páez CENSYS S.A.	Inicialmente cambie un poco el documento propuesto, ya que estábamos necesitando definir una metodología de trabajo para el análisis funcional de cada uno de los pedidos de nuestros clientes, las modificaciones no se realizaron por qué no estuviese claro el documento original, sino más bien por una necesidad de adaptarlo a la estructura de nuestro negocio y debo reconocer que la grilla de preguntas de la parte 1 y 2 está muy bien fundamentada y es muy sólida y que el documento sirvió de base y se adapto sin problema alguno.
Pablo Folgar EVERIS.	No hay observaciones adicionales.
Nicolás Garrido Gobierno Provincia de Neuquén.	Inquietudes: ¿Cómo se aplicaría tu guía o como adaptarla/seguirla para que sea útil tanto para un software simple a uno complejo? ¿Cómo integrar el costo de "asegurar" una aplicación sin que duplique o triplique el valor final del producto?
Claudia Ghisolfi ANSES.	Si bien no me fue posible darle la detenida lectura que un trabajo de tal envergadura merece, fue suficiente para apreciar su calidad no sólo por su nivel académico sino por su claridad tanto en su organización como en su contenido. Su aplicación no se ve inalcanzable, máxime si se tiene en cuenta una de tus acertadas recomendaciones: Aplicación progresiva. Claro que, para efectuarse, es necesario que todo el equipo de trabajo y, fundamentalmente, quienes tienen a cargo la asignación de horas a los proyectos entiendan que en tiempos en que, cada vez más, las aplicaciones están siendo el centro de ataques la adopción de estas metodologías contribuirá a llevar a la mínima expresión la posibilidad de que sean exitosos; lo cual, redunda, en el caso de ANSES, fundamentalmente, en la mejora de su imagen.
Carlos Fontela UBA.	Desde mi particular punto de vista, la veo muy adecuada, sin embargo yo no soy un especialista en seguridad informática, por lo tanto, mi visión está referida a claridad de los objetivos, nivel de detalle de la guía, forma en que se presenta la información.

Tabla No. 8: Proceso de validación. Resultados.

Frecuencias acumuladas / Pregunta	Totalmente de acuerdo	De acuerdo	En desacuerdo	Totalmente en desacuerdo	Sin información	Total respuestas favorables[150]
1	2	3	0	0	0	5
2	2	3	0	0	0	5
3	1	3	0	0	1	4
4	2	3	0	0	0	5
5	2	3	0	0	0	5
6	0	3	2	0	0	3
7	2	2	1	0	0	4
8	1	3	1	0	0	4
Promedio de respuesta favorable:						4,375
Porcentaje de respuesta favorable:						87,5%

[150] La definición de respuesta favorable se presenta en el capítulo 10: Validación de la Guía, sección 10.1.1: Criterio de evaluación.

13. FUENTES DE INFORMACIÓN.

En la elaboración del presente trabajo se toman como base normas y estándares internacionales, relacionados con procesos del ciclo de vida del software, seguridad de la información, modelos de madurez y capacidad, y criterios de evaluación del producto de software.

Los controles sugeridos están apoyados en diferentes propuestas hechas por la industria de las TIC, respecto a lineamientos y buenas prácticas de diseño e implementación de aplicaciones seguras, así como en normas y metodologías para el análisis y evaluación de riesgos de seguridad informática.

Así mismo el diseño y estructura de la guía tiene en cuenta las percepciones de equipos de desarrollo respecto a la incorporación de controles de seguridad en las diferentes etapas del ciclo de vida del software; dichas percepciones son recopiladas a partir de la aplicación de una entrevista semi-estructurada.

14. BIBLIOGRAFIA

[1] Bernard, H. R. (2000). Social Research Methods: Qualitative and Quantitative Aproaches. Sage Publications.

[2] Calero, C., Moraga, M. A., & Piattini, M. (2010). Calidad del producto y proceso software. Madrid: Ra-Ma.

[3] Dougherty, C. e. (Marzo de 2009). Secure Design Patterns. Consultado el Noviembre de 2012, de http://www.cert.org/

[4] Gillham, B. (2004). The Research Interview. London: Continuum.

[5] Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (1991). Metodología de la investigación. Mexico: McGraw Hill.

[6] Howard, M., & Leblanc, D. (2002). Writing Secure Code. (Second. ed.). Microsoft Press.

[7] International Organization for Standardization, International Electrotechnical Commission. (2005.). ISO/IEC 27002. Tecnología de la Información – Técnicas de Seguridad - Código para la práctica de la seguridad de la información.

[8] International Organization for Standardization, International Electrotechnical Commission (2008). ISO/IEC 21827 Information technology - Security techniques - Systems Security Engineering - Capability Maturity Model® (SSE-CMM®).

[9] International Organization for Standardization, International Electrotechnical Commission. (2008). ISO/IEC 12207 Systems and software engineering - Software life cycle processes.

[10] International Organization for Standardization, International Electrotechnical Commission. (2011). ISO/IEC 27034 Information technology - Security techniques - Application security - Part 1: Overview and concepts.

[11] International Organization for Standardization, International Electrotechnical Commission. (2001). ISO/IEC 9126-1:2001 Software engineering - Product quality - Part 1: Quality model.

[12] International Organization for Standardization, International Electrotechnical Commission. (2004). ISO/IEC 13335-1 Information technology -- Security techniques -- Management of information and communications technology security -- Part 1: Concepts and models for information and communications technology security management.

[13] International Organization for Standardization, International Electrotechnical Commission. (2011). ISO/IEC 27005 Tecnología de la información – Técnicas de seguridad. Gestión de Riesgos de seguridad de la información.

[14] ISO25000 Calidad de producto. (s.f.). ISO25000 Calidad de producto. Consultado en abril de 2012, de http://iso25000.com/index.php/25000.html

[15] IT Governance Institute. (2007). COBIT 4.1. Obtenido de http://www.itgi.org/

[16] Kaspersky Lab. (Diciembre de 2012). Kaspersky Security Bulletin 2012 The overall statistics for 2012. Consultado en enero de 2013, de https://www.securelist.com/en/analysis/204792255/Kaspersky_Security_Bulletin_2012_The_overall_statistics_for_2012

[17] Kvale Steinar, B. S. (2009). InterViews: Learning the Craft of Qualitative Research Interviewing. Sage Publications.

[18] Microsoft Corporation. (2005). The STRIDE Threat Model. Consultado en enero de 2013, de http://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

[19] Microsoft Corporation. (Junio de 2006). Code review. Consultado en marzo de 2013, de http://msdn.microsoft.com/en-us/library/ff648637.aspx

[20] Microsoft Corporation. (s.f.). Secure Coding Guidelines. Consultado en marzo de 2013, de http://msdn.microsoft.com/en-us/library/d55zzx87(v=VS.90).aspx

[21] Microsoft Corporation. (Junio de 2003). Design Guidelines for Secure Web Applications. Consultado en octubre de 2012, de http://msdn.microsoft.com/en-us/library/aa302420.aspx

[22] Microsoft Corporation. (Enero de 2006). Improving Web Application Security: Threats and Countermeasures. Consultado en octubre de 2012, de http://msdn.microsoft.com/en-us/library/ff648650.aspx

[23] Microsoft Corporation. (Noviembre de 2010). Microsoft Security Development LifeCycle. Consultado en octubre de 2012, de http://www.microsoft.com/security/sdl/default.aspx

[24] Ministerio de Administraciones Públicas. España. (Junio de 2006). MAGERIT – versión 2. Metodología de Análisis y Gestión de Riesgos, I - Método. Consultado en febrero de 2011, de http://publicaciones.administracion.es/

[25] Ministerio de Administraciones Públicas. España. (Junio de 2006). MAGERIT – versión 2. Metodología de Análisis y Gestión de Riesgos, II - Catálogo de Elementos. Consultado en febrero de 2011, de http://publicaciones.administracion.es

[26] National Institute of Standards and Technology. (Febrero de 2011). Glossary of Key Information Security Terms. Consultado en marzo de 2013, de http://csrc.nist.gov/publications/nistir/ir7298-rev1/nistir-7298-revision1.pdf

[27] National Institute of Standards and Technology. U.S. Department of Commerce. (2002). Risk Management Guide for Information Techonlogy Systems. Consultado enero de 2011, de http://csrc.nist.gov/: http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf

[28] Open Web Application Security Project. (Mayo de 2009). Data Validation. Consultado en octubre de 2012, de https://www.owasp.org/index.php/Data_Validation

[29] Open Web Application Security Project. (2009). OWASP Application Security Security Verification Standard - Web Application Standard. Consultado en marzo de 2013, de https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project_Proposal

[30] Open Web Application Security Project. (Marzo de 2009). OWASP Software Assurance Maturity Model. Consultado en enero de 2013, de http://www.opensamm.org

[31] Open Web Application Security Project. (Noviembre de 2010). OWASP Secure Coding Practices Quick Reference Guide. Consultado en febrero de 2013, de https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

[32] Open Web Application Security Project. (Marzo de 2013). OWASP Risk Rating Methodology. Consultado en marzo de 2013, de https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology

[33] Open Web Application Security Project. (Junio de 2009). Guide to authentication. Consultado en octubre de 2012, de https://www.owasp.org/index.php/Guide_to_Authentication

[34] Open Web Application Security Project. (Octubre de 2009). Session Management. Consultado en febrero de 2013, de https://www.owasp.org/index.php/Session_Management

[35] Open Web Application Security Project. (Febrero de 2012). Guide to Cryptography. Consultado en enero de 2012, de https://www.owasp.org/index.php/Guide_to_Cryptography

[36] Open Web Application Security Project. (Noviembre de 2012). OWASP Testing Guide v3. Consultado en marzo de 2013, de

https://www.owasp.org/index.php/Testing_Guide_Introduction#Developers.27_Security_Tests

[37] Open Web Application Security Project. (Febrero. de 2013). Industry: Citations. Consultado en abril de 2013, de https://www.owasp.org/index.php/Industry:Citations#National_.26_International_Legislation.2C_Standards.2C_Guidelines.2C_Committees_and_Industry_Codes_of_Practice

[38] Open Web Application Security Project. (Febrero de 2013). OWASP Top 10 Application Security Risks – 2013. Consultado en marzo de 2013, de https://www.owasp.org/index.php/OWASP_Top_Ten_Project

[39] Oracle. (2012). Administering Authentication. Consultado en febrero de 2013, de http://docs.oracle.com/cd/B19306_01/network.102/b14266/admnauth.htm

[40] Oracle. (s.f.). Secure Coding Guidelines for the Java Programming Language. Consultado en febrero de 2013, de http://www.oracle.com/technetwork/java/seccodeguide-139067.html

[41] Organización Internacional de Estandarización, Comisión Electrotécnica Internacional. (2009). GUIA ISO/IEC 73 Gestión de riesgos – Terminología – Líneas directrices para el uso en las normas.

[42] Organización Internacional para la Estandarización, Comisión Electrotécnica Internacional. (2005.). IRAM-ISO/IEC 14598-3. Tecnología de la información. Ingeniería de Software. Evaluación del producto de software. Parte 3: Proceso para desarrolladores. Instituto Argentino de normalización y certificación.

[43] Paul, M. (s.f.). The Ten Best Practices for Secure Software Development. (International Information Systems Security Certification Consortium.) Consultado en enero de 2013, de https://www.isc2.org/uploadedFiles/(ISC)2_Public_Content/Certification_Programs/CSSLP/ISC2_WPIV.pdf

[44] Software Engineering Institute. Carnegie Mellon. (Marzo de 2012). Static Source Code Analysis Tools. Consultado el Febrero de 2013, de http://www.cert.org/secure-coding/tools.html

[45] Software Engineering Institute. Carnegie Mellon. (Marzo de 2013). CERT Secure Coding Standards. Consultado en marzo de 2013, de https://www.securecoding.cert.org/confluence/display/seccode/CERT+Secure+Coding+Standards

[46] U.S. Deparment of Homeland Security. (Mayo de 2012). Software Security Testing. Software Assurance Pocket Guide Series. Development, Volume III. Consultado en marzo de 2013, de https://buildsecurityin.us-cert.gov/swa/downloads/SoftwareSecurityTesting_PocketGuide_1%200_05182012_PostOnline.pdf

15. BIBLIOGRAFÍA GENERAL

Dei, H. Daniel (2006). La tesis, Editorial Prometeo, Buenos Aires.

Department of Homeland Security. Build Security In. Setting a higher standard for software assurance. https://buildsecurityin.us-cert.gov.

International Organization for Standardization, International Electrotechnical Commission. (2005). ISO/IEC 15408-1:2005 Information Technology – Security Techniques – Evaluation Criteria for IT Security. Part 1: Introduction and general model.

International Organization for Standardization, International Electrotechnical Commission. (2005). ISO/IEC 15408-2:2005 Information Technology – Security Techniques – Evaluation Criteria for IT Security. Part 2: Security functional requirements.

International Organization for Standardization, International Electrotechnical Commission. (2005). ISO/IEC 15408-2:2005 Information Technology – Security Techniques – Evaluation Criteria for IT Security. Part 3: Security assurance requirements.

Marroquín Ortiz, Karin Xiomara. (2011). Gestión de Riesgos en Seguridad Informática. Trabajo final Especialización en Seguridad Informática. Universidad de Buenos Aires. Argentina.

Software Engineering Process Management Program. Carnegie Mellon University. (2012). CMMI® para Desarrollo, Versión 1.3. http://cmmiinstitute.com/cmmi-solutions/translations/cmmi-dev-spanish.

16. INDICE DE GRÁFICOS

17. INDICE DE TABLAS